NASA CR-

147801

MISSION PAYLOADS SUBSYSTEM DESCRIPTION
REVISION 2

JOB ORDER 81-197

(NASA-CR-147801) MISSION PAYLOADS SUBSYSTEM DESCRIPTION, REVISION 2 (Lockheed Electronics Co.) 186 p HC $7.50 CSCL 22A

N76-27301

Unclas
G3/13 45767

Prepared by

Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Under Contract NAS 9-12200

For

Mission Planning and Analysis Division

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

June 1976

LEC-7390

MISSION PAYLOADS SUBSYSTEM DESCRIPTION

REVISION 2

Job Order 81-197

PREPARED BY

J M Williams

J. M. Williams
Support Software Section

APPROVED BY

J. Philip Davis

J. P. Davis
Support Software Section

F N Barnes

F. N. Barnes
Dynamic Systems Department

Prepared By

Lockheed Electronics Company, Inc.

For

Mission Planning and Analysis Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

June 1976

LEC-7390

TECHNICAL REPORT INDEX/ABSTRACT

(See instructions on reverse side.)

1. TITLE AND SUBTITLE OF DOCUMENT Mission Payloads Subsystem Description Revision 2	2. JSC NO. JSC- 10787
3. CONTRACTOR/ORGANIZATION NAME Lockheed Electronics Co., Inc.	4. CONTRACT OR GRANT NO. NAS 9-12200
5. CONTRACTOR/ORIGINATOR DOCUMENT NO. LEC-7390	6. PUBLICATION DATE (THIS ISSUE) June 1976
7. SECURITY CLASSIFICATION Unclassified	8. OPR (OFFICE OF PRIMARY RESPONSIBILITY)
9. LIMITATIONS GOVERNMENT HAS UNLIMITED RIGHTS ☐ YES ☐ NO IF NO, STATE LIMITATIONS AND AUTHORITY	10. AUTHOR(S) J. M. Williams
11. DOCUMENT CONTRACT REFERENCES	12. HARDWARE CONFIGURATION
WORK BREAKDOWN STRUCTURE NO.	SYSTEM
CONTRACT EXHIBIT NO.	SUBSYSTEM
DRL NO. AND REVISION	MAJOR EQUIPMENT GROUP
DRL LINE ITEM NO.	

13. ABSTRACT

The Mission Payloads Subsystem (MPLS) utilizes a simplified trajectory model to generate a list of missions for the Scheduling Algorithm for Mission Planning and Logistics Evaluation (SAMPLE) program. MPLS is the mechanism that forms the basis of input for the other subsystems of SAMPLE and various post processors.

ORIGINAL PAGE IS
OF POOR QUALITY

14. SUBJECT TERMS

CONTENTS

Section Page

TABLES

FIGURES

DEFINITIONS

Term	Definition
ETR launch	A launch performed from the Eastern Test Range
Feasible combination	A collection of payloads that meet certain system constraints (e.g., Shuttle weight-to-orbit capability)
Flight schedule	A traffic model to which Shuttle resources are assigned
Load factor	The maximum value of two ratios found by comparing the actual weight-to-orbit of a mission to its theoretical capability and the total cargo weight at landing versus the maximum allowed
Mission type	The Orbiter assignment for a payload. The payloads either accompany the Shuttle to orbit (deployment), from orbit (retrieval), or both (in the case of attached and service payloads).
MPLS	The Mission Payloads Subsystem
OMS fuel	The fuel required for the Orbital Maneuvering System
Payload	One or more integrated experiment packages and associated third stage(s), if any, or simply a third stage itself
Payload margin	The minimum value of two calculations found by subtracting the total Shuttle weight at insertion from its theoretical weight-to-orbit capability, and the total cargo weight at landing subtracted from the maximum allowed
Payload type	The mission type for a payload
RCS fuel	The fuel required for the Reaction Control System

Term	Definition
Traffic model	A subset of the list of feasible combinations such that there are no redundant loads which can cover all the payloads. (The same payload is not assigned to two or more distinct flights.)
TSV	Third stage vehicle
Up/down payload	A payload that is deployed and retrieved in the same year
WTR launch	A launch performed from the Western Test Range

1. INTRODUCTION

This document describes the Mission Payloads Subsystem (MPLS) program and includes a general and technical program description as well as subroutine documentation and program functional flow.

The MPLS, a subsystem of the Scheduling Algorithm for Mission Planning and Logistics Evaluation (SAMPLE), was designed to generate a list of feasible combinations (LFC) from a payload model for a given calendar year. The Set Covering Algorithm (SCA), another subsystem of SAMPLE, uses the LFC to determine an optimum traffic model. The Operations Simulation and Resources Scheduling Subsystem (OSARS), the last subsystem, was designed to determine traffic model feasibility for available resources; however, as of this writing a proper SCA/OSARS interface has not been established.

The technique used by the MPLS to determine the validity of a combination is based on payload sequence dependent and independent constraint tests. The independent constraints are performed first to eliminate those missions which fail due to simple parameters tests; the dependent constraints are tested to determine the feasibility of the combination with respect to delta velocity (ΔV) requirements. The specific order of the tests tends to minimize the computer time required to examine the data.

Since the MPLS began to evolve in 1973, the logic has been modified wherever necessary to incorporate new requirements and capability. The changes significantly decrease program run time and increase the number of feasible solutions found.

2. PROGRAM DESCRIPTION

2.1 GENERAL DESCRIPTION

The MPLS generates and evaluates payload combinations from an input payload model for a specific year. The feasibility of each combination is evaluated by flight sequence tests, both dependent and independent. The flight sequence is a series of maneuvers designed to achieve the orbital requirements of each payload in the combination; its validity is contingent upon the accumulative impact of the payload characteristics compared to the system constraints. A combination is infeasible when a constraint or set of constraints have been violated.

2.1.1 FLIGHT SEQUENCE INDEPENDENT TESTS

The flight sequence independent tests are controlled by subroutine CPTEST. Tests are made for payload redundancy, mission type, discipline mix, weight, length, Reaction Control System (RCS) fuel, third stage vehicle (TSV) cargo capacity, and miscellaneous constraints. Each constraint is a gross check of a combination against performance limits of the Orbiter.

2.1.1.1 Payload Redundancy

Each payload combination is examined for redundant payloads, rejecting those combinations with duplicate payload names.

2.1.1.2 Mission Type

Each combination's mission type is compared to an internal list, rejecting those that do not match. This test is optional.

2.1.1.3 Discipline Mix

The combinations discipline mix is compared to an allowable list, rejecting those that do not match. This test is optional.

2.1.1.4 Weight

The total chargeable weight in the Orbiter's cargo bay is compared to a limit of 65,000 lb at launch and 32,000 lb at landing. The chargeable payload weight is composed of the individual payloads, Electrical Power System (EPS) kits, payload adapters, and the TSV used on the mission. A combination that exceeds the limit is rejected.

2.1.5 Length

The total length of the payloads at launch and landing is compared to a limit of 60 feet. A combination that exceeds the limits is rejected.

2.1.1.6 RCS Fuel

The RCS fuel is a function of the number of rendezvous. If more than 9000 lb of fuel is required, the combination is rejected.

2.1.1.7 TSV Cargo Capacity

The total payload plus TSV length and weight at launch and landing are compared against system limits. If a TSV cannot be found which meets the combination's requirements, it is rejected.

2.1.1.8 Miscellaneous Constraints

The total number of payloads on a TSV must be less than four. Only one dedicated TSV payload is permitted per combination.

2.1.2 FLIGHT SEQUENCE DEPENDENT TESTS

The flight sequence dependent tests are controlled by subroutine SEQTST. These tests consider the ΔV fuel usage by using a form of the ideal rocket equation as well as payload weight and length as a function of its center-of-gravity (CG). The fuel requirements of the TSV are not known initially, so the Orbiter and TSV are considered separately. The total TSV configuration is then used as an Orbiter Payload.

2.1.2.1 Orbiter Requirements

The Orbiter has been designed to satisfy the requirements of nominal (low altitude) payloads; high altitude payloads are serviced by the use of a TSV. The TSV is considered to be an Orbiter payload deployed at the first orbit or at an altitude of 150 miles (whichever is less) and retrieved at 20 miles above the last Orbiter orbit. The Orbiter requirements are determined by computing the orbit-to-orbit ΔV's, the TSV requirements, and the ΔV fuel used for deorbit. The fuel requirements may exceed the maximum Orbital Maneuvering System (OMS) main tank fuel capacity, so the OMS capability may be extended by the use of up to three OMS kits. The use of OMS kits reduces the cargo bay capacity, as the dry weight of the kits at landing must be included with payload down weight, assuming that all OMS fuel is exhausted. Furthermore, the OMS kits physically intrude into the cargo bay envelope.

2.1.2.2 TSV Requirements

A combination requires a TSV if at least one payload has a desired orbit of 700 miles or more. The type of TSVs available are:

a. Expendable - Used to deploy a payload but not retrieved

b. Reusable - Used to deploy a payload and retrieved

In making an assignment, the first TSV meeting all requirements is used, then the TSV requirements are included in the Orbiter's payload sequence. If the Orbiter's capability is exceeded, the next TSV is examined. If all available TSVs fail to meet the requirements of the payload sequence, the combination is rejected.

2.2 TECHNICAL DESCRIPTION

2.2.1 ANALYSIS

The solutions generated by the MPLS tend to compute the minimum ΔV for a particular mission without regard to payload type; the marginal cases may be rejected as only the initial sequence is examined. (An exhaustive numeration technique has been inhibited in the current program version.) The ΔV computations are performed by the following techniques.

2.2.1.1 Hohmann Transfers

The Hohmann transfer algorithm is used to compute the ΔV required to change orbits in all instances except deorbit. The initial (insertion) transfer is made from an elliptical orbit; the other transfers are made to and from circular orbits.

2.2.1.2 Empirical Equation

The deorbit is simulated by using an empirical formula which computes the deorbit ΔV from the inclination and altitude of the last parking orbit so as to satisfy reentry conditions.

Fuel usage is computed for both the Orbiter and TSV by a form of the ideal rocket equations, incorporating ΔV requirements.

The MPLS assumes a 20 ft/sec ΔV for any rendezvous maneuver; hence, the best possible phasing required to accomplish the rendezvous is considered. This optimistic philosophy may be justified because of the built-in program penalties:

a. An OMS fuel reserve of 450 lb is carried from insertion to landing. This reserve requires an additional 80 lb of fuel for a mission requiring a total ΔV of 1600 ft/sec.

b. An RCS fuel reserve of 3400 lb is carried from insertion to landing. This reserve requires an additional 580 lb of fuel for a mission requiring a total ΔV of 1600 ft/sec.

The deorbit ΔV is also the ΔV that would be required for an abort. That is, if an abort were to occur from the first parking orbit, a deorbit burn would be made. An examination of the ΔV required for a deorbit and a Hohmann transfer indicates a higher ΔV requirement for the Hohmann transfer. This implies that an abort from any point in the orbit is always considered. Also, the launch sites used have a minimum abort fuel requirement which is always carried. In other words, the minimum fuel carried to orbit is sufficient for an abort.

2.2.2 METHOD OF SOLUTION

The Orbiter/TSV mission is composed of three distinct segments:

a. Insertion of the Orbiter into initial parking orbit and deployment of the TSV

b. Orbiter/TSV maneuvers performed in earth orbit

c. Orbiter reentry

The function of the trajectory calculations in the MPLS is to compute the ΔV requirements for each segment. In general, the ΔV is computed using a Hohmann transfer algorithm at the insertion to parking orbit and maneuver phase; the reentry ΔV is computed using an empirical equation (ref. 1). The specific algorithms for computing the ΔV are:

a. Segment 1 - Entry point DVIPK of subroutine DLTAV is used to compute the ΔV required to transfer from insertion ellipse to the initial circular parking orbit.

b. Segment 2 - Functions DLTAV and entry point DVEL compute the ΔV between two circular parking orbits.

c. Segment 3 - Entry point DVDORB of subroutine DLTAV computes the deorbit ΔV as a function of the altitude and inclination of the last parking orbit.

The mass and length history computation require discrete weight, length, and velocity changes at each maneuver. The basic approach considers the total payload down (deorbit) and then computes the fuel requirement backwards to insertion. The equation used is the ideal rocket equation

$$W_S = W_E \cdot e^{\Delta V/(gI_{sp})}$$

where

W_S = Weight at the start of the maneuver

W_E = Weight at the end of the manuever

g = Acceleration due to gravity

I_{sp} = The specific impulse for the Orbiter OMS engines

The total weight at landing is

$$W_L = W_V + W_{RCS} + W_{OMS} + W_{KITS} + W_{CRW} + W_{EPS} + PL_D$$

where

W_V = Vehicle dry weight

W_{RCS} = Weight of the RCS fuel reserve

W_{OMS} = Weight of the OMS reserve fuel

W_{KITS} = OMS kit dry weight

W_{CRW} = EPS deadweight and crew requirements not charged to the payload weight

W_{EPS} = EPS kit dry weight

PL_D = Weight of the payload carried down

Once the weight of the vehicle is determined at insertion, the total weight of the OMS fuel is computed as

$$W_{FUEL} = W_I - W_V - W_{UP} - W_{TRCS} - W_{CRW}$$

where

W_{UP} = Weight of the Orbiter payloads at insertion, including the TSV, its fuel, and payloads

W_{TRCS} = Total weight of the RCS fuel used

W_I = Weight of the vehicle at insertion

Assuming that the OMS fuel is within the maximum 59750 lb allowed, the total weight of the vehicle must be readjusted to allow for OMS kits, if they are required. A kit is needed if the OMS fuel required exceeds 23900 lb, (table 2.2.2-I) and the additional fuel used to carry the dry kit is computed as

$$W_{DRY} = W_{KITS}\left(e^{T_{DV}/gI_{sp}} - 1\right)$$

and

$$W_{UP} = W_{KITS} + W_{DRY}$$

where T_{DV} = total Orbiter delta velocity required. The down weight is then adjusted as

$$W_{DOWN} = W_{KITS} + W_{DRY} + \sum_{i=1}^{n} W_{PLD_i}$$

where W_{PLD_i} = the weight of the ith payload

The length history is maintained by the summation of the discrete length changes for each payload activity. The length of any OMS kits will reduce the length available for the payloads.

Both the total insertion weight of the Orbiter and the total length requirements for the payloads must be reverified. The total insertion weight is compared to the total weight-to-orbit capability of the Orbiter. The insertion weight-to-orbit is computed empirically as a function of the initial payload orbit inclination.

When all other constraints have been met, then it is necessary to determine if the cargo CG is acceptable. The basic form of the equation used to compute CG is

$$CG = \frac{W_1 \left(d + C_1\right) + \sum_{L=2}^{N} \left(d + \ell_i + C_i\right)}{\sum_{L=1}^{N} W_i}$$

where

W_i = Weight of the ith payload in the cargo bay

d = Distance from the front wall to the first payload

ℓ_i = Total length of the payloads and reserved gaps between payloads

C_i = The CG of the ith payload in the cargo bay

The CG is tested against the cargo minimum and maximum length constraints to determine mission feasibility.

TABLE 2.2.2-I. - SHUTTLE CHARACTERISTICS

Description	Length (ft)	Weight (lb)	Capacity (lb)	ΔV (ft/sec)
Main OMS tank			23,900	
1 OMS kit	4.75	2350	11,950	
2 OMS kits	8.6	3505	23,900	
3 OMS kits	12.1	4715	35,850	
RCS main tank			3,400	
RCS tank 2			3,600	
RCS spillover			2,000	
Reserve OMS at landing			450	
RCS rendezvous requirements		Unique for each payload		20
TSV RCS rendezvous requirements		1800		20
Spacecraft				
Dry weight			163,300	
Cargo bay	60		65,000	

3. PROGRAM USAGE

3.1 INPUT DESCRIPTION

Input to the MPLS is a subset of the input required for the SAMPLE. Information in this section supplements the SAMPLE Users Guide (ref. 2) as applied to the MPLS. Details of the TSV and payload data sets are provided. All input is represented as data card images, regardless of whether they originate on cards or line images from a demand terminal.

The first user input, an executive request, identifies the payload model; the input format of the payload model is described in detail in table 3.1-I. The remaining inputs specify user options. The user options are input, starting in column 1, as responses to prompts. Each prompt may be explained in detail by entering a zero and a carriage return. A sample job stream is given in section 5.4.

3.2 OUTPUT DESCRIPTION

3.2.1 NORMAL OUTPUT

The output of the MPLS can be classified as input data and trajectory data. The initial output consists of approximately three pages of information pertaining to the payload model and is optional. The next set of output pertains to the relationship of the payloads to their mission type, discipline mix code, the TSV, and the number of flights per year.

The trajectory section prints the following parameters for feasible combinations:

- Flight number
- Launch site
- Payload identification number/name
- Orbiter sequence

- Inclination
- Total weight up/down
- Up/down length
- TSV name
- TSV sequence
- Altitude
- Payload type
- Orbiter and TSV ΔV
- Number of OMS kits
- Load factor
- Payload margin
- Percentage used of the first OMS kit

The total number of combinations generated is printed at the end of the trajectory section, as well as the number of feasible and infeasible combinations. The Feasible Mission File is also output; refer to table 3.2.1-I.

3.2.2 ABNORMAL OUTPUT

The following is a list of diagnostic messages and subprogram references.

Diagnostic Message	Subroutine
** STORING ERROR **†	LOAD5
DOWN WEIGHT CONSTRAINT VIOLATED	ERRPRT
MISSION TYPE NOT ALLOWED	ERRPRT

†An error caused by storing data beyond the assigned storage area; this error will terminate program execution.

