

AFIT/GSO/ENS/91D-05

AD-A243 653



DTIC
ELECTE
DEC 27 1991
S C D

1

VERIFICATION AND VALIDATION OF THE
COMPREHENSIVE OPERATIONAL SUPPORT
EVALUATION
MODEL FOR SPACE

THESIS

Lawrence A. Cooper, Captain, USAF

AFIT/GSO/ENS/91D-05

Approved for public release; distribution unlimited

91-19020

91 12 24 0 0

AFIT/GSO/ENS-91D-05

VERIFICATION AND VALIDATION OF THE COMPREHENSIVE
OPERATIONAL
SUPPORT MODEL FOR SPACE

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Space Operations

Lawrence A. Cooper, B.S., M.B.A..
Captain, USAF

Accession For	
REF ORAL	<input checked="" type="checkbox"/>
DTIC Tab	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

December 1991

Approved for public release; distribution unlimited

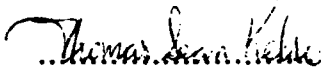
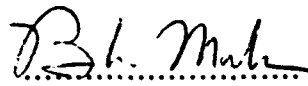
THESIS APPROVAL

STUDENT: Capt Lawrence A. Cooper

CLASS: GSO-91D

THESIS TITLE: VERIFICATION AND VALIDATION OF THE COMPREHENSIVE
OPERATIONAL SUPPORT MODEL FOR SPACE

DEFENSE DATE: 22 November 1991

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Major Thomas S. Kelso/ENS	
Co-Advisor	Major Bruce W. Morlan/ENC	

Preface

The purpose of this paper was to verify and validate the COSEMS for the US Air Force Space Systems Division at Los Angeles Air Force Base, California. The goal was to go over the code developed by PRC, Inc. and ascertain how well it performs and how well it models spacecraft launches, spacecraft in orbit, space support concepts, and spacecraft mission planning.

Since this verification and validation was performed at the very end of the model's development, extensive time was spent getting acquainted with the program's source code and the Ada programming language. To ease the burden of examining over 100,000 lines of code, the scope of the study was narrowed to those functions most essential to the model: random number generation, reliability, orbital mechanics, and mission planning.

A lesson to be gleaned from this paper is that the process of verification and validation should begin when the development of the model starts. By performing verification and validation throughout the design implementation of the model, defects and omissions can be more readily found. Law (17) and Sargent (25) make this plain and detail the necessary steps.

This analysis was made infinitely easier by the cooperation of Dr. Ron Janz, Frank Cheng, Dillap Vallabh, and David Luders at PRC, Inc. Through their unceasing willingness to answer my questions on the nature of their model, the implementation of the code, and the location of specific parts of the code, I was able to complete this study. Finally, I would like to thank my wife Mary for marrying me despite my bringing ten pounds of COSEMS listings with us on our honeymoon.

Lawrence A. Cooper

Table of Contents

	Page
Preface	iii
Table of Contents	iv
List of Figures.....	vi
List of Tables.....	viii
List of Acronyms	ix
Abstract	x
I. Introduction	1
Background	1
Problem Statement.....	3
Sub-Objectives.....	3
Scope.....	4
II. Literature Review.....	5
Simulation.....	5
Random Number Generation.....	7
Verification and Validation	10
Reliability.....	13
Probability Distributions.....	14
Series and Parallel Circuits.....	16
Orbital Mechanics.....	18
Orbit Perturbations	23
Orbit Transfer.....	28
Rendezvous Windows.....	31
Mission Planning.....	34
COSEMS	36
COSEMS Functions and Capabilities.....	38
COSEMS Assumptions and Limitations.....	42

	Page
III. Methodology.....	45
Random Number Generation.....	46
Frequency Tests	46
Randomness Tests.....	46
Parameter Evaluation.....	51
Reliability.....	53
Orbital Mechanics.....	54
Orbit Transfer	55
Mission Planning	56
COSEMS Functions and Capabilities	57
IV. Results and Analysis	58
Random Number generation	58
Parameter Evaluation.....	62
Reliability.....	63
Orbital Mechanics.....	72
Orbital Model.....	72
Orbit Transfer	73
Mission Planning	78
COSEMS Functions and Capabilities	79
Graphics and Output	81
COSEMS Assumptions and Limitations.....	84
V. Conclusions and Recommendations.....	86
Conclusions	86
Recommendations	87
Further Research.....	89
Appendix A: COSEMS Source Files Used for Verification & Validation	91
Appendix B: Bi-Elliptic Transfer Code.....	93
Appendix C: Sample COSEMS Output File	105
Bibliography.....	124
Vita	126

List of Figures

Figure	Page
1. Ways to Study a System	6
2. Relationships of Validation, Verification, and Establishing Credibility	12
3. The Bath Tub Curve.....	13
4. Exponential Distribution Characteristics	15
5. Weibull Distribution Characteristics	15
6. Example of Series and Parallel Blocks	17
7. Reliability Block Diagrams	18
8. Kepler's Elliptical Orbit Geometry.....	20
9. True Anomaly Relation to Mean Anomaly.....	21
10. Orbital Elements and Orbit Geometry	22
11. Effect of the Earth's Oblateness.....	24
12. Hohmann Transfer.....	29
13. Bi-Elliptic Transfer	30
14. Geometry for Rendezvous Windows.....	31
15. Plane Change Through Angle θ	34
16. COSEMS Inputs and Outputs	37
17. COSEMS Support Concept Options.....	40
18. Availability in COSEMS	41
19. Frequency Distribution of Random Numbers from Seed #1	59
20. Frequency Distribution of Random Numbers from Seed #10.....	60

Figure	Page
21a. SLAM II Reliability Program Flowchart.....	65
21b. SLAM II Reliability Program Flowchart.....	66
21c. SLAM II Reliability Program Flowchart.....	68
22. Plane Change Comparison for the Bi-Elliptic Transfer.....	77
23. Availability as a Function of Time	82
24. Proposed Availability Plot.....	83

List of Tables

Table	Page
1. Attributes of Different Pseudorandom Number Generators.....	9
2. Validation Techniques	12
3. Orbital Elements.....	22
4. Perturbative Terms for the Classical Orbit Elements.....	26
5. Mission Planning Activities/Questions	35
6. P -Values from Statistical Analysis of the COSEMS Random Number Generator	61
7. Average and Overall P-Values For COSEMS's Random Number Streams.....	61
8. Parameter Evaluation	62
9. Satellite Subsystem Failure Parameters.....	64
10. Satellite Failures and Availability Comparison	69
11. Comparison of SLAM II and COSEMS Reliability Output	70

List of Acronyms

Acronym	Meaning
BFT.....	Bulk Fuel Tanker
COSEMS.....	Comprehensive Operational Support Evaluation Model for Space
ETR.....	Eastern Test Range
GPS.....	Global Positioning System
LOP.....	Long-Term Orbit Predictor
OMV.....	Orbital Maneuver Vehicle
ORU.....	Orbital Replacement Unit
SASS.....	Space Asset Support System
SBSP.....	Space-Based Support Platform
SC.....	Spacecraft
SDS.....	Space Defense System
STS.....	Space Transportation System
TOF.....	Time-Of-Flight
TT&C.....	Telemetry, Tracking, and Control
V&V.....	Verification and Validation
WTR.....	Western Test Range

Abstract

This study details the verification and validation (V&V) of the Comprehensive Operational Support Evaluation Model for Space (COSEMS). COSEMS is an Ada-based simulation which models spacecraft constellation support concepts such as support from the ground and on-orbit support. While the model is intended for use in analyzing Strategic Defense System concepts, it can easily evaluate non-military satellite constellations. The V&V was confined to a subset of the over 200 subprograms which comprise COSEMS. This subset covered random number generation, reliability, orbital mechanics, and mission planning. The study used traces and comparison to other models to perform the V&V. An input/output analysis was also performed to ascertain the ease of use of COSEMS and the utility of its output. The analysis showed that the areas under investigation performed according to the model and that the model approximated real-world behavior except for orbital motion. The part of the model governing orbital perturbations due to the non-spherical earth omitted rotation of the line-of-apsides. The analysis also revealed that the Ada code and the input/output format are highly machine dependent, which restricts the program from coming into widespread use and limits the usefulness of the output.

VERIFICATION AND VALIDATION OF THE COMPREHENSIVE OPERATIONAL SUPPORT EVALUATION MODEL FOR SPACE

1. Introduction

Background

As part of the Strategic Defense Initiative, several alternatives to the current abandon/replace and on-orbit spare activation methods of satellite constellation maintenance have been proposed (9:143). These alternative concepts include using orbital maneuver vehicles (OMVs) to replace electronics modules and replenish satellites' coolant and fuel systems either by maneuvering an OMV from an on-orbit supply depot or by launching it from ground-based depots. The constellations served by such systems can involve a thousand or more satellites in different orbital inclinations and altitudes and must be maintained in a perpetual high state of readiness (5:v). Because of the high cost involved in building and launching satellites into orbit (upwards of \$5000/lb to launch a satellite into orbit) and the complex orbital dynamics involved in moving satellites from one orbit to another, the different Strategic Defense System (SDS) architectures under consideration may be best served by a particular logistical support concept. Since constellations of such size have never before been deployed and operated as a single system, and the modular designs which will allow on-orbit repair are still under development, the Air Force commissioned the development of the Comprehensive Operational Support Evaluation Model for Space (COSEMS), an Ada program designed to analyze the logistical support

required to maintain a Strategic Defense System (9:143).¹ The current version of COSEMS is Version 5.1. The model's output and predictions have not been verified and validated independently from its developers. Testing must be accomplished for the users to have confidence in the program's output.

According to its documentation, COSEMS models the individual satellites that comprise a user-defined set of constellations, orbital maneuver vehicles (OMVs), space-based support platforms (SBSPs) for on-orbit support, and the launch vehicles used to put satellites into orbit. COSEMS also models the mission planning, as well as the launch vehicle and payload preparation activities. The model ties these systems together by simulating the operation and maintenance of the satellites, the operation of OMVs and SBSPs, and the use of expendable launch vehicles. COSEMS simulates the discrete failure events of modules in individual satellites based upon specific reliability parameters and schedules support missions to repair or replace those satellites. The actual support type varies depending on the support concept being modeled—abandon/replace, on-orbit support, ground-based support, and specific variations of these concepts. The program accounts for the complex dynamics of objects in orbit about the earth and reports on the number of failure events, the status of the constellation at user-specified intervals, and the number and types of resources consumed in supporting the failed satellites (5:3-1).

It should be noted that while COSEMS has been developed for evaluating SDS support concepts, it can potentially perform support analysis for a wider variety of space system concepts. Currently COSEMS can be used to evaluate non-SDS concepts, but the analysis would be a best-case analysis given the differences between a SDS and a space system with lower mission requirements.

¹ Abandon/replace refers to launching a replacement satellite from the ground in lieu of attempting to repair or refuel the malfunctioning satellite. On-orbit spare activation involves storing the replacement satellites in a parking orbit instead of on the ground. When a satellite fails, the spare is activated and moved into position to take up the duties of the failed satellite.

Problem Statement

Before the Consolidated Operational Support Evaluation Model for Space can be used to aid in deciding between different, multi-billion dollar space logistical support concepts, its users in the Air Force and the Department of Defense must have a basis for confidence in the output of this complex simulation model. This requires verification that COSEMS's code and its implementation of mission planning, reliability, and orbit/orbital transfer maneuver concepts are correct. COSEMS must be shown to have valid assumptions and that its orbital mechanics, reliability, and mission planning functions provide output consistent with the actual behavior of space systems. To solve this problem, appropriate verification and validation techniques for COSEMS must be selected and applied to the model to accurately evaluate how well the code performs and how meaningful the output will be to choosing between different, costly, space support concepts. Favorable evaluation will build the user's confidence in the output of the program for use in comparing and selecting between competing orbital support concepts.

Sub-Objectives

In order to perform the verification and validation (V&V) of COSEMS, the source code was installed on a VAX/VMS computer system. Running the program with its default cases and using the COSEMS Design Document and COSEMS Users Guide provided familiarity with the program and its use. These runs allowed an inspection of the utility and ease of use of the user interface, the simulation functions, and the output.

Several case studies were developed for use as benchmarks in verifying and validating COSEMS. These cases consisted of several different constellations with varied orbital parameters and design constraints which tested the limits of the simulation. Varying the orbital characteristics revealed how well the orbital model worked, while varying the design constraints revealed how well the mission planning and reliability function worked.

COSEMS is composed of over two hundred subprograms. The function of these sub-programs can be classified into three areas: discrete-event simulation (reliability), constellation architecture and support control (orbital mechanics), and scheduling (mission planning). The main functions of the reliability, orbital mechanics, and mission planning modules were verified and validated to determine whether the model's output is accurate and useful for its intended purpose of aiding in the evaluation of different support concepts.

Scope

Due to the the large size of COSEMS (over 100,000 lines of code) the entire program was not verified and validated. Only the particular subprograms which perform critical functions necessary for accurate simulation were inspected and tested. These subprograms include event generation, orbit propagation, launch vehicle selection, and orbital transfer. The names of the specific subprograms are listed in Appendix A. COSEMS's output files were evaluated for their ability to convey useful information to the user. All the program files and procedures dealing with COSEMS's preprocessor menus and post-processor graphics generation were not reviewed, verified, or validated.

This paper is not intended as an in-depth study of how COSEMS works and what it does, only as an evaluation of how well it works and its utility to a user. The *COSEMS Design Document* (5) and *COSEMS Users Guide* (6) are required reading for a full understanding of COSEMS's capabilities, functions, and use.

II. Literature Review

Introduction

The Comprehensive Operational Support Evaluation Program (COSEMS) can be thought of as three discrete sections which work together to simulate an entire space support system: a reliability module, an orbital mechanics module, and a mission planning module. As a discrete-event simulation program, COSEMS determines when failure events occur, the failed satellite's location in orbit, and which logistical assets will be used to repair or replace the failure. COSEMS then determines the time until the repair will be completed, the total assets consumed in performing the repair, and then performs the repair/replacement according to the schedule.

Simulation

Simulations use computers to imitate processes, operations, or events of special interest (17:1). A model is a mathematical or logical representation of the rules and assumptions which describe how the process of interest works. Simulation models are extremely useful in providing detailed representation of complex events and their interaction with other events and the world. Many processes can be determined exactly through the application of basic and advanced mathematic principles, however the complex nature of many processes and their interaction with their environment and other processes, operations, and events makes it difficult or impossible to exactly predict their behavior.

As shown by Figure 1 below, many different methods exist for studying systems. A system is a collection of parts or processes which act and interact to accomplish some defined goal. Researchers may experiment with the actual system or with some sort of model of the system. If the system is small or easily managed, one may actually work with a system, for example an electrical device or an automobile. However for abstract, very small, very large, spatially, or temporally spread out systems such as financial processes,

machinery, or chemical processes, it is easier to work with scale models or abstract mathematical models. Machinery lends itself to actual physical experimentation, but complex processes such as economic systems or atomic interactions require mathematical modeling.

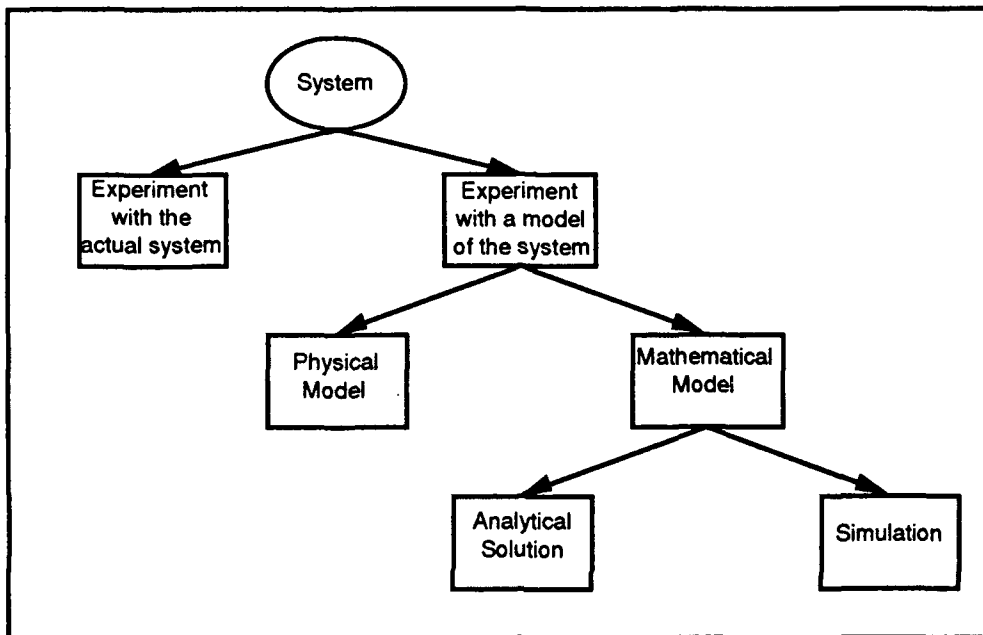


Figure 1. Ways to Study a System (17:4)

For simple models or for those in which relationships are determined precisely, analytical calculations provide exact solutions to the problem. However, complex analytical solutions require vast computing resources. Examples of such solutions are fuel consumption or the prediction of aerodynamic flow about an aircraft, which requires state-of-the-art, costly computers. For many systems the processes are so complex that there is no closed-form solution. Simulation of these processes provides the means to investigate how such complex systems' performance are affected by different inputs and variation of parameters.

Many models such as COSEMS utilize discrete-event simulation because they are temporally, as well as spatially dispersed. Discrete-event simulation refers to modeling a

system over time by changing from state to state or moving from one simulated action to another by discrete time increments (17:7). Although some simulations use fixed time increments, others including COSEMS use variable time increments. Instead of stepping through the process at specific time intervals, COSEMS determines the time of each event and then steps through the simulation one event at a time. At each event, all processes related to that event are completed and then COSEMS moves on to the next scheduled event.

Random Number Generation. Any simulation that does not completely rely on deterministic equations (it utilizes random processes) must have some means to generate random numbers in order to simulate a stochastic process. Generally, programmers inject random numbers into a simulation through two methods: 1) built-in operating system utilities available on computer operating systems and 2) library routines accessed from high-level programming languages (4:28). The first approach is highly machine dependent. Different machines may use different generators, while different machines utilizing the same generators can still draw different random numbers. Identical random number generators on a 32-bit machine will generate a totally different number stream than a 16-bit machine.

Most commonly, a uniform distribution of random numbers is generated in the range [0,1) (17:420). The numbers produced by the generator are then transformed into other distributions including exponential distributions, normal distributions, and Weibull distributions. For the simulation to be valid, these uniformly distributed numbers must appear to be drawn from identical, independent distributions. The numbers are produced using a deterministic equation and are therefore called “pseudorandom” numbers (23:717). Pseudorandom number generators should have the following characteristics:

1. The numbers should be uniformly distributed in the interval [0,1).
2. The numbers should be independent, therefore no correlation should exist in the sequence of random numbers.

3. Many numbers should be generated before the same number is obtained. This is referred to as the period or cycle length of the generator.
4. A random number sequence should be reproducible. This implies that different starting values or seeds should be permitted to allow different sequences or streams to be generated.

(23:717)

The majority of random number generators in use today are linear congruential generators (17:424). A sequence of pseudorandom numbers Z_1, Z_2, Z_3, \dots is generated by the formula

$$Z_{i+1} = (aZ_i + b) \text{ mod } c \quad i = 0, 1, 2, \dots \quad (1)$$

$$R_{i+1} = Z_{i+1} / c \quad (2)$$

Z_0 is the seed value, R_{i+1} is the normalized pseudorandom number, and $a, b,$ and c are constants (23:717). Selection of the constants is extremely important to drawing a good stream of pseudorandom numbers. Ideally the selection of c determines the period of the generator and the density of the random numbers (17:425). Since R_i is bound between 0 and 1, and Z_i is confined to integers, the random variates are confined to values of $0, 1/c, 2/c, 3/c, \dots$. This implies that c must be chosen to be very large (usually $>10^9$) to ensure that there are many possible values for the random numbers (17:425). Additionally, c must be chosen very large, since once a number is repeated the entire sequence is repeated. As a rule c is chosen to be 2^B , where the computer has B bits/word (23:719).

It can be seen that by properly choosing the seed and the constant a , the value $(aZ_i + b)$ will exceed the word size of the machine resulting in an overflow condition. If the integer's binary representation exceeds B digits, the leftmost digits in excess of B will be lost and the value $(aZ_i + b) \text{ mod } c$ will be retained without division by c (17:427). While this simplifies and speeds up computations, it is difficult to rely on this condition when the host language or the computer does not contain overflow handling procedures (4:28). The VAX Ada compiler raises exceptions when these overflow conditions occur and requires

extra code to suppress the checks and prevent execution termination (4:28). This severely limits the code portability and highlights the desirability of a pseudorandom number generator that resides within the program itself. Chandrasekaran (4) provides a listing of Ada pseudorandom number generators for exponential, normal, uniform, and Poisson distributions.

Once the pseudorandom number stream has been generated, it is a simple task to use a reverse transform to map the variates to a distribution other than the uniform distribution. As shown by Table 1, these transforms are often easily coded. Using these equations, any number taken from a uniform [0,1) random number generator can be transformed to another distribution for use in a simulation. For the normal distribution shown below in Table 1, standardized random numbers are created via a normalization algorithm (4:39). The normalization algorithm used by COSEMS is contained within the subprogram *MATHPAK.ADA* (7).

TABLE 1
ATTRIBUTES OF DIFFERENT PSEUDORANDOM NUMBER GENERATORS

Function	Parameters	Computation
Uniform	Interval (a,b)	$(b - a) \times RND + a$
Weibull	Scale β Shape α	$-\beta/\alpha \times \log(RND)$
Exponential	Mean λ	$-\lambda \times \ln(RND)$
Poisson	λ	$e^{-\lambda}$
Normal	Mean a Std. Dev. b	$a + RNDN \times b$
RND—random number between 0 and 1 RNDN—standardized normal number with mean 0 and standard deviation 1		

(4:28)

Replication. In order to approximate the random behavior of systems in the real world, simulations must provide results which exhibit good statistical characteristics

(17:242). One run of a simulation will not ensure that the results appear to come from a random distribution. Since the random samples drawn from the distributions shown in Table 1 provide the data which drives the simulation events, enough random numbers must be drawn so that the random number streams appear random. This is called replication.

The simulation should allow multiple replications, each using a different set of random numbers. Each replication should start from the same initial state with the statistical counters reset to zero (17:243). The results of these independent replications are tabulated into confidence intervals in order to provide statistical precision for the simulation's output. When the number of replications is small (less than 50), a *t* distribution is used to adjust the confidence interval approximation. If the number of replications is sufficiently large, the Central Limit Theorem states that the results will approximate the normal distribution. This implies that large numbers of replications increase the statistical precision of the calculated confidence interval (17:287–289).

Verification and Validation (V&V)

V&V of computer programs ensures that the simulation model's users have confidence in using the model as an aid in decision making (25:33). Verification ensures that a simulation implements the model concepts correctly. Simply put, a model is verified when the code matches the data flow and manipulation specified in the model design. According to Sargent, validation ensures that a model fulfills the purpose for which it is intended. Valid models provide output within acceptable accuracy limits and which conforms to the behavior being modeled (25:33).

Two methods of verification can be used, static and dynamic testing. Static testing uses correctness proofs and structured walk-throughs to determine whether a computerized model works as intended. Static testing is conducted through the examination of listings and output and reveals syntax errors, misspellings, missing statements, and improper sequencing of expressions. Correctness proofs involve actual mathematical proofs that a

code is correct, while structured walk-throughs involve analyzing the flow of the code and comparing it to the conceptual flowchart (25:35).

Models are dynamically tested by executing runs with different parameters and determining whether the program performs correctly. This manner of testing reveals errors through the program's response to specific inputs. Inputs are planned to exercise the limits of the program by taking on both extreme values and improbable values. Dynamic testing uses traces and the checking of the internal consistency of the program's modules with each other (25:35).

Validation techniques include comparison to other models, event validity, face validity, traces, and predictive validation as defined by Sargent and excerpted in Table 2 below. Traces can be used for both verification and validation.

As shown by Figure 2, validation and verification are not accomplished separately from software development, but are continually accomplished parallel to the software development. Verification, validation, and confidence building are perhaps the most important elements of simulation development, since it is unlikely that the results will be used if the program cannot be shown to work correctly. Validation assures that the conceptual model truly represents the system under study and the simulation program functions in a similar manner to the real system, affording proper results for known cases. Verification assures that that the simulation program follows the principles outlined in the conceptual model. When the function and output of the simulation are shown to accurately represent the system under study, the model is called credible (17:299). This demonstration means that the results of a simulation run are accepted as credible by the user and have a very reasonable expectation of providing information to aid in decision making.