NO FEASIBLE SEQUENCE FOUND.	ERRPRT
NO TUGs SATISFY LENGTH AND WEIGHT CONSTRAINTS	ERRPRT
NUMBER OF PAYLOADS ON A TUG GREATER THAN 3	ERRPRT
PAYLOAD iiii CAN ONLY BE ON A DEDICATED TUG	ERRPRT
PAYLOAD DISCIPLINE MIX TYPE NOT ALLOWED iiiiii	ERRPRT
THE RCSWT IS GREATER THAN THE CAPACITY FOR THIS CASE	ERRPRT
TOTAL LENGTH GREATER THAN BAY LENGTH, DOWN TOTAL LENGTH = rrrrr.rr	ERRPRT
TOTAL LENGTH GREATER THAN BAY LENGTH, UP TOTAL LENGTH = rrrrr.rr	ERRPRT
UP WEIGHT CONSTRAINT VIOLATED	ERRPRT

3.3 PROGRAM UNITS

English units of measure are used for the MPLS. Units given for altitudes, inclinations, ΔV, specific impulse, weight, and length are nautical miles (n. mi.), degrees (deg), feet/second (ft/sec), seconds (sec), pounds (lb) and feet (ft).

TABLE 3.1-I.- PAYLOAD MODEL CARDS

The first set of cards in the payload model identifies the solid-propellant Interim Upper Stage (IUS) data. The second set of cards pertains to the Liquid-Propellant Upper Stage (LUS) data, and the remaining cards identify individual payload characteristics.

Card	Word	Symbol	Type	Units	Format	Column	Description
1	1	N	I	-	Free	-	Number of unique stages in the model
2-L ($L \leq 15$)	1	TISP	R	sec	F10.0	1-10	Specific impulse of the stage
	2	TTSVWT	R	lb	F10.0	11-20	Total weight of the stage
	3	FUEL	R	lb	F10.0	21-30	Total fuel available for the stage
	4	TSVLN	R	ft	F10.0	31-40	Length of the stage
L+1	1	NTSVS	I	-	Free	-	Number of IUS vehicles to be used ($NTSVS \leq 10$)
L+2	1	N	I	-	Free	-	Number of stages on the IUS ($1 \leq N \leq 5$)
	2	NUNQS(NSTVS,1)	I	-	Free	-	First stage identified by the first set of cards
	3	NUNQS(NSTVS,2)	I	-	Free	-	Second stage number
		⋮					
	N+1	NUNQS(NSTVS,N)	I	-	Free		Last stage number
	N+2	YRAVAL	I	-	Free	-	A two-digit number which represents the year of availability of this vehicle

TABLE 3.1-I.- CONTINUED

The second set of cards pertains to the LUS vehicles. The order in which the data are input is the order each LUS is considered. For simplicity, the next card in the sequence is denoted as "k".

Card	Word	Symbol	Type	Units	Format	Column	Description
k	1	N1	I	-	Free	-	Number of LUS vehicles to be input
k+1	1	TUGLN	R	ft	F10.3	1-10	Length of the LUS
	2	TUGWT	R	lb	F10.3	11-20	Weight of the LUS
to	3	TUGCAP	R	lb	F10.3	21-30	Capacity of the LUS
	4	TUGISP	R	sec	F10.3	31-40	Specific impulse of the LUS
k+N1	5	TUGTYP	I	-	I2	44-45	The LUS type =1, expendable =3, reusable
	6	YRAVAL	I	-	I2	47-48	First year available for the LUS
k+N1+1	1	NUMPL	I	-	Free field	-	Number of payloads in the model
	2	MKS	I	-	Free	-	A flag specifying the internal units of the payload model =1, the units are in mks =2, the units are in fps

TABLE 3.1-I.- CONTINUED

The rest of the cards are identified in sets of three; and identify individual payload characteristics.

Card	Word	Symbol	Type	Units	Format	Column	Description
1	1	NUMB	A	-	2A6	4-15	Payload alphanumeric identification label
	2	NDISP	A	-	2A6	16-27	Payload discipline
	3	NAME	A	-	6A6	28-63	Payload description
	4	LEN	R	ft	F5.0	64-68	Total payload length, including the pallet and/or lab
	5	WT	R	lb	F6.0	69-74	Total weight of the payload at lift-off, including the pallet and/or lab, if applicable
	6	WT1	R	lb	F6.0	75-80	Total weight of the payload at landing, including the pallet and/or lab, if applicable
2	1	DIAM	R	ft	F4.1	4-7	Diameter of the payload
	2	HA	R	n.mi.	F9.0	8-16	Desired circular altitude
	3	INCL	R	deg	F5.1	17-21	Desired inclination
	4	C3	R	ft^2/sec^2	F5.0	22-26	C3 energy, this number will be multiplied by 100,000.

TABLE 3.1-I.- CONTINUED

Card	Word	Symbol	Type	Units	Format	Column	Description
2	5	PMT	I	-	I2	27-28	Payload mission type flag =1, attached =2, servicing =3, deploy =4, retrieved
	6	FLTPYR	I	-	13I3	29-67	Flight frequency for 1979 to 1991. Each word contains a flag and denotes the number of times a payload goes up and/or down in a given year. The word is entere as XYZ, where X=1 The up and down trips for this payload can be combined on a flight X=2 The up and down trips for this payload cannot be combined on a flight X is ignored if the mission type is 1 or 2, or if the payload is always deployed or retrieved. In these situations, X is set to zero (or blank) If X is nonzero, Y is the number of deplo ments ($Y \leq 9$) and Z is the number of retrievals If X is zero, YZ is the number of deploys retrieves, sorties, or services
	7	IRPT	I	-	I3	68-70	A flag which indicates the repeat conditions of a payload =0 Payloads to be repeated in a given year cannot be flown on the same flight =1 Payloads flown can be repeated in a given year on the same flight

TABLE 3.1-I.- CONCLUDED

Card	Word	Symbol	Type	Units	Format	Column	Description
2	8	PLDUR	R	hrs	F3.1	71-73	Desired time on-orbit
	9	OPTIME	R	hrs	F3.1	74-76	Nominal duration of payload operation/ day of time onboard the orbiter
	10	IFREQ	I	-	I3	77-79	Number of times/day the payload is operated while onboard the orbiter
	11	MODE	I	-	I1	80	Preferred delivery mode (attached payloads only) =1 lab =2 pallet =3 lab and pallet
3	1	RCS	R	lb	F6.1	4-9	Reaction Control System (RCS) fuel requirements based on individual payload requirements
	2	OXEPS	R	lb	F6.1	10-15	Electrical Power System (EPS) O_2 requirements based on individual payload demands
	3	HEPS	R	lb	F5.1	16-20	EPS H_2 requirements based on individual payload demands
	4	CGPOS	R	ft	F4.1	21-24	Distance of the payload center of gravity from the front end of the payload
	5	FTSV	R	-	I1	25	A flag when set nonzero forces the use of a TSV

TABLE 3.1-II. - THE PAYLOAD DISCIPLINE CODES

Symbol	Experiment
AS	Astronomy
OP	Earth and Ocean Physics; same as EP
CN	Communication and Navigation
ST	Space Technology
LS	Life Sciences
EO	Earth Observation
PL	Planetary
SP	Space Processing
PH	Physics
LU	Lunar
OA	Office of Applications
AP	Atmospheric
SO	Solar Physics
HE	High Energy Astrophysics
EP	Earth and Ocean Physics
NN	Non-NASA/non-DOD

TABLE 3.2.1-I. - DESCRIPTION OF THE FEASIBLE MISSION FILE

[The Feasible Mission File contains fixed length records, each 83 words, and is unformatted.]

Sequence no.	Symbol	Dimension	Type	Description
1	MM	1	I	Flight number
2	M	1	I	Number of payloads on this flight
3	IC	6	I	Payload numbers
4	IB	6	I	Optimal payload sequence
5	NAME1	6	A	Alphanumeric mission type
6	LAUNCH	1	I	An integer set to 1 or 2 to indicate ETR or WTR
7	NTUGPL	1	I	Number of TSV payloads
8	IFTUG	1	I	TSV number used for this mission
9	ICTUG	6	I	TSV payload numbers
10	NOKITS	1	I	Number of OMS kits used
11	PWMARG	1	R	Additional payload weight the Shuttle could carry on this flight
12	PCTUSE	1	R	Percentage of the first OMS kit used
13	FLOAD	1	R	Load factor
14	POINC	1	R	Inclination of the first orbit
15	POALT	1	R	Altitude of the first orbit
16	ALT	6	R	Orbital altitude of each payload
17	XINC	6	R	Orbital inclination of each payload

TABLE 3.2.1-I. - DESCRIPTION OF THE FEASIBLE MISSION FILE (Continued)

Sequence no.	Symbol	Dimension	Type	Description
18	TOTLU	1	R	Total length up
19	TOTLD	1	R	Total length down
20	TOTWU	1	R	Total weight up
21	TOTWD	1	R	Total weight down
22	TUGDV	1	R	Total TSV ΔV used
23	ORBDV	1	R	Total Shuttle ΔV used
24	IG	6	A	Array of alphabetic mission types
25	ICC	12	I	Cost coefficients
26	IDENT	12	A	A list of two-word payload names; 12 character identification for the payloads

4. EXECUTION CHARACTERISTICS

4.1 RESTRICTIONS

The MPLS has the following limitations which apply to the analysis of any year under investigation:

a. The maximum number of payloads allowed on a flight is six.

b. The maximum number of single payloads which may be investigated is 100.

c. A limit of 6200 feasible missions are allowed for a specific number of payloads in a combination.

d. The maximum amount of data which may be written onto a mass storage file before the program terminates is 50 positions (3200 tracks).

e. The numbers of payloads per combination is restricted to three when both the mission type and discipline constraints are activated.

Further restrictions are specified in the subprogram documentation.

4.2 VALIDITY

Validation of the MPLS has been accomplished primarily by the comparison of results obtained from other programs and by hand calculations. Reference 3 gives a detailed explanation of the validation performed.

5. REFERENCE INFORMATION

5.1 DETAILED FLOWCHART

Figure 5.1-1 illustrates the subprogram interaction for the MPLS. Figure 5.1-2 illustrates the flow of the MPLS executive logic.

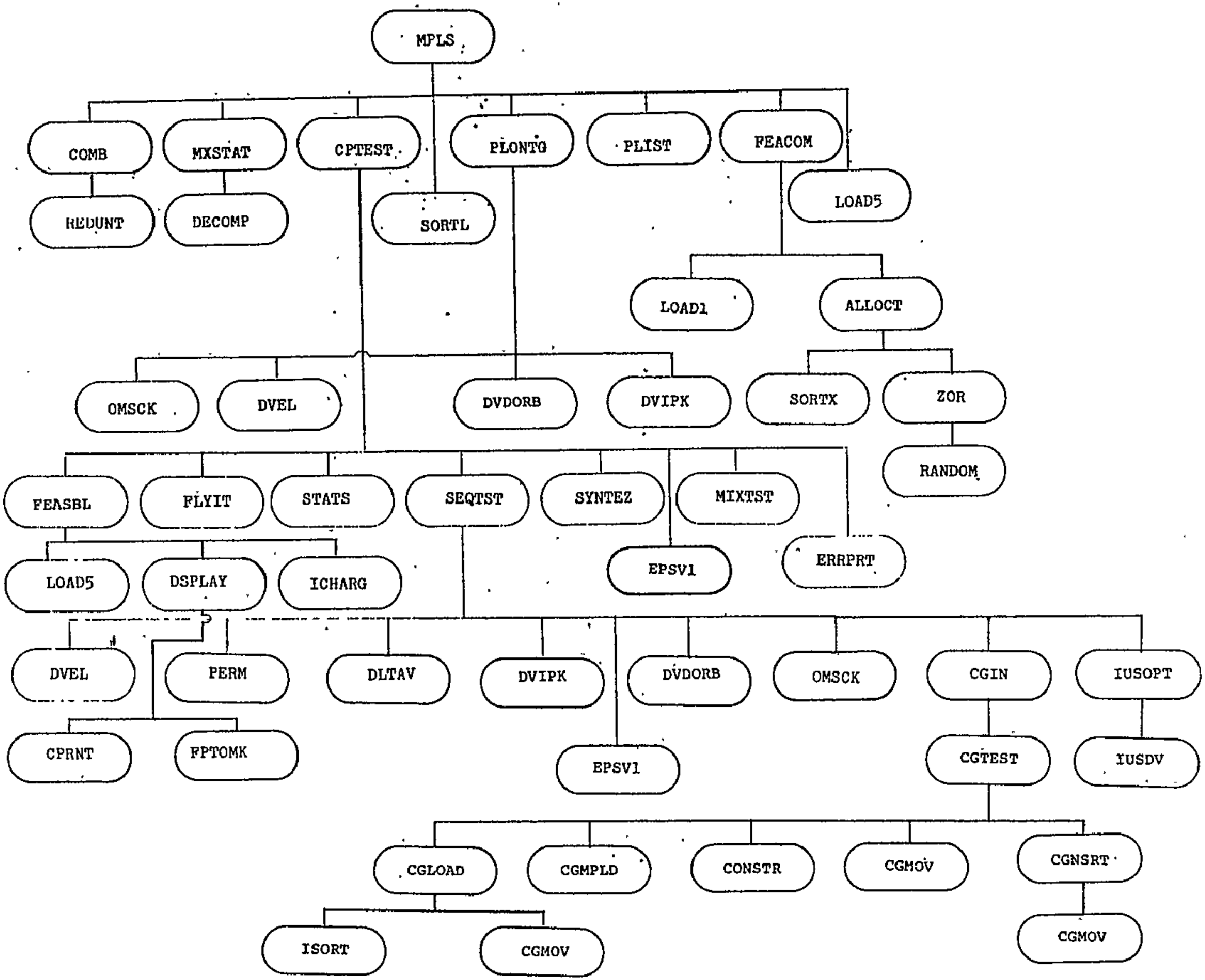
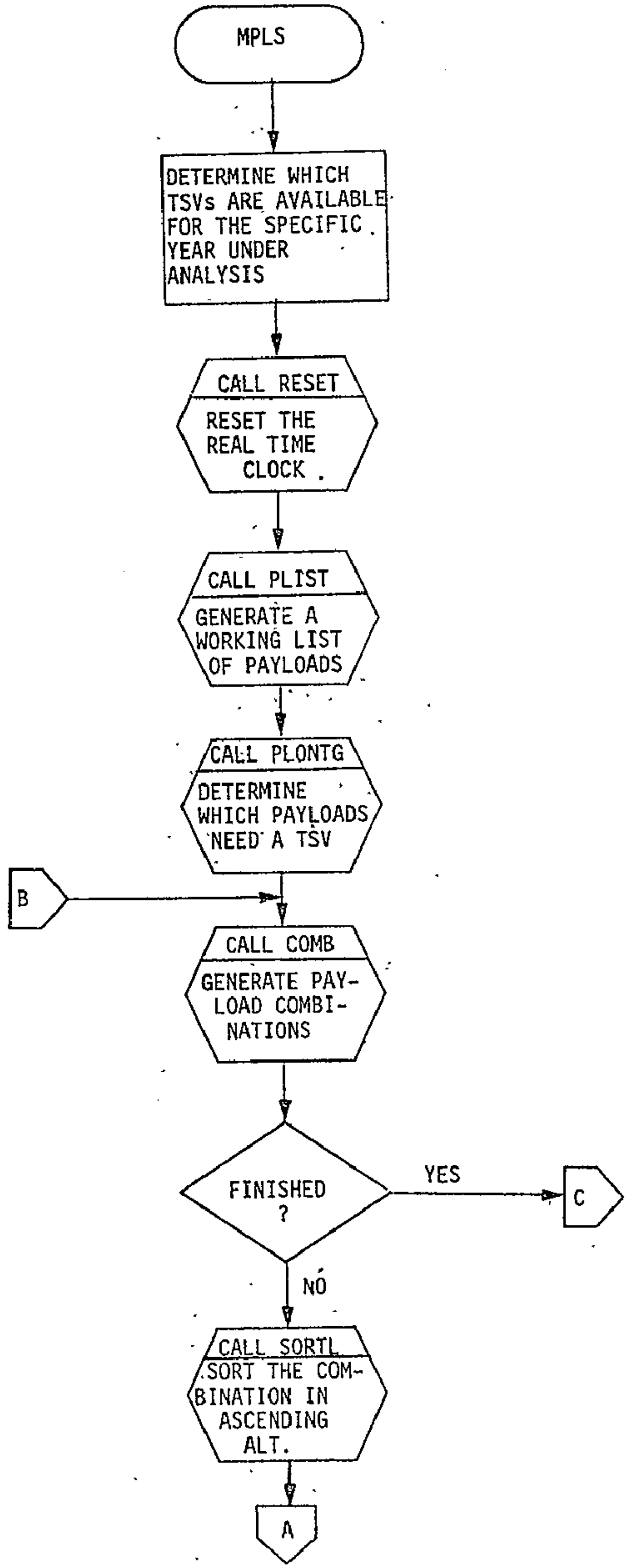


Figure 5.1-1.- MPLS subroutine interaction.



ORIGINAL PAGE IS OF POOR QUALITY

Figure 5.1-2.- MPLS functional flow.

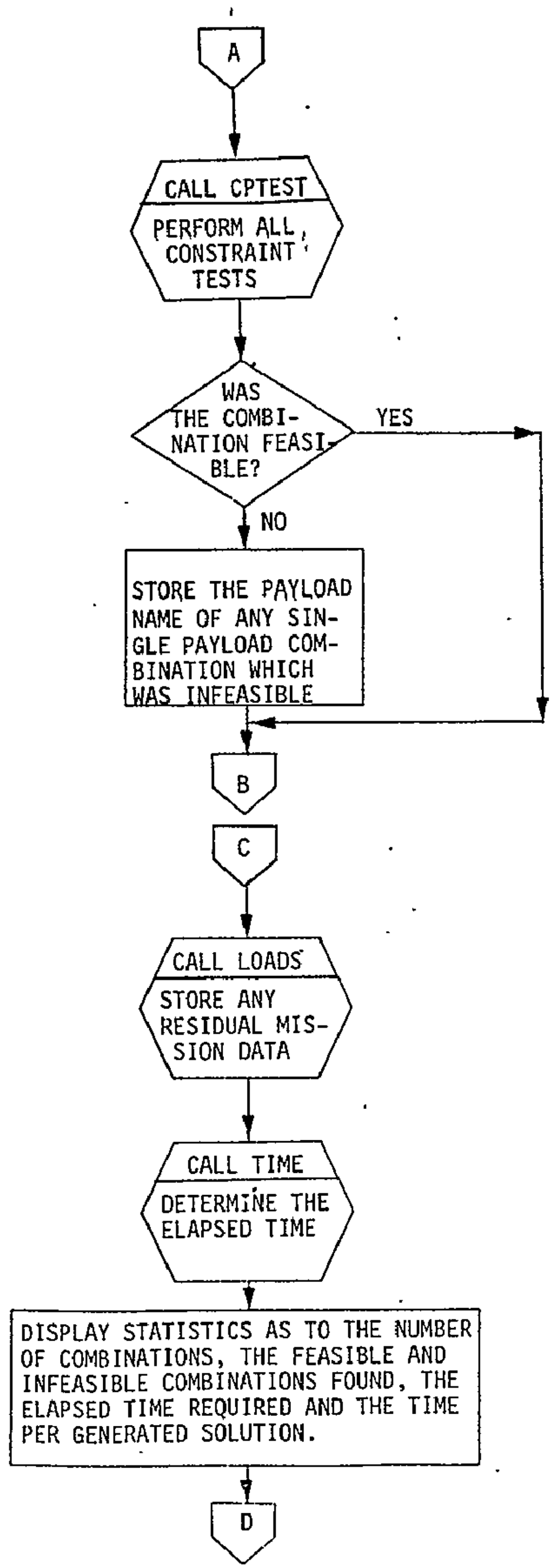


Figure 5.1-2.- MPLS functional flow (continued).

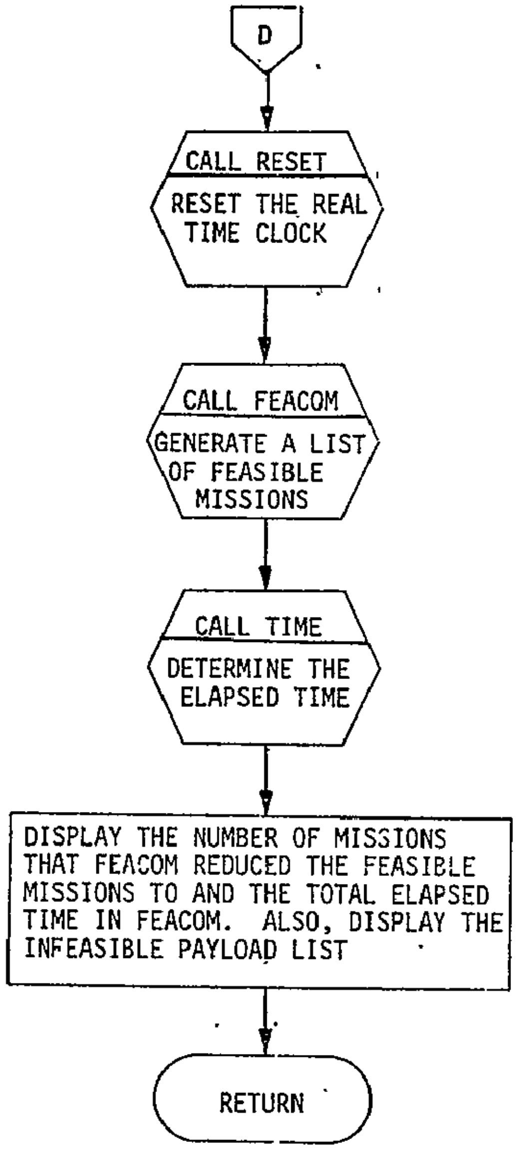


Figure 5.1-2.- MPLS functional flow (concluded).

5.2 LABELED COMMON

Table 5.2-I provides descriptions for all variables in labeled COMMON.

TABLE 5.2-I. - VARIABLES IN LABELED COMMON

- COMMON block name: C1

  Description: Labeled COMMON C1 transmits the payload model data to various subprograms of MPLS.

  Storage required: 8201

Location	Name	Dimension	Type	Description
1-400	NUMB	200x2	A	Hollerith payload identification names
401-800	NDISP	200x2	A	Hollerith payload disciplines
801-2000	NAME	200x6	A	Hollerith array describing the payload
2001-2200	LEN	200	R	Array containing the total payload length including the pallet and/or lab.
2201-2400	WT	200	R	Array containing the total weight of the payload at lift-off
2401-2600	WT1	200	R	Array containing the total weight of the payload at landing
2601-2800	DIAM	200	R	Payload diameter
2801-3000	HA	200	R	Desired circular altitude
3001-3200	INCL	200	R	Desired orbital inclination
3201-3400	C3	200	R	C_3 energy
3401-3600	PMT	200	I	Payload mission type flag: =1 attached =2 servicing =3 deploy =4 retrieve
3601-6200	FLTPYR	200x13	I	Array of the flight frequencies; each of the 13 words represents the data for an entire year starting at 1979. Each word contains a flag and denotes

TABLE 5.2-I. - VARIABLES IN LABELED COMMON (Continued).

Location	Name	Dimension	Type	Description
				the number of times a payload goes up and/or down in a given year. The word is entered as XYZ where X = 1 The up and down trips for this payload can be combined on a flight. X = 2 The up and down trips for this payload cannot be combined on a flight. X is ignored if the mission type is always deployed or always retrieved. In these situations X is set to zero. If X is nonzero, Y is the number of deployments (Y<9) and Z is the number of retrievals. If X is zero, YZ is the number of deploys, retrieves, sorties, or services.
6201-6400	IRPT	200	I	A flag which indicates the repeat conditions of a payload =0 Payloads to be repeated in a given year cannot be flown on the same flight =1 Payloads to be repeated in a given year can be flown on the same flight
6401-6600	PLDUR	200	R	Desired time on-orbit
6601-6800	OPTIME	200	R	Nominal duration of payload operation per day of time onboard the Orbiter.

TABLE 5.2.I. - VARIABLES IN LABELED COMMON (Concluded)

Location	Name	Dimension	Type	Dimension
6801-7000	IFREQ	200	I	Number of times per day the payload is operated while on-board the Orbiter
7001-7200	MODE	200	I	Preferred delivery mode for attached payloads =1, lab =2, pallet =3, lab and pallet
7201-7400	RCS	200	R	RCS fuel requirements based on individual payload requirements
7401-7600	OXEPS	200	R	EPS O_2 requirements based on individual payload demands
7601-7800	HEPS	200	R	EPS H_2 requirements based on individual payload demands
7801-8000	CGPOS	200	R	Distance of payload center of gravity from the front end of the payload
8001-8200	FTSV	200	I	A flag when set non-zero forces the use of a TSV
8201	NUMPL	1	I	Number of payloads in the model

- COMMON block name: C2

Description: Labeled COMMON C2 transmits the EPS data to various subprograms of MPLS.

Storage required: 46

Location	Name	Dimension	Type	Description
1	EPSWT	1	R	The total EPS weight carried to insertion
2	TEPSRV	1	R	The total EPS required for payloads retrieved by the TSV and loaded onto the Orbiter
3	TEPSDP	1	R	The total EPS required for payload deployed by the TSV
4	W	1	R	The weight of the Orbiter excluding fuel, payloads, and consumables
5-13	DVORB	9	R	The ΔV requirements for each maneuver in the mission
14-22	DWORB	9	R	The discrete weight changes corresponding to DVORB
23-31	ORBMR	9	R	An array which is computed as $e^{\Delta V/G \cdot I_{sp}}$ where ΔV corresponds to DVORB and the terms $g \cdot I_{sp}$ denote the gravity term and specific impulse
32	NORBP3	1	I	The number of Orbiter payload plus 3
34	EPSKIT	1	I	The number of EPS kits used
35	EPSDRY	1	R	The total EPS dry tank weight

Location	Name	Dimension	Type	Description
36	EDRY	1	R	The weight of one dry EPS kit
37	EO2RZ	1	R	The EPS O_2 deadweight requirements
38	EH2RZ	1	R	The EPS H_2 deadweight requirements
39	EPNKT	1	I	The number of EPS kits initially loaded and not charged to the cargo payload weight
40	EPCRW	1	R	The crew and deadweight requirements not chargeable to the Orbiter's payload weight
41	WTO2	1	R	The weight of one EPS O_2 kit, not including the tank
42	WTH2	1	R	The weight of one EPS H_2 kit, not including the tank
43	O2SL	1	R	The O_2 crew and deadweight requirements not charged to the payload weight
44	H2SL	1	R	The H_2 deadweight requirements not charged to the payload weight
45	EPSO2	1	R	The O_2 EPS deadweight requirements charged to the payload weight
46	EPSH2	1	R	The H_2 EPS deadweight requirements charged to the payload weight

- COMMON block name: C3

  Description: Labeled COMMON C3 transmits a debug print flag to various MPLS routines and loading information the CG routines.

  Storage required: 10

Location	Name	Dimension	Type	Description
1	IPRNT	1	I	A flag when set nonzero causes debug information to be printed from MPLS routines
2-7	IJSORB	6	I	An array which contains the orbiter payload sequence being flown. This array is used as an index into ICORB.
8-10	IJSTUG	3	I	An array which contains the TSV payload sequence being flown. This array is used as an index into ICTUG.

- COMMON block name: C5

Description: Labeled COMMON C5 transmits a working list of the payload model data to various subprograms of MPLS.

Storage required: 504

Location	Name	Dimension	Type	Description
1-100	PAYNO	100	I	List of payloads which fly during the year under analysis
101-200	PLREP	100	I	Information for duplicate payloads where even indices indicate the number of duplicates of the payload number identified by the odd indices
201	NUMRPT	1	I	Total number of repeated payloads found for the year under analysis
202	NPLNYR	1	I	Number of payloads which fly during the year under analysis
203-302	IDOWN	100	I	Identification of payloads which are both deployed and retrieved in the same year
303	NUPDN	1	I	Number of up/down payloads found
304-403	IREP	100	I	A working storage area used to keep track of the repeated payloads as they are output
404-503	PMT1	100	I	Payload mission types defined as = 1 attached = 2 servicing = 3 deploy = 4 retrieve

Location	Name	Dimension	Type	Description
				= -3 An up/down payload; deployed = -4 An up/down payload; retrieved
504-603	NEEDTG	100	I	Array used to indicate whether a payload in the PAYNO list requires a TSV as: = 0 , no TSV required = 1 , a TSV is required

- COMMON block name: C6

  Description: Labeled COMMON C6 transmits the TSV data from the payload model to various subprograms of MPLS.

  Storage required: 106

Location	Name	Dimension	Type	Description
1-15	TUGLN	15	R	Array containing the lengths of the TSVs
16-30	TUGWT	15	R	Array containing the dry weight of each TSV
31-45	TUGISP	15	R	Specific impulse of each of the TSV's main engines
46-60	TUGCAP	15	R	Maximum amount of propellant for each TSV
61-75	YRAVAL	15	I	First year available for each of the TSVs
76-90	STAGE3	15	I	Array used to identify which TSVs are available for the year under analysis; a nonzero word in the array specifies that the ith TSV is available
91-105	TUGTYP	15	I	The TSV type =1, expendable LUS =2, expendable IUS =3, reusable LUS
106	NTUGN	1	I	The number of TSV's used for the analysis

- COMMON block name: C7

Description: Label COMMON C7 transmits data pertaining to the feasible missions. Other parameters identifying the payload sequence and the number of single mission payloads that have been rejected are also transmitted.

Storage required: 111

Location	Name	Dimension	Type	Description
1	NCC	1	I	Number of combinations generated
2	M	1	I	Number of payloads in the IA array
3-8	IA	6	I	Initial payload sequence, used to index into the PAYNO array.
9	NFS	1	I	Number of single payload missions rejected.
10-109	NFLS	100	I	An array containing the names of the single payload missions rejected
110	NOPL	1	I	An option which causes repeated payloads to be flown on the same mission; ignored if zero
111	KOPT	1	I	A flag, set nonzero, used to inhibit permutations of a payload sequence

- COMMON block name: C8

Description: Labeled COMMON C8 transmits information pertaining to the Orbiter/TSV payloads carried up and down.

Storage required: 50

Location	Name	Dimension	Type	Description
1	OPWU	1	R	Orbiter payload weight carried to insertion
2	TPWU	1	R	TSV payload weight carried to insertion
3	OPWD	1	R	Orbiter payload weight at landing
4	TPWD	1	R	TSV payload weight at landing
5	OPLU	1	R	Orbiter payload length at launch
6	TPLU	1	R	TSV payload length at launch
7	OPLD	1	R	Orbiter payload length at landing
8	TPLD	1	R	TSV payload length at landing
9	NORBPL	1	I	Number of payloads on the Orbiter
10	NTUGPL	1	I	Number of payloads on the TSV
11-16	ICORB	6	I	Identification number for each Orbiter payload
17-22	ICTUG	6	I	Identification number for each TSV payload
23-28	IEORB	6	I	An array containing the numerical mission type for each Orbiter payload

Location	Name	Dimension	Type	Description
29-34	IETUG	6	I	An array containing the numerical mission type for each TSV payload
35	IREUSE	1	I	A flag set to indicate that dedicated TSVs are required.
36	NREUSE	1	I	Number of reusable TSVs that are available
37	NTUGS	1	I	Number of TSVs which meet the requirements of the mission
38-47	LTUGS	10	I	Indices of available TSVs which meet the requirements of the mission
48	RCSWT	1	R	Total RCS fuel used in pounds
49	TOTPLU	1	R	Total Orbiter and TSV payload weight at launch, not including the OMS kits or TSV dry weight.
50	TOTPLD	1	R	Total Orbiter and TSV payload weight at landing, not including the OMS kits or TSV dry weight

- COMMON block name: C9

  Description: Labeled COMMON C9 transmits the majority of user options to the MPLS

  Storage required: 54

Location	Name	Dimension	Type	Description
1	YFAIL	1	I	A diagnostic message flag, when set nonzero, causes the reason a mission is infeasible to be printed
2	MM	1	I	Number of feasible missions found for the year under analysis
3-47	NMCLD	45	I	Cumulative total of occurrences for each mission type over all years processed
48	MLIST	1	I	A flag, when set nonzero, compares the payload combinations to the allowable mission types. Unmatched combinations are declared infeasible.
49	NN	1	I	Number of mission type parameters in NMCLD
50	FEASOP	1	I	A flag, when set nonzero, causes the printout of data associated with feasible combinations
51	COSTOP	1	I	Not used
52	STATOP	1	I	A flag, when set nonzero, causes the mission type occurrence statistics for feasible combinations to be generated

Location	Name	Dimension	Type	Description
53	MIXDIS	1	I	A flag, when set nonzero, causes the statistics of the mission types to be printed
54	NOTAB	1	I	A flag, when set nonzero, causes the SCA occurrence table to be printed

- COMMON block name: C10

Description: Labeled COMMON C10 transmits the Orbiter/TSV data to various routines after it has been evaluated by subroutine SEQTST.

Storage required: 25

Location	Name	Dimension	Type	Description
1	POALT	1	R	Altitude of the initial parking orbit
2	POINC	1	R	Inclination of the initial parking orbit
3	ORBDV	1	R	Orbiter total ΔV for the mission
4	TUGDV	1	R	TSV total ΔV for the mission
5	OMSWT	1	R	Weight of the Orbiter propellant needed to form the sequence
6	TUGOMS	1	R	Weight of the TSV propellant needed to form the sequence
7	TTWU	1	R	Total TSV weight at launch
8	TWD	1	R	TSV weight at landing
9	TLU	1	R	TSV length at launch
10	TLD	1	R	TSV length at landing
11	XLMAX	1	R	Length of the cargo bay used for the Orbiter payloads
12-17	IFSORB	6	R	Final sequence of the Orbiter payloads in the combination
18-20	IFSTUG	3	I	Final sequence of the TSV payloads

Location	Name	Dimension	Type	Description
21	IFTUG	1	I	Index of the TSV used to fly this mission
22	DOALT	1	R	Altitude of the orbit at which the TSV is deployed
23	DOINC	1	R	Inclination of the orbit in which the TSV is deployed
24	ROALT	1	R	Altitude of the TSV being retrieved
25	ROINC	1	R	Inclination of the orbit in which the TSV is to be retrieved

- COMMON block name: C11

Description: COMMON C11 transmits the resultant output of subroutines SEQTST and OMSCK to various other modules for output.

Storage required: 17

Location	Name	Dimension	Type	Description
1	PLPOMS	1	R	Theoretical maximum weight the Shuttle can carry to the initial parking orbit
2	WT2ORB	1	R	Maximum payload weight the Shuttle can carry to the initial parking orbit
3	TOTWU	1	R	Total cargo weight at launch; includes payloads OMS kits, etc.
4	TOTWD	1	R	Total cargo weight at landing; includes payloads, OMS kits, etc.
5	PWMARU	1	R	Additional payload weight the Orbiter can carry up
6	PWMARD	1	R	Additional payload weight the Orbiter can carry down
7	PWMARG	1	R	Additional payload weight the Orbiter can carry
8	TOTLU	1	R	Total length of the cargo at launch
9	TOTLD	1	R	Total length of the cargo at landing
10	CLMAX	1	R	Maximum length used by any permutation of the sequence
11	TOTLMX	1	R	Greatest length referenced, either TOTLU, TOTLD, or CLMAX

Location	Name	Dimension	Type	Description
12	PLMARG	1	R	Additional payload cargo length the Orbiter can use for this flight
13	OMSTNK	1	R	Total weight of the OMS fuel used by the Orbiter
14	OMSKIT	1	R	Weight of the OMS fuel carried in the kits
15	WTKITS	1	R	Weight of the dry OMS kits used
16	XLKITS	1	R	Length of the stacked OMS kits used
17	NOKITS	1	R	Number of OMS kits used

- COMMON block name: C12

  Description: Labeled COMMON C12 transmits the mission type code related data to various subroutines.

  Storage required: 55

Location	Name	Dimension	Type	Description
1-9	MSSTYP	9	I	Symbols for the nine mission types considered
10-54	MSCLCD	45	I	List of the allowable mission types for all combinations
55	NMTYP	1	I	Number of allowable mission types in MSCLCD

- COMMON block name: C13

Description: Labeled COMMON C13 transmits IUS TSV data from the payload model to various subprograms of MPLS.