TABLE 2
VALIDATION TECHNIQUES

Comparison to Other Models

Various results (e.g., outputs) of the simulation model being validated are compared to results of other (valid) models. For example simple cases of a simulation model may be compared to known results of analytic models...

Event Validity

The "events" or occurrences of the simulation model are compared to those of the real system to determine if they are the same...

Face Validity

Face validity is asking people knowledgeable about the system whether the model and/or its behavior is reasonable. This technique can be used in determining if the logic in the model flowchart is correct and if a model's input-output relationships are reasonable.

Predictive Validation

The model is used to predict (forecast) the system behavior and comparisons are made to determine if the system behavior and the model's forecast are the same. The system data may come from an operational system...

Traces

The behavior of different types of specific entities in the model are traced (followed) through the model to determine if the model's logic is correct and if the necessary accuracy is obtained. This technique is more commonly used in verification.

(25:33-34)

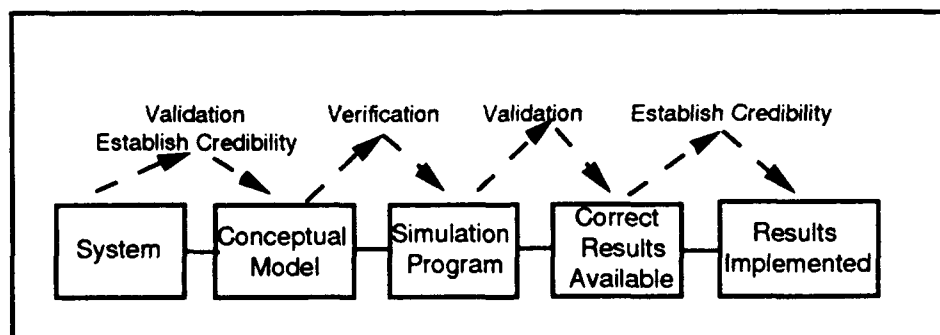


Figure 2. Relationships of Validation, Verification, and Establishing Credibility (17:299)

Reliability

In the design of any system, it is desirable that it operates for as long as possible without breaking down. Reliability is the probability that a device will operate for a given time period. Reliability or designing for sustained performance can be modeled through probabilistic analysis and predicted through the application of these models using measured failure rates from the system components' past performance or from desired performance goals.

If the time between successive failures is a continuous random quantity, the time between failures can be determined when the distribution function is known (18:105). Through the analysis of failure data, the distribution parameters can be estimated and a distribution fit to the data. In practice, the physical failure rate of electronics often follows the curve shown in Figure 3. Region ab has a high and decreasing failure rate due to inherent flaws in the devices. Zone bc shows a relatively constant failure rate in which the

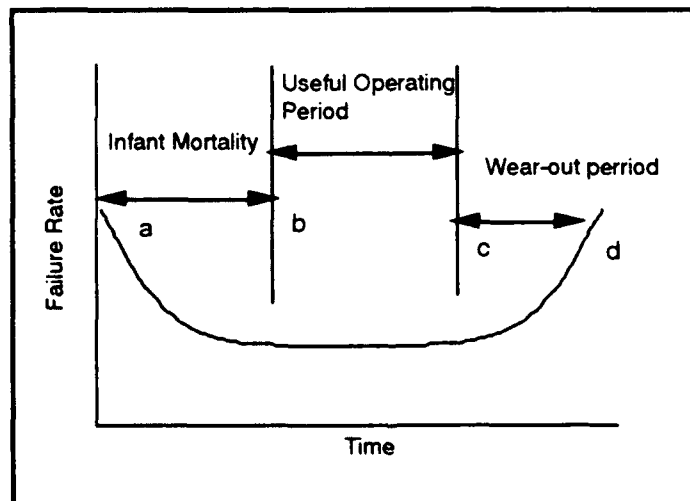


Figure 3. The Bath Tub Curve (18:106)

majority of operating life occurs, and the last region has an increasing failure rate because devices wear out and break down. While the above bath-tub curve provides good insight into understanding the life-cycle and failure of devices, Ling (18) points out the time between failures of complex systems can be modeled by a few well-understood probability

distributions: the exponential, the Rayleigh, the normal, the gamma, and the Weibull distributions. The exponential and Weibull distributions are most applicable to understanding the reliability of electronic systems (10:5).

Probability Distributions. In the past, reliability prediction for spacecraft assumed the exponential distribution (10:1). The exponential distribution is unique because of its memoryless property; the fact that a particular device or system has survived for a particular period of time in no way alters the probability that the device will survive for another random time period. For example, a particular light bulb has an exponential failure distribution with a mean lifetime of 1000 hours. The probability that it will survive for 1000 hours given that the bulb has already burnt for 1000 hours is the same probability that it will survive for 1000 hours— $P(\text{failure time} > 1000 + \text{time used so far} | \text{time used so far} > 1000) = P(\text{failure time} > 1000)$.

If the mean time to failure for an electronic component is defined by μ and the failure rate by λ , the probability density $f(t)$, hazard function $Z(t)$ (instantaneous rate of failure), and reliability $R(t)$ for the exponential distribution are shown in Figure 4 and related by the equations

$$\mu = \frac{1}{\lambda} \quad (3)$$

$$f(t) = \lambda e^{-\lambda t} \quad (4)$$

$$Z(t) = \lambda \quad (5)$$

$$R(t) = e^{-\lambda t} \quad (6)$$

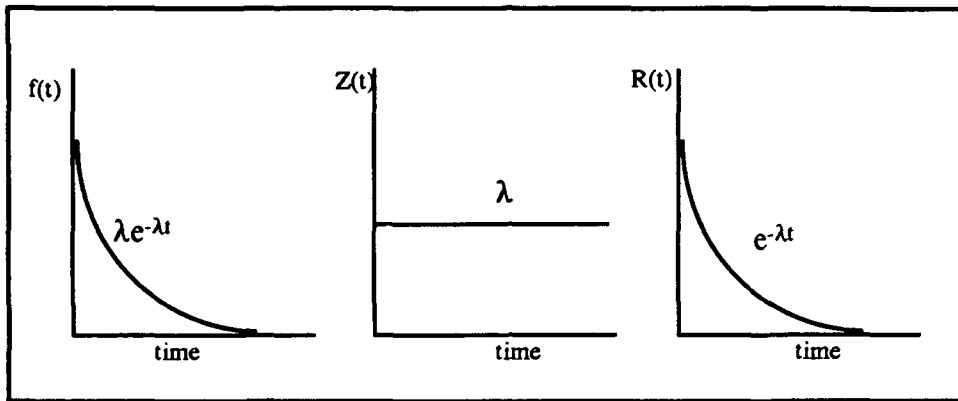


Figure 4. Exponential Distribution Characteristics

The exponential distribution is unique in having a constant failure rate and is very useful in predicting the reliability of individual electronic parts (8:6). However, as illustrated by the bath-tub curve, it only allows modeling of one part of the system lifetime. When most known distribution functions cannot be fit to failure data, the Weibull distribution is an appropriate function to use (18:111).

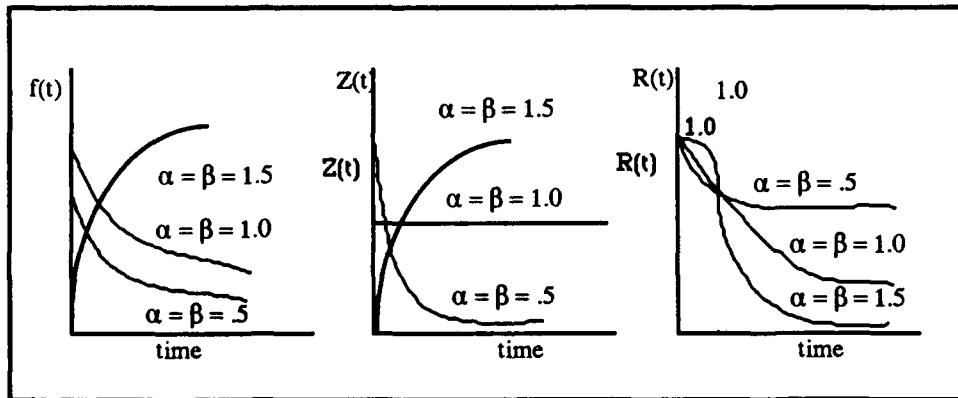


Figure 5. Weibull Distribution Characteristics (18:112)

The Weibull distribution can approximate other distributions by the adjustment of the shape and scale parameters α and β (18:113). As shown in Figure 5, for a given shape parameter, changing the scale parameter alters the vertical amplitude of the hazard function $Z(t)$. The probability, hazard, and reliability functions are given as

$$f(t) = \alpha \beta t^{\alpha-1} e^{-\beta t^\alpha} \quad (7)$$

$$Z(t) = \alpha \beta t^{\alpha-1} \quad (8)$$

$$R(t) = e^{-\beta t^\alpha} \quad (9)$$

For $\alpha > 1$, the hazard function increases with time, while for $\alpha < 1$, the hazard function decreases with time. For $\alpha = 1$, the hazard function is constant, and the Weibull behaves like an exponential distribution with $\beta = \lambda$. This distribution is useful because varying the parameters at different times in a system's life-cycle and superpositioning the different distributions upon each other creates a model with the desired shape of the bathtub curve (18:113).

Recent studies have implied that the use of exponential distributions for predicting spacecraft reliability may be overly conservative. Collected data from on-orbit spacecraft component failures shows a decreasing hazard rate which fits a Weibull distribution (10:13). Apparently parts, quality, and operational failures are still modeled best by the exponential distribution, but over time, the mission type and orbit environment modify the failure rate to follow a Weibull distribution (10:2). This new model provides higher reliabilities relative to the more conservative exponential distribution.

COSEMS uses the Weibull distribution for simulating spacecraft component failures, but defines the reliability function differently:

$$R(t) = e^{-(\beta t)^\alpha} \quad (10)$$

By comparing this expression to Equation (9), it would seem that COSEMS uses a distribution that is different from the Weibull distribution. However the only difference is the units of the α parameter (9:146). The *COSEMS Design* documentation (5) confirms the use of this equation (5:A-12).

Series and Parallel Circuits. While the distributions detailed in the previous section allow the modeling of the failure of individual components, spacecraft are composed of many different components with varying failure rates. The relationship

between the reliability of the spacecraft as a whole and its individual components is complex and extremely dependent on how the spacecraft design connects those components. Electronic components may be connected in series and in parallel. Generally, parallel components are used to provide redundancy and increase reliability.

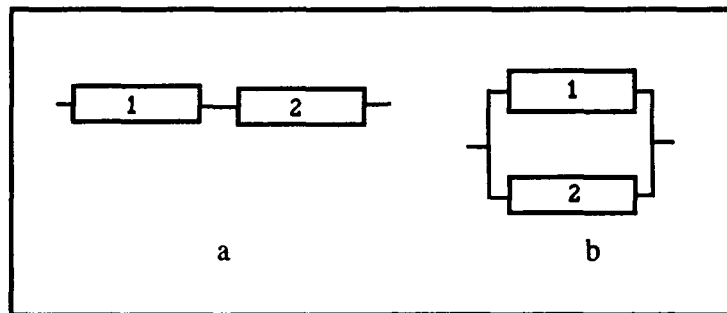


Figure 6. Example of Series and Parallel Blocks(15:56–57)

Series configuration of components (Figure 6a) is the most frequently encountered model and is easy to analyze. By assuming each component is independent of the others, the reliability of the series is the product of the reliability of each component (15:56). If you have n components in series, each with different reliability, the model for the reliability of the whole system is expressed as

$$R_s = R_1 R_2 \dots R_{n-1} R_n \quad (11)$$

$$R_s = \prod_{i=1}^n R_i \quad (12)$$

Parallel systems are analyzed differently. The parallel system shown in Figure 6b cannot be considered to have failed unless all the components in parallel have failed. Thus the unreliability, Q , of a parallel system is the product of the unreliability of each of n components (15:58). The reliability of the system is one minus the unreliability of the system.

$$Q_s = (1-R_1)(1-R_2)\dots(1-R_{n-1})(1-R_n) \quad (13)$$

$$R_s = 1 - \prod_{i=1}^n (1-R_i) \quad (14)$$

The combination of series and parallel subsystems into one system involves repeated application of these two concepts. The reliability of each subsystem of parallel modules is first calculated and then each subsystem is treated as a series module connected to other series modules. Figure 7 shows how a complicated design can be treated as five series modules once the system reliability of three groups of parallel modules is calculated.

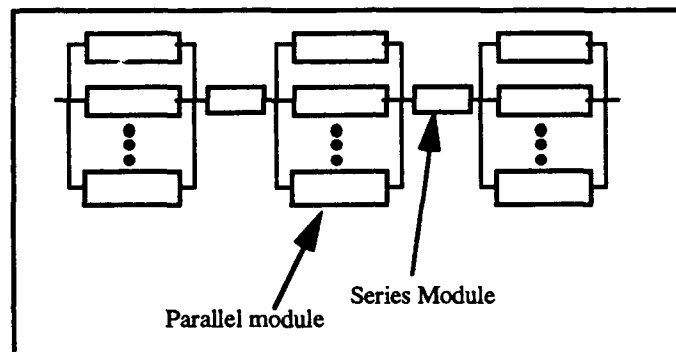


Figure 7. Reliability Block Diagrams (9:146)

COSEMS treats the satellites it models with this type of analytical approach (9:146). The user specifies the scale and shape parameters for each module in the satellite, the number of subsystems, and how many parallel modules make up each subsystem (6:6-24). While COSEMS uses the Weibull distribution for modeling the individual modules, the default parameters provided by the programmers (unless overridden by the user) reduce the Weibull to the exponential distribution (5:A-13).

Orbital Mechanics

Simulating and predicting the orbits of satellites involves complicated equations accounting for the gravitational attraction of the satellite to the body it is orbiting, the gravitational attraction of other bodies, and many non-gravitational sources of small

perturbations. A basic understanding of the orbital mechanics involved is possible by ignoring all bodies but the satellite and the central body it orbits.² Such simplifications make closed-form solutions to orbit equations possible (3:4). A full treatment, beyond the scope of this paper, requires complicated analysis and iterative solution techniques.

By assuming conservation of energy and momentum (bold characters are vectors), and assuming that the central body and satellite are perfect spheres, Newton's universal law of gravitation states

$$\mathbf{F} = -\frac{GMm}{r^2} \frac{\mathbf{r}}{r} \quad (15)$$

where \mathbf{F} is the central force vector, G is the gravitational constant, M is the mass of the central body, m is the mass of the satellite, and r is the distance between the bodies.

Kepler's laws add more information important to understanding orbits:

1. The orbit of each satellite is an ellipse with the central body at a focus.
2. The line joining the satellite to the central body sweeps out equal areas in equal time.
3. The square of the period of a planet is proportional to the cube of its mean distance from the central body.

(3:2)

Working with these laws and equations, it can be shown that a satellite in orbit around the earth follows an elliptical trajectory with the earth at a focus.

As illustrated by Figure 8, periapsis, r_p , is the orbit radius at closest approach for a satellite and apoapsis, r_a , is the radius at furthest retreat. For a circular orbit the apoapsis and periapsis are equal; the radius is constant in time. The eccentricity, e , measures how much the orbit departs from a circle. r and v indicate the radius and velocity vectors, while the true anomaly, ν , is the angular measurement of the radius vector from periapsis.

² A central body attracts objects to its center of mass. In this case, the earth is a central body and exerts a gravitational force which attracts satellites towards itself.

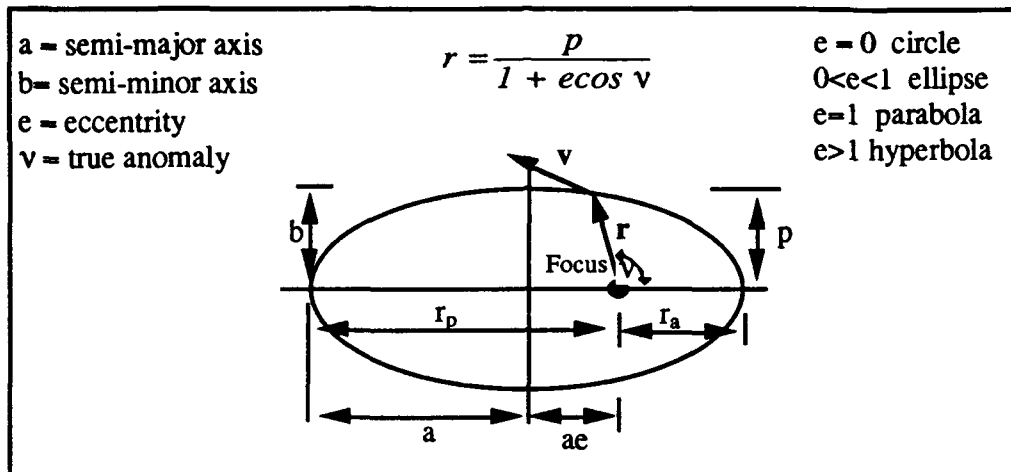


Figure 8. Kepler's Elliptical Orbit Geometry (3:21)

The equations for conservation of momentum (\mathbf{h}) and energy (E) are

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \quad (16)$$

$$E = \frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \quad (17)$$

where μ is the gravitational constant and depends on the central body (3:16–17).

Figure 8 shows how the motion about the earth relates to the true anomaly. According to Kepler's Second Law, conservation of energy dictates that a satellite in an elliptical orbit must speed up as it approaches periapsis and slow down as it reaches apoapsis. Only in a circular orbit will the orbital speed remain constant. By using a reference circle as shown in Figure 9, the constant motion about a circular orbit can be related to an elliptical orbit. The eccentric anomaly, E , is used to calculate the mean anomaly, M , of the satellite in its elliptical orbit. This mean anomaly is based upon the period, P , the semi-major axis, a , and the time since periapsis passage, T . With these equations and an initial position for a satellite, it is possible to calculate the satellite's position at a later time. Iteration techniques provide a value for the eccentric anomaly, E , which is used to calculate the true anomaly, giving the satellite's position in the orbit.

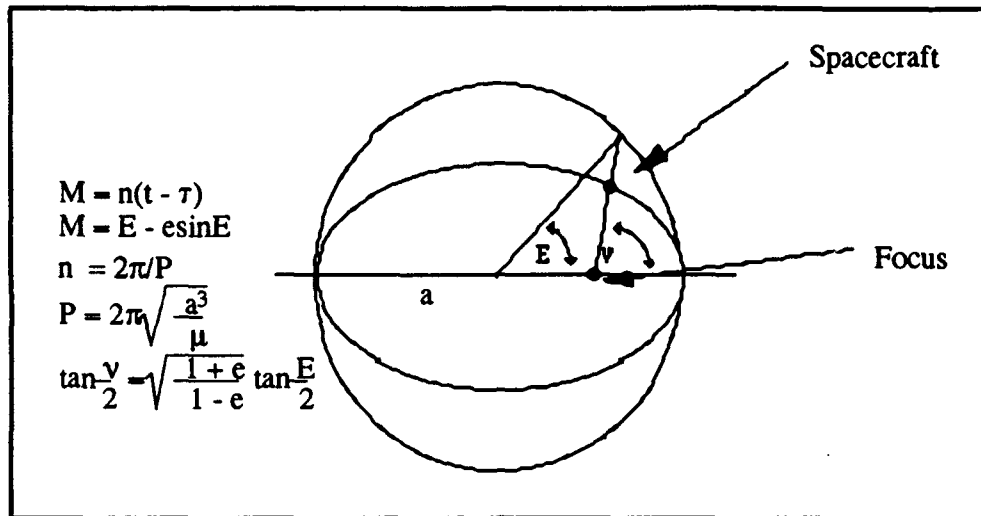


Figure 9. True Anomaly Relation to Mean Anomaly (22:33-34)

While Figure 8 illustrates the shape and governing equation of motion about an orbit, and Figure 9 shows the equations relating the motion to time, more equations are needed in order to account for the motion of the satellite and the earth in space and the orientation of the orbit to the earth. This motion can be solved by picking a non-inertial (unaccelerated and non-rotating) reference frame in which the earth and the satellite both move. Five independent quantities describe the size, shape, and orientation of the orbit, and a sixth element specifies the satellite's position in the orbit (3:58).

The relationships of the six elements defined by Table 3 are shown in Figure 10 and fully describe any individual orbit. The inertial frame IJK is fixed in space with the K axis passing through the north pole and the IJ plane passing through the equator. The earth rotates about the K axis. The inclination, i , is measured from K to the momentum vector and denotes the inclination of the orbit plane in relation to the equator. Inclination is always measured as less than 180° —when it is greater than 90° , the satellite travels in a retrograde fashion of from east to west. Ω , the longitude of the ascending node, is measured from I and denotes the point at which the satellite crosses the equator in a northerly direction.

TABLE 3

ORBITAL ELEMENTS

a	semi-major axis
e	eccentricity
i	inclination of orbit plane
Ω	longitude of the ascending node
ω	argument of periaapsis
T	time of periaapsis passage

The argument of periaapsis, ω , is measured from Ω and fixes the location of periaapsis. T , the time of periaapsis passage, provides the true anomaly, ν , as shown in Figure 10 and fixes the satellites angular separation from periaapsis.

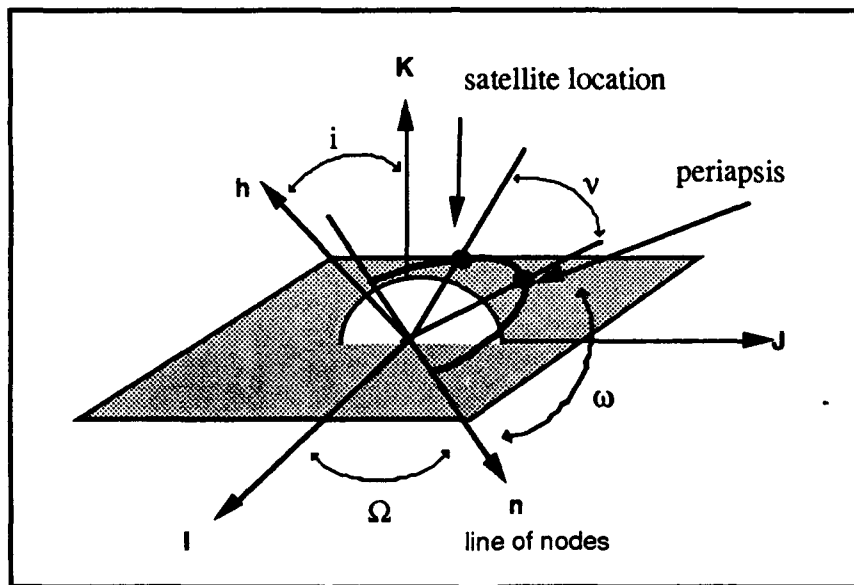


Figure 10. Orbital Elements and Orbit Geometry (3:59)

With the aid of Figure 10 and Equation (16), orbital elements may be calculated from the spacecraft position and velocity vectors r and v using the following equations

$$\mathbf{n} \equiv \mathbf{K} \times \mathbf{h} \quad (18)$$

$$\mathbf{e} = \frac{1}{\mu} \left[\left(v^2 - \frac{\mu}{r} \right) \mathbf{r} - (\mathbf{r} \cdot \mathbf{v}) \mathbf{v} \right] \quad (19)$$

$$e = |\mathbf{e}| \quad (20)$$

$$\cos i = \frac{h_K}{h} \quad (21)$$

$$\cos \Omega = \frac{n_I}{n} \quad (22)$$

$$\cos \omega = \frac{\mathbf{n} \cdot \mathbf{e}}{ne} \quad (23)$$

$$\cos \nu_0 = \frac{\mathbf{e} \cdot \mathbf{r}}{er} \quad (24)$$

where boldface indicates a vector and the subscript the vector's projection onto the indicated coordinate (h_K is \mathbf{h} projected onto the K coordinate axis) (3:61-63).

It should be noted that for circular orbits, ω and ν_0 are undefined and for equatorial orbits ($i = 0$), Ω is undefined. A computer model must switch to a different coordinate system to keep from obtaining undefined values or use alternate coordinates. These alternate coordinates are not important for this analysis. Further elaboration on coordinates is discussed in Reference (3).

Orbit Perturbations. The equations and concepts on orbital mechanics detailed in the last section assume that only the orbiting body and its central body interact and that the central body behaves as a point mass. Just as aircraft encounter course and speed deviations due to wind, spacecraft experience perturbations to their orbital elements due to effects from the non-spherical shape of the earth, drag from the earth's atmosphere, gravitational attraction of the sun and other celestial bodies, and radiation pressure from the solar wind (1:53). This paper will confine itself to perturbations due the non-spherical shape of the earth and atmospheric drag.