Storage required: 181

Location	Name	Dimension	Type	Description
1-15	TTWT	15	R	An array containing the total weight of the IUS's being used
16-30	TISP	15	R	The specific impulse of each stage
31-45	NTSVA	15	I	The number of stages used on each IUS vehicle
46-60	FUEL	15	R	The total fuel available for each stage
61-75	TTSVWT	15	R	The total weight of each stage
76-150	NUNQS	15×5	I	An array which contains the stage numbers used for an IUS. This array is used to index into TISP, FUEL, TSVLN, and TTSVWT.
151	NTSVS	1	I	The number of IUS vehicles to be used ($NTSVS \leq 10$)
152-166	TSVLN	15	R	The length of each stage
167-181	TLS	15	R	The total length of each IUS being used

- COMMON block name: C14

  Description: Labeled COMMON C14 transmits information pertaining to the payload combination to various subroutines in MPLS.

  Storage required: 30

Location	Name	Dimension	Type	Description
1-6	IB	6	I	Payload sequence as formed from the ICTUG and ICORB array in labeled COMMON C8
7-12	IC	6	I	Payload identification numbers in the combination; the indices are pointers into each array of COMMON C1
13-18	ID	6	I	Discipline mix code; the numbers are pointers into the DISPNM array of COMMON C30
19-24	IE	6	I	Payload mission type; the array is stored as a function of the IC array pointing into the PMT1 array of COMMON C5
25-30	IG	6	A	Alphanumeric payload mission type; the array is stored as a function of the absolute value of the IE array as it points into the MSSTYP array of COMMON C12

- COMMON block name: C15

  Description: Labeled COMMON C15 retains two words used to identify the launch window and whether the WTR is available.

  Storage required: 2

Location	Name	Dimension	Type	Description
1	NWTR	1	I	A flag which indicates, if nonzero, that the WTR is available; otherwise an ETR launch is assumed
2	NYAV	1	I	An integer used to denote the year of availability for WTR launches

- COMMON block name: C17

  Description: Labeled COMMON C17 transmits information to the statistical subprograms.

  Storage required: 1440

Location	Name	Dimension	Type	Description
1-675	MCT	45x15	I	An array which contains the unique mission type codes for all years investigated
676-1350	MCTNO	45x15	I	The accumulative mission class codes for all years under investigation
1351-1395	LOC	45	I	An array of accumulative mission class codes
1396-1440	JORD	45	I	A working array used to store the particular class code for the year under investigation

- COMMON block name: C25

  Description: Labeled COMMON C25 retains the majority of the system constants used by the program.

  Storage required: 19

Location	Name	Dimension	Type	Description
1	UPLBS	1	R	Maximum payload chargeable weight with which the Orbiter can launch
2	DNLBS	1	R	Maximum payload chargeable weight with which the Orbiter can land
3	SBOWT	1	R	Orbiter dry weight
4	OMSINT	1	R	Weight of the OMS fuel in the main tank
5	SPI	1	R	Specific impulse of the Orbiter's OMS engines
6	BAYLN	1	R	Length of the Orbiter cargo bay
7	FTKIT	3	R	Length of the stacked OMS kits; the first word represents one kit; the second is the accumulative length of two kits and the third the accumulative length of three kits
8	WTKIT	3	R	Weight of the stacked OMS kits; the accumulative weights of one, two and three kits
9	RESOMS	1	R	Amount of reserve OMS fuel carried for contingencies
10	REDZDV	1	R	The ΔV required for rendezvous maneuvers
11	RCSRDZ	1	R	Weight of the RCS fuel used for each rendezvous

ORIGINAL PAGE IS
OF POOR QUALITY

Location	Name	Dimension	Type	Description
12	GRAV	1	R	Acceleration due to gravity
13	CORB	1	R	Acceleration of gravity times the specific impulse of the OMS engines
14	RCSCAP	1	R	Total fuel capacity of the RCS system
15	RESRCS	1	R	Reserve RCS fuel

- COMMON block name: C30.

Description: Labeled COMMON C30 transmits information pertaining to the payload type and discipline to various statistics routines.

Storage required: 223

Location	Name	Dimension	Type	Description
1-20	DISPNM	20	A	A list of two-character alphanumeric discipline names allowed
21-120	MIXLST	100	I	The allowed discipline mix codes as pertain to the payload sequence
121-220	MIXCNT	100	I	Cumulative count of the payload discipline mix codes
221	MIXCHK	1	I	A flag, when set nonzero, causes the discipline mix constraint to be applied to a payload sequence
222	NMIX	1	I	Current number of mission types
223	NODIS	1	I	Number of discipline mix combinations considered

- COMMON block name: C33

  Description: Labeled COMMON C33 retains information pertaining to the number of combinations generated.

  Storage required: 7

Location	Name	Dimension	Type	Description
1-6	KCOMB	6	I	An array which indicates the starting numbers of the mission payload sets
7	NUM	1	I	The maximum number of payloads generated in a combination

- COMMON block name: C34

  Description: Labeled COMMON C34 transmits information within the combination generator routines to track the data as it is generated.

  Storage required: 12403

Location	Name	Dimension	Type	Description
1-6200	SETIN	6200	I	Reference set of payload combinations generated
6201-12400	NXCOMB	6200	I	Current set of payload combination generated
12401	M2	1	I	Number of combinations in SETIN
12402	K	1	I	Index of the current payload combination being generated
12403	LAST	1	I	First nonzero combination in SETIN, set to 1

- COMMON block name: C39

  Description: Labeled COMMON C39 transmits the optional print information.

  Storage required: 1

Location	Name	Dimension	Type	Description
1	MKS	1	I	A flag specifying the printed output units =1, the units are mks =2, the units are fps

5.3 SUBPROGRAM DOCUMENTATION

Individual subprogram documentation is given in alphabetical order on the following pages. Functions RANDOM and ZOR are available from the MSC*LOCALIB on EXEC 8. Their documentation is found in reference 4.

Subroutine ADAPTR

Identification

Name/Title	- ADAPTR (payload adapter routine)
Author/Date	- J. Williams, April 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

Purpose

Subroutine ADAPTR computes the payload adapter weight as a function of payload weight.

Usage

- Calling Sequence

  CALL ADAPTR (WT, WT1)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
WT	In/Out	1	R	The payload weight carried to earth parking orbit
WT1	In/Out	1	R	The payload weight at landing

Method

Subroutine ADAPTR uses a linear interpolation technique to compute the payload adapter weight as a function of payload weight. This computation is ignored if WT1 is nonzero when ADAPTGR is entered. The payload weight at landing is initialized as WT, and WT is augmented by the adapter weight.

Subroutine ALLOCT

Identification

Name/Title - ALLOCT (mission allocation routine)

Author/Date - J. Williams, Aug. 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine ALLOCT generates a list of random numbers in ascending order for a specific interval defined in common.

Usage

- Calling Sequence

  CALL ALLOCT (M, LIST, IOPT)

Calling arguments:

Parameters Name	In/Out	Dimension	Type	Description
M	Out	1	I	Number of words stored in LIST
LIST	Out	1	I	A list of random numbers generated in ascending order for a particular interval
IOPT	In	1	I	An initialization flag used as = 1 initialization = 2 causes a list of random numbers to be generated as a function of the intervals specified in COMMON C33

PRECEDING PAGE BLANK NOT FILMED

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C33	1-7	1-6

Method

Subroutine ALLOCT is used to generate a list of random numbers within an interval. The routine solves the problem of reducing the number of feasible missions found by the MPLS for use by the SCA. The list is then sorted into ascending order and checked for redundant numbers.

Restrictions

- Operational

  Function ZOR and subroutine SORTX are required.

Subroutine CGIN

Identification

Name/Tile - CGIN (CG initialization routine)

Author/Date - J. Williams, April 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine CGIN provides an interface between subroutine SEQTST and the CG model.

Usage

- Calling Sequence

CALL CGIN (LMODE, KTUGS, $)

Parameter Name	In/Out	Dimension	Type	Description
LMODE	In	1	J	A dummy flag which is set to unity to insure a proper routine interface from SEQTST
KTUGS	In	1	I	Not used
$N	-	-	-	The statement in the calling program to which control is transferred if an error occurs

- Data In/Out

Labeled COMMON (Refer to the labeled COMMON description section):

Block Name	In	Out
C1	2001-2400, 7801-8000	
C3	All	

Block Name	In	Out
C6	1-30	
C7	2-8	
C8	11-22	
C10	5-21	
C11	15	
C12	1-9	
C25	4	

Method

Subroutine CGIN is used to initialize the arguments for a call to subroutine CGTEST. If it has been determined by CGTEST that the mission failed, then an error return is made.

Subroutine CGLOAD

Identification

Name, Title - CGLOAD (initial loading of payloads in cargo bay for CG model)

Author, Date - E. H. Perrenot, February 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine CGLOAD creates the initial loading sequence of payloads in the shuttle cargo bay prior to the mission events.

Usage

- Calling Sequence

  CALL CGLOAD (NPLDS, ITNL, ISEQ, NTUGPL, ITUG, IPMT, PLDWT, NOMSKT, NSEQ)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
NPLDS	In	1	I	The number of payloads involved in the candidate mission
ITNL	In	1	I	=0, no round-trip payloads in ISEQ requiring a tunnel; =n (≠0), payload number n in ISEQ requires a tunnel
ISEQ	In	Dimensioned in calling program	I	The identification numbers of the payloads involved in the mission (in the order flown)

Parameter name	In/Out	Dimension	Type	Description
NTUGPL	In	1	I	The number of payloads in ISEQ that require a third stage
ITUG	In	Dimensioned in calling program	I	The identification numbers of the payloads requiring a third stage
IPMT	In	Dimensioned in calling program	A	Array containing payload mission types for the payloads in ISEQ: A = attached, S = service, D = deploy, R = retrieve
PLDWT	In	Dimensioned in calling program	R	The weights of the payloads in the payload model plus adapter weights, if required
NOMSKT	In	1	I	The number of OMS kits on board for this mission
NSEQ	Out	Dimensioned in calling program	I	The sequence of payloads as loaded (front to back of the cargo bay)

Method

Subroutine CGLOAD examines all payloads on the candidate mission with regard to payload mission type. If the payload is to be retrieved, whether by the shuttle or by a third stage, it is not considered for loading. All payloads are ordered in the cargo bay from front to rear by increasing weight. The exceptions are that a third stage and its payloads are loaded in the rear and that other payloads are grouped into deploys and round trips, the heavier group loaded in back of the other. In the case of the third stage, it is loaded to the rear of the bay with its payloads stacked immediately in front in the reverse order of deployment. A flag is set in the eighth word of the array NSEQ if OMS kits are present. These kits will be loaded against the rear wall of the cargo bay.

Example:

Flight sequence: (1) deploy third stage (TS) with payloads A and B (B to be deployed first), (2) retrieve C, (3) round trip D, (4) deploy E, and (5) deploy F.

Weights - D - 1,500 lb
E - 700 lb
F - 600 lb

Order in cargo bay, front to rear:
F, E, D, B, A, TS

Restrictions

- Operational
  Subroutine ISORT is required.

Subroutine CGMOV

Identification

Name, Title - CGMOV (shift payloads in payload array)
Author, Date - E. H. Perrenot, February 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

Purpose

Subroutine CGMOV shifts elements in an array when an element is being inserted in the array.

Usage

- Calling Sequence
  CALL CGMOV (IBUF, IDIM, I, J, K)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
IBUF	In/Out	Dimensioned in calling program	I	Array
IDIM	In	1	I	Dimension of IBUF
I	In	1	I	The word in IBUF where the shift is to begin
J	In	1	I	The number of places (words) to shift the array
K	In	1	I	$\leq$ 0, shift to the left; > 0, to the right

PRECEDING PAGE BLANK NOT FILMED

Method

Subroutine CGMOV uses another array to store words from IBUF instead of actually shifting them in IBUF itself. It tests for zeroed words and does not include them in the shift. Thus, if the calling arguments specify a two-word shift to the right, the first word indicated in IBUF is indeed moved two words to the right, but the shifting process will be terminated if two zeroed words are encountered.

Subroutine CGMPLD

Identification

Name, Title	- CGMPLD (multipayload center-of-gravity computation)
Author, Date	- E. H. Perrenot, February 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

Purpose

Subroutine CGMPLD computes the center of gravity for a group of payloads in the shuttle cargo bay.

Usage

- Calling Sequence

  CALL CGMPLD (IPLD, PLDWT, PLDLEN, PLDCG, TUGWT, TUGLEN, F, OMSWT, OMSLEN, D, NSEQXT, TWT, WTXLEN, CG)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
IPLD	In/Out	Dimensioned in calling program	I	Array containing identification numbers of payloads
PLDWT	In	Dimensioned in calling program	R	The weights of the payloads in the payload model plus adapter weights, if required
PLDLEN	In	Dimensioned in calling program	R	Array of payload lengths, corresponding to PLDWT
PLDCG	In	Dimensioned in calling program	R	Array of payload centers of gravity, corresponding to PLDWT

Parameter name	In/Out	Dimension	Type	Description
TUGWT	In	1	R	The dry weight of third stage, if used
TUGLEN	In	1	R	The length of the third stage
F	In	1	R	Weight of the fuel on board the third stage
OMSWT	In	1	R	Total weight of OMS kits, if used
OMSLEN	In	1	R	Length of OMS kits
D	In	1	R	The distance (in feet) from the front of the cargo bay to the first payload in the cargo bay, a function of center-of-gravity constraints in previous mission phases (events)
NSEQXT	In/Out	Dimensioned in calling program	I	Array containing the amount of available space in the cargo bay created by previous payload deployments
TWT	Out	1	R	Total weight of payloads in the cargo bay (plus third stage and fuel, if applicable)
WTXLEN	Out	1	R	For all payloads in the cargo bay, a summation of the following (for each payload): the sum of payload center of gravity and total distance from the front of the bay multiplied by payload weight, used in computing center-of-gravity of retrieval payloads loaded in the rear of the cargo bay.

Parameter name	In/Out	Dimension	Type	Description
CG	Out	1	R	The center of gravity for the group of payloads in IPLD, expressed in feet from the front of the cargo bay

Method

- Model

  Subroutine CGMPLD calculates the distance from the front of the cargo bay to the front of each payload in the bay. This distance includes the sum of the lengths of payloads in front of it as well as the extra space left by deployed payloads. This length is used in the basic center-of-gravity equation:

$$CG = \frac{w_1\left(d + s_1\right) + \sum_{i=2}^{n} w_i\left(d + L_i + s_i\right)}{\sum_{i=1}^{n} w_i}$$

where: CG = center of gravity,

w_i = dry weight of ith payload in the cargo bay,

d = distance to the first payload in the cargo bay (calling argument D),

s_i = center of gravity of ith payload in the cargo bay,

L_i = total length of payloads and available space in front of the ith payload in the cargo bay $\left(L_1 = 0\right)$.

Subroutine CGNSRT

Identification

Name, Title - CGNSRT (inserts payload into gap in cargo bay)

Author, Date - E. H. Perrenot, February 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine CGNSRT checks the cargo bay for an available area in which to insert a retrieval payload after it has been determined by subroutine CGTEST that the payload will not fit in the front or rear of the bay.

Usage

- Calling Sequence

  CALL CGNSRT (NP, ITNL, PLEN, TOTLEN, PWT, TWT, NSEQ, NSEQXT, MIND1, MAXD1, NPINB, $a, $b)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
NP	In	1	I	Identification number of payload to be inserted
ITNL	In	1	I	=0, no round-trip payloads in ISEQ requiring a tunnel; =n (>0), payload number n in ISEQ requires a tunnel.
PLEN	In/Out	1	R	Length of the payload
TOTLEN	In/Out	1	R	Total length of the payloads in the cargo bay

PRECEDING PAGE BLANK NOT FILMED

Parameter name	In/Out	Dimension	Type	Description
PWT	In	1	R	Weight of the payload
TWT	In/Out	1	R	Total weight of the payloads in the cargo bay
NSEQ	In/Out	Dimensioned in calling program	I	The sequence of payloads in the cargo bay (front to back)
NSEQXT	In/Out	Dimensioned in calling program	I	Array containing the amount of available space in the cargo bay created by previous payload deployments
MIND1	In/Out	1	R	The minimum distance from the front of the cargo bay to the first payload
MAXD1	In/Out	1	R	The maximum distance from the front of the cargo bay to the first payload
NPINB	In/Out	1	I	The number of payloads in the cargo bay
$a				Returns to a statement numbered a in the calling program if the payload cannot be loaded
$b				Returns to a statement numbered b in the calling program if it can be loaded

Method

Subroutine CGNSRT searches the array of available space (in the form of "gaps" in the payload bay) from the rear of the cargo bay to the front to determine if the candidate retrieval payload will fit in such a location. If such a space is found, the payload number will replace the flag (-5) in the array NSEQ and the length of the payload is considered to be, for center-of-gravity purposes, that of the gap in the cargo bay that it occupies. This is necessary because the other payloads in the bay cannot be moved about and "squeezed" against the new addition. Actually, the payload is loaded in the forwardmost part of the gap it occupies. When the payload is loaded, the total payload length and weight in the cargo bay is updated, as well as the number of payloads in the bay.

Restrictions

- Operational

  Subroutine CGMOV is required.

Subroutine CGTEST

Identification

Name, Title - CGTEST (main routine for the center-of-gravity model)

Author, Date - E. H. Perrenot, February 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine CGTEST is the driver for the center-of-gravity model. It tests the payloads in the shuttle cargo bay against computed center-of-gravity constraints for all events in a given mission.

Usage

- Calling Sequence

  CALL CGTEST (NPLDS, ISEQ, ITNL, NTUGPL, ITYPE, ITUG, TUGLEN, TUGWT, TUGF, IPMT, PLDCG, PLDWT, PLDLEN, NOMSKT, OMSLEN, OMSWT, IFLAG)

  Arguments:

Parameter Name	In/Out	Dimension	Type	Description
NPLDS	In	1	I	The number of payloads involved in the candidate combination
ISEQ	In	Dimensioned in calling program	I	Array containing the identification numbers of the payloads in the combination (in the order flown)
ITNL	In	1	I	=0, no round-trip payloads in ISEQ requiring a tunnel; =n (>0), payload number n in ISEQ requires a tunnel.

PRECEDING PAGE BLANK NOT FILMED

Parameter name	In/Out	Dimension	Type	Description
NTUGPL	In	1	I	The number of payloads in ISEQ that require a third stage
ITYPE	In	1	I	The type number of the third stage
ITUG	In	Dimensioned in calling program	I	The identification numbers of the payloads requiring a third stage
TUGLEN	In	1	R	Length of the third stage, if used
TUGWT	In	1	R	Dry weight of the third stage
TUGF	In	1	R	Weight of fuel onboard the third stage
IPMT	In	Dimensioned in calling program	A	Array containing payload mission types for the payloads in ISEQ: A = attached, S = service, D = deploy, R = retrieve
PLDCG	In	Dimensioned in calling program	R	The centers of gravity of the payloads in the payload model
PLDWT	In	Dimensioned in calling program	R	Array of payload weights plus adapter weights, if required, corresponding to PLDCG
PLDLEN	In	Dimensioned in calling program	R	Array of payload length corresponding to PLDCG
NOMSKT	In	1	I	The number of OMS kits onboard for this mission
OMSLEN	In	1	R	Length of OMS kits, if used

Parameter name	In/Out	Dimension	Type	Description
OMSWT	In	1	R	Total weight of OMS kits
IFLAG	Out	1	I	> 0, combination passed; = 0, failed loading: = - n, failed in mission event n

- Data In/Out: (refer to the labeled COMMON description):

Block Name	Input	Output
C3	1	

Method

- Model

  Subroutine CGTEST tests the payloads in the cargo bay against center-of-gravity constraints for all events in the mission, starting with initial loading. The events are determined by the payloads in ISEQ and their respective payload mission types in IPMT. First, initial loading is accomplished with a call to subroutine CGLOAD. Then, for the initial loading and for each combination of payloads in the bay following an event, the center of gravity for the group of payloads is determined by subroutine CGMPLD. Two centers of gravity are actually determined, a minimum and maximum with respect to allowable loading in the cargo bay for the event. These centers of gravity are tested against the center-of-gravity constraints as a function of total payload weight in the bay for that event. If neither the minimum nor maximum centers of gravity fall within the allowable constraints, then the mission fails in that event and the calling program is notified by the value in argument IFLAG.

CGTEST also makes provision for the retrieval of a reusable third stage and attached payloads. Reference 5 outlines the center-of-gravity process in detail.

Restrictions

- Operational
  Subroutines CGLOAD, CGMPLD, CONSTR, CGMOV, and CGNSRT are required.

Subroutine COMB

Identification

Name/Title - COMB (combination generator)

Author/Date - J. M. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine COMB generates the possible combinations for MPLS; the feasible combinations are kept and used to generate the remaining combinations.

Usage

- Calling Sequence

  CALL COMB ($N, NPLNYR, MAXPL, IPASS, IMM, K1, IJ)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
$N	In	-	-	Statement number in the calling program to which control is transferred when all the combinations have been generated
NPLNYR	In	1	I	Number of single mission payloads
MAXPL	In	1	I	Maximum number of payloads allowed on a combination

Parameter Name	In/Out	Dimension	Type	Description
IPASS	In/out	1	I	A flag, when set to zero, causes the single mission payloads to be generated; if nonzero, the remaining combinations wil be generated
IMM	Out	1	I	Number of payloads in the current combinations
K1	In/out	1	I	Index number of the current combination; if the combination is rejected, this number is decremented by 1
IJ	Out	6	I	The payload combination. The array is used as a pointer into COMMON C1. The IJ array is stored in the IC array of COMMON C14

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C7	110	
C34		All

Method

Subroutine COMB generates the combination of payloads for MPLS evaluation by using the results of previous combinations. The combinations are generated in sets; each set is generated from the previous set. This scheme allows only those combinations to be generated as a result of successful feasible combinations. For example:

Initial single payloads	Doubles	Triples
1	12*	235
2	13*	
3	14	
4	15*	
5	23	
	24*	
	25	
	34*	
	35*	
	45*	

*Indicates a rejected combination of payloads.

Restrictions

- Operational

  A maximum of 4000 combinations may be generated for each set with M payloads.

  Subroutine REDUNT is required.

Subroutine CONSTR

Identification

Name, Title - CONSTR (center-of-gravity constraint)
Author, Date - E. H. Perrenot, February 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

Purpose

Subroutine CONSTR supplies minimum and maximum allowable centers of gravity for a given total payload weight.

Usage

- Calling Sequence

  CALL CONSTR (TWT, CMIN, CMAX)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
TWT	In	1	R	Total weight of payloads in the cargo bay (plus third stage and fuel, if applicable)
CMIN	Out	1	R	Minimum center-of-gravity constraint (in feet from the front of the cargo bay)
CMAX	Out	1	R	Maximum center-of-gravity constraint

Method

- Model

  Subroutine CONSTR performs a table lookup on cargo center-of-gravity limits (in feet from the front of the cargo bay) as a function of total payload weight. The subroutine stores in

DATA statements the minimum and maximum constraints for various payload weights (500 lb to 65,000 lb, a set of constraints for various payload weights (500 lb to 65,000 lb, a set of constraints every 1,500 lb). For a total payload weight from 0 to 65,000 lb, a linear interpolation is performed to obtain the appropriate set of constraints. A representation of these limits appears in figure 2 of reference 5.

Subroutine CPRNT

Identification

Name/Title - CPRNT (detailed print routine)
Author/Date - J. Williams, March 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

Purpose

Subroutine CPRNT is a special purpose print routine used to display detailed information pertaining to feasible combinations.

Usage

- Calling Sequence
  CALL CPRNT

- Data In/Out:

  Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C2	All	
C8	3	
C10	8	
C11	3-4	
C25	9,15	

Method

Subroutine CPRNT displays information pertaining to the EPS, the missions ΔV requirements, and discrete weight changes.

Subroutine CPTEST

Identification

Name/Title - CPTEST (combination preliminary testing routine).

Author/Date - J. Williams, July 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine CPTEST performs the flight sequence independent tests for a given payload combination.

Usage

- Calling sequence

  CALL CPTEST ($N, IYEAR)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement in the calling program to which control is transferred if an error is indicated
IYEAR	In	1	I	The year that the combination of payloads is to be flown

PRECEDING PAGE BLANK NOT FILMED

- Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	2001-2600, 2801-6400	1-3, 34-35
C2	36-44	45-46
C5	All	
C6	All	
C7	2-110	1
C8	11-34	All
C9	1,3-48,50,53	3-47
C10	12-17	
C12	(Not required)	
C14		All
C17	(Not required)	
C25	All	
C30	221	

Method

Subroutine CPTEST performs flight sequence independent tests and initializes parameters for use in statistical analysis and flight dependent tests. The method is as follows:

1. Payload combination conformity constraint tests are made. The first constraint test is a function of redundant payloads on the same flight; the remaining conformity tests are a function of mission type and the payload discipline mix.

2. Orbiter conformity constraint tests are made to determine if each combination is within Orbiter limits. The Orbiter limits are a function of the cargo bay length, the maximum weight allowed on the TSV, the number of TSV payloads, and number of payloads which require a dedicated TSV.

3. Subroutine SEQTST is called to perform the flight sequence dependent tests.

4. Subroutines STATS and MIXTST are called to tabulate data for statistical analysis of payload mission types and discipline mix.

Restrictions

- Operational

  Subroutines FLYIT, STATIS, SYNTEZ, MIXTST, ERRPRT, SEQTST, and FEASBL are required.

Subroutine DECOMP

Identification

Name/Title - DECOMP (decompose)
SYNTEZ (compose)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Language - FORTRAN V

Purpose

Subroutine DECOMP is used to separate each digit of a multi-digit integer number into a list of single digit integer words; entry point SYNTEZ is used to generate a single integer word from an input list.

Usage

- Calling Sequence

```
CALL DECOMP (N, M, NOYES, NARRAY)
CALL SYNTEZ (N, M, NOYES, NARRAY)
```

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
N	In/out	1	I	Number of words in NARRAY
M	In	1	I	Number of digits each word (NARRAY) will occupy in NOYES
NOYES	In/out	1	I	An integer formed from the integer list NARRAY
NARRAY	In/out	6	I	A list of integers which range in value from 1 to 9

PRECEDING PAGE BLANK NOT FILMED

Method

Subroutine DECOMP uses the MOD function to separate each integer digit from the word NOYES. Entry point SYNTEZ forms the integer word from a list by first sorting the integers into descending order.

Restrictions

- Operational

  Subroutine SORT is required.

  The largest integer that can be decoded is 9 digits.

Card - C-2

Subroutine DISPLY

Identification

Name/Title - DISPLY (display routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine DISPLY prints the payload characteristics and associated information for a feasible mission.

Usage

- Calling Sequence

```
CALL DISPLY (MM, M, IC, IB, NAME1, LAUNCH,
* NTUGPL, TUG, ITUG, NOKITS, PWMARG, PCTUSE, FLOAD,
* POINC, POALT, ALT, XINCI, TOTLU, TOTLD, TOTWU,
* TOTWD, TUGDV, ORBDV, NOYES, IDENT)
```

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
MM	In	1	I	Flight number
M	In	1	I	Numbers of payloads on the flight
IC	In	6	I	Payload numbers which represent the combination
IB	In	6	I	Payload numbers ordered with respect to TSV and Orbiter events
NAME1	In	6	I	Numerical mission type

Parameter Name	In/Out	Dimension	Type	Description
LAUNCH	In	1	I	An integer set to 1 or 2 to indicate an ETR or WTR launch
NTUGPL	In	1	I	Number of TSV payloads
TUG	In	1	I	TSV number used in this mission
ITUG	In	6	I	Payload numbers of the TSV payloads
NOKITS	In	1	I	Number of OMS kits used
PWMARG	In	1	R	Additional payload weight the Orbiter could carry on this flight
PCTUSE	In	1	R	Percentage of the first OMS kit used
FLOAD	In	1	R	Load factor
POINC	In	1	R	Inclination of the first orbit
POALT	In	1	R	Altitude of the first orbit
ALT	In	1	R	Altitude of each payload in the payload combination
XINC	In	1	R	Inclination of each payload in the combination
TOTLU	In	1	R	Total payload length at launch
TOTLD	In	1	R	Total payload length at landing
TOTWU	In	1	R	Total payload weight at launch
TOTWD	In	1	R	Total payload weight at landing

Parameter Name	In/Out	Dimension	Type	Description
TUGDV	In	1	R	Total TSV ΔV
ORBDV	In	1	R	Total Orbiter ΔV
NOYES	In	1	I	(Not used)
IDENT	In	12	A	A list of two-word payload names in the combination

- Data In/Out:

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C3	1	
C10	1	

Method

Subroutine DISPLY prints a table of data related to each feasible combination; the subroutine contains no special computations.

Restrictions

- Operational

  Subroutine CPRNT is required.

Function DLTAV

Identification

Name/Title	- DLTAV (DELTA velocity function 1) DEL (DELTA velocity function 2) DVIPK (ΔV from insertion to the parking orbit) DVDORB (deorbit ΔV to landing)
Author/Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

Purpose

Function DLTAV and its three entry points are used to compute the ΔV requirements for the MPLS. DLTAV computes the ΔV between orbits for the ith and jth payloads. DEL computes the ΔV between orbits for two payloads by specifying their altitudes and inclinations. DVIPK computes the ΔV from insertion to the first parking orbit and DVDORB computes the deorbit ΔV to landing.

Usage

- Calling Sequence

```
X = DLTAV (I,J)
X = DEL (E,B,C,D,·)
X = DVIPK (HH)
X = DVDORB (H,XIN)
```

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
I	In	1	I	Index of the ith payload
J	In	1	I	Index of the jth payload
E	In	1	R	Orbital altitude of the ith payload

PRECEDING PAGE BLANK NOT FILMED

B	In	1	R	Orbital altitude of the jth payload
C	In	1	R	Inclination of the ith payload
D	In	1	R	Inclination of the jth payload
HH	In	1	R	Altitude of the first parking orbit
H	In	1	R	Altitude of the last parking orbit before landing
XIN	In	1	R	Inclination of the last parking orbit before landing

- Data In/Out:

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	2801 - 3200	

Method

Function DLTAV and its three entry points calculate the ΔV requirements for the MPLS. The functions use a Hohmann transfer algorithm at the insertion and orbital phases of the mission; the deorbit ΔV is computed using an empirical equation.

1. ENTRY DVIPK

$$a = \left(r_{p_i} + r_{a_i}\right)/2$$

where

a = semimajor axis

r_{p_i} = perigee radius at insertion

r_{a_i} = apogee radius at insertion

$$V_{p_i} = \sqrt{\frac{\mu \ r_{a_i}}{a \ r_{p_i}}}$$

where

μ = gravitational constant

V_{p_i} = perigee velocity of the insertion orbit

$$a_1 = \left(r_{p_i} + r_{t_e}\right)/2$$

where

a_1 = the semimajor axis of the transfer ellipse

r_{t_e} = the apogee radius of the transfer ellipse

$$V_p = \sqrt{\frac{\mu \ r_{t_e}}{a_1 \ r_{p_i}}}$$

where V_p = the perigee velocity of the transfer ellipse

$$V_a = \sqrt{\frac{\mu \ r_{p_i}}{a_1 \ r_{t_e}}}$$

where V_a = the apogee velocity of the transfer ellipse

$$V_c = \sqrt{\mu / R_{t_e}}$$

where V_c = the circular velocity of the transfer ellipse

and

$$\Delta V_1 = |V_p - V_{p_i}| + |V_c - V_a|$$

where ΔV_1 = the delta velocity required to transfer from insertion to the initial parking orbit

2. Function DLTAV and entry point DEL

These two functions perform identical tasks and differ only in their calling arguments. Function DLTAV computes the ΔV to transfer from the ith payload orbit to the jth payload orbit. DEL permits direct entry of orbital altitudes and inclinations; the transfer goes from the E to B altitude.

$$a = \left(r_{a_1} + r_{a_2}\right)/2$$

where

r_{a_1} = the apogee radius of the initial orbit (E)

r_{a_2} = the apogee radius of the final orbit (B)

a = the semimajor axis of the transfer ellipse

$$\Delta i = |i_1 - i_2|$$

where

i_1 = the inclination of the initial orbit

i_2 = the inclination of the final orbit

If the altitude of the initial orbit is greater than the final orbit, then

$$\Delta i_a = \Delta i$$

and

$$\Delta i_b = 0$$

where

Δi_a = the change in inclination at the first impulse point

Δi_b = the change in inclination at the second impulse point

If the altitude of the initial orbit is less than or equal to the final orbit, then

$$\Delta i_a = 0$$

and

$$\Delta i_b = \Delta i$$

Then

$$V_{c_1} = \sqrt{\mu / r_{a_1}}$$

and

$$V_{e_1} = \sqrt{\frac{\mu \, r_{a_2}}{a \, r_{a_1}}}$$

where

V_{c_1} = the circular velocity at the first impulse point

V_{e_1} = the elliptical velocity at the first impulse point

The delta velocity of the first impulse is computed as

$$\Delta V_1 = \sqrt{\left(V_{c_1} - V_{e_1}\right)^2 + 4V_{c_1}V_{e_1}\left[\sin\left(\frac{i_1}{2}\right)\right]^2}$$

The circular and elliptical velocity of the second impulse point is computed as

$$V_{c_2} = \sqrt{\frac{\mu}{r_{a_2}}}$$

$$V_{e_2} = \sqrt{\frac{\mu\ r_{a_1}}{a\ r_{a_2}}}$$

where

V_{c_2} = the circular velocity of the second impulse point

V_{e_2} = the elliptical velocity of the second impulse point

The total velocity change required for the transfer is computed as

$$\Delta V = \Delta V_1 + \sqrt{\left(V_{c_2} - V_{e_2}\right)^2 + 4V_{c_2}V_{e_2}\left[\sin\left(\frac{i_2}{2}\right)\right]^2}$$

3. Entry DVDORB computes the ΔV required for the deorbit maneuver. The ΔV is computed as a function of inclination and altitude.

For altitudes (H) at the last orbit less than 140 n. mi.

$$\Delta V_1 = -.05 \cdot H + 256.0$$

For altitudes greater than 140 but less than 457 n. mi.

$$\Delta V_1 = 1.4 \cdot H + 52.0$$

For altitudes greater than 457 n. mi

$$\Delta V_1 = 1.443 \cdot H + 32.0$$

If the inclination of the last orbit is greater than or equal to 28.5°, the ΔV as computed is output. Assuming the inclination is less than 28.5 degrees, then

$$\Delta V = \Delta V_1 + 2V_c \sin\left(\frac{C_1 - i}{2C_2}\right)$$

where

C_1 = 28.5 degrees

C_2 = 57.29578, a conversion factor used to convert degrees to radians

V_c = the circular velocity at the altitude of the last orbit

The V_c parameter is computed as

$$V_c = \sqrt{\frac{\mu_e}{R + H \cdot C_3}}$$

where

μ_e = 1.40765392E16, gravitational constant in ft^3/sec^2

R = the radius of the earth

C_3 = 6076.11548, a conversion factor used to convert nautical miles to feet

Restrictions

- Analytical

  1. All orbital ΔV's are computed for circular orbits

  2. All plane change maneuvers are performed at the higher altitude orbit

Subroutine EPSV1

Identification

Name/Tile - EPSV1 (EPS model)

Author/Date - J. Williams, April 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine EPSV1 computes the EPS requirements for MPLS.

Usage

- Calling Sequence

  Call EPSV1 (KMODE, LMOD)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
KMODE	In	1	I	A flag used to compute the initial EPS requirements for a mission
LMOD	In	1	I	An input parameter set to zero if the payload is a TSV retrieve or deploy

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	7401-7800	
C2		All

Block Name	Input	Output
C5	1-100	
C7	2-8	
C8	1,3	
C10	(not used)	

Method

Subroutine EPSV1 computes the EPS requirements of a mission for MPLS. At launch, the orbiter is loaded with all the initial EPS requirements which includes crew/deadweight and payload requirements. The first two EPS kits are charged to the weight of the spacecraft. The use of more than two kits is charged to the payload weight carried up. The use of more than seven kits is charged against the payload bay length at the rate of four feet per pair (the assumption is that two kits are loaded side by side). EPSV1 computes the total EPS used, the payload chargeable EPS weight and kit length, as well as the EPS required for a TSV deploy or retrieve.