The Non-Spherical Earth. Extra gravitational bodies are accounted for by treating the system as a system of N bodies consisting of point masses. However, the earth itself is not a point mass because of its proximity to the orbiting spacecraft and because its rotation causes the earth to bulge at the equator. Additionally, the different distributions of mountains, oceans, and continents cause further deviation from a point mass (1:63). Due to the earth's non-spherical shape, the center of mass of the earth does not coincide with its physical center. This is the primary cause of perturbative effects on the six orbital elements.

The principal effects of the non-spherical earth are the rotation of the line-of-apsides (the major axis, $2a$) and the regression of the line-of-nodes, n (3:156). By considering the earth's bulges and other departures from sphericity to be a massive band around the earth's equator, Figure 11 illustrates how the mass will impart a torque on the satellite's orbit. A torque on the h vector shown in Figure 10 causes the plane of orbit to precess in a similar manner as that of a gyroscope (3:156). Orbits with inclinations between 0 and 90 degrees

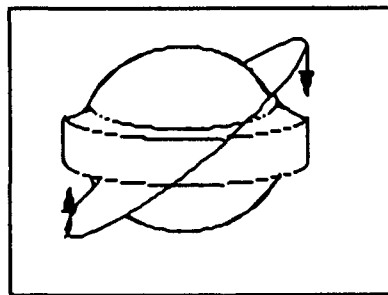


Figure 11. Effect of the Earth's Oblateness (3:156)

regress westward, while orbits with inclinations greater than 90 degrees progress eastward. Higher orbital altitudes experience this to a lesser degree. At sufficiently high altitudes, the earth may be treated as a point mass and the non-spherical earth effects can be ignored.

The earth's non-spherical shape also causes rotation of the line-of-apsides for non-circular orbits (3:159). Any elliptical orbit will experience the rotation of its major axis resulting in a change in the relative position of the apoapsis and periapsis. Satellites with inclinations less than 63.4° or greater than 116.6° experience rotation of periapsis in their

direction of motion, while those with inclinations between those two critical inclinations experience rotation opposite that of their motion (3:159). The rate of rotation is dependent on altitude and inclination. Spacecraft in geosynchronous orbit also experience perturbations. The non-spherical earth causes east/west while the effects of the sun and moon cause north/south drift.

In the absence of any forces other than classical Newtonian gravitational attraction, the six classical orbital elements shown in Table 3 do not change. However, we have just discussed how the non-spherical shape of the earth affects these orbital elements. The exact equations detailing the changes to the orbital elements is complicated and depends on a mathematical model of the earth's gravitational (geopotential) field. The departure from a point mass is detailed in a complex equation defining the geopotential of the earth. Due to the structure of the earth and the distribution of its lands and oceans, this geopotential has many terms. The J_2 term is the predominant term of the equation and accounts for the earth's non-spherical shape (1:87). Table 4 shows the disturbing potential, R , which accounts for the effects of the non-spherical earth. The disturbing potential may be changed to add other terms and model additional effects.

Table 4 presents the equations for the changes in the classical orbital elements over time. Most of the perturbation effects are small or are sinusoidal and average out to zero over time. All terms that involve sines and cosines will average out to zero (1:86). However as shown in Figure 9, the mean anomaly, M , increases with time and produces effects that increase over time (1:88). This means that any perturbative term dependent on the terms with a , e , and i are affected and thus only the elements Ω , ω , and M grow with time (1:87). These are terms which cause apsidal rotation and nodal regression.

TABLE 4

PERTURBATIVE TERMS FOR THE
CLASSICAL ORBIT ELEMENTS

$$R = -\frac{\mu R_{\oplus}^2 J_2}{2a^3} \left(1 + \frac{3}{2}e^2\right) \left(\frac{3}{2}\sin^2 i - 1\right)$$

$$\frac{da}{dt} = \frac{2}{na} \frac{\partial R}{\partial M}$$

$$\frac{de}{dt} = \frac{1-e^2}{na^2 e} \frac{\partial R}{\partial M} - \frac{\sqrt{1-e^2}}{na^2 e} \frac{\partial R}{\partial \omega}$$

$$\frac{di}{dt} = \frac{\cot i}{na^2 \sqrt{1-e^2}} \frac{\partial R}{\partial \omega} - \frac{1}{na^2 \sqrt{1-e^2} \sin i} \frac{\partial R}{\partial \Omega}$$

$$\frac{d\Omega}{dt} = \frac{1}{na^2 \sqrt{1-e^2} \sin i} \frac{\partial R}{\partial i}$$

$$\frac{d\omega}{dt} = \frac{\sqrt{1-e^2}}{na^2 e} \frac{\partial R}{\partial e} - \frac{\cot i}{na^2 \sqrt{1-e^2}} \frac{\partial R}{\partial i}$$

$$\frac{dM}{dt} = \frac{2}{na} \frac{\partial R}{\partial a} - \frac{1-e^2}{na^2 e} \frac{\partial R}{\partial e}$$

R_{\oplus} is the radius of the earth and J_2 is the the secular term in the geopotential of the earth accounting for the earth's non-spherical earth. The equations of Table 4 suffer when inclination is zero and eccentricity is zero. Other forms of the equations must be used for orbits with these elements.

(1:50)

It should be noted that these affects can be used to a satellite designer's advantage. By selecting orbits with specific elements, the nodal regression can keep the satellite's orbit in the exact same orientation to the sun continuously. This is of special interest to satellites with photographic missions, since the sun stays at the same angle to the satellite's orbit. This allows the comparison of photographs taken on different days, since all shadows cast by the sun will be identical. Molniya satellites ("news flash" in colloquial Russian, so-called because of their use in television broadcasts), take advantage of the apsidal rotation; by choosing a highly elliptical orbit with an inclination of 63.4° , the apoapsis of the satellite

remains over the same spot on the earth and allows a satellite to loiter over a particular geographic location for most of its orbital period. These communications satellites use this effect to spend most of their time servicing a specific geographic location (1:91). East/west drift can be used to move geosynchronous satellites from one latitude to another without having to use a lot of fuel.

Drag. Atmospheric drag interferes primarily with low orbiting satellites. Air molecules produce drag effects which decrease the energy of a satellite. This causes the satellite to drop to a lower orbit with a smaller period. In effect, drag actually increases the satellite's speed. Unchecked, the satellite will dip lower and lower into the atmosphere, experiencing more and more drag, until it can no longer stay in orbit. Another effect is the circularizing of orbits. For elliptical orbits, the satellite at periapsis experiences more drag than at apoapsis, which eventually lowers the apoapsis and circularizes the orbit.

Drag is difficult to predict due to the constantly changing atmospheric density. As the sun heats the earth and the earth revolves about the sun, the earth's atmosphere expands and contracts. Solar flares, sunspots, and other solar phenomena can cause unpredictable heating of the earth's atmosphere and dramatically affect low orbiting satellites and even some higher satellites (1:70). Drag is proportional to the satellite's cross sectional area. The larger the satellite, the more drag it will experience.

Station-Keeping. Perturbative effects are important because station-keeping thrusters must be used to counteract unwanted changes in a satellite's orbital elements. Station-keeping for low orbiting satellites requires more fuel to counteract the increased effects of the non-spherical earth of the earth and atmospheric drag. Thrusters can be used at periodic intervals—allowing more drift to accumulate by having long periods between drift corrections dramatically increases fuel consumption. Fuel consumption is highly dependent on inclination and eccentricity as a result of the perturbative effects.

COSEMS accounts for station-keeping by assuming that a satellite uses its fuel to counteract all perturbations except for apsidal and nodal rotation. Fuel is expended at a constant rate to account for this station-keeping.

Orbit Transfer. Now that the general equations governing motion within an orbit have been presented, the equations governing the transfer between orbits and rendezvous between spacecraft in orbit can be dealt with. While it is possible to transfer between any two orbits, for brevity, this paper will only go into detail on transfers between circular orbits. Transfers between coplanar circular orbits will be accomplished with a bi-elliptic transfer unless a Hohmann transfer is immediately possible.

The Hohmann transfer orbit, illustrated in Figure 12, is generally considered the most energy efficient transfer and requires the minimum fuel (27:165). The basic idea is to impart a change in velocity to increase the orbital speed (when moving from a low orbit to a higher orbit) and change the orbit's shape to an elliptical orbit (Figure 12, Point 1). When the elliptical orbit crosses onto the desired circular orbit (Point 2) another velocity change is performed to circularize the orbit at the new altitude. From Equation (17) the velocity of the initial and final circular orbits can be found to be

$$v_{cs} = \sqrt{\frac{\mu}{r_{cs}}} = \sqrt{\frac{\mu}{r_p}} \quad (25)$$

The required velocity for the elliptical transfer orbit can be found from Equations (16) and (17), since the difference between the two velocities gives the required change in velocity that must be made at Point 1.

$$v_1 = \sqrt{2\left(\frac{\mu}{r_p} + E_t\right)} \quad (26)$$

$$E_t = \frac{-\mu}{r_a + r_p} = \frac{-\mu}{r_1 + r_2} \quad (27)$$

$$\Delta v_1 = |v_{cs} - v_1| \quad (28)$$

Using the same means, it can be shown that the velocity at the apoapsis of the transfer orbit is less than the required circular velocity at the high orbit. The same equations will provide the velocity increment required at Point 2 to circularize the orbit.

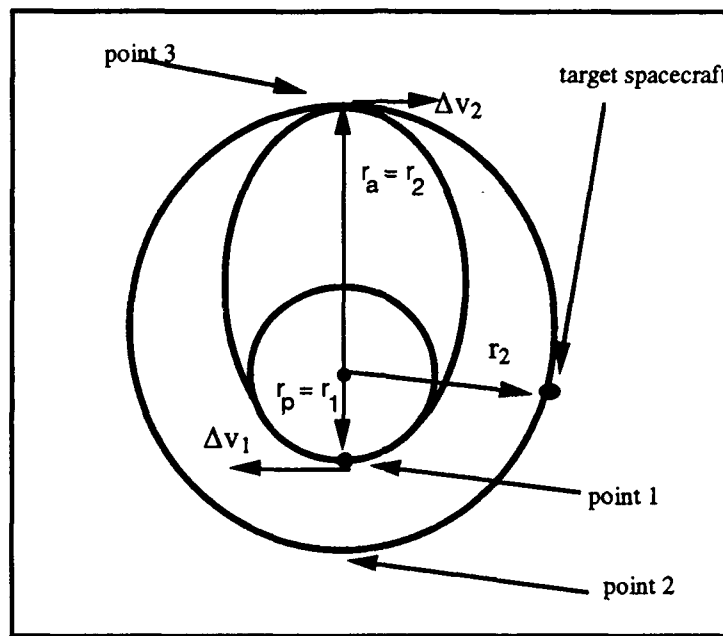


Figure 12. Hohmann Transfer

Additionally, using the equation for the period shown in Figure 9 and by recognizing that the transfer trajectory is one-half of the ellipse, the time of flight for the transfer is

$$TOF = \pi \sqrt{\frac{a^3}{\mu}}, \quad a = r_a + r_p = r_1 + r_2 \quad (29)$$

The bi-elliptic transfer orbit, shown in Figure 13, differs from the Hohmann transfer because it uses three velocity changes rather than two (19:27). In effect, two Hohmann transfers are patched together. It has been shown that the bi-elliptic transfer will be more economical than the Hohmann transfer when the ratio of the two orbits' semi-major axes is greater than 15.6 (2:439). This generally only occurs when the initial elliptical trajectory takes the transfer spacecraft out beyond the target orbit and the second trajectory brings the spacecraft back down to the desired orbit.

Bi-elliptic transfers are desirable not because of economy, but because of the time savings. This is discussed in detail in the next section, *Rendezvous Windows*. Since performing a Hohmann transfer in order to rendezvous with a spacecraft at another orbit requires very strict timing, the transfer must be initiated at such a time that the target spacecraft and transfer spacecraft are in phase. In order to perform a Hohmann transfer (Figure 12), the spacecraft must start at Point 1, and the target spacecraft must start at Point 2 if they are both to rendezvous at Point 3. The time of flight for the transfer orbit must be such that the target arrives at the elliptical transfer orbit's apoapsis simultaneously with the transfer spacecraft (21:5). The bi-elliptic transfer, when confined to the region between the two orbits, will be less economical than the Hohmann, but provides a range of phasing values for the rendezvous (21:6). In reference to Figure 13, the initial velocity change

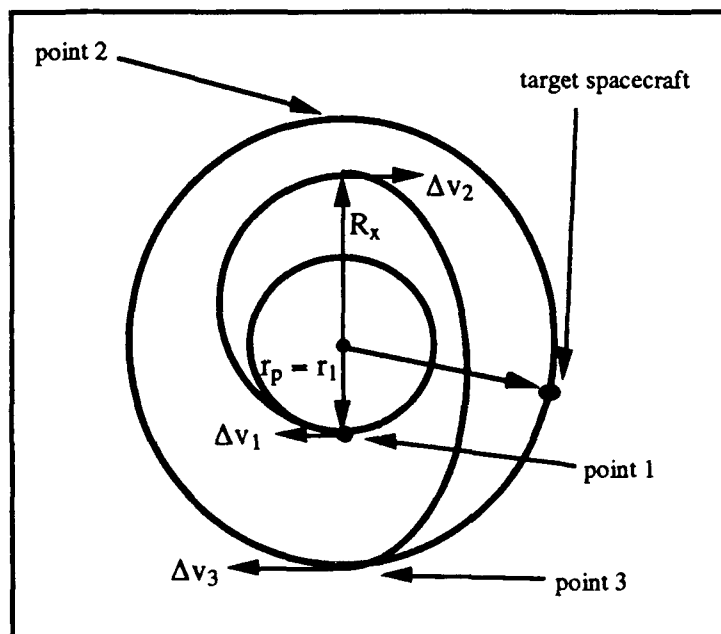


Figure 13. Bi-Elliptic Transfer (21:6)

(Point 1) takes the transfer spacecraft approximately half the distance to its target and through an angle of 180° . Depending on the phase, the second burn (Point 2) is adjusted to make the two spacecraft meet after the transfer spacecraft travels through another angle of

180°. A third velocity change (Point 3) circularizes the orbit, and the two spacecraft rendezvous. COSEMS uses a bi-elliptic transfer unless a Hohmann transfer is immediately possible (21:5).

Rendezvous Windows. Figure 14 assumes a space vehicle (SV) at θ at time t_0 has decided to rendezvous with a spacecraft (SC). The following discussion and equations reference Figure 14. Rendezvous can only be initiated at Point 1 or 2. This requires the spacecraft to wait until it reaches either of those points (21:8). If $0 < \theta < \pi$, the next rendezvous initiation is at Point 1. If $\pi < \theta < 2\pi$, the initiation must occur at Point 2. The waiting time until SV is at one of these points is either

$$t_{wait1} = \left(\frac{\pi - \theta}{2\pi}\right) P_0, \quad 0 < \theta < \pi \quad (30)$$

$$t_{wait1} = \left(\frac{2\pi - \theta}{2\pi}\right) P_0, \quad \pi < \theta < 2\pi \quad (31)$$

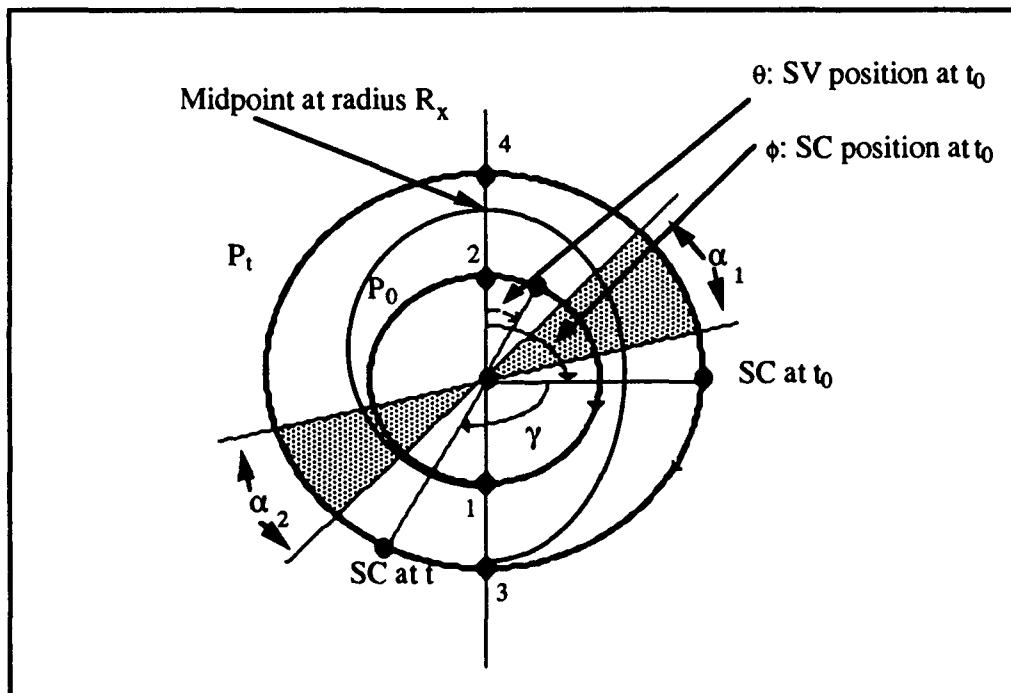


Figure 14. Geometry for Rendezvous Windows (21:8)

During this waiting period, SC moves through an angle γ , defined as

$$\gamma = t_{wait1} \frac{2\pi}{P_t} \quad (32)$$

where P_t is the period of SC.

If SV started at $0 < \theta < \pi$, at $t = t_{wait1}$, SV is at $\theta = \pi$, while SC is at $\phi_1 = \phi + \gamma$. If SC is in α_2 , the transfer can be performed. Otherwise the transfer is delayed for a half revolution and initiated when SC is in the α_1 region. At this point $\theta = 2\pi$, and SC would have continued to travel through an angle β (21:8).

$$\left(\frac{\beta}{2\pi}\right)P_t = \frac{P_0}{2}, \quad \beta = \pi \frac{P_0}{P_t} \quad (33)$$

At this point, the spacecraft have waited for $time = t_{wait1} + P_0/2$. SV is at $\theta = 2\pi$, while SC is at $\phi_2 = \phi + \gamma + \beta$. If SC is in the α_1 region, the transfer is initiated, otherwise SC waits another half revolution. Eventually SC will be in the appropriate α region and the transfer will be performed (21:9). Therefore the waiting process can continue for many half revolutions with t_{wait2} defined as

$$t_{wait2} = M \left(\frac{P_0}{2}\right), \quad M = 0, 1, 2, 3, \dots \quad (34)$$

and the total waiting time as

$$t_{wait} = t_{wait1} + t_{wait2} \quad (35)$$

It should be noted that when the two orbits of SC and SV are close together, this phase waiting period can become inconveniently long since the rendezvous window decreases considerably (21:12).

The actual time of transfer and intermediate point at which the second velocity change is initiated, R_x , (see Figure 14) depends on where in the α region the transfer is initiated. In other words every point in the α region corresponds to a unique transfer orbit (21:10). This α region is entirely dependent on mission planning constraints. Once the SC is positioned in the α region, the intermediate orbit can be found numerically using the equation

$$\alpha = \frac{\pi}{2} \left[\sqrt{\frac{(R_0 + R_x)^3}{2R_T}} + \sqrt{\frac{(R_x + R_T)^3}{2R_T}} \right], \quad R_0 \leq R_x \leq R_T \quad (36)$$

where R_0 is the initial orbit radius, R_x is the intermediate orbit radius, and R_T is the final orbit radius.

While the transfer time (TOF) to move from the initial orbit to the final orbit is found using Equation (29), the equation is used to calculate the half-period of the elliptical orbit at the mid-point and the half-period of the elliptical orbit from the mid-point to the final orbit.

$$TOF = t_{12} + t_{23} \quad (37)$$

$$TOF = \pi \left[\sqrt{\frac{a_{12}^3}{\mu}} + \sqrt{\frac{a_{23}^3}{\mu}} \right]$$

$$a_{12} = \frac{R_0 + R_T}{2}, \quad a_{23} = \frac{R_0 + R_T}{2} \quad (38)$$

In the discussion of orbital transfer, it has been shown that a change in velocity in the orbital plane can change its size and eccentricity. Sometimes it is desirable to change the orientation of the orbital plane with respect to the earth. Such a change requires a velocity change outside the plane of the orbit. Such plane changes may take place in conjunction with an orbital transfer or separate from it.

A simple plane change from an inclined orbit to an equatorial orbit is illustrated by Figure 15. Since the two orbits are circular, the initial and final orbit velocities are equal, allowing the total change of velocity, Δv , to be solved using the Law of Cosines and is expressed by Equation (39) (3:169).

$$\Delta v = 2v \sin \frac{\theta}{2} \quad (39)$$

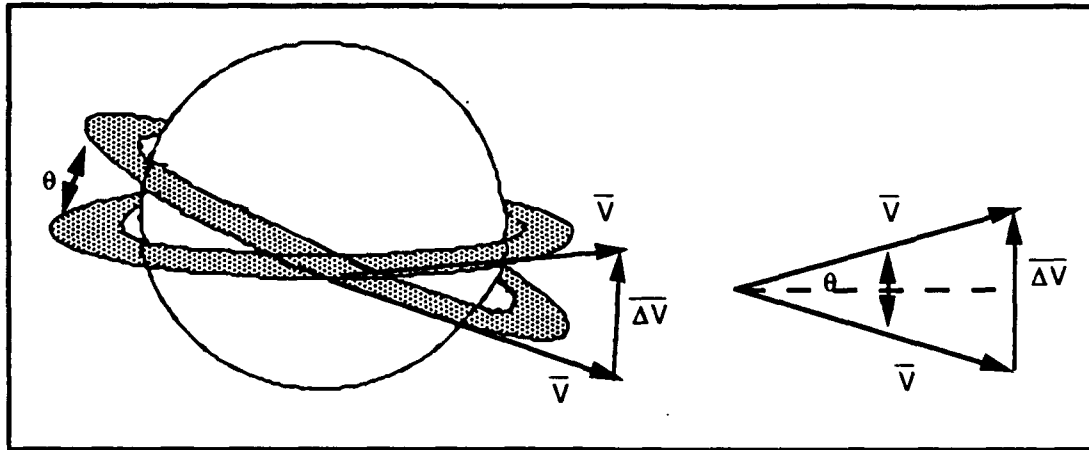


Figure 15. Plane Change Through Angle θ (3:170)

When plane changes are performed at other than ideal times and include altitude changes, the vector diagram shown in Figure 15 is different. The legs of the triangles shown are of different lengths v and v_t , the original orbital velocity and the transfer orbit velocity. The change in velocity required may then be expressed in its general form

$$\Delta v = \sqrt{v^2 + v_t^2 - 2vv_t \cos \theta} \quad (40)$$

Equations (39) and (40) are identical when v is equal to v_t .

In order to equatorialize the orbit, the change in velocity must be applied at one of the nodes (3:169). For elliptical orbits, it is usually best to perform a plane change near apoapsis, when the satellite is moving its slowest in order to minimize the required velocity change.

Plane changes may be made in conjunction with orbit transfers in order to rendezvous with satellites at different altitudes and inclinations. In this case, whether for Hohmann or bi-elliptic transfers, the total velocity required may be expressed as

$$\Delta v_{total} = \Delta v_{transfer} + \Delta v_{plane\ change} \quad (41)$$

Mission Planning

Mission planning is an important aspect of space system support and COSEMS because mission planning responds to the constellation's demands in order to make repairs

and maximize system availability. Initial planning prior to fielding the system determines the size and the number of spacecraft in the constellation, as well as which logistical support concept will be used. Typical mission planning after the constellation has been fully deployed consists of assessing failures and attempting a telemetry fix by activating redundant systems. If the telemetry fix fails, the mission planner schedules a support mission to repair or replace the satellite. If the planner chooses replacement, the planner must then also choose to either use a spare satellite already in orbit or launch a new satellite. Mission planning must also anticipate the depletion of consumables (e.g., spacecraft fuel, coolant) and schedule resupply missions (14:4). Table 5 details some of the activities and questions which affect mission planning decisions.

TABLE 5

MISSION PLANNING ACTIVITIES/QUESTIONS

Determine when spacecraft will deplete their consumables.

Determine which satellites have failed and which of their components have failed.

Determine what replacement parts are required and where they are located.

Determine if the repair or resupply can be accomplished in a timely fashion and whether a spare needs to be activated or a replacement mission scheduled.