Subroutine ERRPRT

Identification

Name/Title - ERRPRT (error printing routine)

Author/Date - J. Williams, Aug. 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine ERRPRT is used to display the diagnostic messages associated with infeasible missions generated by MPLS.

Usage

- Calling Sequence

  CALL ERRPRT (KERR, MIXCOD)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
KERR	In	1	I	The error number to indicate the type of constraint that was violated
MIXCOD	In	1	I	The discipline mix code

- Data In/Out:

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	3201-3400	
C7	2	

C8	10,17-22,49-50
C11	Not used
C14	7-18,25-30

Method

Subroutine ERRPRT is a logic routine that prints a diagnostic as a function of an error number. Refer to section 3.2.2 for the diagnostic messages.

Restrictions

- Operational

  Only 11 diagnostic messages are available.

Subroutine FEACOM

Identification

Name/Title - FEACOM (feasible combination routine)

Author/Date - J. M. Williams, Aug. 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine FEACOM generates a data file of feasible missions by randomly selecting missions from the feasible mission file. The reduced data is used by the SCA to form traffic models.

Usage

- Calling Sequence

  CALL FEACOM (MM, FEASOP, NP)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
MM	In/Out	1	I	The number of feasible missions; MM is the number generated by the MPLS when input; when output it represents the number of missions kept.
FEASOP	-	1	I	Not used
NP	-	1	I	Not used

• Data In/Out

Labeled COMMON (refer to the labeled COMMON descrition section):

Block Name	Input	Output
C33	All	

Method

Subroutine FEACOM generates a reduced list of missions from the feasible mission file for use in the SCA. The missions to be retained are selected by subroutine ALLOCT, which generates a list of missions for a specific interval. Subroutine LOAD1 is called with each mission number to retrieve it from mass storage in order to restore that mission on unit 2. The intervals used to select the mission numbers are obtained from COMMON C33, in the array KCOMB.

Restrictions

• Operational

Subroutines ALLOCT and LOAD1 are required.

FEACOM will limit the number of feasible missions to 500.

Subroutine FEASBL

Identification

Name/Title - FEASBL (feasible mission output routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine FEASBL is used to output data related to a feasible mission both on mass storage and the printer.

Usage

- Calling Sequence

  CALL FEASBL (INCR, ICNT, IYEAR, IEE)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
INCR	In/Out	1	I	A flag used to count the number of missions written to a mass storage file, set to zero before each call.
ICNT	In	1	I	A flag used to indicate if repeated payloads are not on this mission; ignored if zero.
IYEAR	In	1	I	The year that the combination of payloads is to be flown.
IEE	In	6	I	The payload mission types for the combination of payloads.

• Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	1-400,2801-3200	
C5	101-201	
C7	1-2	
C8		
C9	2, 50	
C10	1-4	
C11	2-4, 12, 14	
C12	1-9	
C14	1-12, 19-30	
C15	1	
C25	1-2	
C33		All
C39	All	

Method

Subroutine FEASBL is used to set up data related to a feasible mission for purposes of output. The following procedure is used for any feasible combinations.

1. The load factor for the Orbiter is computed as the maximum of the ratio of total weight up versus the weight to orbit capability and the total weight down versus the maximum down weight allowed.

2. Function ICHARG is referenced to initialize the cost array.

3. The up/down payloads are coded for print.

4. The combination is scrutinized for repeated payloads.

Restrictions

- Operational

  Subroutines LOAD5, FPTOMK, and DISPLY are required.

Subroutine FLYIT

Identification

Name/Title - FLYIT (compatible payloads routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine FLYIT determines if repeated payloads can fly on the same mission.

Usage

- Calling Sequence

  CALL FLYIT (ICNT, ICM1, ICM2, $N, IYEAR, ICM3)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
ICNT	In	1	I	Number of unique repeated payloads found
ICM1	In	6	I	Index of the payload found
ICM2	In	6	I	Number of payloads which have been duplicated for each duplicated payload
$N	-	-		The statement number to which control is passed if an error occurs
IYEAR	In	1	I	A two-digit number which represents the year under analysis

PRECEDING PAGE BLANK NOT FILMED

Parameter Name	In/Out	Dimension	Type	Description
ICM3	In	6x6	I	An array of payload mission types. Each row in the matrix represents a set of mission types for a specific redundant payload.

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	3601-6200	
C5	1-100	

Method

Subroutine FLYIT compares the compatibility of all redundant payloads in a combination to the mission flight parameter, FLTPYP, of the payload model. The following procedure is used for any specific payload:

1. The flight frequency parameter is decoded into a flag and two parameters which represent the number of up/down payloads flown this year. The flag is used to indicate whether up/down payloads can fly together.
2. If the number of payloads of the same name exceed three, the combination is rejected.
3. If the number of redundant payloads of the same name exceed that allowed by the payload model, the combination is rejected.
4. If both up and down payloads of the same name are on the combinations, the flag parameter must be checked to determine if the flight is allowed.

Subroutine FPTOMK

Identification

Name/Title - FPTOMK (foot-pound-second system to meter-kilogram-second system)

Author/Date - H. Chang, March 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine FPTOMK transforms the measurement of output data of MPLS from the English to the metric system.

Usage

- Calling Sequence

  CALL FPTOMK(POALT,ALT,TOTLN,TOTLN1,TOTWTU,TOTWTD,PLMARG, CURDV,TOTDV)

Calling arguments:

Parameter name	In/Out	Dimension	Type	Description
POALT	In/Out	1	R	Altitude of the initial parking orbit
ALT	In/Out	6	R	Orbital altitude of each payload
TOTLN	In/Out	1	R	Total length up
TOTLN1	In/Out	1	R	Total length down
TOTWTU	In/Out	1	R	Total weight up
TOTWTD	In/Out	1	R	Total weight down
PLMARG	In/Out	1	R	Additional payload weight the Shuttle can carry on this flight

Parameter name	In/Out	Dimension	Type	Description
CURDV	In/Out	1	R	Total Shuttle ΔV used
TOTDV	In/Out	1	R	Total TSV ΔV used

Method

Subroutine FPTOMK changes the value of those variables in the calling arguments from the foot-pound-sec system to the meter-kilogram-sec system by multiplying by conversion constants.

Function ICHARG

Identification

Name/Title - ICHARG (cost coefficient routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Function ICHARG computes a selected cost coefficient for a specific feasible payload combination.

Usage

- Calling Sequence

J(I) = ICHARG (I, FLOAD)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
I	In	1	I	Index of the cost coefficient being computed
FLOAD	In	1	R	Load factor, the ratio of the total Orbiter weight up to its capability, or the down weight to its capability.

- Data In/Out

Labeled COMMON (refer to the labeled COMMON description):

Block Name	Input	Output
C10	5	
C11	7, 11	
C25	6	

Method

Function ICHARGE computes one of 12 integer cost coefficients of a mission for use in the SCA. The technique used allows the initialization or computation of a coefficient based on the input argument I. The function has the following meanings for I:

I = 1, the function is set to unity

I = 2, the function is a three-digit integer, representing the percentage of the load factor not used

I = 3, the function represents the integer value of the OMS weight

I = 4, the function represents the integer value of the payload margin

I = 5, the function is a three-digit integer representing the unused cargo bay length versus the cargo bay length

I = 6 to 12, the function is set to unity (reserved for additional capability)

The function scales the parameters for options 2 and 5 by multiplying the real number by 1000 because SCA assumes integer cost coefficients.

Subroutine ISORT

Identification

Name, Title	- ISORT (sort array and rearrange additional array(s))
Author, Date	- E. H. Perrenot, November 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

Purpose

Subroutine ISORT sorts an array in either ascending or descending order and rearranges up to three arrays in the same sequence.

Usage

- Calling Sequence

  CALL ISORT (A, N, B, C, D, K, SWITCH)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
A	In/Out	N	R	Array to be sorted
N	In	1	I	The number of words in array A (and B, C, and/or D, if used)
B C D	In/Out	N	R	Arrays to be rearranged
K	In	1	I	0 = sort A only; 1 = rearrange B; 2 = rearrange B and C 3 = rearrange B, C, and D
SWITCH	In	1	I	> 0, sort will be in ascending order; $\leq$ 0, in descending order

Method

Subroutine ISORT uses a binary search technique to reorder array A. If the options for K are exercised, B, C, and/or D will be rearranged in the same sequence as A. When a test is made between two elements of the A array to determine which is larger, a switch is made depending upon whether the sort is in ascending or descending order. If a switch is made, the same respective elements in B, C, and/or D are switched.

Subroutine IUSDV

Identification

Name/Title - IUSDV (IUS ΔV routine)
INSTG (initialize IUS vehicles)

Author/Date - J. Williams, April 22, 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine IUSDV is used to compute the ΔV for a specific IUS which was predefined by entry point INSTG.

Usage

- Calling Sequence

```
CALL IUSDV (L, WT, TDV, V)
CALL INSTG (IOUT)
```

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
L	In	1	I	The index of the IUS being used
WT	In	1	R	The payload weight carried on the IUS
TDV	Out	1	R	The total IUS ΔV required
V	Out	Dimensioned in the calling program	R	An array of ΔV's computed for each stage of the IUS
IOUT	Out	1	I	The number of IUS vehicles which may be used

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C6	91-105	1-46, 61-75
C13	1-166	167-181

Method

Entry point INSTG computes the weight and length of an IUS vehicle defined by the payload model; refer to table 3.1-I. The computation is performed by a summation of all IUS stage weights and lengths defined for a specific IUS.

Subroutine IUSDV uses the total loaded weight of the IUS to compute the ΔV requirements for each stage. The equation used to compute the ΔV requirement is a form of the ideal rocket equation and is:

$$\Delta V = g \cdot Isp \cdot \ell n \left(W_I / W_F \right)$$

where

g - the acceleration due to gravity

Isp - the specific impulse of the stage

W_I - the vehicle weight before the burn

W_F - the vehicle weight after the burn

The fuel requirement for each stage is known; therefore, the problem solution is simplified.

Subroutine IUSOPT

Identification

Name/Title - IUSOPT (IUS option routine)

Author/Date - J. Williams, April 1976

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine IUSOPT determines feasibility of a mission for a specific IUS vehicle, given the weight of the payload to be carried.

Usage

- Calling Sequence

  CALL IUSOPT (HA1, HA2, XINC1, XINC2, PYLDWT, TW, ISN, TL, TUGDV, $)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
HA1	In	1	R	The altitude of the circular orbit at which the IUS and its payloads are deployed from the shuttle.
HA2	In	1	R	The altitude of the payload final circular orbit. If HA1 < 0 then HA2 is the C3 energy required rather than altitude.
XINC1	In	1	R	The inclination of the IUS deployment orbit
XINC2	In	1	R	The desired inclination

Parameter Name	In/Out	Dimension	Type	Description
PYLDWT	In	1	R	The weight of the payload
TW	Out	1	R	The weight of the IUS
ISN	In	1	I	The index of the IUS used
TL	Out	1	R	The length of the IUS
TUGDV	Out	1	R	The ΔV required by the IUS
$N	–			A statement number in the calling program to which control is transferred when an error occurs.

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C3	1	
C13	1-15, 167-181	

Method

Subroutine IUSOPT verifies whether a payload(s) can be flown on a specific IUS. The logic optionally allows the use of C_3 energy cases or permits the user to specify altitudes and inclinations. The method is as follows:

1. Subroutine IUSDV is called to compute the IUSΔV available.

2. The Hohmann transfer ΔV is computed.

3. The ΔV from each stage of the IUS is summed until it exceeds the transfer ΔV, or until there are no more stages. If the IUS stage ΔV is less than the transfer ΔV, the case fails.

4. Next, the transfer ΔV for the second burn is computed by readjusting the plane change.

5. If the remaining IUS stages are within 10 ft/sec of the transfer ΔV, the case is converged. If not, the initial

plane change is modified and the program transfers to step 2. The problem is considered infeasible if 200 iterations have been exceeded.

This basic form of the Hohmann equations used are given in the documentation of subroutine DLTAV.

Restrictions

- Operational

  Subroutine IUSDV is required.

Subroutine LOAD1

Identification

Name/Title - LOAD1 (unload and retrieve routine)

Author/Date - J. Williams, Aug. 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine LOAD1 is used to retrieve a feasible mission record by specifying a mission number; the data is used by subroutine FEACOM to reduce the number of feasible missions.

Usage

- Calling Sequence

  CALL LOAD1 ($N, II, LIST)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
$N	In	-	-	The statement number in the calling program to which control is passed if an error occurs
II	In	1	I	The number of the flight to be retrieved
LIST	Out	Dimensioned in the calling program	I	The output record

PRECEDING PAGE BLANK NOT FILMED

Method

Subroutine LOAD1 is used to retrieve a record from a blocked record containing 10 logical records. The record number to be retrieved is compared to the first word of each logical record; if a match was not found, another block record is retrieved and the process is repeated. Once a record has been located, the data are stored in the calling argument and return is made to the calling routine. However, if the record does not exist, a nonstandard return is made to the calling routine.

Restrictions

- Operational

  Unit 1 must be assigned.

Subroutine LOAD5

Identification

Name/Title - LOAD5 (output record blocking routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine LOAD5 stores 10 logical records into a buffer before transferring the data to mass storage.

Usage

- Calling Sequence

```
CALL LOAD5 (MM, M, IC, IB, NAME1, LAUNCH, NTUGPL,
*  IFTUG, ICTUG, NOKITS, PWMARG, PCTUSE,
*  FLOAD, POALT, POINC, ALT, XINC1, TOTLU,
*  TOTLD, TOTWU, TOTWD, TUGDV, ORBDV, IG,
*  ICC, IDENT, IOPT)
```

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
MM	In	1	I	Flight number
M	In	1	I	Number of payloads on the flight
IC	In	6	I	Payload numbers comprising the combination
IB	In	6	I	Final payload sequence
NAME1	In	6	I	Numerical mission type
LAUNCH	In	1	I	An integer set to 1 or 2 to indicate ETR or WTR

Parameter Name	In/Out	Dimension	Type	Description
NTUGPL	In	1	I	Number of TSV payloads
TUG	In	1	I	TSV number used in this mission
ITUG	In	6	I	Payload numbers of the TSV payloads
NOKITS	In	1	I	Number of OMS kits used
PWMARG	In	1	R	Additional payload weight the Orbiter could carry on this flight
PCTUSE	In	1	R	Percentage of the first OMS kit used
FLOAD	In	1	R	Load factor
POINC	In	1	R	Inclination of the first orbit
POALT	In	1	R	Altitude of the first orbit
ALT	In	1	R	Altitude of each payload in the payload combination
XINC	In	1	R	Inclination of each payload in the combination
TOTLU	In	1	R	Total payload length at launch
TOTLD	In	1	R	Total payload length at landing
TOTWU	In	1	R	Total payload weight at launch
TOTWD	In	1	R	Total payload weight at landing
TUGDV	In	1	R	Total TSV ΔV
ORBDV	In	1	R	Total Orbiter ΔV

Parameter Name	In/Out	Dimension	Type	Description
IG	In	6	A	Array of alphabetic mission types
ICC	In	1	I	Cost coefficients
IDENT	In	12	A	List of two-word payload names
IOPT	In	1	I	A flag, when set nonzero, causes the routine to store the internal buffer's contents on mass storage

Method

Subroutine LOAD5 stores the contents of each logical record it receives into a buffer. Once the buffer contains 10 logical records, it is transferred to mass storage and the buffer is zero filled. Since the total number of records will not always be a multiple of 10, an option is available to transfer any remaining information to mass storage. The option is exercised only after all the feasible missions have been generated, and immediately before a call is made to subroutine FEACOM.

Restrictions

- Operational

  Unit 1 must be assigned.

Subroutine MIXTST

Identification

Name/Title	- MIXTST (discipline mix testing routine) - MXSTAT (display discipline frequency routine)
Author/Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

Purpose

Subroutine MIXTST is used to generate a list of up to 100 payoad disciplines; entry point MXSTAT is used to display the frequency of occurrence of each discipline mix.

Usage

- Calling Sequence

  CALL MIXTST ($N, KK, MIXCOD)

  CALL MXSTAT

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement number in the calling program to which control is passed if an error occurs
KK	In	1	I	A flag, when set nonzero causes the discipline mix to be checked against a list in COMMON
MIXCOD	In	1	I	The discipline mix to be verified or stored

PRECEDING PAGE BLANK NOT FILMED

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C30	1-221	21-220

Method

Subroutine MIXTST cumulates numeric payload discipline mix codes (PDMC) for use by CPTEST as a constraint. The information pertaining to the PDMC is displayed from MXSTAT from subroutine MPLS after the analysis of a particular year has been completed.

Restrictions

- Operational

  Subroutine DECOMP is required.

Subroutine MPLS

Identification

Name/Title - MPLS (the MPLS executive routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine MPLS is the executive routine to control the initialization of payload data, to generate combinations referencing the payload numbers, and to cause each combination to be tested for feasibility.

Usage

- Calling Sequence

  CALL MPLS (IYEAR, MAXLP)

Calling arguments:

Parameter Name	InOut	Dimension	Type	Description
IYEAR	In	1	I	The last two digits of the year under analysis
MAXLP	In	1	I	The maximum number of payloads allowed on a combination

• Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	3401-3600	
C5	1-100,202	
C6	61-90	
C7	1-109	
C9	2	
C10	(Not used)	
C11	(Not used)	
C15	2	

Method

Subroutine MPLS is the executive for the MPLS. It uses the following procedure:

1. The number of TSVs which are available for use are determined by examining their year of availability.
2. Subroutine PLIST is referenced to generate a working list of payloads.
3. Subroutine PLONTG is referenced to determine which payloads in the list require TSVs.
4. Subroutine COMB is called to generate a unique payload combination.
5. Subroutine SORTL is referenced to arrange the payload combination in order of ascending altitude.