Are the required consumables or components already on-orbit or on the way to orbit and can they be rerouted?

Is a launch vehicle or orbit transfer vehicle available? Can the repair use an already scheduled mission or must a launch or orbit transfer be accomplished just for the repair/resupply?

(6:3-16-3-34)

For support of a failed satellite, mission planning involves determining which electronics modules are required for the repair and whether they are located on-orbit or on the ground. A mission planner determines the mission payload weight and selects an appropriate booster to put the payload in orbit or schedules a transfer vehicle to rendezvous with the failed spacecraft and make repairs (13:37). For resupply missions, a launch or a rendezvous from on-orbit must be scheduled to arrive before or immediately after one or more of the satellite's expendables have been depleted.

Another aspect of mission planning not often discussed is the juggling of schedules. As discussed in the *COSEMS Design Document* (5), not only must special consideration be made for anticipating the depletion of satellite consumables to maintain constellation availability, but additional considerations must be made to repair failed satellites. Assigning and manifesting ground-based support and on-orbit support requires the assignment of priority to spacecraft, launch vehicles, and cargo and then requires action based upon those assignments. The only components or consumables available to repair a satellite may be awaiting ground launch to another satellite or may be already in-flight on its way to a different spacecraft. Mission planners must weigh the importance of the scheduled mission against that of the need to repair the failed satellite. If the scheduled mission is non-critical or the satellite is of less importance, the resupply or repair may be reassigned and rerouted, if possible (5:D-6-D-8). COSEMS uses algorithms to assess such situations and choose among alternatives (13:36).

COSEMS

The Comprehensive Operational Support Evaluation Model for Space (COSEMS) is a discrete-event simulation model as described in the *Simulation* section. It was designed to simulate different support concepts for the Space Defense System (12:1), although it can also be used to simulate other types of constellations. It is written in Ada and has over 100,000 lines of code. Figure 16 illustrates the relationship of COSEMS to its inputs and

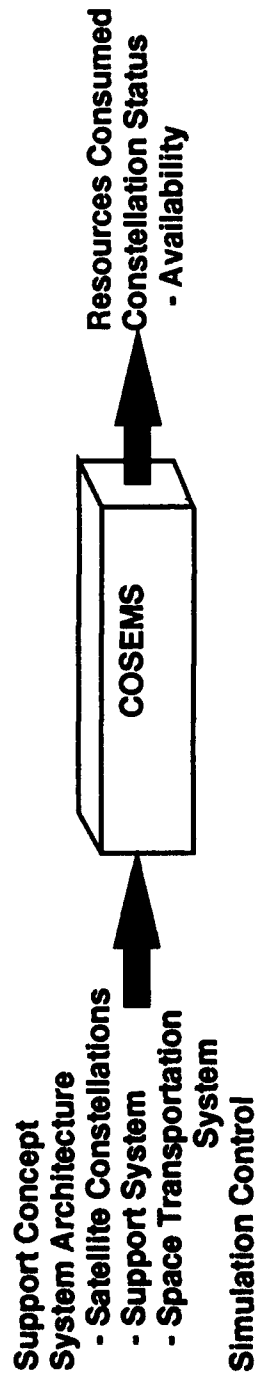


Figure 16. COSEMS Inputs and Outputs (5:2-2)

outputs. The model consists of three elements, the preprocessor, the core simulation, and the postprocessor. The preprocessor allows the user to provide specific input values for several satellite designs, constellation architecture, model parameters, launch vehicles and a host of other functions (6:3-2). The core simulation performs all the modeling of the interactions of the constellation and its support elements. The postprocessor provides summary output of all the statistical data generated by the core simulation and can create data files for use graphics packages and assessment of system costs (6:3-2).

COSEMS simulates all elements of a constellation and its support system. It simulates the satellites and their subsystems, the space asset support system (SASS), its elements, and the space transportation system (STS). SASS consists of space-based support platforms (SBSPs) where orbital replacement units (ORUs) are stored, Orbital Maneuver Vehicles (OMVs) which move between satellites and the SBSPs, and Bulk Fuel Tankers (BFTs) where propellants and other consumables are stored for the OMV to transfer to satellites. The STS consists of the many different classes of launch vehicles, the launch sites at the Eastern and Western Test Ranges (ETR and WTR), and the processing facilities at the sites.

COSEMS Functions and Capabilities. As illustrated in Figure 16, COSEMS has three categories of inputs, the support concept, the architectures of the space system's constellations, and the control parameters for the simulation (5:2-1). These control parameters include the random number seed selection, the level of confidence for statistical calculations, the simulation scenario duration, and the number of replications over which the statistics will be averaged (5:2-1). Outputs consist of the numbers and types of cargo deployed, used, and lost due to launch failures, the number of launches by vehicle type, and the status of the space system including availability and number of failures by subsystem element or consumable depletion. COSEMS does not perform mission effectiveness or cost analysis, only the time varying utilization of resources which influence cost and time varying availability which impacts mission effectiveness (12:37).

COSEMS provides several options for support concepts depending on which support concept is chosen. Figure 17 shows three of the four primary support concepts which may be chosen, 1) on-orbit support from space, 2) on-orbit support from the ground, and 3) satellite replacement. A fourth option is available by choosing none of the three options, no support & no replacement. Within the first three options are further options as shown in the figure below.

For on-orbit support from space, the SASS consisting of SBSPs, BFTs, and OMVs. When a satellite requires servicing, an OMV moves from a SBSP stationed within the orbital plane (for service of in-plane satellites only) or from a separate plane (from which the OMV services several orbital planes). Ground support replenishes the BFTs and the ORUs stored at the SBSP. When SBSPs are located in separate planes, they utilize nodal regression to help make the plane change (5:2-4).

On-orbit support from the ground depends primarily on the space transportation system. COSEMS simulates user specified numbers of launch sites at the Eastern and Western Test Ranges including many different launch vehicles, their turnaround times and failure rates (5:2-6). Cargo manifests take into account launch vehicle lift capabilities as they vary with altitude and inclination. When OMVs are placed into orbit for a ground support, they remain in orbit and can make as many repairs and servicings as their fuel and cargo of ORUs permits.

Satellite replacement refers to no servicing. When a satellite fails, its replacement is launched from the ground utilizing the space transportation system. No replacement or resupply allows for constellations to degrade in capability over time as individual satellites fail. Satellites that have failed modules are labeled "degraded," while those with no remaining back-up modules are labeled "at risk."

As previously stated, COSEMS provides statistics on the constellations' availability over the scenario duration. Figure 18 illustrates how the model accounts for availability. Over the discrete time intervals, satellites fail and are repaired. COSEMS accounts for the

OPTIONS FOR ON-ORBIT SUPPORT FROM SPACE						
	1	2	3	4	5	6
ORU REPLACEMENT SUBSEQUENT TO CRITICAL FAILURE	•	•	•	•	•	•
CONSUMABLE REPLENISHMENT PRIOR TO DEPLETION		•		•	•	•
SATELLITE REPLACEMENT WHEN CONSUMABLES DEPLETED				•		•
SATELLITE REPLACEMENT SUBSEQUENT TO CRITICAL FAILURE DUE TO NONREPLACEABLE MODULE(S)			•		•	•

OPTIONS FOR ON-ORBIT SUPPORT FROM GROUND						
	1	2	3	4	5	6
ORU REPLACEMENT SUBSEQUENT TO CRITICAL FAILURE	•	•	•	•	•	•
CONSUMABLE REPLENISHMENT PRIOR					•	•
SATELLITE REPLACEMENT WHEN CONSUMABLES DEPLETED		•		•		
SATELLITE REPLACEMENT SUBSEQUENT TO CRITICAL FAILURE DUE TO NONREPLACEABLE MODULE(S)			•	•		•

OPTIONS FOR SATELLITE REPLACEMENT			
	1	2	3
SATELLITE REPLACEMENT WHEN CONSUMABLES DEPLETED	•		•
SATELLITE REPLACEMENT SUBSEQUENT TO CRITICAL FAILURE		•	•

Figure 17. COSEMS Support Concept Options (5:2-5)

time in service for each satellite and the time during which each satellite is unavailable.

When a satellite is repaired, COSEMS begins tracking its time-in-service once again.

COSEMS characterizes the reason for the failure (subsystem failure or consumable depletion) and calculates constellation availability (as shown in Figure 18) over each time interval as specified by the user.

$$\bar{A}_0 = \frac{\text{Mean Time Between Failures}}{\text{Mean Down Time} + \text{Mean Time Between Failures}} \quad (42)$$

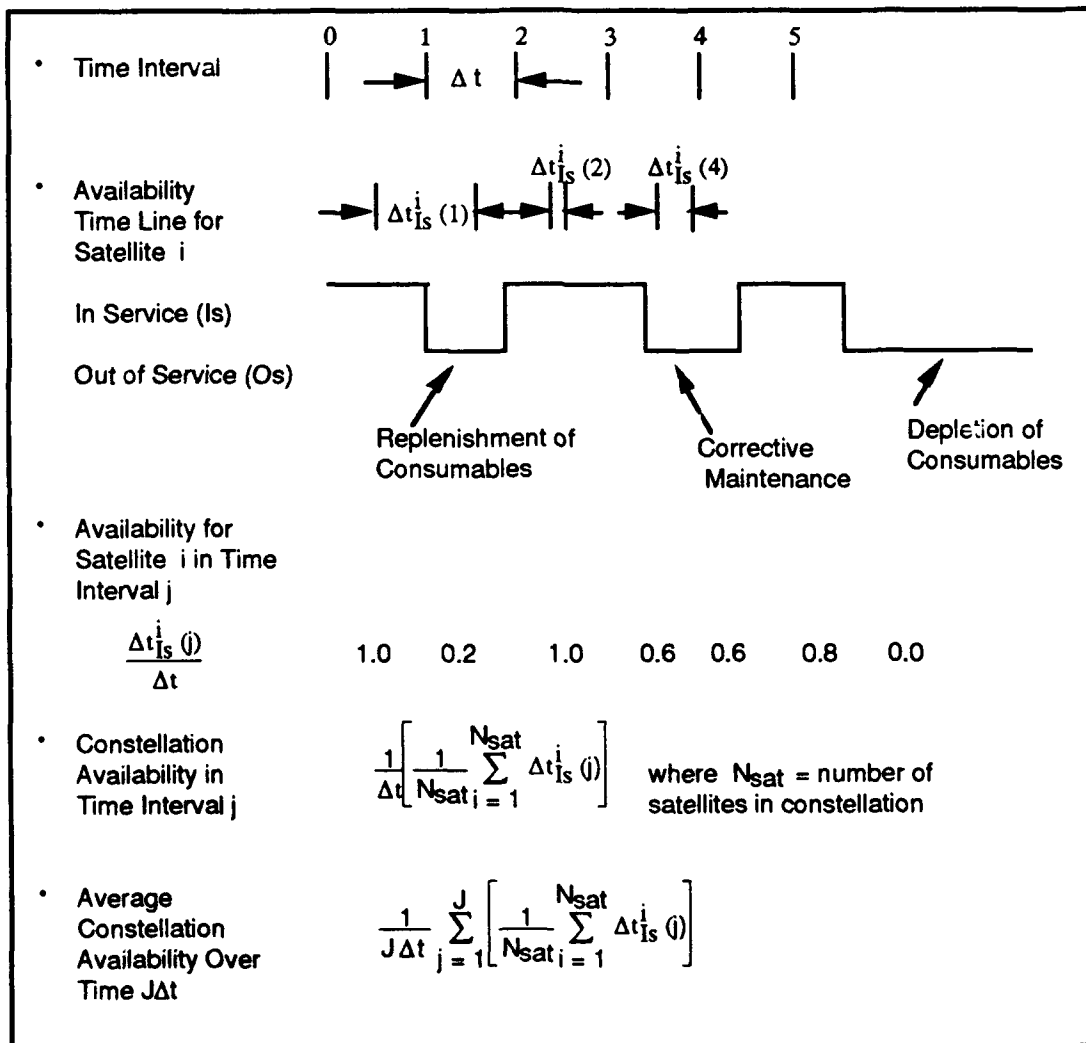


Figure 18. Availability in COSEMS (5:2-3)

After running the simulation for the specified time period, COSEMS produces output consisting of constellation availability per time interval specified by the user (week, month, quarter, or year), cumulative availabilities, ring operational availabilities, number of satellite failures and failure cause, hardware failure by subsystem type, SASS failures, number of missions, launch failures, and resources consumed (see Appendix C). Graphics files can be utilized by another program called COSPLOT to produce bar charts of which include availability, failures, down time, number of missions, and weight of resources consumed (6:8-16-8-18). This program also provides snapshots of the constellation's availability and status at specified times in the simulation. These snapshots are accompanied by latitude and longitude representations of the different satellites' locations in space above the earth. (6:8-12, 8-20).

COSEMS Assumptions and Limitations. In order for COSEMS to perform its simulation as quickly as possible and to limit its complexity, many simplifying or clarifying assumptions were made. Due to computational constraints COSEMS has certain limitations. These include: (6:4-1-4-4)

1. Satellite replacement/No support are the only support options available for simulation of 1,000-10,000 satellites. All support options are available for simulations involving less than 1,000 satellites.
2. All SBSPs and BFTs are 100 percent reliable.
3. The refueling of OMVs and the loading/unloading and docking of OMVs is 100 percent reliable.
4. SASS elements are deployed in a specific time period for space-based support concepts. They are deployed according to a time schedule for ground-based concepts.
5. SBSPs do not use consumables.
6. BFTs are co-orbital to their SBSPs.
7. Command, control, communications, and telemetry support is available on demand.
8. Launch vehicles and their payloads are always ready for integration and launch.
9. The time from launch until orbit is negligible when compared to launch processing times.

10. No consideration is given to payload and launch fairing sizes. Weight is the only restriction for loading cargo.
11. All launches for COSEMS simulation have priority.
12. There is no limit to launch vehicle inventory.
13. Satellite modules are not passively redundant. As soon as the satellite is operational, all modules including backup modules may fail. Spares located on SBSPs are passive and their "failure clock" does not start until they are installed onto a satellite.
14. Satellite module failures are uncorrelated.
15. The true anomalies of satellites are not recorded. When critically failed satellites are replaced, the replacements occupy the same position as the failed satellite.
16. Nodal regression of OMVs is not available for non-circular orbits.
17. Hohmann transfers are only used when a window is immediately available. Otherwise bi-elliptic transfers are always used. When a Hohmann transfer is initiated, any associated plane change occurs at the highest possible altitude to conserve Δv .
18. If the phasing delay for a bi-elliptic transfer exceeds a user-specified maximum, the simulation terminates immediately.
19. The non-spherical earth of the earth's effect of nodal regression and apsidal rotation are the only perturbations accounted for by COSEMS.
20. Launch sites are restricted to the Eastern and Western Test Ranges.
21. OMVs cannot be refueled in the ground support mode. Otherwise ground support would eventually transition to space-based support.

The assumptions and limitations affect how the constellations and their support systems are modeled. Those detailed above are the primary assumptions and limitations which will affect the user's input parameters and the analysis of that output. The complete description of assumptions and analysis is available in the Reference (6).

Summary

Using or verifying and validating the Comprehensive Operational Support Evaluation Model for Space requires an understanding of reliability, orbital mechanics, and mission planning. By using known probability functions, the reliability of system components can be predicted and factored together to predict and model the reliability of an entire system.

The physics of orbital mechanics involves complicated equations which can be simplified to reveal the major characteristics of orbits and allow the modeling of the orbital motion of satellites and the transfer orbits of OMVs. Mission planning synthesizes these concepts and requires determination of the state of the constellation and the disposition of the logistical resources available. Repairs and required resources must be anticipated through an understanding of reliability, while orbital mechanics allows the determination of the feasibility of getting satellites, their components, and consumables into orbit for repair and resupply missions.

All these concepts provide the basis for the simulation. In order to properly choose the parameters in COSEMS for support simulation, the user must be familiar with this basis, as well as the assumptions and limitations of the model and the concepts which the simulation relies upon.

III. Methodology

Introduction

COSEMS contains default cases for low, medium, and high-altitude constellations. The model was run with these default cases and with other test cases, so that the input and output parameters of the different program menus provided the means to verify the program's functions and execution of case studies and validate its use.

In order to develop benchmark cases for COSEMS verification and validation, several orbital architectures were chosen whose reliability and orbital geometry parameters and life-cycle behavior are known. Additional models were chosen with extreme orbital parameters to test the limits of the program. The reliability, orbital mechanics, and mission functions were verified using both structured traces and repetitive testing with different input parameters. These testing methods were used to determine how well the modules interact and pass data to each other.

Tracing the flow of the program and determining which sub-programs are called for different tasks was accomplished with the Source Code Analyzer (SCA) utility resident on the VAX computer. By compiling the COSEMS code with compilation flags added for use with the SCA utility, a listing was obtained of the program's calls to each subprogram. This allowed investigation of the exact order in which certain coding calls were made by the COSEMS's main program and the order in which special subroutine and functions were utilized by the program.

The reliability module was validated using comparison to other models and face validation techniques. The validation used the expertise of space system engineers and a comparison of COSEMS's output against a SLAM II reliability model.

The orbit mechanics module was validated by comparison to other models and tracing techniques. The output was compared to output of an existing valid model used by the Air Force and NASA in order to test the module's ability to calculate orbital dynamics and the

position of spacecraft in orbit over long periods of time. The orbital mechanics and orbit transfer transfer algorithm was tested by tracing the code and comparing it to behavior predicted by established orbital transfer mathematical techniques.

The mission planning module was validated with face and trace validation techniques. The assumptions and decision routines that form the basis of the mission planning modules was validated using trace and face validity techniques.

The actual determination of whether COSEMS is a valid and accurate decision making tool was made by a synthesis of the results of the module testing and an evaluation of the ease of use of the simulation output.

Random Number Generation

One cannot simply use a random number generator in a simulation and assume that it provides a number stream that appears truly random. For the uniform [0,1) generator, we require that the numbers appear to come from identical, independent uniform distributions. If it appears to do anything else, the program which relies on these numbers does not reflect the real world and produces skewed results. Tests must be performed on the generator to show that its frequency distribution looks uniform and also that the numbers appear random. Groupings of numbers and apparent trends in the number generation is undesirable. Several tests were accomplished to provide a more confident analysis of how well the generator worked.

Many other tests were detailed by Knuth (16), Chandrasekaran (4), Ravindran (24), and the IMSL Stat/Library (26), but the following tests were the most common and work well at detailing how well a random number generator worked. Using more tests or different ones is simply a matter of discretion.

Frequency Tests. Plots were made for 1000 random numbers for each of the ten seed numbers used by COSEMS in order to visually check for uniform distribution. The 10

streams of 1000 numbers were generated by writing an Ada program that utilized COSEMS own code which in turn accessed the VAX random number generator.

Additionally, two types of frequency tests were performed on each of the 10 streams of 1000 numbers. The Kolmogrov-Smirnov (K-S) one-sample test for continuous distributions and the chi-square (χ^2) goodness-of-fit test was performed. For the chi-square test, 3 runs for each stream were performed, one with 10 cells, one with 20 cells, and one with 30 cells. Several different rules of thumb for choosing cell size were found, depending on which source was cited. The tests could be performed with cells of constant size or of equal probability. Since the probability of any number from a uniform [0,1) distribution is that number, $P(x) = x$, cells of equal size also have equal probability. Therefore several tests with differing cell size were chosen, in the same manner as detailed in Chandrasekaran (4), in order to demonstrate that the test results were independent of cell size, to test local behavior, and to support the randomness hypothesis for the global properties of the generator (4:31). The K-S test was included in the battery of tests to increase the confidence in the test results.

In each case the tests were performed by writing FORTRAN code which accessed the data files containing the streams of 1000 random numbers generated by the Ada program and which then called on IMSL Stat/Library routines to accomplish statistical analysis. For these tests the null hypothesis was that the generator produces numbers from a uniform distribution. When cell size became a factor of the test, more than one test was performed to demonstrate robustness and an insensitivity to the cell size.

The Chi-Square Test. Chi-square tests apply to both continuous and discrete distributions. To perform this test, n observations were divided into k categories. For $i = 1, 2, 3, \dots, k$, f_i is the number of random numbers that fall into the i^{th} category, $\frac{n}{k}$ is the expected number of counts in each category, and the chi-square statistic is

$$\chi^2 = \frac{k}{n} \sum_{i=1}^k (f_i - \frac{n}{k})^2 \quad (43)$$

For large n , χ^2 approximates a chi-square distribution with $k-1$ degrees of freedom (17:437). These statistics provide a measure of how well the distribution in question matches the desired distribution. χ^2 statistics are tabulated for specific probabilities and degrees of freedom. By choosing a certain probability of error, α , a tabulated statistic can be chosen for a distribution with a specified degree of freedom. If the calculated statistic is less than the tabulated statistic, α is the probability that the calculated value occurred by chance. The lower the statistic, the more likely the null hypothesis (the distribution matches the uniform distribution) (4:31).

For the purposes of validating the random number generator the χ^2 test was performed for 10 categories with 9 degrees of freedom, 20 categories with 19 degrees of freedom, and 30 categories with 29 degrees of freedom.

The Kolmogrov-Smirnov Test. The Kolmogrov-Smirnov test applies to continuous distributions such as the uniform distribution and is based upon the difference between the real distribution function, $F(x)$, and $F_n(x)$, the function being scrutinized (16:41). This test has certain advantages over the χ^2 test: 1) the grouping of data does not affect the test, 2) sample size does not affect the test, and 3) the test is more powerful (17:387). The main disadvantage is the requirement for complicated sets of formulae required to calculate the critical values needed for forming the statistic. (17:387).

For a uniform [0,1) distribution the probability of x , $P(x)$, is x for $0 \leq x \leq 1$. By making n independent observations of the quantity x , we obtain the empirical distribution function $F_n(x)$ where

$$F_n(x) = \frac{\text{number of } X_1, X_2, \dots, X_n \text{ which are } \leq x}{n} \quad (44)$$

To make the test, we use the two statistics

$$K_n^+ = \sqrt{n} \left(\max_{-\infty < x < \infty} (F_n(x) - F(x)) \right) \quad (45)$$

$$K_n^- = \sqrt{n} \left(\max_{-\infty < x < \infty} (F(x) - F_n(x)) \right)$$

K_n^+ measures the greatest amount of deviation when F_n is greater than F , while K_n^- measures the maximum deviation when F_n is less than F (16:43). \sqrt{n} magnifies the statistic in such a manner that the standard deviation is independent of n . This makes the K-S test desirable for testing of continuous functions, since that calculation of statistics is not dependent on n . As in the chi-square test, the values for K^+ and K^- are looked up in a table to determine whether the statistic is too high or too low, indicating a poor fit.

Randomness Tests. Two types of randomness tests were performed, the runs test and the d^2 test. Two different runs for these were performed. A runs-up and a runs-down test was performed for each random number stream, while d^2 tests were performed with 10 cells, 20 cells, and 30 cells. In these tests the null hypothesis is that the generator produces a stream of random numbers. Multiple tests were performed to demonstrate robustness and insensitivity to cell size.

In each case the tests were performed by writing FORTRAN code which accessed data files containing the streams of 1000 random numbers generated by the Ada program and which then used the IMSL Stat/Library routines to accomplish the statistical analysis.

The Runs Test. Sequences of random numbers may be tested for “runs up” and “runs down” in order to make a statistical inference on its randomness (16:61). In order to accomplish this test the numbers are examined in sequence $x_1, x_2, x_3, \dots, x_j, x_{j+1}, \dots, x_n$. For a “runs up” test one breaks up the sequence whenever the x_{j+1}^{th} number is greater than the x_j^{th} (24:624).

For example a sequence of

$$1\ 3\ 10\ 2\ 3\ 9\ 8\ 4 \quad (46)$$

would be broken up to look like

$$1\ 3\ 10\ |2\ 3\ 9|\ 8\ |4 \quad (47)$$

This example has two runs up of length 3 and two of length 1. For a "runs down," the break up of the sequence is done for decreasing sequences. The run shown in the above expressions would then have three runs of length 1 and one of length 2 and two of length 3.