6. A call to subroutine CPTEST is made to perform the sequence dependent and independent test.

7. Once all the combinations have been generated, subroutine FEACOM is called. generate a list of missions.

8. If required, the number of feasible combinations is reduced to 500 or less.

Subroutines STATIS and MXSTAT are called to output tables related to the discipline mix and mission types.

Restrictions

- Operational

  Subroutines RESET, PLIST, PLONTG, COMB, SORTL, CPTEST, LOAD5, TIME, FEACOM, STATIS, and MXSTAT are referenced.

Subroutine OMSCK

Identification

Name/Title - OMSCK (OMS checking routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine OMSCK is used to determine the number and weight of the OMS kits plus fuel for a flight sequence. Also, the total weight-to-orbit capability is computed and compared to the total weight of the vehicle at insertion.

Usage

- Calling Sequence:

  CALL OMSCK

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C8	1, 3	
C10	2, 5, 7-10	5
C11		All
C25	1	

Method

Subroutine OMSCK computes various parameters associated with the Orbiter OMS and OMS kits; the data are used to adjust the weights and length parameters required for the payload sequence. The following procedure is used:

1. The weight-to-orbit capability is calculated by an empirical equation as a function of the orbital inclination.

2. The minimum OMS weight, the abort weight, is initialized as a function of the launch inclination for either ETR or WTR.

3. The weight of the OMS fuel carried in the kits is computed as the difference between the OMS required for the mission and the capacity of the main OMS tank.

4. The number of OMS kits required is a function of any residual OMS fuel which cannot be carried in the main OMS tanks.

5. The additional OMS fuel required to carry the dry weight of the kits from insertion to landing is computed and added to the total OMS weight.

6. The weight to orbit is recomputed as the minimum of the weight-to-orbit capability or 65,000 pounds.

7. The up/down parameters for weight, length, and the payload margin are computed.

Subroutine PERM

Identification

Name/Title - PERM (permutation generator)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine PERM generates a permutation of a set of N variables taken N at a time. One permutation is generated for each call.

Usage

- Calling Sequence

```
CALL PERM (N, IEND, IPERM, ICNT)
```

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
N	In		I	Number of variables in the permutation
IEND	In/Out		I	An initialization flag = 0, initialize the permutation generator ≠ 0, generate the next permutation
IPERM	Out	Dimensioned in the calling program	I	A list which contains the new permutation
ICNT	Out	Dimensioned in the calling program	I	An array of counters used to generate a permutation

Method

Each permutation is formed by a cyclic permutation of all or part of a previous permutation. The logic is structured such that a single call to PERM generates a permutation.

Subroutine PLIST

Identification

Name/Title	- PLIST (payload list routine)
Author/Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

Purpose

Subroutine PLIST searches the payload model and selects a list which can be flown in a particular year.

Usage

- Calling Sequence

  CALL PLIST (IYEAR)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
IYEAR	In	1	I	The last two digits of the year under analysis

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	3401-6200,8201	
C5		1-363
C9	50	

Method

Subroutine PLIST forms a working list of payloads from the payload model by eliminating those payloads which do not fly in the year under analysis.

In addition, payloads that are repeated or those which are up/down payloads are identified. The technique used is based on the definition of the variable FLTPYR which is described in detail in the labeled COMMON description section.

A distinction is made between nominal payloads and up/down payloads by using negative numbers to represent the mission type.

Subroutine PLONTG

Identification

Name/Title - PLONTG (payload on TSV routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine PLONTG determines the TSV requirements for payloads to be flown in a particular year.

Usage

- Calling Sequence

  CALL PLONTG

- Data In/Out

  Labeled COMMON (refer to labeled COMMON description section):

Block Name	Input	Output
C1	2001-2600, 2801-3400, 8001-8200	
C5	1-100, 202, 264-363	364-463
C8		1, 3, 5, 7
C9	50	
C10		1-3, 4
C11		7
C25	10-11, 13, 15	

Method

Subroutine PLONTG is used to determine which payloads in the working list require a TSV. The following procedure is used to determine if a TSV is required:

1. If the dedicated TSV velocity parameter associated with that payload is nonzero, a TSV is required.

2. If the payload's orbit exceeds 700 n. mi., a TSV is required.

3. If the payload when flown alone requires more than three OMS kits or has a payload margin less than zero, a TSV is required.

If all the conditions are met, then a 1 is stored in the NEEDTG array.

Restrictions

- Operational

  Functions DVIPK, DVEL, DVDORB and subroutine OMSCK are required.

Subroutine REDUNT

Identification

Name/Title - REDUNT (redundancy check routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine REDUNT is used to eliminate payload combinations that have an infeasible subset.

Usage

- Calling Sequence

  CALL REDUNT ($N)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement number in the calling program to which control is transferred if an error occurs

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C34	All	

Method

Subroutine REDUNT evaluates a candidate payload combination by comparing a subset of the combination to the previous generated set in the feasible mission file. The technique eliminates the second payload from the combination and compares it to a previously generated set, for example:

Candidate combination	Temporary set	Feasible missions previous set
ABCDE	ABCE	ABCD ABCE ACDE ACDF ADFG BCDE

Since the combination was generated by adding the last payload, E, then the subset ABCE must exist before the combination can be successful. This method avoids complex testing of a combination which has an infeasible subset.

Subroutine SEQTST

Identification

Name/Title - SEQTST (payload sequence testing routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine SEQTST performs the flight sequence dependent tests for a given payload combination.

Usage

- Calling Sequence

  CALL SEQTST ($N)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement in the calling program to which control is transferred if the combination is infeasible.

- Data In/Out

  Labeled COMMON (Refer to the labeled COMMON description section):

Block Name	In	Out
C1	2001-2600, 2801-3600 7201-7800	
C2	2-3, 40	4-32
C3	1	2-10
C6	1-60, 91-105	
C7	111	
C8	1-47	
C10	1-6	2, 7, 11-25
C11	7, 12	
C25	3, 9, 10-13	

Method

Subroutine SEQTST verifies whether or not a payload sequence is feasible. The logic optionally allows one or all permutations of a particular combination considering all available TSVs. The method is as follows:

1. The combination is divided into those payloads which are required on the TSV and the Orbiter.
2. If payloads are to be carried in the Orbiter, then the ΔV's to each orbit and the bay length required are computed.
3. If payloads are to be carried on the TSV, then the TSV ΔV's to each orbit and the weight changes are computed. The weight calculations are performed backwards from the last maneuver to the first, yielding the total TSV weight required for launch. The fuel required by the TSV is computed as

$$W_{TOMS} = W_{TOT} - W_{TSV} - W_{PLDS}$$

where

W_{TOMS} = Total TSV OMS fuel requirement

W_{TOT} = Total TSV weight at launch

W_{TSV} = Dry weight of the TSV

W_{PLDS} = Weight of the TSV payloads

The weight change at each maneuver is computed as

$$W_S = W_E\left(e^{\Delta V/g \cdot T_{isp}} - 1\right)$$

where

W_E = Total weight at the end of the maneuver

g = Acceleration due to gravity

T_{isp} = TSV engine's specific impulse

ΔV = TSV delta velocity requirement

4. The deorbit and launch ΔV's must be computed for the Orbiter. The ΔV's are used to compute OMS fuel requirements based on weight at landing and weight changes at each maneuver. The total weight of the vehicle at launch is

$$W_L = W_V + W_{RCS} + W_{OMS} + W_{KITS} + W_{CRW} + W_{EPS} + PL_D$$

where

W_V = Dry weight of the vehicle

W_{RCS} = Weight of the RCS fuel

W_{OMS} = Weight of the OMS fuel

W_{KITS} = Dry weight of the kits used

W_{CRW} = EPS deadweight and crew requirements not charged to payload weight

W_{EPS} = EPS kit dry weight

PL_D = Weight of the Orbiter payloads carried up

The weight change at each maneuver is computed as

$$W_S = W_E \left(e^{\Delta V / g \cdot I_{sp}} - 1 \right)$$

where

W_E = Weight of the vehicle at the end of the maneuver

ΔV = ΔV required for the maneuver

g = Acceleration due to gravity

I_{sp} = OMS engine's specific impulse

5. A normal program RETURN is made if parameters fall within the Orbiter limits as they are computed. If the constraints have been violated, subroutine PERM will generate the next unique permutation and program control is passed to step 1 or 3. Once all the permutations have been attempted and the constraints are still violated, a nonstandard return is made to the calling program.

Restrictions

- Operational

  Subroutines PERM, IUSOPT, CGIN, and OMSCK and functions DVEL, DLTAV, DVIPK, DVDORB are required.

Subroutine SORT

Identification

Name/Title - SORT (descending sort module)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine SORT sorts a list of N integers into descending order.

Usage

- Calling Sequence

  CALL SORT (N, X, Y)

Calling arguments:

Parameter Name	In/Out	Dimensions	Type	Description
N	In	1	I	Number of variables to be sorted
X	In	Dimensioned in the calling routine	I	List of integers to be sorted
Y	Out	Dimensioned in the calling routine	I	List of integers sorted in descending order

Method

Subroutine SORT arranges a list in descending order by searching the list to find the maximum number in the array; the tests are repeated N times.

Subroutine SORTL

Identification

Name/Title - SORTL (ascending order sort)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine SORTL sorts the HA array into ascending order and rearranges the IA array accordingly.

Usage

- Calling Sequence

```
CALL SORTL (M, IJ, HA, IA)
```

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
M	In	1	I	Number of variables to be sorted
IJ	In	Dimensioned in the calling program	I	An array which is stored into IA
HA	In/Out	Dimensioned in the calling program	I	An array sorted into ascending order
IA	Out	Dimensioned in the calling program	I	The IJ array arranged in an order corresponding to the sorted HA array

PRECEDING PAGE BLANK NOT FILMED

Method

Subroutine SORTL arranges the HA array into ascending order by incrementing through the list to determine if it was less than the previous minimum value. If the ith word is greater than the previous word, the word values are switched and a counter is decremented. The procedure continues until all the words have been tested. Since each element of the IA array corresponds to the HA array, it is arranged in an order corresponding to the sorted HA array.

Subroutine SORTX

Identification

Name/Title - SORTX (sort ascending order)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine SORTX sorts a list of M integers into ascending order.

Usage

- Calling Sequence

  CALL SORTX (M, IA)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
M	In	1	I	Number of variables to be sorted
IA	In/Out	Dimensioned in the calling program	I	Array to be sorted into ascending order

Method

Subroutine SORTX arranges the IA array into ascending order by incrementing through the list to determine if it was less than the previous minimum value. If the i^{th} word was greater than the previous word, the word values are switched and a counter is decremented. The procedure continues until all the words have been tested.

Subroutine STATS

Identification

Name/Title - STATS (mission type status routine)

Author/Date - J. Williams, August 1975

Machine Identification - UNIVAC 1110

Source Language - FORTRAN V

Purpose

Subroutine STATS uses the mission type list to reject payload combinations as a function of the mission types.

Usage

- Calling Sequence

  CALL STATS ($N, IB, M, INCR, MATCH, ISTART, IEND, IALT)

Calling arguments:

Parameter Name	In/Out	Dimension	Type	Description
$N	In	-	I	Statement number in the calling program to which control is transferred if an error occurs
IB	In	6	I	Mission type numbers of the payload sequence being evaluated
M	In	1	I	Number of elements in IB
INCR	In	1	I	Number of missions generated with the same sequence; the repeated missions stored
MATCH	Out	1	I	A flag which denotes if nonzero that the mission types in IB correspond to the mission class codes in COMMON C12.

Parameter Name	In/Out	Dimension	Type	Description
ISTART	In	1	I	First nonzero word in the mission type list
IEND	In	1	I	Last nonzero word in the mission type list
IALT	In	1	I	Not used

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C9	3-49	
C12	1-54	

Method

Subroutine STATS uses an internal allowable list of mission-type codes to constrain candidate missions. If a candidate missions-type's code doesn't correspond to the MSCLCD array in COMMON, then it is rejected. Otherwise the mission is accepted:

5.4 SAMPLE INPUT/OUTPUT

5.4.1 SAMPLE INPUT

The job stream given below indicates the operations necessary to execute the MPLS. To gain a clear understanding of the input, consult the user's guide (reference 2).

Card Image	Description
@RUN JWLXIA	Run card
@USE SAMPLE., FM3-L71194*SAMPLE.	Specifies an internal file name for an external file name
@XQT SAMPLE.SAMPLE	Starts execution
@ADD SAMPLE.DATAIS	Adds the payload model to the run
1	Selects the display option
7	Requests all displays
2	Allows the selection of an analysis type
1	Selects the MPLS only option
80	Specifies the year of analysis (1980)
8	No data base options
5	Terminate
@FIN	Sign off the system

5.4.2 SAMPLE OUTPUT

The printed output generated by the input data described in Section 5.4.1 follows.

```
@XQT SAMPLE
 INPUT TUG CHARACTERISTICS AND MISSION MODEL DATA:
 (FOR EXAMPLE @ADD SAMPLE.DATA99)
 SELECT AN OPTION: ( 3 TO TERMINATE )
 *********************  MISSION MODEL DISPLAY  **********************
```

NO.	PAYLOAD DISCIPLINE	PAYLOAD ID		NAME
1	ASTRONOMY	AS-1A CDR	A	LOW EARTH ORBIT EXPLORER
2	ASTRONOMY	AS-03 CDR	A	SOLAR PHYSICS SATELLITE
3	PHYSICS	PH-1A CDR	A	EXPLORER-UPPER ATMOSPHERE
4	PHYSICS	PH-1B CDR	A	EXPLORER-MEDIUM ALTITUDE
5	PHYSICS	PH-04 CDR	A	HELIO AND INTERSTEL. SPACECRAFT
6	PHYSICS	PH-5V CDR	A	REVISIT
7	LUNAR	LU-03 CDE	A	LUNAR ROVER
8	LIFE SCI.	LS-C1 LCR	A	LIFE SCIENCE RESEARCH MODULE
9	EARTH OBS.	EO-5D LCE	A	SPECIAL PURPOSE SATELLITE-D
10	N NASA/DOD	NN-01 CDR	A	INTEL SAT.
11	N NASA/DOD	NN-05 CDR	A	FOREIGN COMMUNICATIONS
12	N NASA/DOD	NN-09 CDR	A	FOREIGN SYN. MET. SATELLITE
13	N NASA/DOD	NN-11 LCR	A	LOW ORBIT EARTH RES.
14	ASTRONOMY	AS-10K30	S	STELLAR
15	PHYSICS	PH-6D	S	HIGH ENERGY ASTROPHYSICS

NO.	HP N.MI.	DHPP N.MI.	DHPM N.MI.	HA N.MI.	DHAP N.MI.	DHAM N.MI.	DELTAV
1	297.	0.	0.	297.	0.	0.	0.
2	270.	0.	0.	270.	0.	0.	0.
3	847.	0.	0.	847.	0.	0.	0.
4	4028.	0.	0.	4028.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.	20000.
6	200.	0.	0.	200.	0.	0.	0.
7	0.	0.	0.	0.	0.	0.	20000.
8	300.	0.	0.	300.	0.	0.	0.
9	400.	0.	0.	400.	0.	0.	0.
10	19323.	0.	0.	19323.	0.	0.	0.
11	19323.	0.	0.	19323.	0.	0.	0.
12	19323.	0.	0.	19323.	0.	0.	0.
13	300.	0.	0.	300.	0.	0.	0.
14	162.	0.	0.	162.	0.	0.	0.

ORIGINAL PAGE IS
OF POOR QUALITY

15 120. 0. 0. 120. 0. 0. 0.

NO.	INCL DEG.	DINCP DEG.	DINCM DEG.	LAUNCH LENGTH, FT.	LAUNCH WT. INCL. ADAPTER	ADAPTER WT., LB.	PMT
1	28.5	.0	.0	12.2	640.0	640.0	3
2	28.5	.0	.0	13.1	4146.0	4146.0	3
3	90.0	.0	.0	13.3	1046.0	1046.0	3
4	28.5	.0	.0	12.8	852.0	852.0	3
5	28.5	.0	.0	10.5	635.0	635.0	3
6	28.5	.0	.0	5.0	3500.0	3500.0	2
7	28.5	.0	.0	24.0	8700.0	8700.0	3
8	28.5	.0	.0	13.0	682.0	682.0	3
9	90.0	.0	.0	9.7	676.0	676.0	3
10	.0	.0	.0	12.2	4498.0	4498.0	3
11	.0	.0	.0	12.2	982.0	982.0	3
12	.0	.0	.0	10.3	807.0	807.0	3
13	97.0	.0	.0	36.0	6213.0	6213.0	3
14	28.5	.0	.0	55.0	42702.0	31190.0	1
15	28.5	.0	.0	27.0	20720.0	18138.0	1

FLIGHTS PER YEAR

NO.	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991
1	0-0	1-1	1-2	1-1	1-0	1-0	1-0	1-0	1-1	1-1	1-1	1-1	1-1
2	0-0	1-1	0-0	1-1	0-0	1-1	0-0	1-1	0-0	1-1	0-0	1-1	0-0
3	0-0	0-0	0-0	0-0	0-0	1-1	0-0	0-0	0-0	1-0	1-1	1-1	1-1
4	0-0	0-0	1-0	0-0	0-0	1-1	0-0	0-0	0-0	1-0	1-1	1-1	1-1
5	0	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0	0	1	1	1	1
7	0	0	0	0	0	0	0	0	1	1	0	0	0
8	0-0	2-2	2-2	2-2	2-2	2-2	2-2	2-2	2-2	2-2	2-2	2-2	2-2
9	0	0	0	0	0	0	1	0	0	1	0	0	1
10	0-0	0-0	0-0	0-0	2-0	3-0	2-0	2-0	0-1	0-1	2-2	3-3	2-0
11	0-0	0-0	1-0	1-0	1-0	1-3	1-1	1-0	1-0	1-1	1-1	1-1	1-1
12	0-0	0-0	1-0	1-0	0-0	1-1	0-1	1-1	0-0	1-1	0-0	1-1	0-0
13	0-0	0-0	0-0	0-0	1-2	1-1	1-1	1-1	1-1	1-1	1-1	1-1	1-1
14	0	0	0	0	0	0	0	0	0	1	1	1	1
15	0	0	1	1	1	1	1	1	1	1	1	1	1

SEQUENCE NO.,PAYLOAD ID,PMT
1 1-3 2 1-4 3 2-3 4 2-4 5 8-3 6 8-4

DUPLICATED PAYLOADS
PAYLOAD ID,TIMES DUPLICATED
8 2 8 2
SEQ,PAYLOAD NO,TUG
1 1 0 2 1 0 3 2 0 4 2 0 5 8 0
6 8 0

FLT. NO. 1 LAUNCH SITE: ETR

PAYLOADS: AS-1A CDR A
1000

SHUTTLE SEQUENCE 1-0
ALTITUDE 297.
INCLINATION 28.5
TOTAL LENGTH UP: 12. TOTAL WEIGHT UP: 640.0

PAYLOAD MARGIN: 64360.   LOAD FACTOR: .00985
SHUTTLE DELTAV: 1241.