A chi-square test cannot be directly applied because the adjacent runs are not independent of each other (16:60). Instead the following statistic is used to calculate a chi-square statistic with 6 degrees of freedom:

$$V = \frac{1}{n} \sum_{i=1}^6 \sum_{j=1}^6 (\text{count}(i) - nb_i)(\text{count}(j) - nb_j) a_{ij} \quad (48)$$

where the coefficients of a_{ij} and b_j are

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} = \begin{bmatrix} 4529.4 & 9044.9 & 13568 & 18091 & 22615 & 27892 \\ 9044.9 & 18097 & 27139 & 36187 & 45234 & 55789 \\ 23568 & 27139 & 40721 & 54281 & 67852 & 83685 \\ 18091 & 36187 & 54281 & 72414 & 90470 & 111580 \\ 22615 & 45234 & 67852 & 90470 & 113262 & 139476 \\ 27892 & 55789 & 83685 & 111580 & 139476 & 172860 \end{bmatrix} \quad (49)$$

$$[b_5 \ b_2 \ b_1 \ b_3 \ b_4 \ b_5] = \left[\frac{1}{6} \ \frac{5}{24} \ \frac{11}{120} \ \frac{19}{720} \ \frac{29}{5040} \ \frac{1}{840} \right]$$

Examination of the statistic shown in Equation (48), illustrates that counts are taken for runs of lengths one up through runs of length six. Runs of six and greater are grouped together (16:60). While these coefficients are approximate, the runs test routine from the IMSL Stat/Library contains the complete and exact algorithm detailed by Knuth (26:466).

For the runs-down test, the numbers in the stream were multiplied by -1, allowing the runs test algorithm to examine the streams for "runs down."

The d^2 Test. The d^2 test is a serial test which specifically tests for randomness in uniform distributions. This test was performed on the 10 number streams under the assumption that the frequency tests showed that they were truly uniform. The d^2 test computes a comparison statistic for succeeding quadruples of random numbers which are then used to compute an approximate chi-square statistic.

Consider four random numbers $x_1, x_2, x_3,$ and x_4 and that

$$D^2 = (x_3 - x_1)^2 + (x_4 - x_2)^2 \quad (50)$$

The probability of D^2 is given as

$$P(D^2 \leq d^2) = d^2\pi - \frac{8d^3}{3} + \frac{d^4}{2} \quad D^2 \leq 1$$

or

$$P(D^2 \leq d^2) = \frac{1}{3} + (\pi - 2)d^2 + 4\sqrt{d^2 - 1} + 8 \frac{(d^2 - 1)^{\frac{3}{2}}}{3} - 4d^2 \arctan\left(\frac{\sqrt{1 - \frac{1}{d^2}}}{\frac{1}{d}}\right) \quad D^2 > 1 \quad (51)$$

For each set of quadruples in the number stream, the cumulative probability is calculated using Equation (51) and placed into one of k categories as in the chi-square test (26:473). If the expected value of d^2 values in each category is $\frac{n}{k}$ and f_i is the observed count of values, an approximate chi-square statistic with $k-1$ degrees of freedom can be calculated with Equation (43). The d^2 test was performed for 10 categories with 9 degrees of freedom, 20 categories with 19 degrees of freedom, and 30 categories with 29 degrees of freedom.

Parameter Evaluation. For an added measure, a parameter evaluation of the random number generator was performed by calculating the mean and variance of the streams generated by each seed. The 10 statistics were then combined into one average

mean and variance. These statistics were then compared to their theoretical value using a two-tailed t-test with 999 degrees of freedom. For the COSEMS random number generator [a, b) is [0,1). The null hypothesis is that the parameters indicate a uniform distribution with

$$\text{mean} = \frac{a+b}{2}, \quad \text{variance} = \frac{(b-a)^2}{12} \quad a \leq x \leq b \quad (52)$$

Interpretation of Data. Interpreting the statistics for the frequency and randomness tests and the parameter evaluation can be subjective depending on the confidence interval chosen. In order to make it easier to infer the “goodness” of the random number generator and to combine statistics with differing degrees of freedom, all of the test statistics were converted back to their percentile point value or p-value. In essence, the lower the statistic, the more exact the specified distribution is to uniformity or to randomness, that is the more likely the null hypothesis is to occur. The percentile point is the probability that the observed values of the statistic or higher values can occur by chance (4:31). The percentile point will be large for small values of χ^2 . Additionally, too high or too low a p-value should be considered unlikely. For this analysis the null hypothesis is accepted if the p-value lies between 0.05 and 0.95.

Average values for the frequency and randomness tests were accomplished by averaging the p-values. The overall performance is measured as a percentile point of the following statistic:

$$\chi_{\text{overall}}^2 = 2 \sum_{i=1}^n -\ln P_i, \quad 2n \text{ degrees of freedom} \quad (53)$$

where n is the total number of tests and P_i is the percentile point for the i^{th} test. This test is used if the experimenter has designed a series of experiments designed to test the same hypothesis and wishes to make an over-all evaluation of the experiments (28:43). The test is valid given that the observed probabilities are a random sample from a population with a

mean of 0.50 (28:44). By calculating this approximate chi-square distribution statistic with $2n$ degrees of freedom, the overall performance of the random number generator can be determined for each stream.

Reliability

COSEMS performs reliability computations for several elements including spacecraft modules, Orbital Maneuver Vehicles (OMVs), repair missions, and launch vehicles. Whereas any single failure for a launch vehicle, OMV, or resupply mission consists of calculating a single time-between-failures, the failure mode for spacecraft is more complicated. The reliability functions for launch vehicle failure, resupply/repair, and OMV failure was verified and validated using a trace. Since the simulated spacecraft have several parallel modules operating in series, verification was accomplished by tracing the code and validation was done using face validation and comparison to other models. Face validation was performed by first making several runs of COSEMS simulating constellations of 10 and 20 satellites with on-orbit support judging whether the output was reasonable.

Comparison to other models was performed by writing a program in SLAM II. SLAM II is a FORTRAN-based simulation language which supports discrete-event modeling (23:63). The model assumes that the constellation is fully deployed on Day One and runs for ten years and that there is no repair of satellites. Fuel consumption of the satellites is kept low so that they do not run out of fuel for those ten years. The SLAM II program was designed to take yearly failure and ten-year average failure statistics in the same manner as COSEMS. Statistics were collected over 200 replications for a constellation of 4 satellites.

After the runs were completed, the SLAM II output was compared to the COSEMS output using the Z statistic. Due to the large samples taken, the Central Limit Theorem states that the population of all the replications combined is approximately normally distributed (20:317). Using a null hypothesis that the means of COSEMS and the SLAM II

program are identical and assuming that the population variances are identical, a statistic for comparison can be calculated using

$$Z = \frac{(\bar{Y}_1 - \bar{Y}_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (54)$$

where \bar{Y}_1 and \bar{Y}_2 are the calculated means output by the programs, σ_1 and σ_2 are the standard deviations of the two outputs, and n_1 and n_2 are the number of observations from each of the two models. The Z statistic can then be converted to a p-value. Values between 0.05 and 0.95 indicate that the models provide similar reliability estimation.

Orbital Mechanics

The orbital mechanics model of COSEMS was verified through trace verification and by dynamic testing utilizing parameter variation. Validation was performed using trace validation and comparison to other models.

Dynamic testing and comparison to other models was accomplished by adding code to COSEMS's orbital mechanics module which would output to a data file a satellites orbital elements over time. Dynamic testing was accomplished by running COSEMS for one satellite in a 400-km periapsis orbit with eccentricity of 0.1 and for runs with inclinations of 0°, 30°, 63.4°, and 90°. The same cases were also run for circular orbits. This tested the model's ability to model different types of orbits. Simple models cannot normally model both 0° and 90° orbits with the same code because the calculation of orbital elements will usually fall apart at either of those extremes unless additional coordinates are used to define the orbit. The 63.4° orbit was performed to check apsidal rotation.

Comparison to other models was accomplished by using the Long-Term Orbit Predictor (LOP), a code from the Jet Propulsion Laboratory. LOP can simulate orbits about any planet and include the gravitational effects of third bodies, air drag, and non-spherical geopotentials. LOP was run for the same cases with all perturbative effects turned off, and

the orbital elements of the two models were compared. Since COSEMS uses a very simple orbit model, no statistics of comparison are needed. If the COSEMS output is similar to that of LOP, the handling of orbital mechanics for the satellites should not vary greatly from LOP. LOP cannot model equatorial orbits, so the COSEMS runs for 0° were only used to verify how well COSEMS handled equatorial orbits.

Orbit Transfer. The orbit transfer routines, intra-orbit and bi-elliptic transfer, were both verified using traces. Additionally, the bi-elliptic transfer algorithm was coded in FORTRAN and run for a plane change 5° during a transfer from a 100-km orbit to a 1000-km orbit. The run was repeated for transferring from the high orbit to the low orbit. A mathematical derivation of the plane split equations was performed to verify the code.

In order to verify that making a 3-split orbit transfer—where each change in velocity for the transfer is accompanied by a plane change and the sum of the plane changes equals the total plane change required—the code was run for the case of transferring from a 100-km altitude orbit to a 1000-km altitude orbit while making a 5-degree plane change. Additional runs were made after the code was altered to force a single plane change—three runs were made where the plane change was made entirely on the first change in velocity, the second, and then the third. Three runs were done for a 2-split transfer—equal plane changes were performed during two of the changes in velocity. For this last case each plane change was one-half the total plane change. Any angle optimization for a 2-split provides a total velocity change with an upper bound of the velocity provided by two plane changes of one-half the required plane change. Runs were made for plane changes on the first and second changes in velocity, the first and third, and the second and third.

Plots of total Δv against the intermediate transfer orbit radius were done for 1, 2, and 3-split plane changes to verify that the 3-split provides the optimum total change in velocity required. Furthermore, additional runs for the 3-split were done for a 5 degree plane change with a transfer from a 100-km orbit to a 200-km orbit, a 100-km orbit to a 500-km

orbit, and a 100-km orbit to a 150-km orbit. The output was checked to make sure that it seemed reasonable given the input parameters and orbit characteristics.

Mission Planning

Verification for the mission planning routines for launch vehicle selection, resupply, and repair missions were performed by traces. Since a space transportation system with priorities and the ability to rapidly respond to critical failures does not exist, all validation was done by face and trace validation. The names of the COSEMS mission planning subprograms verified and validated are listed in Appendix A.

The subprograms governing launch vehicle selection, repair mission planning, and resupply planning were verified by checking the code; traces were done to verify that launch vehicles were selected properly and that the scheduling of resupply and repair missions were accomplished as detailed in the *COSEMS Design Document (5)*. Several runs were made with satellites of different masses and orbiting at different altitudes to check that these subprograms worked properly.

Validation was performed by determining whether the model's assumptions seemed reasonable and realizable. The logic of the mission planning flowchart's as detailed in the *Design Document (5)* was examined to see if its behavior was reasonable and that the program actually performed as detailed by the flowcharts. Further face validation was performed by examining whether the outputs of all the runs performed for the reliability section and orbital mechanics section were reasonable when compared to the input parameters. Finally the assumptions were checked to make sure that they are realizable; while the output might reflect the assumptions, it may still not make sense to utilize such an assumption (operating with certain assumptions may be counter-intuitive, not possible in real world situations, or may preclude certain solutions by excluding them from a simulation).

The validation of the assumptions for mission planning included all the assumptions upon which the entire model is based. If these are not reasonable, do not make sense, or cannot be implemented in a real world situation, the model is not valid. Face validation was therefore performed upon the base assumptions of COSEMS to show whether the input/output relationships make sense and can be utilized as an aid in decision making.

COSEMS Functions and Capabilities

A determination of the utility of COSEMS output and a validation of the model as a whole was performed by an analysis of its input and output. The evaluation was performed by running COSEMS for several different scenarios and determining how easily the simulation was to set up the input parameters. This is a subjective evaluation of how intuitive the interface is, how easily case studies are run with the help of the *COSEMS Users Guide* (6), and how useful the simulation's output is to evaluating alternative space support concepts.

IV. Results and Analysis

Introduction

This chapter is devoted to detailing and explaining the results garnered from the methodology described in the preceding chapter. In all cases what the code and the trace of the code reveals is discussed first, while the performance of the code and the tests performed on the code is detailed second. A complete listing of the COSEMS subprograms (7) used in this investigation are listed in Appendix A.

It should be noted that while COSEMS Version 5.1 runs properly on the most current version of the VAX operating system, during compilation numerous informational errors were flagged. These errors only affect print routines, and the VAX Ada compiler successfully expands the code to eliminate the problems. However, this indicates that the COSEMS code is not only machine dependent, but may also be dependent on the version of the operating system is in use by the VAX computer.

Random Number Generation

An examination of COSEMS shows that it uses the random number generator resident in the VAX computer's Run-Time Library (VAX RTL). All probability distributions, whether they are uniform, exponential, Weibull, or normal, come from this generator (7). COSEMS uses the computations presented in Table 1 to perform the inverse transform. A trace of the code shows that the inverse transforms are properly done.

It should be noted that the use of the VAX RTL specifically ties the COSEMS code to a particular type of machine. In spite of any other modifications to the program's code, it cannot run on another type of machine unless the code for the random number generator is updated. Moreover, should Digital Corporation make any major changes in their VAX RTL code or should the random number generator on a user's VAX become altered or corrupted, the random number generator would no longer be valid. Chandrasekaran details

specific Ada random number generators that can easily be incorporated into any Ada program (4:35). These number generators have been extensively validated and can be ported to any machine running Ada.

A histogram of 1000 random number variates from COSEMS's random number generator provides a first glimpse at its uniformity. While Figures 19 and 20 show that the frequency distributions from the seeds used by COSEMS appear uniform, this is not conclusive.

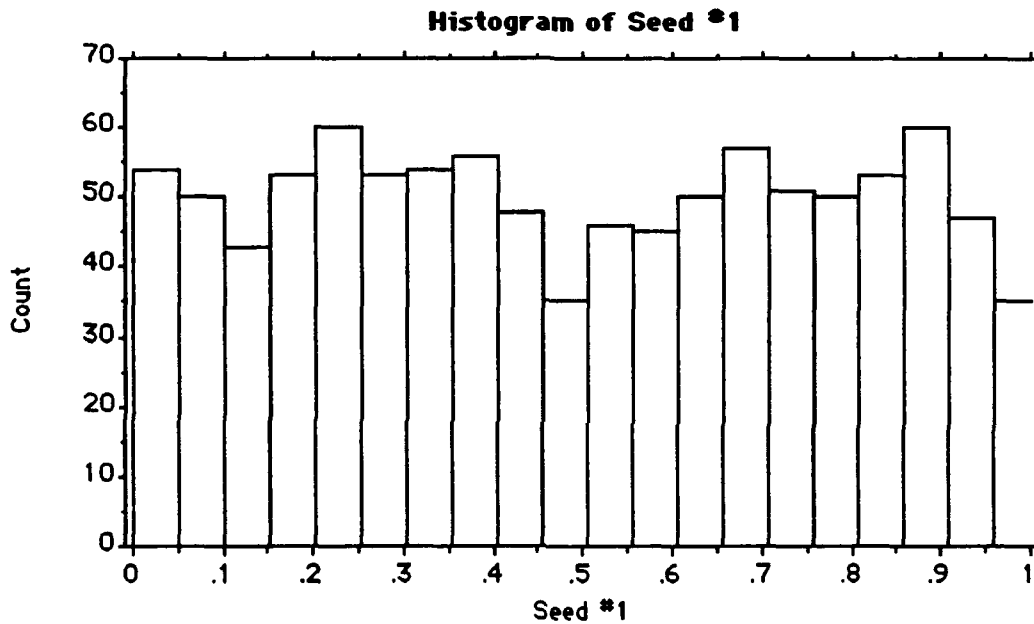


Figure 19. Frequency Distribution of Random Numbers from Seed #1

The uniformity of the random number distribution for these two figures is representative for all ten seeds. Increasing the number of the cells over ten did not change the observed uniformity. Varying the number to less than ten improperly imposes the appearance of uniformity. Since the histograms were inconclusive, further statistical analysis was performed using the Kolmogrov-Smirnov (K-S) and chi-square tests to test for uniformity, and the runs-up, runs-down, and d^2 tests to test for randomness.

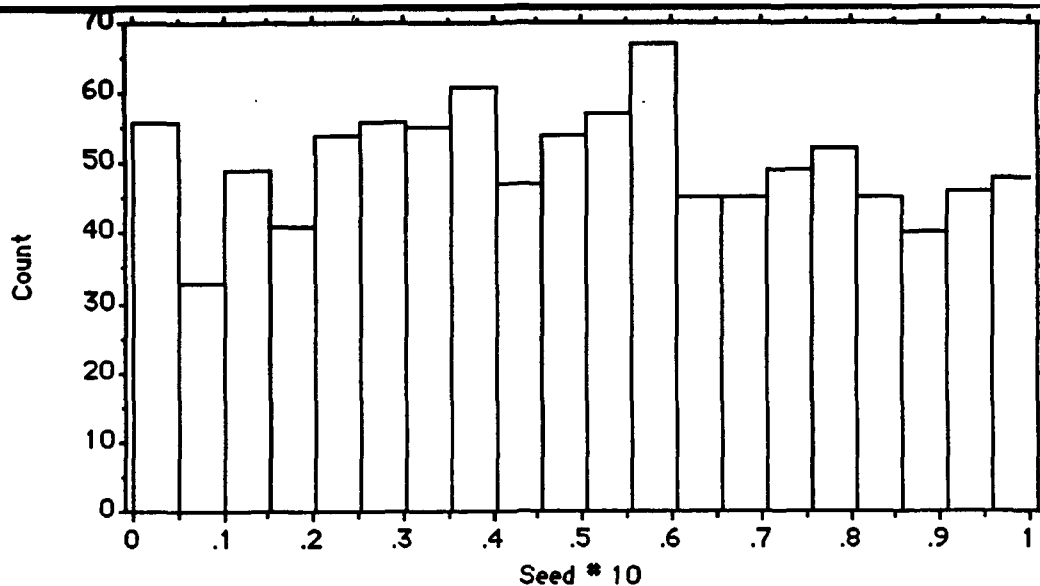


Figure 20. Frequency Distribution of Random Numbers from Seed #10

The percentiles calculated by the frequency and randomness tests are shown below in Table 6. As shown, three values for the K-S test could not be calculated because the test requires a continuous distribution and the random number stream produced duplicate values (ties) which are not allowed by the algorithm. High and low percentile values are ignored (these are shown in boldface) as unlikely; therefore the observed distribution is considered to match the uniform distribution or appear random if the statistic is between 0.05 and 0.95. A more stringent confidence interval requiring statistics to be between 0.1 and 0.9 can be used as done by Chandrasekaran, but this does not affect the results of the test.

The indications of the individual tests performed on each of the ten random number seeds show that the numbers come from a uniformly distributed random number stream. Furthermore, varying the cell size for the χ^2 and d^2 tests does not significantly change the acceptance of the null hypothesis, that the random number generator produces random, uniformly distributed numbers.

TABLE 6
P-VALUES FROM STATISTICAL ANALYSIS OF THE
COSEMS RANDOM NUMBER GENERATOR

Seeds	K-S	Frequency			Randomness					
		χ^2	χ^2	χ^2	Runs	Runs	d ²	d ²	d ²	
		(10 cells)	(20 cells)	(30 cells)	Up	Down	(10 cells)	(20 cells)	(30 cells)	
1	19880801	0.4674	0.6163	0.8344	0.9208	0.0516	0.4262	0.7399	0.9401	0.4588
2	2000720065	0.3740	0.5811	0.9111	0.6282	0.0370	0.2678	0.8237	0.5654	0.4713
3	-1158007263	0.0327	0.7055	0.0322	0.2774	0.0574	0.2393	0.7238	0.8751	0.7009
4	1626622849	0.1264	0.5811	0.1700	0.0772	0.0001	0.4145	0.3191	0.7647	0.7928
5	969330913	tie	0.3473	0.5198	0.8567	0.1711	0.0994	0.2676	0.1992	0.2335
6	1376372289	tie	0.7849	0.3324	0.3986	0.6711	0.0385	0.5022	0.5978	0.8235
7	-229299295	tie	0.2743	0.1382	0.3739	0.0306	0.1054	0.8237	0.4906	0.7820
8	-1623129855	0.2557	0.0972	0.3151	0.3533	0.3731	0.4255	0.5997	0.7550	0.1071
9	426067553	0.2480	0.7299	0.8733	0.6926	0.3323	0.2872	0.4484	0.4594	0.5868
10	1566178241	0.2626	0.1365	0.2119	0.1799	0.7466	0.0729	0.7943	0.4492	0.5997

tie—Kolmogrov-Smirnov algorithm does not allow ties

TABLE 7
AVERAGE AND OVERALL P-VALUES
FOR COSEMS 'S RANDOM NUMBER STREAMS

Seed	Frequency	Randomness	Overall (w/o K-S Test)
1	0.8266	0.5233	0.8939
2	0.7171	0.4330	0.6334
3	0.2706	0.5193	0.3376
4	0.2703	0.4582	0.0308
5	0.5746	0.1942	0.2051
6	0.5053	0.5266	0.6878
7	0.2621	0.4465	0.2061
8	0.2692	0.4521	0.4049
9	0.6980	0.4228	0.8756
10	0.2634	0.5325	0.4535

For further indication of the random number generator's uniformity and randomness, the average percentiles for the frequency and randomness and for the overall performance was calculated and is shown in Table 7. The overall performance was calculated using Equation (53) on page 52, but does not include the percentiles generated from the

Kolmogrov-Smirnov test. This was done because the K-S test is totally different than the χ^2 test. Since a linear series of χ^2 numbers approximates a χ^2 distribution, the K-S percentiles must be excluded.

As the p-values in Table 7 indicate, the null hypothesis is accepted. This strongly indicates that the random number generator produces random and uniformly distributed numbers.

Parameter Evaluation. As a final test, the uniform number generator was tested by parameter evaluation. By calculating the mean and standard deviation from several of the number streams, it is shown in Table 8 that the random number generator produces numbers with the correct mean and standard deviation for a uniform distribution. The percentile point indicates that the generator produces numbers from a random uniform distribution.

TABLE 8
PARAMETER EVALUATION
OF THE RANDOM NUMBER GENERATOR

Parameter	Given	Computed	Percentile Point
Mean	0.5000	0.498	0.8123
Variance	0.0833	0.082	

In the final analysis, all tests on the random number generator indicate that it has excellent characteristics and produces good pseudorandom numbers. COSEMS uses pseudorandom numbers and performs proper inverse transforms to turn the uniformly distributed numbers into whichever distribution the simulation requires. The only drawback is the code's reliance on a random number generator exterior to the program itself. The incorporation of a short procedure would allow COSEMS to produce random numbers independent of the computer the code resides on.

Reliability

The trace of the COSEMS code which handles module, launch vehicle, OMV, SBSP, and BFT failures indicates that it correctly determines failure events. Failure events are properly identified and placed in a queue from which repair/replacement missions are scheduled. The time increments are based upon these events; the program moves to the next event in the queue, determines the actions required, and then performs those actions. Furthermore, the program tracks how many parallel backup modules each subsystem contains. If a module fails, a telemetry switch to a working backup is attempted. When the primary module and all its backups have failed, the satellite fails. The trace of the programs governing these functions indicates that COSEMS correctly implements reliability as explained in its design document.

The trace also reveals that reliability is modeled correctly. The trace of the subprograms that handle reliability revealed hints at how to properly implement a SLAM II program to simulate reliability in a constellation of four satellites. The flowchart for this program is presented in Figures 21a, 21b, and 21c. These figures show the program flow for one satellite in the constellation and for the ancillary networks which track time-in-service, number and type of failures, and overall constellation availability.

The SLAM II program simulates 4 satellites. Each satellite is identical and has four subsystems. The first subsystem has only one module, the second and third have two redundant modules, and the fourth has four redundant modules. The failure of all the modules in any subsystem causes the failure of the satellite. The scale and shape parameters for the Weibull distribution used by the SLAM II simulation are shown in Table 9 and are identical to the default values used by COSEMS. Counters keep track of the total time each satellite is available in accordance with the time-lines and formulae shown in Figure 18, page 41. Availability is tracked on a year-to-year basis for a total of ten years, and the program outputs statistics on the number of hardware failures for each system and the

availability of the constellation as a whole. These statistics are taken over 200 replications of the simulation to ensure that the failures occur in a pseudorandom manner.

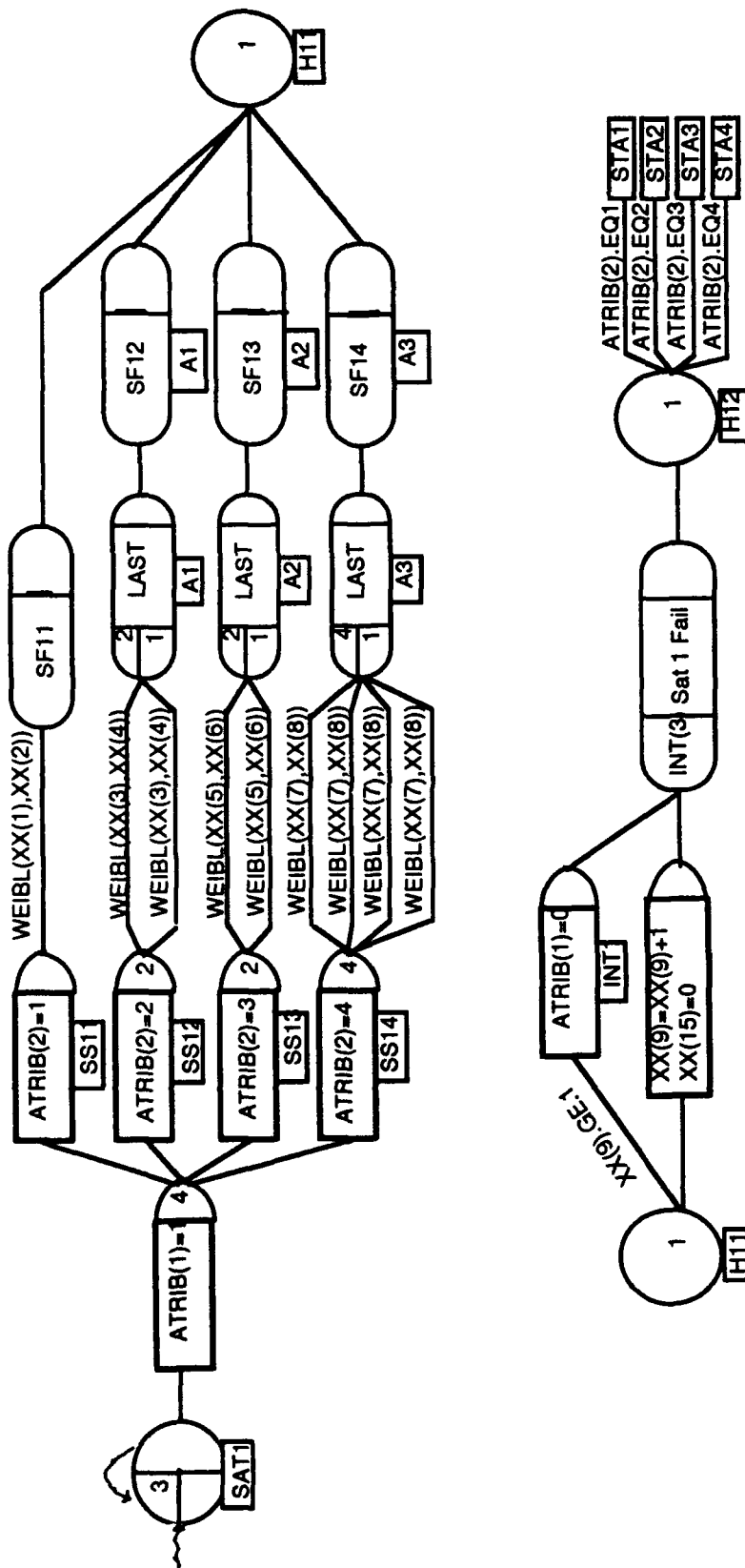
TABLE 9

SATELLITE SUBSYSTEM FAILURE PARAMETERS

	Scale (yrs)	Shape	# Redundant Modules
Subsystem 1	67.31	1.00	1
Subsystem 2	14.25	1.00	2
Subsystem 3	14.25	1.00	2
Subsystem 4	25.95	1.00	4

The type of support simulated was no satellite repair/replacement in order to simplify the coding required to simulate satellites in orbit. The simulation assumes that all satellites go into service at the exact same time, that no failures occur due to depletion of consumables, and that all telemetry repairs happen instantaneously. When all redundant modules in a subsystem have failed the entire satellite fails. Once the satellite fails, the simulation does not care about any subsequent failures aboard that satellite. The networks used to simulate the constellation of four is specific to a particular design of spacecraft. Changing the numbers of redundant modules requires changing most of the code. The time unit for this simulation is years.

Coding for Figure 21a and the upper half of 21b is duplicated for each satellite. All satellites in the constellation share the code, depicted in Figure 21c, which gathers statistical information on all the spacecraft. All failure times are generated by a Weibull distribution. Each satellite starts out with four entities which are identified with their respective subsystems by ATRIB(2). Identical brother entities for each redundant module are created and are passed to an ACTIVITY with a duration of the time-until-failure for that particular



SF11	OPEN	1
SF12	OPEN	2
SF13	OPEN	3
SF14	OPEN	4

INTLC, XX(9)=0, XX(10)=0, XX(11)=0, XX(12)=0, XX(13)=0, XX(14)=0, XX(15)=0, XX(16)=4, XX(17)=1,
 XX(23)=0, XX(24)=0, XX(25)=0, XX(26)=0

Figure 21a. SLAM II Reliability Program Flowchart

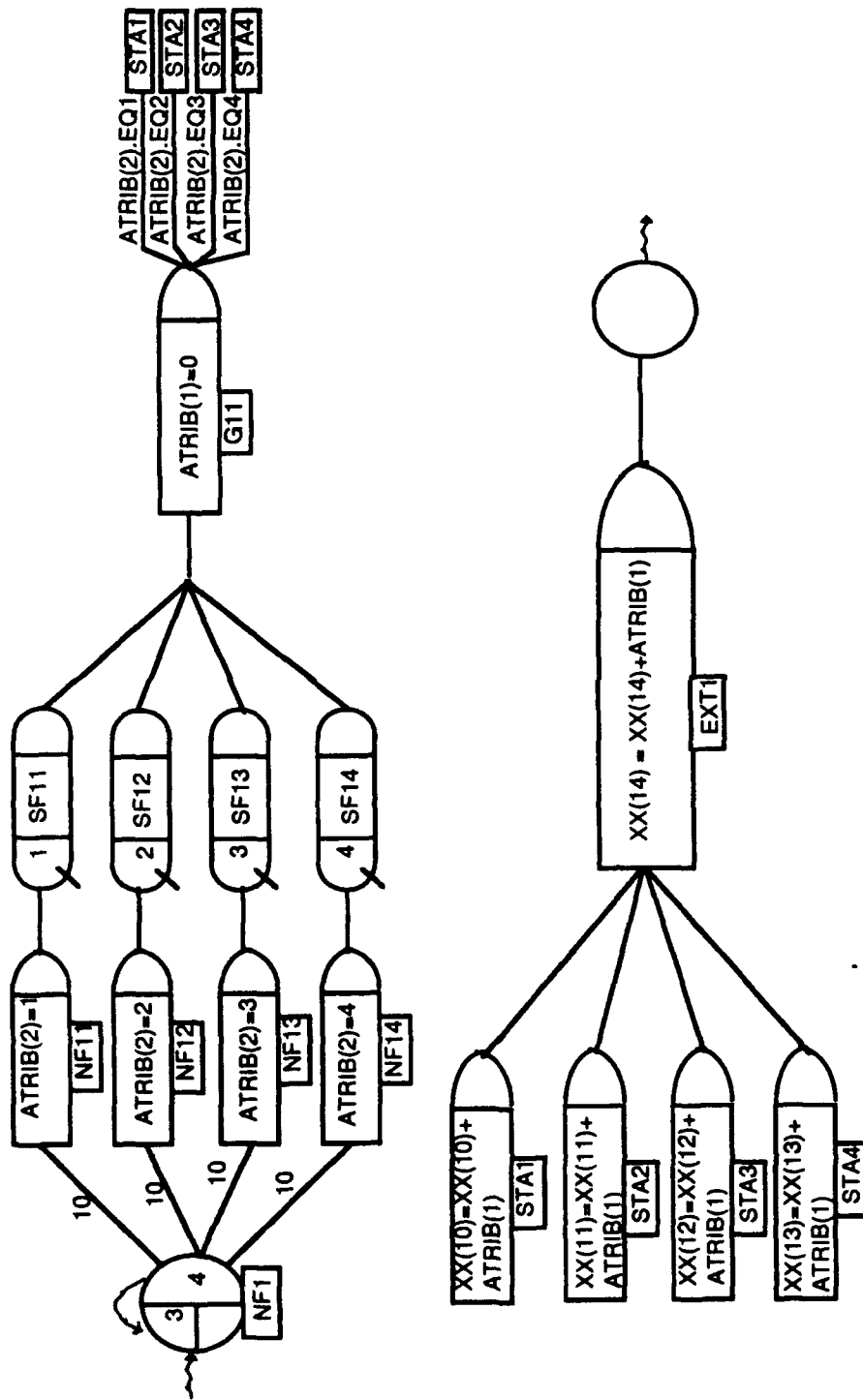


Figure 21b. SLAM II Reliability Program Flowchart

module. An ACCUMULATE node waits until all redundant modules have failed and then passes one entity with the attributes of the last arrival to a GATE node which closes the subsystem gate, signifying subsystem failure. The first failure entity increments counter XX(9) and sets XX(15), the in-service counter, to zero. The time of the failure is collected and then the entity is routed to data counting nodes STA1, STA2, STA3, or STA4 depending on the value of ATRIB(2). These nodes increment the failure counter for the particular subsystem type and the counter for total number of satellite failures. Once the data is gathered, the entities pass to the TERMINATE node.

The network detailed in the upper half of Figure 21b accounts for the occurrence of zero subsystem failures over the timespan of the simulation. Four entities are created and pass through an ACTIVITY of duration 10 time units. This makes sure that the “no-failure” entities do not get to any other node until the end of the simulation. If a particular subsystem has not failed, the GATE node is still open and the entity passes through the gate and into the data count nodes as detailed previously. If the particular subsystem has failed, the gate is closed and the entity is not counted.

Figure 21c shows the parallel network which tracks time for calculating availability of the constellation. The time network consists of a loop that takes 0.01 time units. If the satellite is active, XX(15) has a value of one, and the time-in-service is incremented by 0.01. If the satellite has failed, XX(15) has a value of zero, and the time-in-service stops incrementing. XX(30) collects the time in service for all four satellites.

The last network shown in Figure 21c collects failure and availability information. Time increments in one year intervals during which availability is calculated for each year. The number of hardware failures and satellite failures is only collected at the end of the simulation.

COSEMS was run for the same scenario with 50 replications. The support concept was no repair/no replacement, consumable consumption rates were reduced so that no satellite would fail due to consumable depletion in the first ten years, and telemetry repair

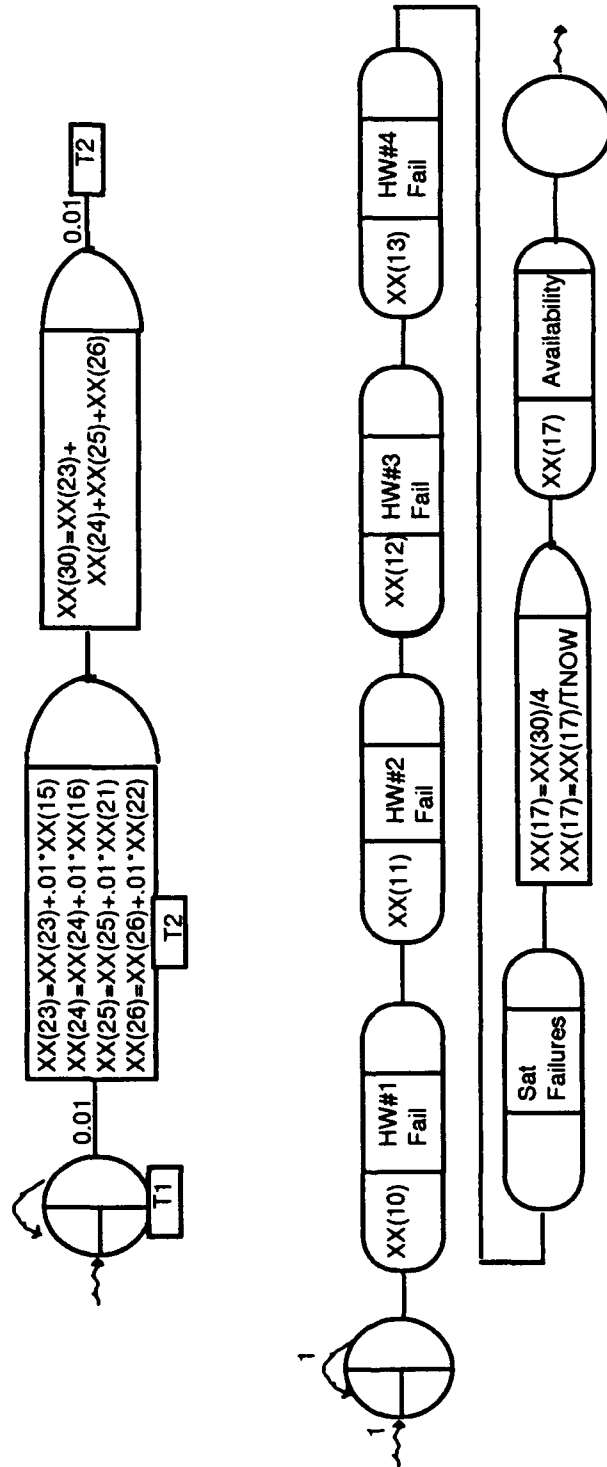


Figure 21c. SLAM II Reliability Program Flowchart

was set for 0.01 days. Additionally the satellite weight was reduced so that all four satellites could be deployed on one launch.

The failure statistics for both the SLAM II and COSEMS runs, as shown in Table 10, were compared using the Z statistic calculation from Equation (54). The results of these calculations are shown Table 11. The comparison was performed using n_1 equal to 200 and n_2 equal to 50 for comparing the hardware failures and n_1 equal to 2000 and n_2 equal to 500 for comparison of availability. This was done by simply by taking the number of replications (200 and 50 respectively) and multiplying by the number of simulation replications. Failures were sampled once at the end of the simulation, while availability was sampled every year for ten years.

TABLE 10
SATELLITE SUBSYSTEM FAILURES AND
AVAILABILITY COMPARISON

	SLAM II		COSEMS	
	Mean	Std Dev	Mean	Std Dev
Subsystem 1 Failures	0.455	0.600	0.42	0.20
Subsystem 2 Failures	0.820	0.788	0.88	0.25
Subsystem 3 Failures	0.785	0.769	0.96	0.22
Subsystem 4 Failures	0.010	0.100	0.38	0.15
Hardware Failures	2.070	0.985	2.64	0.27
Availability %	88.50	13.20	70.89	3.99

As the results shown in Table 11 indicate, the overall SLAM II simulation is similar to the COSEMS output. However, for a 95 percent confidence interval, p-values of zero are unlikely small, indicating that the SLAM II results are inconclusive. The evaluation of inconclusiveness is also shown by the large standard deviations measured by SLAM II. While the means for the subsystem failures and availability are close, the standard

deviations are large enough to make them appear to agree with the COSEMS output, whether this is actually true or not. Several different programs were tried, but the standard deviations could not be reduced.

The differences in standard deviation could be a result of the satellite deployment in COSEMS taking a finite period of time or from SLAM II trying to fit a non-normal distribution to a normal distribution. The difference may also be explained by deviations of the SLAM II program from how COSEMS actually calculates the statistics, differences in the actual number samples taken in each replication, or the fact that COSEMS measures time in satellite-hours while the SLAM II program measured time in 0.01 satellite-years. This would skew the SLAM II output from the COSEMS output because COSEMS takes data with finer time increments.

TABLE 11
COMPARISON OF SLAM II AND COSEMS
RELIABILITY OUTPUT

	Z Statistic	P-Value
Subsystem 1 Failures	0.6864	0.2462
Subsystem 2 Failures	0.9092	0.1816
Subsystem 3 Failures	2.7934	0.0026
Subsystem 4 Failures	16.5469	0.0000
Hardware Failures	7.1761	0.0000
Availability %	51.0574	0.0000

While the number of failures of each subsystem and total number of satellite failures is calculated at the end of each SLAM II replication and does not change in value because only one measurement is taken, the average availability depends on the time interval size. As shown in Figure 18, page 41, the simulation time-line is divided into time intervals of Δt , and availability is calculated over each interval. These interval availabilities are then

used to calculate the average constellation availability. Several runs of the SLAM II program were performed, each with a finer time sampling interval. Availability was taken at 0.1 and 0.01 year intervals. The program was also run with the time increment used for calculating time-in-service changed from 0.01 to 0.1 and to 0.001. With each different run, the average availability increased asymptotically towards its limit. This was due to the proliferation of time intervals which added more data points with availabilities of 100 percent. In other words, the first simulation would have ten measurements of availability, the second one hundred, and the third a thousand. It is obvious from the change in measured availability that data points of 100 percent availability are not necessarily balanced by 0 percent data points. The end result is a biased value for availability.

These results show that the measure of average availability is dependent on the unit time and how often data is taken. It can be gathered that this number will increase to its ultimate limit through an inspection of Equation (42). As the time increments approach zero, the value for Mean Time Between Failures will approach its limit. This means that the value for mean availability will also approach its limit.

The differences in calculating the standard deviation may lie in that the SLAM II program implements the equations shown in Figure 18, while COSEMS ultimately calculates availability from Equation (42). Perhaps writing a reliability simulation in FORTRAN and treating availability with finer time increments might eliminate the problem, allow the calculation of smaller standard deviations, and validate COSEMS calculations.

While this exercise in validation by comparison to other models has been inconclusive, it has ultimately revealed the shortcomings of calculating average availability. In the final analysis, average availability and the availability over any particular time interval conveys no useful information to the user. This is discussed in more detail in the section on *Mission Planning*, page 78. The inconclusiveness of this particular validation method in no way detracts from the trace and walk-through of the code. Failure events are properly

generated, subsystems fail only when all their redundant modules fail, and time-in-service is properly calculated for each satellite.

Orbital Mechanics

Orbital Model. The trace of the orbital mechanics program indicates that correct equations are used with two exceptions. The code which accounts for orbital motion has a minus sign (-) in the apsidal rotation rate equation, making the satellites periapsis perturbate in the wrong direction. The latest version of COSEMS (Version 5.2) already has accounted for this error, and the error was corrected for all runs of COSEMS 5.1. The other exception is that the term accounting for apsidal rotation is not used anywhere in the program. This means that the inter-orbit transfer procedures which rely on the location of periapsis to determine satellite location for bi-elliptic transfers are not accurate.

Code was inserted into COSEMS's orbital procedures to output orbital elements to a separate data file. A comparison of this output to that of Long-Term Orbit Propagator (LOP) for circular orbits of 400 km at inclinations of 0°, 30°, and 90° and for elliptical orbits (0.1 eccentricity) with a periapsis of 400 km at inclinations of 0°, 30°, 63.4°, and 90° shows that there is no significant deviation from proper orbital behavior for the mean anomaly, M , and the Longitude of the Ascending Node, Ω . Plotting these orbital elements over time shows that they are nearly identical to those output by the LOP and are offset only by a time shift accounting for the spacecraft's initial position.

The orbital model is therefore incomplete and does not entirely reflect the effects of the earth's oblateness on orbiting bodies. This will impact all calculations relying on on-orbit support from space with nodal regression. Scenarios relying only on ground support or co-orbital support from space will not have errors in orbital calculations. These errors will also affect COSEMS's graphical output since calculation of the satellite's latitude relies on the argument of periapsis and the apsidal rotation rate (5:A-3)

Orbit Transfer. Traces of the COSEMS code for the bi-elliptic and intra-orbit transfers show that the code works. Orbit transfers are initiated after the proper phasing time, and the spacecraft makes a rendezvous at the proper point in orbit. When the spacecraft are in proper phase for immediately initiating a Hohmann transfer, the code allows the Hohmann transfer in lieu of the bi-elliptic transfer.

Since the bi-elliptic transfer uses three velocity impulses, any plane change required as a part of the orbital transfer should be made concurrent with the change in altitude. This means that there will be at most three plane changes during the entire transfer. The technique of Lagrange multipliers can be used to minimize the velocity change required to perform those plane changes.

The total change in velocity can then be expressed as the sum of magnitudes of the three changes in the velocity performed for the bi-elliptic transfer, where

$$\Delta V_{TOT}(\theta) = \Delta V_1(\theta_1) + \Delta V_2(\theta_2) + \Delta V_3(\theta_3) \quad (55)$$

and θ_1 , θ_2 , and θ_3 are the plane changes performed at each of the three changes in velocity done for the bi-elliptic transfer. The sum of these three plane changes is the total plane change required and is expressed as

$$\theta = \theta_1 + \theta_2 + \theta_3 \quad (56)$$

By expanding Equation (55) as a power series, an approximate expression is found to be

$$\Delta V_{TOT}(\theta) \cong \Delta V_1(\theta) + \Delta V_2(\theta) + \Delta V_3(\theta) + \frac{1}{2} \left[\left. \frac{\partial^2 \Delta V_1}{\partial \theta_1^2} \right|_{\theta_1=0} \theta_1^2 + \left. \frac{\partial^2 \Delta V_2}{\partial \theta_2^2} \right|_{\theta_2=0} \theta_2^2 + \left. \frac{\partial^2 \Delta V_3}{\partial \theta_3^2} \right|_{\theta_3=0} \theta_3^2 \right] \quad (57)$$

The minimum change in velocity is found by using the method of Lagrange multipliers on Equations (56) and (57), where the function to be minimized is expressed as

$$\min F(\Delta V_{TOT}, \lambda) = \Delta V_{TOT}(\theta) + \lambda(\theta - \theta_1 - \theta_2 - \theta_3) \quad (58)$$

Equations (59) through (61) are found by taking the partial differential of Equation (58) with respect to each of the plane change angles.

$$\frac{\partial F}{\partial \theta_1} = \frac{\partial^2 \Delta V_1}{\partial \theta_1^2} \theta_1 + \lambda = 0 \quad (59)$$

$$\frac{\partial F}{\partial \theta_2} = \frac{\partial^2 \Delta V_2}{\partial \theta_2^2} \theta_2 + \lambda = 0 \quad (60)$$

$$\frac{\partial F}{\partial \theta_3} = \frac{\partial^2 \Delta V_3}{\partial \theta_3^2} \theta_3 + \lambda = 0 \quad (61)$$

Solving Equations (56) and (61) for λ and substituting back into the Equations (59) and (60) results in

$$\frac{\partial^2 \Delta V_1}{\partial \theta_1^2} \theta_1 + \frac{\partial^2 \Delta V_3}{\partial \theta_3^2} (\theta_1 + \theta_2 - \theta) = 0 \quad (62)$$

and

$$\frac{\partial^2 \Delta V_2}{\partial \theta_2^2} \theta_2 + \frac{\partial^2 \Delta V_3}{\partial \theta_3^2} (\theta_1 + \theta_2 - \theta) = 0 \quad (63)$$

Solving for θ_1 and θ_2 gives the expressions

$$\theta_1 = \frac{\theta \frac{\partial^2 \Delta V_2}{\partial \theta_2^2} \frac{\partial^2 \Delta V_3}{\partial \theta_3^2}}{NUM} \quad (64)$$

$$\theta_2 = \frac{\theta \frac{\partial^2 \Delta V_1}{\partial \theta_1^2} \frac{\partial^2 \Delta V_3}{\partial \theta_3^2}}{NUM} \quad (65)$$

where NUM represents the equation

$$NUM = \frac{\partial^2 \Delta V_1}{\partial \theta_1^2} \frac{\partial^2 \Delta V_2}{\partial \theta_2^2} + \frac{\partial^2 \Delta V_1}{\partial \theta_1^2} \frac{\partial^2 \Delta V_3}{\partial \theta_3^2} + \frac{\partial^2 \Delta V_2}{\partial \theta_2^2} \frac{\partial^2 \Delta V_3}{\partial \theta_3^2} \quad (66)$$

and θ_3 is expressed as

$$\theta_3 = \theta - (\theta_2 + \theta_3) \quad (67)$$

Adding subscript notation to Equation (40) to represent each individual change in velocity results in Equation (68).

$$\Delta V_i = \sqrt{V_i^2 + V_{ii}^2 - V_i V_{ii} \cos \theta_i} \quad (68)$$

Taking the first and second partial derivatives with respect to the plane change angle gives the two equations

$$\frac{\partial \Delta V_i}{\partial \theta_i} = \frac{V_i V_{ii} \sin \theta_i}{\sqrt{V_i^2 + V_{ii}^2 - V_i V_{ii} \cos \theta_i}} \quad i = 1, 2, 3 \quad (69)$$

$$\frac{\partial^2 \Delta V_i}{\partial \theta_i^2} = \frac{V_i V_{ii}}{|V_i - V_{ii}|} \quad i = 1, 2, 3 \quad (70)$$

Substituting Equation (69) back into Equations (64) and (65) produces

$$\theta_1 = \frac{\theta \frac{V_2 V_{i2}}{|V_2 - V_{i2}|} \frac{V_3 V_{i3}}{|V_3 - V_{i3}|}}{NUM} \quad (71)$$

$$\theta_2 = \frac{\theta \frac{V_1 V_{i1}}{|V_1 - V_{i1}|} \frac{V_3 V_{i3}}{|V_3 - V_{i3}|}}{NUM} \quad (72)$$

Further substitution of Equation (70) into Equations (71) and (72) provides the final equations for finding θ_1 and θ_2 , while θ_3 is found using Equation (67).

$$\theta_1 = \frac{\theta |V_1 - V_{i1}| V_2 V_{i2} V_3 V_{i3}}{NUM |V_1 - V_{i1}| |V_2 - V_{i2}| |V_3 - V_{i3}|} \quad (73)$$

$$\theta_2 = \frac{\theta |V_2 - V_{i2}| V_1 V_{i1} V_3 V_{i3}}{NUM |V_1 - V_{i1}| |V_2 - V_{i2}| |V_3 - V_{i3}|} \quad (74)$$

A trace of the COSEMS code which handles the bi-elliptic transfer shows that these equations are mathematically identical to those used by COSEMS for calculating the proper plane change angles for minimum change in velocity.