FLT. NO. 2   LAUNCH SITE: ETR

PAYLOADS: AS-1A CDR A
1050
SHUTTLE SEQUENCE 1-R
ALTITUDE 297.
INCLINATION 28.5
TOTAL LENGTH DOWN: 12.   TOTAL WEIGHT DOWN: 640.0
PAYLOAD MARGIN: 64360.   LOAD FACTOR: .00985
SHUTTLE DELTAV: 1261.

FLT. NO. 3   LAUNCH SITE: ETR

PAYLOADS: AS-03 CDR A
2000
SHUTTLE SEQUENCE 2-D
ALTITUDE 270.
INCLINATION 28.5
TOTAL LENGTH UP: 13.   TOTAL WEIGHT UP: 4146.0
PAYLOAD MARGIN: 60854.   LOAD FACTOR: .06378
SHUTTLE DELTAV: 1113.

FLT. NO. 4   LAUNCH SITE: ETR

PAYLOADS: AS-03 CDR A
2050
SHUTTLE SEQUENCE 2-R
ALTITUDE 270.
INCLINATION 28.5
TOTAL LENGTH DOWN: 13.   TOTAL WEIGHT DOWN: 4146.0
PAYLOAD MARGIN: 60854.   LOAD FACTOR: .06378
SHUTTLE DELTAV: 1133.

FLT. NO. 5   LAUNCH SITE: ETR

PAYLOADS: LS-01 LCP A
8001
SHUTTLE SEQUENCE 8-D
ALTITUDE 300.
INCLINATION 28.5
TOTAL LENGTH UP: 13.   TOTAL WEIGHT UP: 682.0
PAYLOAD MARGIN: 64318.   LOAD FACTOR: .01049
SHUTTLE DELTAV: 1255.

FLT. NO. 6   LAUNCH SITE: ETR

PAYLOADS: LS-01 LCP A
8002
SHUTTLE SEQUENCE 8-D
ALTITUDE 300.
INCLINATION 28.5
TOTAL LENGTH UP: 13.   TOTAL WEIGHT UP: 682.0
PAYLOAD MARGIN: 64318.   LOAD FACTOR: .01049
SHUTTLE DELTAV: 1255.

```
FLT. NO.    7      LAUNCH SITE: ETR

 PAYLOADS:    LS-01    LCR  A
                8051
 SHUTTLE SEQUENCE         8-R
 ALTITUDE                 300.
 INCLINATION              28.5
 TOTAL LENGTH DOWN:  13.      TOTAL WEIGHT DOWN:    682.0
 PAYLOAD MARGIN: 64318.    LOAD FACTOR:  .01049
 SHUTTLE DELTAV:  1275.

FLT. NO.    8      LAUNCH SITE: ETR

 PAYLOADS:    LS-01    LCR  A
                8052
 SHUTTLE SEQUENCE         8-R
 ALTITUDE                 300.
 INCLINATION              28.5
 TOTAL LENGTH DOWN:  13.      TOTAL WEIGHT DOWN:    682.0
 PAYLOAD MARGIN: 64318.    LOAD FACTOR:  .01049
 SHUTTLE DELTAV:  1275.

FLT. NO.    9      LAUNCH SITE: ETR

 PAYLOADS:    AS-03    CDP  A   AS-1A    CDP  A
                2000      1000
 SHUTTLE SEQUENCE         2-D          1-D
 ALTITUDE                 270.         297.
 INCLINATION              28.5         28.5
 TOTAL LENGTH UP:  25.      TOTAL WEIGHT UP:  4786.0
 PAYLOAD MARGIN: 60214.    LOAD FACTOR:  .07363
 SHUTTLE DELTAV:  1241.

FLT. NO.   10      LAUNCH SITE: ETR

 PAYLOADS:    AS-03    CDR  A   AS-1A    CDP  A
                2050      1000
 SHUTTLE SEQUENCE         2-R          1-D
 ALTITUDE                 270.         297.
 INCLINATION              28.5         28.5
 TOTAL LENGTH UP:  12.      TOTAL LENGTH DOWN:  13.
 TOTAL WEIGHT UP:    640.0   TOTAL WEIGHT DOWN:  4146.0
 PAYLOAD MARGIN: 60854.    LOAD FACTOR:  .06378
 SHUTTLE DELTAV:  1261.

FLT. NO.   11      LAUNCH SITE: ETR

 PAYLOADS:    AS-1A    CDP  A   LS-01    LCP  A
                1000      8001
 SHUTTLE SEQUENCE         1-D          9-D
 ALTITUDE                 297.         300.
 INCLINATION              28.5         28.5
 TOTAL LENGTH UP:  25.      TOTAL WEIGHT UP:  1322.0
 PAYLOAD MARGIN: 63678.    LOAD FACTOR:  .02034
 SHUTTLE DELTAV:  1255.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
FLT. NO.   12      LAUNCH SITE: ETR

 PAYLOADS:    AS-1A     CDR  A    LS-01     LCR  A
                 1000      8051
 SHUTTLE SEQUENCE        1-D          8-R
 ALTITUDE                297.         300.
 INCLINATION             28.5        28.5
 TOTAL LENGTH UP:   12.     TOTAL LENGTH DOWN:  13.
 TOTAL WEIGHT UP:    640.0   TOTAL WEIGHT DOWN:   682.0
 PAYLOAD MARGIN: 64318.     LOAD FACTOR:   .01049
 SHUTTLE DELTAV:   1275.

FLT. NO.   13      LAUNCH SITE: ETR

 PAYLOADS:    AS-03     CDR  A    AS-1A     CDR  A
                 2000      1050
 SHUTTLE SEQUENCE        2-D          1-R
 ALTITUDE                270.         297.
 INCLINATION             28.5        28.5
 TOTAL LENGTH UP:   13.     TOTAL LENGTH DOWN:  12.
 TOTAL WEIGHT UP:   4146.0   TOTAL WEIGHT DOWN:   640.0
 PAYLOAD MARGIN: 60854.     LOAD FACTOR:   .06378
 SHUTTLE DELTAV:   1261.

FLT. NO.   14      LAUNCH SITE: ETR

 PAYLOADS:    AS-03     CDR  A    AS-1A     CDR  A
                 2050      1050
 SHUTTLE SEQUENCE        2-R          1-R
 ALTITUDE                270.         297.
 INCLINATION             28.5        28.5
 TOTAL LENGTH DOWN:  25.     TOTAL WEIGHT DOWN:  4786.0
 PAYLOAD MARGIN: 60214.     LOAD FACTOR:   .07363
 SHUTTLE DELTAV:   1281.

FLT. NO.   15      LAUNCH SITE: ETR

 PAYLOADS:    AS-1A     CDR  A    LS-01     LCR  A
                 1050      8002
 SHUTTLE SEQUENCE        1-R          8-D
 ALTITUDE                297.         300.
 INCLINATION             28.5        28.5
 TOTAL LENGTH UP:   13.     TOTAL LENGTH DOWN:  12.
 TOTAL WEIGHT UP:    682.0   TOTAL WEIGHT DOWN:   640.0
 PAYLOAD MARGIN: 64318.     LOAD FACTOR:   .01049
 SHUTTLE DELTAV:   1275.

FLT. NO.   16      LAUNCH SITE: ETR

 PAYLOADS:    AS-1A     CDR  A    LS-01     LCR  A
                 1050      8052
 SHUTTLE SEQUENCE        1-R          8-R
 ALTITUDE                297.         300.
 INCLINATION             28.5        28.5
 TOTAL LENGTH DOWN:  25.     TOTAL WEIGHT DOWN:  1322.0
 PAYLOAD MARGIN: 63678.     LOAD FACTOR:   .02034
 SHUTTLE DELTAV:   1295.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
FLT. NO.   17      LAUNCH SITE: ETR

PAYLOADS:    AS-03      CDR  A    LS-01     LCR   A
                 2000        8001
SHUTTLE SEQUENCE          2-D          8-D
ALTITUDE                  270.         300.
INCLINATION               28.5        28.5
TOTAL LENGTH UP:   26.      TOTAL WEIGHT UP:   4828.0
PAYLOAD MARGIN: 60172.      LOAD FACTOR:   .07428
SHUTTLE DELTAV:  1255.

FLT. NO.   18      LAUNCH SITE: ETR

PAYLOADS:    AS-03      CDP  A    LS-01     LCR   A
                 2000        8051
SHUTTLE SEQUENCE          2-D          8-R
ALTITUDE                  270.         300.
INCLINATION               28.5        28.5
TOTAL LENGTH UP:   13.      TOTAL LENGTH DOWN:   13.
TOTAL WEIGHT UP:   4146.0    TOTAL WEIGHT DOWN:    682.0
PAYLOAD MARGIN: 60854.      LOAD FACTOR:   .06378
SHUTTLE DELTAV:  1275.

FLT. NO.   19      LAUNCH SITE: ETR

PAYLOADS:    AS-03      CDP  A    LS-01     LCP   A
                 2050        8002
SHUTTLE SEQUENCE          2-R          8-D
ALTITUDE                  270.         300.
INCLINATION               28.5        28.5
TOTAL LENGTH UP:   13.      TOTAL LENGTH DOWN:   13.
TOTAL WEIGHT UP:    682.0    TOTAL WEIGHT DOWN:   4146.0
PAYLOAD MARGIN: 60854.      LOAD FACTOR:   .06378
SHUTTLE DELTAV:  1275.

FLT. NO.   20      LAUNCH SITE: ETR

PAYLOADS:    AS-03      CDP  A    LS-01     LCR   A
                 2050        8052
SHUTTLE SEQUENCE          2-R          8-R
ALTITUDE                  270.         300.
INCLINATION               28.5        28.5
TOTAL LENGTH UP:    5.      TOTAL LENGTH DOWN:   31.
TOTAL WEIGHT UP:   2713.4    TOTAL WEIGHT DOWN:   7178.0
NO. OF KITS NEEDED: 1
                             % USE OF FIRST KIT:    3.04
PAYLOAD MARGIN: 57822.      LOAD FACTOR:   .11043
SHUTTLE DELTAV:  1295.

FLT. NO.   21      LAUNCH SITE: ETR

PAYLOADS:    AS-03      CDP  A    AS-1A     CDP   A   LS-01     LCR   A
                 2000        1000        8001
SHUTTLE SEQUENCE          2-D          1-D          8-D
ALTITUDE                  270.         297.         300.
INCLINATION               28.5         28.5         28.5
```

```
TOTAL LENGTH UP:  38.      TOTAL WEIGHT UP:  5468.0
PAYLOAD MARGIN: 59532.     LOAD FACTOR:  .08412
SHUTTLE DELTAV:  1255.

FLT. NO.   22      LAUNCH SITE: ETR

PAYLOADS:    AS-03     CDR  A    AS-1A     CDP  A    LS-01     LCR  A
                2000        1000        8051
SHUTTLE SEQUENCE           2-D           1-D            8-R
ALTITUDE                   270.          297.           300.
INCLINATION                28.5          28.5           28.5
TOTAL LENGTH UP:  25.       TOTAL LENGTH DOWN:  13.
TOTAL WEIGHT UP:  4786.0     TOTAL WEIGHT DOWN:   682.0
PAYLOAD MARGIN: 60214.      LOAD FACTOR:  .07363
SHUTTLE DELTAV:  1275.

FLT. NO.   23      LAUNCH SITE: ETR

PAYLOADS:    AS-03     CDR  A    AS-1A     CDP  A    LS-01     LCP  A
                2050        1000        8002
SHUTTLE SEQUENCE           2-R           1-D            8-D
ALTITUDE                   270.          297.           300.
INCLINATION                28.5          28.5           28.5
TOTAL LENGTH UP:  25.       TOTAL LENGTH DOWN:  13.
TOTAL WEIGHT UP:  1322.0     TOTAL WEIGHT DOWN:  4146.0
PAYLOAD MARGIN: 60854.      LOAD FACTOR:  .06378
SHUTTLE DELTAV:  1275.

FLT. NO.   24      LAUNCH SITE: ETR

PAYLOADS:    AS-03     CDR  A    AS-1A     CDP  A    LS-01     LCR  A
                2050        1000        8052
SHUTTLE SEQUENCE           2-R           1-D            8-R
ALTITUDE                   270.          297.           300.
INCLINATION                28.5          28.5           28.5
TOTAL LENGTH UP:  17.       TOTAL LENGTH DOWN:  31.
TOTAL WEIGHT UP:  3405.7     TOTAL WEIGHT DOWN:  7178.0
NO. OF KITS NEEDED: 1
                            % USE OF FIRST KIT:    3.48
PAYLOAD MARGIN: 57822.      LOAD FACTOR:  .11043
SHUTTLE DELTAV:  1295.

FLT. NO.   25      LAUNCH SITE: ETR

PAYLOADS:    AS-03     CDP  A    AS-1A     CDP  A    LS-01     LCR  A
                2000        1050        8001
SHUTTLE SEQUENCE           2-D           1-R            8-D
ALTITUDE                   270.          297.           300.
INCLINATION                28.5          28.5           28.5
TOTAL LENGTH UP:  26.       TOTAL LENGTH DOWN:  12.
TOTAL WEIGHT UP:  4828.0     TOTAL WEIGHT DOWN:   640.0
PAYLOAD MARGIN: 60172.      LOAD FACTOR:  .07428
SHUTTLE DELTAV:  1275.

FLT. NO.   26      LAUNCH SITE: ETR

PAYLOADS:    AS-03     CDP  A    AS-1A     CDP  A    LS-01     LCR  A
```

```
              2000       1050       8051
SHUTTLE SEQUENCE        2-D        1-R           8-R
ALTITUDE                270.       297.         300.
INCLINATION             28.5       28.5         28.5
TOTAL LENGTH UP:  18.     TOTAL LENGTH DOWN:  30.
TOTAL WEIGHT UP:  6933.6   TOTAL WEIGHT DOWN:  3672.0
NO. OF KITS NEEDED: 1
                           % USE OF FIRST KIT:   3.66
PAYLOAD MARGIN: 56066.    LOAD FACTOR:  .10667
SHUTTLE DELTAV:  1295.

FLT. NO.   27    LAUNCH SITE: ETR

PAYLOADS:   AS-03    CDP  A   AS-1A    CDP  A   LS-01    LCP  A
              2050       1050       8002
SHUTTLE SEQUENCE        2-R        1-P           8-D
ALTITUDE                270.       297.         300.
INCLINATION             28.5       29.5         28.5
TOTAL LENGTH UP:  18.     TOTAL LENGTH DOWN:  30.
TOTAL WEIGHT UP:  3449.9   TOTAL WEIGHT DOWN:  7136.0
NO. OF KITS NEEDED: 1
                           % USE OF FIRST KIT:   3.50
PAYLOAD MARGIN: 57864.    LOAD FACTOR:  .10978
SHUTTLE DELTAV:  1295.

    INFEASIBLE MISSION:      2-R    1-R    8-R
        NO PAYLOADS ON TUG
THE RCSWT IS GREATER THAN THE CAPACITY FOR THIS CASE
******************   STATISTICAL ANALYSIS FOR 1980   ******************
TOTAL NUMBER OF COMBINATIONS GENERATED:          40
      NUMBER OF FEASIBLE COMBINATIONS:           27
      NUMBER OF INFEASIBLE COMBINATIONS:         13

TOTAL ELAPSED TIME:          477
      (ALL TIMES ARE IN MILLISECONDS)
      AVERAGE TIME PER FEASIBLE COMBINATION:        17
      AVERAGE TIME PER GENERATED COMBINATION:        11
* TIME IN FEACOM *          40
MISSION TYPE ANALYSIS

                    FREQUENCY OF OCCURANCE             DISCIPLINE MIX

CHOOSE COST COEFFICIENT FOR EACH FLIGHT:

1980     OCCURRENCE TABLE
    PAYLOAD     MISSIONS
 1)   1000     1    9   10   11   12   21   22   23   24
 2)   1050     2   13   14   15   16   25   26   27
 3)   2000     3    9   13   17   18   21   22   25   26
 4)   2050     4   10   14   19   20   23   24   27
 5)   8001     5   11   17   21   25
 6)   8002     6   15   19   23   27
 7)   8051     7   12   18   22   26
 8)   8052     8   16   20   24
 ** MAX NO. SINGLES =     8
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
OCCURRENCE TABLE AND SCA INTERFACE REQUIRED     76 MILLISECONDS
SELECT AN OPTION: ( 3 TO TERMINATE )
```

6. REFERENCES

1. Babb, G. R.: Space Shuttle Performance Capabilities; JSC IN 71-FM-350; Sept. 1971.

2. Williams, J. M.: SAMPLE User's Guide (Revision 2); Document No. LEC-6642, Lockheed Electronics Company, Houston Aerospace Systems Division; Houston, Texas; August 1975.

3. Gonzales, L.: Validation of the Mission Payloads (MPLS) of the Scheduling Algorithm for Mission Planning and Evaluation (SAMPLE), FM33 (75-74); NASA, Houston, Texas; May 1975.

4. Anon.: IDSD Procedures Manual - MSC, Part 20 (Revision 0), MSC; Oct. 1973.

5. FM3/Chief, Mission Analysis Branch, Proposed Design of a Center of Gravity and Geometric Fit Constraint Check in SAMPLE, JSC Memorandum no. FM34 (75-122), Aug. 20, 1975.