COSEMS uses the bi-elliptic transfer code detailed by Morrison (21) and derived in the preceding equations. While these equations estimate the minimum change in velocity

required, they do not address the calculation of the rendezvous windows. Morrison details in his *Bi-Elliptic Transfer Orbit Users Guide* all of the theory and code required to perform the orbital transfer (21:35–46). His entire code and an excerpt from the output used to generate Figure 22 are contained in Appendix B.

Tracing and running Morrison's code (21) caused determination that one minor error existed. This error does not exist in the code used by COSEMS. The plane change angle is mistakenly changed from degrees to radians and then assumed by the code to remain in degrees. By altering two lines of Morrison's bi-elliptic transfer code, the mistake was corrected. It is this corrected code which is detailed in Appendix B. Several runs of this code were made to show that making a 3-split plane change—a partial plane change at each change of velocity made for the bi-elliptic transfer—provides the minimum change in velocity. Appendix B lists the corrected Morrison code's output for a 3-split bi-elliptic transfer.

Figure 22 shows the results of these runs and shows the total change in velocity required versus the intermediate transfer orbit radius for the 1-split, 2 split, and the 3-split transfer. The 3-split is plotted against the optimum transfer for both the 1-split and 2-split transfers, where these optimums are the minimum of the three combinations graphed for the each transfers. As shown, the 3-split bi-elliptic transfer always provides the minimum total change in velocity. This results holds true for any selection of initial and final orbit altitudes.

This analysis of the bi-elliptic code used by COSEMS conforms to the model described by the COSEMS Design Document and indeed works as described. The bi-elliptic transfer moves a designated spacecraft from one orbit to another and optimizes the plane change angles to provide the minimum possible total change in velocity for the bi-elliptic transfer.

$h_0 = 100 \text{ km}$
 $h_t = 1000 \text{ km}$
 $i = 5 \text{ degrees}$

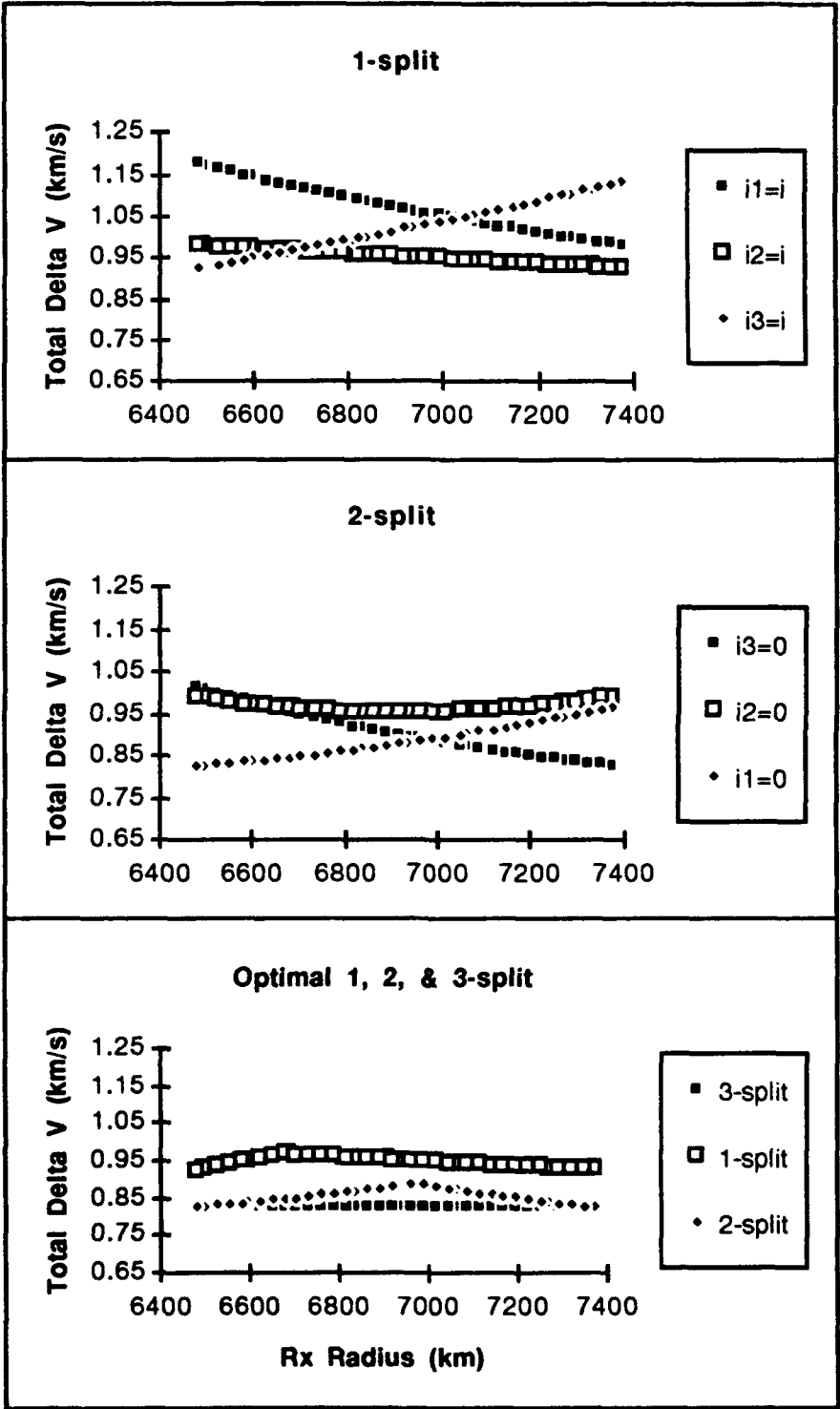


Figure 22. Plane Change Comparison for the Bi-Elliptic Transfer

Mission Planning

A trace of the COSEMS code shows that the algorithms and decision flow occur as detailed by the *COSEMS Design Document* (5). The code determines when failures have occurred and whether telemetry repairs are possible. Satellite failures are determined and repair missions assigned and performed dependent on the support concept. Spare ORUs, replenishment of consumables, and the loading of launch vehicles and OMVs is determined and performed according to the time-lines specified by the user.

The one problem detected is in the use of the specific impulse of the OMV as specified by user input. This quantity fixes the OMV's rate of fuel consumption and its velocity. In the subprogram *MISSION_PROP_ROUTINES.ADA*, subroutine *CALC_FUEL*, the specific impulse of the OMV is fixed at its default value, 300 seconds. This is the only occurrence of this error, but it causes deviations from the proper results if the user has specified a different specific impulse for the OMV. Depending on the input, orbital transfers would take more/less time and fuel consumption will be greater/less. This could cause some orbital transfers to not occur because the program might determine that insufficient fuel is available. The ultimate results are that more spacecraft failures could occur because repair missions could not reach the spacecraft in time, more spares are activated than should be, or that more satellite replacements occur because the program determines that a spacecraft cannot be repaired before its cutoff time and no spare is available.

The questionable piece of coding was found in *GROUND_SERVICE_FUNCTIONS.ADA*. In this subprogram the procedures *FIND_OMV*, *START_DOCKING_AT_SATELLITE*, and *START_SORTIE* all contain hard code which forces a specific minimum time before an OMV launch. Forcing a minimum wait or turn-around time is not in itself bad, but it would be better if it were incorporated into the menu-driven preprocessor to allow for user input. This would flag to the user that the minimums exist and allow the user to increase or decrease them to suit a particular case study. If these minimums are required to eliminate

infinite loops or decision/time-line conflicts, it should then be noted in the *Design Document* (5) or *User's Guide* (6).

While there is no problem with the mission planning decision criteria and time-lines, it is questionable that mission planning should always occur exactly according to the time-line specified by the user. If the user specifies that the processing of launch vehicles takes ten hours and that turnaround of thirty days exists between launches, then those activities always occur in that specified manner. Those acquainted with NASA space launches, contracted work, or most any scheduling activity know that schedules often are delayed for many different reasons. Since weather, loading foul-ups, or other mishaps are very likely to occur, it seems prudent to make the mission planning activities more stochastic. This addition to the code would allow a launch pad turn-around to occur within a specified time frame such as 30 days \pm 1 week. The duration of the different mission planning activities would then be user-specified and allow the duration to follow the Weibull, normal, or some other desired probability distribution with user specified parameters. This change would allow mission planning to simulate the real-world with greater accuracy.

COSEMS Functions and Capabilities

COSEMS has many different options, settings, and capabilities which are detailed in the *COSEMS User's Guide* (6). While the *User's Guide* details how to navigate through the different menus and what they are used for, the *User's Guide* should contain several examples and a trouble shooter's guide on how to set up the simulation for different types of scenarios. Although the *Guide* contains a complete description of what to do, learning the peculiar nuances of navigating COSEMS menus and implementing a particular scenario would be easier with more examples to make sure that a case study will run on COSEMS.

COSEMS was run for many different cases in order to learn the effect of different menu selections on output and the ease in getting the simulation to run. The output for a

case study is listed in Appendix C. Getting these case studies to execute was not always easy and required trial-and-error, as well as guidance from COSEMS's programmers.

A particular example is the selection of launch vehicles. A new user of COSEMS would have trouble selecting launch vehicles for launch, repair, and replacement even though the total weight to orbit seems to be within the weight constraints of many more launchers than is listed. The solution is to alter the minimum payload capacity to a much lower number than the default value. Another problem is getting the default OMV to high altitudes and inclinations. Although an OMV with the default weight should be able to be inserted into such orbits, it is not possible unless the OMV's weight is reduced. Such solutions would be beyond anyone simply reading the *User's Guide* and would require guidance from someone well versed in running COSEMS or a lot of trial and error.

Another example is the confinement of Pegasus launchers to the Eastern and Western Test Ranges although the vehicle actually has an unlimited range of launch sites because it is air launched. Operationally, Pegasus can be flown to the equator and launched into any inclination. In this case the operational limitation on Pegasus can be worked around by ensuring that its launch capacity to orbit database reflects equator launch information and by increasing the number of Pegasus launch sites to reflect its ease of launch and mobility.

Even though running COSEMS requires a VT200 or similar terminal, a VT100 was used for purposes of this investigation, since a VT200 was unavailable and emulation software for such a terminal was not always available. After some trial and error, it was found that VT100 emulation was more than adequate to run COSEMS. It should be noted that even though VT100 and VT200 emulation was used, the preprocessor menus did not function equally well for all emulation programs. Although emulation modes were the same, it seemed that some emulation programs did not provide the user with equal ability to navigate through COSEMS's menus and make appropriate menu selections. In these cases either it was impossible to make menu selections or input values would not appear on the screen.

for average availability. Figure 23 illustrates that for space support concepts, the availability as a function of time has a periodic nature and can drop off as spacecraft experience failures and rise as they are repaired or replaced. Since the constellation operation is an on-going activity, the mean is meaningless and conveys no useful information.

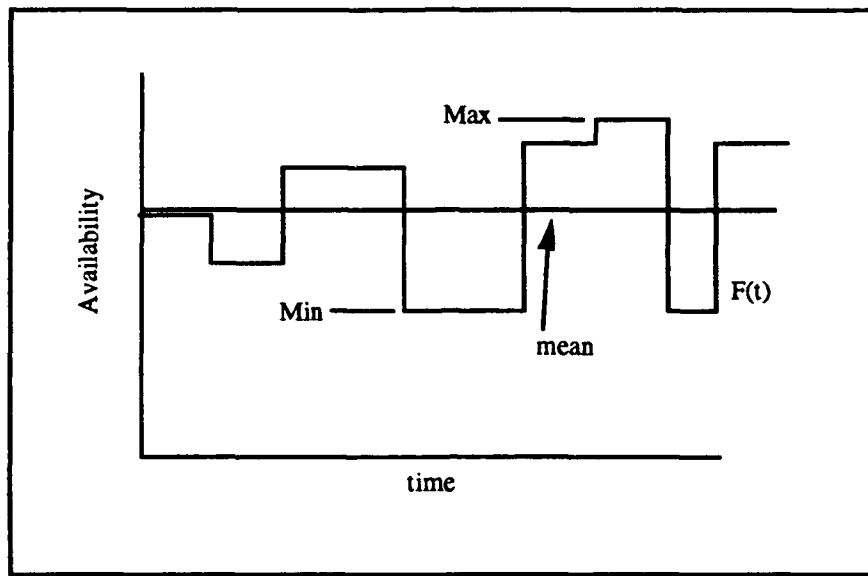


Figure 23. Availability as a Function of Time

If Figure 23 did represent a constellation's operational availability as a function of time, the information that is more useful is the minimum and maximum availabilities and the length of time that these occur. It is considerably more useful to see the plot of the availability function rather than the mean. The same can be said for COSEMS's provision of availability over a time interval of a week, month, quarter, or year. It is important not only to know the average availability over that interval, but also the minimum. When providing snapshots of the constellation's condition at a specific interval in the simulation, that snapshot is just a point in the availability function. Knowing behavior at several points or its aggregate behavior over a long interval does not help characterize the overall behavior of a space system in the same manner that several plotted points of a complicated function can give no true idea of the behavior between those points.

Reliance on VT200 terminals could restrict or discourage potential users from using COSEMS. It would seem prudent to make COSEMS compatible with the most base and easily emulated terminal such as the VT100 so that use is not dependent on finding a particular emulation program or purchasing a particular computer.

Graphics and Output. COSEMS provides exhaustive output on the lifetime characteristics of the space logistical support system. Much of the output is directly available from the case study output file, while other output requires use of the program COSPLOT (6:8-3). COSEMS output on the resources consumed, number of failures, and failure causes provides good descriptive measures of the support concept's performance. However, the information provided by the availability charts needs improvement to be useful in evaluating competing support concepts.

As shown in Appendix C, page 105–107, COSEMS provides constellation operational availability over user specified intervals, cumulative availability, and operational availability over the specified time interval for each ring in the constellation. It should be made clear that the presentation of a value for availability is very deceptive. Availability as measured by COSEMS is actually the average percentage of satellites available at any particular time. For this information to be useful, it must be integrated with each satellite's orbital location relative to the earth and to each other. If the constellation in question is a worldwide communications system, a user must not only know how many satellites are operational, but also if there is enough coverage for the constellation to perform its mission. For a Global Positioning System (GPS), it is important to know how well the operational satellites perform their mission. It is easily possible for a particular satellite constellation to be 85 percent or more operational, but to be totally unable to perform its mission because its satellites are not in the proper position over the earth.

Obviously, COSEMS is just one tool in evaluating space support concepts and cannot provide all this information, but it should be made more clear in COSEMS's documentation that the availability information can be deceptive. One example is the provision of a value

As shown in Appendix C, COSEMS does provide a table of cumulative operational availabilities showing the percent of time that the system is less than specified availabilities. While this is useful, an additional chart such as Figure 24 would significantly aid in understanding the information. A supplementary plot such as this would provide the user with a better idea on how the the space system behaves over time, what the minimum availability is, and how long it stays there or how often the availability gets that low. As shown in the figure, the chart would not only provide a graph of operational availability, but also the availability without “degraded” and “at risk” satellites. This sort of information would provide some measure of the system’s robustness and sensitivity to any additional stress factors (unexpected satellite outages) without performing additional parametric studies.

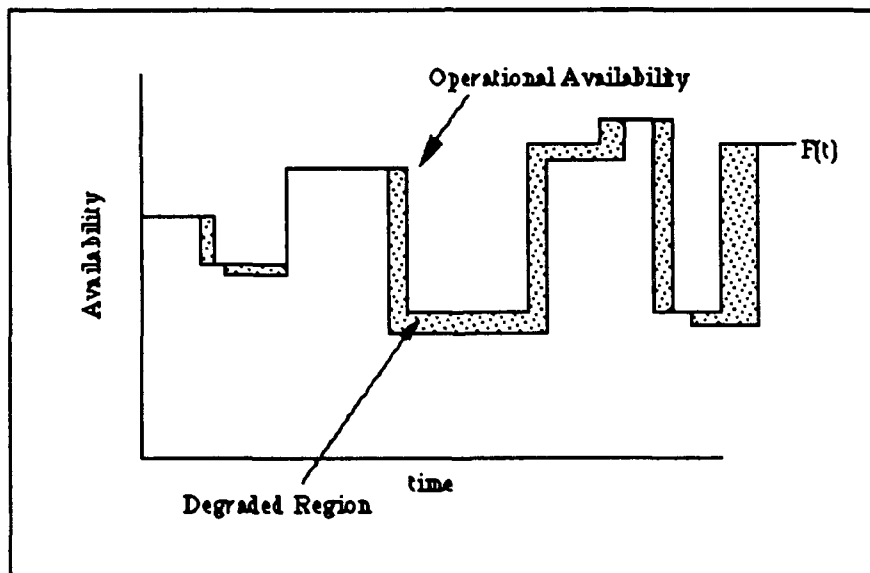


Figure 24. Proposed Availability Plot

COSEMS does provide the capability to provide minimum and maximum ring availabilities such as described above, but these plots are only accessible by running its sister program, COSPLOT (6:8-14). Also, this chart does not give any indication of the constellation’s state of degradation, when some satellites have backup module failures and may completely fail at any moment. The problem in relying on another program to provide

this information is that it also requires the use of specialized graphics terminals and graphics printers (6:8-3). Running COSPLOT and printing its graphs and plots was not possible for this investigation because of the unavailability of such terminals and printers. Although the *COSEMS User's Guide* indicates that COSPLOT is compatible with PostScript® printers, this is not the case (6:8-3). The need for specialized computer equipment restricts the full use of COSEMS to those with access to the equipment detailed in the *COSEMS Users Guide* (6).

COSEMS has several different plots and snapshots available to users by running COSPLOT. Many of these put into graphical terms the information given by the output listed in Appendix C. However, some of the plots provide snapshots of satellites at a given time and provide a bird's-eye view of the satellites' ground tracks over the earth and their location by latitude and longitude. These are interesting graphs, but provide little pertinent information because the snapshot does not reveal the dynamic nature of the space system's status. Since COSEMS cannot be used to measure mission capability of the system, a graphical depiction of satellite location seems unnecessary.

A possible alternative would be to provide the information used to generate the plots and snapshots in tab-delimited files along with the spreadsheet macros necessary to generate the graphs using any commercially available spreadsheet programs. This would allow the user more latitude in the manipulation of the data generated by COSEMS.

COSEMS Assumptions and Limitations. Overall, COSEMS's assumptions and limitations, as listed in Chapter II, are well thought out and simplify the work of a large and complicated program. While most of them are well suited for simulating space support concepts for a SDS, they are invalid for other space systems. These assumptions include 1) telemetry, command, and control support (TT&C) always being available, 2) all COSEMS launches have priority, 3) no limit to launch vehicle inventory, and 4) no nodal regression is allowed for non-circular orbits.

It is easily understood that a SDS would have the utmost priority and demand launch priority and TT&C priority, but if COSEMS were to be used for other military systems, foreign government systems, or commercial systems, this would probably not be the case. If wider use of COSEMS were desired, COSEMS should have the capability to implement a more stochastic TT&C system in which the constellation's priority was not always a certainty. In the same manner, it is not always possible to take launch priority because required launch vehicles are not available or required launch pads are in use.

Unlimited launch vehicle inventory seems to be out of place in current space operations and most probably will remain so for future operations. Although the assumption of unlimited launch vehicles is improbable for even an SDS, it seems prudent to have several launch vehicles of any one type available at any one time. An enhancement to COSEMS would allow a user specified number of launch vehicles as a starting pool and would then simulate the acquisition of replacement and additional launchers according to a specified time-line and probability distribution.

The assumption that nodal regression is not available to non-circular orbits seems to greatly limit COSEMS. It is highly probable that some elliptical orbit scenarios will provide better space support than circular orbit scenarios. Limiting nodal regression to circular orbits eliminates a whole class of orbits from study and limits users to the comparison of circular, nodal regression concepts to elliptical, in-plane support concepts. While it is understood that bi-elliptic transfer calculations between elliptical orbits involves more complicated and time-consuming calculations than presented in this paper, the purpose of COSEMS is to provide analysis and comparison of competing space support concepts. Therefore nodal regression should be an option for elliptical orbits.

V. Conclusions and Recommendations

Conclusions

The methodology and results given in the preceding chapters indicate that COSEMS follows the model detailed in the *COSEMS Design Document* (5). With the exception of the errors in coding the OMV's specific impulse and the omission of apsidal rotation in the orbital model, as well as using deterministic mission planning, the simulation model reflects how the real world operates, given the limitations and assumptions of the program. The two errors cited in this investigation have been reported to PRC Inc. for correction. Notwithstanding the omission of apsidal rotation, after testing the random number generator, running reliability simulations, and testing the code for orbital motion and bi-elliptic transfer, the investigation concludes that the code adheres to the documented model and that that model is consistent with realistic space system behavior. Once the program has been corrected to include apsidal rotation, COSEMS's orbital model will be fully valid and provide output consistent with its assumptions and spacecraft orbital motion.

While the code governing the actual failure of individual modules, subsystems consisting of several modules, and satellites consisting of several subsystems has been verified and validated to ensure that events and calculations are performed correctly, the validation of COSEMS's reliability functions by comparison to other models has been found to be inconclusive. The reliability information generated by the SLAM II simulation approximates the COSEMS output, although the large standard deviations generated by the SLAM II simulation implies that some doubts persist regarding the statistical validity of COSEMS statistical calculations. These results call for further validation of the COSEMS code only in the area of statistical data calculation.

In evaluating the utility of the model and its use by a variety of users, extensive runs of COSEMS using different microcomputers emulating VT100 and VT200 terminals were performed. It was determined that differing emulation programs have enough

idiosyncrasies to impede the analysis of case studies with COSEMS by making it difficult and sometime impossible to run COSEMS. Furthermore, the need for specialized graphics printers and consoles prohibits the full use of COSEMS output for evaluating space support concepts.

Although the output file generated by COSEMS provides detailed information on availability, resources consumed, and constellation status, this output file and the COSPLOT graphics output files lack utility and user friendliness. The actual reports on availability provide a false sense of constellation performance since a static view of average constellation availability over a time interval and at particular intervals does not provide information on the time-varying nature of satellite system availability. The COSPLOT graphs which provide a measure of availability versus time require specialized terminals and printers and therefore restrict access to those capabilities and discourage would-be users of COSEMS from using the program.

Finally, although COSEMS models the stochastic nature of space system failures and launches, it fails to account for the stochastic nature of the mission planning process. Mission planning, launch vehicle manifesting, cargo loading, and OMV repair/resupply missions occur in a strictly deterministic fashion. This does not invalidate the model, but its users must be aware that the mission planning simulated by COSEMS represents a best-case analysis.

Recommendations

While COSEMS, once it is corrected for the previously noted errors, is highly capable of providing analysis of space support concepts, it requires certain enhancements to make it attractive for use in a wide variety of military and non-military cases. COSEMS needs to be divorced from the need for specific brands and types of computer equipment and modified to provide full simulation of elliptical orbit concepts to allow support concepts using bi-elliptic transfers and nodal regression.

Further validation of the calculation of mean and standard deviations for availability and module failures should be undertaken. This should not only include the design implementation of computer code to simulate satellite failures, but to also output the failure data to separate files and allow the calculation of mean and standard deviation statistics for time-in-service, failures, and availability for test case studies by separate computational methods.

Currently COSEMS must run on a VAX system running VMS system 5.1 or later due to the incorporation of code specifically tied to this type of computer processor. This dedicated code is contained in the random number generator, the menu driven preprocessor, and the graphics output files which require COSPLOT. While VAX computers are often available to military and non-military users of COSEMS, this may not always be the case as powerful workstations become less expensive and more available. Additionally, potential users of COSEMS may be discouraged from using the program because they do not have access to VAX computers, VT100 and VT200 terminals, or the graphics terminals and printers. Since Ada is a portable language, COSEMS could be written to eliminate the reliance on a particular brand of computer. Several changes would be required. These would include 1) incorporation of a random number generator as detailed in Chapter IV and listed in Reference (4), 2) either the elimination of machine-dependent graphics or the assurance of full VT100 compatibility, and 3) changing or incorporating as an option the graphics output format from one requiring COSPLOT to generate plots to a tab-delimited format which would enable any commercial spreadsheet to access the data and generate charts.

Since COSEMS is supposed to provide information for the evaluation of alternate space support concepts, the lack of information on availability as a function of time is the one drawback of the program. Instead of providing average availability for specified time intervals and over the entire simulation, the program should have the ability to take the data

used to provide this output and instead provide a data file which would allow plots of availability as a function of time.

Further Research

Further research opportunities with COSEMS include the following:

- 1) Adapting the COSEMS code to object-oriented programming. This would allow additional capabilities to be added on an as-available basis and provide for customized versions of COSEMS. These customized programs would have only the functions the user required and would allow limitation of program size and increases in execution speed. Such added capabilities could include provisions for adding electric propulsion OMV concepts and additional launch sites.
- 2) Further analysis of COSEMS calculation of mean and standard deviations for satellite failures and availability. This would provide conclusive validation of the reliability/availability calculations.
- 3) Development of code to allow COSEMS to perform bi-elliptic transfers between elliptical orbits and use nodal regression for elliptical orbit concepts. This would allow for increased analytical capability which could provide for support concepts utilizing specialized elliptical orbits with unique characteristics. Such orbits might provide better performance for space-based support concepts than circular orbits allow.
- 4) Development of example case studies and manuals which would provide guidance on how to alter COSEMS input values for many possible case studies. Such things might include alternate launch vehicle tables to allow the use of Soviet, European, and Chinese launch vehicles and also to allow for launch capacities that would simulate launches from launch sites other than the Eastern and Western

Test Ranges. Such manuals would include examples of several constellations such as GPS and case study debugging information to help users work out any problems in getting their case studies to run. The debugging information would include common input errors and user oversights which keep the simulation from running as desired.

Appendix A

COSEMS Source Files Used for Verification & Validation (7)

COSEMS_CONSTANTS.ADA	MISSION_PROP_ROUTINES.ADA
CO_AVAILABILITY.ADA	MISSION_PROP_ROUTINES_.ADA
CO_AVAILABILITY_.ADA	MISSION_STATUS_OPERATIONS.ADA
CO_CONTROL.ADA	MISSION_STATUS_OPERATIONS_.ADA
CM_CONTROL.ADA	MODULE_OPERATIONS.ADA
CM_GROUND_REPLACE.ADA	MODULE_OPERATIONS_.ADA
CM_REPLICATION.ADA	MODULE_STATUS_OPERATIONS.ADA
CM_REPLICATION_.ADA	MODULE_STATUS_OPERATIONS_.ADA
CM_RESUPPLY_SERVICE.ADA	OMV_OPERATIONS.ADA
CM_RS_RETURN.ADA	OMV_OPERATIONS_.ADA
CM_RS_TRANSFER.ADA	OOFG_OPERATIONS.ADA
CM_SATELLITE_DOWN.ADA	OOFG_OPERATIONS_.ADA
CM_SPACE_SERVICE.ADA	ORBIT_FUNCS.ADA
CM_SPARE_ACTIVATION.ADA	ORBIT_FUNCS_.ADA
CM_SS_CHECK_RESOURCES.ADA	OSCRS_OPERATIONS.ADA
CM_SS_LAUNCH.ADA	OSCRS_OPERATIONS_.ADA
CM_SS_RETURN.ADA	PAYLOAD_OPERATIONS.ADA
CM_SS_SATELLITES.ADA	PAYLOAD_OPERATIONS_.ADA
ELLIPTICAL.ADA	PLATFORM_OPERATIONS.ADA
ELLIPTICAL_.ADA	PLATFORM_OPERATIONS_.ADA
EVENT_GENERATOR.ADA	PLAT_MOD_RESUP_OPERATIONS.ADA
EVENT_GENERATOR_.ADA	PLAT_MOD_RESUP_OPERATIONS_.ADA
FROM_GROUND_MISSION_PLANNING.ADA	PO_DIVIDE_PAYLOAD.ADA
FROM_GROUND_MISSION_PLANNING_.ADA	PO_GLI.ADA
GLMP_PERFORM_LAUNCH.ADA	PO_GLU.ADA
GROUND_LAUNCH_MISSION_PLANNING.ADA	RESUPPLY_ROUTINES.ADA
GROUND_LAUNCH_MISSION_PLANNING_.ADA	RESUPPLY_ROUTINES_.ADA
GROUND_LAUNCH_STATUS_OPERATIONS.ADA	SATELLITE_OPERATIONS.ADA
GROUND_LAUNCH_STATUS_OPERATIONS_.ADA	SATELLITE_OPERATIONS_.ADA
GROUND_SERVICE_FUNCTIONS.ADA	SAT_STATUS_PACKAGE.ADA

GROUND_SERVICE_FUNCTIONS_.ADA
GSF_END_OF_MISSION.ADA
GSF_FIND_NEXT_SATELLITE.ADA
GSF_SERVICE_COMPLETED.ADA
LAUNCHER_OPERATIONS.ADA
LAUNCHER_OPERATIONS_.ADA
LO_RESCHEDULE_LAUNCH.ADA
MATH_PAK.ADA
MATH_PAK_.ADA

SAT_STATUS_PACKAGE_.ADA
SAT_STATUS_Q_OPERATIONS.ADA
SAT_STATUS_Q_OPERATIONS_.ADA
SCHEDULED_MISSION_PLANNING.ADA
SCHEDULED_MISSION_PLANNING_.ADA
SMP_SPACE_PLANNING.ADA
SMP_TELEMETRY_REPAIR.ADA
STATISTICS_PACKAGE.ADA
STATISTICS_PACKAGE_.ADA

Appendix B

Bi-Elliptic Transfer Code (21)

```
PROGRAM RENDEZ
*
* THIS PROGRAM SIMULATES BI-ELLIPTIC, NON-COPLANAR REBDEZVOUS
* ORBIT FROM A SERVICE VEHICLE IN AN INITIAL ORBIT TO A S/C IN A
* TARGET ORBIT. IT IS DESIRED TO FIND THE DELTA V AND THE
* TRANSFER TIME.
*
* LAST UPDATE: 7/28/88
*
* AUTHOR: S. C. MORRISON
* GENERAL RESEARCH CORPORATION, EL SEGUNDO, CA
*
* COMMON XMU,PI,R0,RT,P0,PT
*
* CHARACTER*40 FNAME
*
* WRITE(6,9998) ' ENTER NAME OF INPUT FILE: '
9998 FORMAT(A,$)
* READ(5,9999) FNAME
9999 FORMAT(A)
* OPEN(10,FILE=FNAME,STATUS='OLD')
* OPEN(11,FILE='REND.OUT',STATUS='NEW')
*
* READ IN THE INPUT FILE
*
* SERVICE VEHICLE POSITION (DEG)
*
* READ(10,*) THMIN,THMAX,DTH
*
* S/C VEHICLE POSITION (DEG)
*
* READ(10,*) FHMIN,PHMAX,DPH
*
* SERVICE VEHICLE ALT (KM), S/C ALT (KM)
*
* READ(10,*) H0,HT
*
* MINIMUM AND MAXIMUM TRANSFER ALTITUDE ALLOWED (KM)
* USUALLY SAME AS H0 AND HT
*
* READ(10,*) HMIN,HMAX
*
* PLANE CHANGE ANGLE (DEG)
*
* READ(10,*) PTHETA
*
* PRINT INPUTS
*
* WRITE(11,500)
500 FORMAT(///,10X,'** INPUT DATA **',//)
```

```

WRITE(11,501) THMIN, THMAX, DTH
501  FORMAT(10X,'S/V POSITION: THETAMIN, THETAMAX, DELTHETA = ',3F9.2)
WRITE(11,502) PHMIN,PHMAX,DPH
502  FORMAT(10X,'S/C POSITION: PHMIN, PHMAX, DPH = ',3F9.2)
WRITE(11,504) H0,HT
504  FORMAT(10X,'INITIAL ALT = ',F10.2, 'KM',5X,
&      'FINAL ALT = ',F10.2, 'KM')
WRITE(11,503) HMIN,HMAX,PTHETA
503  FORMAT(10X,'MIN ALT, MAX ALT, PLANE CHANGE ANGLE (KM, DEG)',
&      '=',3F12.2)
WRITE(11,505)
505  FORMAT(///,10X,'** OUTPUT DATA **',///)
*
XMU = 3.986032E5
PI = 3.14159265359
RE = 6378.165
TWOPI = 2.*PI
RADEG = 180./PI
*
THMIN = THMIN/RADEG
THMAX = THMAX/RADEG
DTH = DTH/RADEG
PHMIN = PHMIN/RADEG
PHMAX = PHMAX/RADEG
DPH = DPH/RADEG
*
THETA = THMIN
PHI = PHMIN
*
R0 = RE + H0
RT = RE + HT
RMIN = RE + HMIN
RMAX = RE + HMAX
*
*   INITIAL TARGET ORBIT PERIODS AND VELOCITIES
*
P0 = 2.*PI*SQRT(R0**3/XMU)
PT = 2.*PI*SQRT(RT**3/XMU)
V0 = SQRT(XMU/R0)
VT = SQRT(XMU/RT)
*
*   CALCULATING THE FIRST RENDEZVOUS WINDOW
*
RX = RMIN
ALPH1 = PI/2.*(SQRT((R0+RX)**3/(2.*RT**3)))+
&      SQRT((RX+RT)**3/(2.*RT**3)))
RX =RMAX
ALPH2 = PI/2.*(SQRT((R0+RX)**3/(2.*RT**3)))+
&      SQRT((RX+RT)**3/(2.*RT**3)))
*
ALP1 = ALPH1
ALP2 = ALPH2
ALP1MIN = 2.*PI-ALPH2
ALP1MAX = 2.*PI-ALPH1
IF (R0.GT.RT) THEN
    ALP1MIN = 4.*PI-ALPH2
    ALP1MAX = 4.*PI-ALPH1
END IF

```

```

ALP1MIN = AMOD(ALP1MIN,TWOPI)
ALP1MAX = AMOD(ALP1MAX,TWOPI)
IF (ALP1MIN.LT.0.0) ALP1MIN = ALP1MIN+TWOPI
IF (ALP1MAX.LT.0.0) ALP1MAX = ALP1MAX+TWOPI
AL1MIN = ALP1MIN
AL1MAX = ALP1MAX
*
IF (ALP1MIN.GT.AL1MAX) ALP1MIN = ALP1MIN-TWOPI
*
*
CALCULATING THE SECOND RENDEZVOUS WINDOW
*
ALP2MIN = 3.*PI-ALPH2
ALP2MAX = 3.*PI-ALPH1
IF (R0.GT.RT) THEN
    ALP2MIN = 3.*PI-ALPH2
    ALP2MAX = 3.*PI-ALPH1
END IF
ALP2MIN = AMOD(ALP2MIN,TWOPI)
ALP2MAX = AMOD(ALP2MAX,TWOPI)
IF (ALP2MIN.LT.0.0) ALP2MIN = ALP2MIN+TWOPI
IF (ALP2MAX.LT.0.0) ALP2MAX = ALP2MAX+TWOPI
AL2MIN = ALP2MIN
AL2MAX = ALP2MAX
*
IF (ALP2MIN.GT.AL2MAX) ALP2MIN = ALP2MIN - TWOPI
*
*
WRITE OUTPUT THE RENDEZVOUS ANGLES
*
WRITE(11,506) ALP1MIN*RADEG,ALP1MAX*RADEG,
& ALP2MIN*RADEG,ALP2MAX*RADEG
506 FORMAT(//,10X,'ALP1MIN, ALP1MAX =',2F9.2,' DEG ',
& 'ALP2MIN, ALP2MAX =',2F9.2,' DEG')
WRITE(11,199)
199 FORMAT(//,5X,'THETA1',3X,'THETA2',4X,'THETA3',3X,
& 'PHI',5X,'PHI2',5X,'M',6X,'RX',7X,'TWAIT1',5X,
& 'TWAIT2',5X,'DVTOT',7X,'I1',5X,'I2',6X,'I3',6X,'TTOT',//)
*
201 CONTINUE
*
THETA = THMIN
*
200 CONTINUE
*
THETA2 = THETA
TWAIT1 = 0.0
*
IF (ABS(THETA - TWOPI).LT.0.001.OR.ABS(THETA-0.0).
& LT.0.001) THEN
    PHINOW = PHI
    GOTO 20
END IF
*
IF (ABS(THETA - PI).LT.0.001) THEN
    PHINOW = PHI
    GOTO 20
END IF
*
*
CASE WHEN S/V LIES WITHIN 0 TO PI

```

```

*
IF (THETA.GT.0.0.AND.THETA.LT.PI) THEN
  GAMMA = (PI-THETA)*P0/PT
  TWAIT1 = (PI-THETA)*P0/TWOPI
  THETA = PI
END IF
*
*
CASE WHEN S/V LIES WITHIN PI TO TWOPI
*
IF (THETA.GT.PI.AND.THETA.LT.TWOPI) THEN
  GAMMA = (TWOPI-THETA)*P0/PT
  TWAIT1 = (TWOPI-THETA)*P0/TWOPI
  THETA = 0.0
END IF
*
PHIGAM = PHI + GAMMA
PHIGAM = AMOD(PHIGAM,TWOPI)
PHINOW = PHIGAM
*
20 CONTINUE
*
*
CASE WHEN R0 >> RT WHERE THE WINDOW OS GREATER THAN TWOPI
*
IF (ABS(ALP1-ALP2).GE.TWOPI) THEN
  IF (THETA.EQ.0.0) ALPHD = PHINOW-ALP1MAX
  IF (THETA.EQ.PI) ALPHD = PHINOW-ALP2MAX
  IF (ALPHD.LT.0.0) ALPHD = ALPHD + TWOPI
  ALPHA = ALPH1 + ALPHD
  TWAIT2 = 0.0
  PHIF = TWOPI*TWAIT1/PT
  PHIF = PHINOW + PHIF
  PHIF = AMOD(PHIF,TWOPI)
  M=0
  GO TO 30
END IF
*
*
CALLING TO FIND S/V WAITIME BEFORE INTIATING A RENDEZVOUS
*
CALL WAITIME(ALP1MIN,ALP1MAX,ALP2MIN,ALP2MAX,THETA,PHINOW,
&            THETAF,PHIF,MX,TWAIT2)
*
*
IF THE WAIT TIME IS GREATER THAN 48 HOURS, SEEK ALTERNATE
*
SOLUTION
*
IF (MX.EQ.999) THEN
  RXMIN = RE + 200.
  AMIN1 = (R0 + RXMIN)/2.
  AMIN2 = (RXMIN +RT)/2.
  T12 = 0.5*(2.*PI*SQRT(AMIN1**3/XMU))
  T23 = 0.5*(2.*PI*SQRT(AMIN2**3/XMU))
  PRX = T12 + T23
  IF (ABS(THETAF-0.0).LT.0.01.OR.ABS(THETAF-TWOPI).LT.0.01) THEN
    TTRAV = PT*(TWOPI-PHIF)/TWOPI
    IF (PRX.LE.TTRAV) THEN
      ALPHA = TWOPI-PHIF
      CALL GETRX(ALPHA, RX)
      CALL FSTSPLT(RX,PTHETA,XI1,XI2,XI3,DVTOT,TTOT,IFLG)
      IF (R0.EQ.RT) THEN

```

```

        IF (THETA F.EQ.PHIF) TTOT = 0.0
      END IF
    GO TO 40
  END IF
  ALPHA = 2.*TWOPI-PHIF
  CALL GETRX(ALPHA,RX)
  CALL FSTSPLT(RX,PTHETA,XI1,XI2,XI3,DVTOT,TTOT,IFLG)
  IF (RO.EQ.RT) THEN
    IF (THETA F.EQ.PHIF) TTOT = 0.0
  END IF
  GO TO 40
END IF
IF (ABS(THETA F-PI).LT.0.01) THEN
  IF (PHIF.GE.0.0.AND.PHIF.LT.PI) THEN
    PHIFP = PHIF + PI
  END IF
  IF (PHIF.GE.PI.AND.PHIF.LT.TWOPI) THEN
    PHIFP = PHIF - PI
  END IF
  TTRAV = PT*(TWOPI-PHIFP)/TWOPI
  IF (PRX.LE.TTRAV) THEN
    ALPHA = TWOPI - PHIFP
    CALL GETRX(ALPHA, RX)
    CALL FSTSPLT(RX,PTHETA,XI1,XI2,XI3,DVTOT,TTOT,IFLG)
    IF (RO.EQ.RT) THEN
      IF (THETA F.EQ.PHIF) TTOT = 0.0
    END IF
    GO TO 40
  END IF
  ALPHA = 2.*TWOPI-PHIFP
  CALL GETRX(ALPHA, RX)
  CALL FSTSPLT(RX,PTHETA,XI1,XI2,XI3,DVTOT,TTOT,IFLG)
  IF (RO.EQ.RT) THEN
    IF (THETA F.EQ.PHIF) TTOT = 0.0
  END IF
  GO TO 40
END IF
END IF
END IF
*
* TOTAL WAITIME IN R0 ORBIT BEFORE RENDEZVOUS INITIATION
*
* TWAIT = TWAIT1 + TWAIT2
*
* CASE WHEN A S/V IS AT PI
*
IF (ABS(THETA F-PI).LT.0.01) THEN
  ALPHA = 3.*PI-PHIF
  IF (ALP1MIN.LT.0.0) THEN
    IF (PHIF.GE.0.0.AND.PHIF.LE.AL P1MAX) ALPHA = PI-PHIF
  END IF
  IF (ALP2MIN.LT.0.) THEN
    IF (PHIF.GE.0.0.AND.PHIF.LE.AL P2MAX) ALPHA = PI-PHIF
  END IF
  IF (RO.GT.RT) THEN
    IF (PHIF.GE.0.0.AND.PHIF.LE.AL P2MAX) PHIF = PHIF+TWOPI
    ALPHA = 5.*PI-PHIF
    PHIF = AMOD(PHIF,TWOPI)
  END IF

```

```

      END IF
*
*
*   CASE WHEN A S/V IS AT 0 OR TWOPI
*
      IF (ABS(THETA2-0.0).LT.0.01.OR.ABS(THETA2-TWOPI).LT.0.01) THEN
        ALPHA = 2.*PI-PHIF
        IF (R0.GT.RT) THEN
          IF (PHIF.GE.0.0.AND.PHIF.LE.ALPIMAX) PHIF = PHIF+TWOPI
          ALPHA = 6.*PI-PHIF
          PHIF = AMOD(PHIF,TWOPI)
        END IF
      END IF
*
30    CONTINUE
*
*   KNOWING ALPHA, FIND THE CORRESPONDING TRANSFER RADIUS RX
*
      CALL GETRX(ALPHA,RX)
*
*   KNOWING RX AND PLANE CHANGE ANGLE PTHETA, FIND DELTA V AND
*   TRANSFER TIME
*
      CALL FSTSPLT(RX,PTHETA,XI1,XI2,XI3,DVTOT,TTOT,IFLG)
*
40    CONTINUE
*
      WRITE(11,53) THETA2*RADEG,THETA*RADEG,THETA2*RADEG,
&   PHI*RADEG,PHIF*RADEG,MX,RX,TWAIT1,TWAIT2,DVTOT,
&   XI1,XI2,XI3,TTOT
53    FORMAT(2X,5F9.2,2X,I3,2X,F9.3,2X,F9.2,2X,F9.2,2X,
&           F7.4,2X,2X,F6.2,2X,F6.2,2X,F6.2,2X,F9.2)
*
600   CONTINUE
*
      THETA = THETA2
      THETA = THETA + DTH
      IF (THETA.GT.THMAX) GO TO 300
      GO TO 200
*
300   CONTINUE
*
      PHI=PHI+DPH
      IF (PHI.GT.PHMAX) GO TO 400
      GO TO 201
*
400   CONTINUE
*
      CLOSE(11)
      WRITE(6,9999) ' PRINT FILE IS IN REND.OUT'
      END
*

```

```

SUBROUTINE WAITIME(A1MIN,A1MAX,A2MIN,A2MAX,THETAG,PHIO,
&
                THETA,PHI,M,TWAIT)
*
COMMON XMU, PI, R0, RT, P0, PT
*
TWOPI = 2.*PI
DEGRAD = PI/180.
RADEG = 1./DEGRAD
*
TAU = PI*P0/PT
*
PHI = PHIO
PHI = AMOD(PHI,TWOPI)
THETA = THETA0
THETA = AMOD(THETA,TWOPI)
M = 0
*
20 CONTINUE
*
IF S/V IS AT 0 OR TWOPI
* CHECK FOR RENDEZVOUS POSSIBILITY
*
PHITEMP = PHI
DIFF1 = ABS(THETA - TWOPI)
DIFF2 = ABS(THETA - 0.0)
IF (DIFF1.LT.0.001.OR.DIFF2.LT.0.001) THEN
IF (A1MIN.LE.0.0.AND.A1MAX.GE.0.0) THEN
    IF (PHITEMP.GT.A1MAX) PHITEMP = PHITEMP - TWOPI
END IF
IF (PHITEMP.GE.A1MIN.AND.PHITEMP.LE.A1MAX) GO TO 30
DIF1 = ABS(PHITEMP-A1MIN)
DIF2 = ABS(PHITEMP-A1MAX)
IF (DIF1.LT.0.001.AND.DIF2.LT.0.001) THEN
    THETA = THETA0
    PHI = PHIO
    M = 999
    TWAIT = 0.
    GO TO 40
END IF
END IF
*
IF S/V IS AT PI
* CHECK FOR RENDEZVOUS POSSIBILITY
*
IF (ABS(THETA-PI).LT.0.001) THEN
IF (A2MIN.LE.0.0.AND.A2MAX.GE.0.0) THEN
    IF (PHITEMP.GT.A2MAX) PHITEMP = PHITEMP - TWOPI
END IF
IF (PHITEMP.GE.A2MIN.AND.PHITEMP.LE.A2MAX) GO TO 30
DIF1 = ABS(PHITEMP-A2MIN)
DIF2 = ABS(PHITEMP-A2MAX)
IF (DIF1.LT.0.001.AND.DIF2.LT.0.001) THEN
    THETA = THETA0
    PHI = PHIO
    M = 999
    TWAIT = 0.
    GO TO 40
END IF

```

```

      END IF
*
*   IF NOT, S/V WAITS HALF REVOLUTION
*   M = NO OF HALF REVOLUTIONS FO S/V
*
*   CHECK IF WAITING IS TOO LONG
*
      IF ((FLOATJ(M)*P0).GE.(48.*3600.)) THEN
        THETA = THETA0
        PHI = PHI0
        M = 999
        TWAIT = 0.
        GO TO 40
      END IF
*
      M = M + 1
      THETA = THETA + PI
      PHI = PHI + TAU
      THETA = AMOD(THETA,TWOPI)
      PHI = AMOD(PHI,TWOPI)
      IF (PHI.LT.0.0) PHI = PHI + TWOPI
*
      GO TO 20
*
30    CONTINUE
*
*   RENDEZVOUS INITIAITON IS POSSIBLE
*
      IF (PHI.LT.0.0) PHI = PHI + TWOPI
      TWAIT = FLOAT(M)*P0/2.
      PHI = AMOD(PHI,TWOPI)
*
      IF (R0.EQ.RT) M = 999
*
40    CONTINUE
      RETURN
      END
*

```



```

SUBROUTINE GETRX(ALPHA,RX)
*
* THE SUBROUTINE USED IN GETTING RX OF TRANSFER ORBIT
* KNOWING ANGULAR POSITION OF A TARGET
*
COMMON XMU, PI, R0, RT, P0, PT
*
F(X) = A*((R0+X)**1.5+(RT+X)**1.5)-ALPHA
DF(X) = DA*(SQRT(R0+X) + SQRT(RT+X))
*
A = 0.5*PI/SQRT(2.*RT**3)
DA=1.5*A
*
N = 0
RX = 0.
IF (F(RX).GE.0.) STOP 'ERROR - NO SOLUTION'
*
10 RXX = RX - F(RX)/DF(RX)
IF (N.GT.50) STOP 'ERROR - NO CONVERGENCE'
IF (ABS(RXX-RX).GT.0.01) THEN
    N = N + 1
    RX = RXX
    GO TO 10
END IF
*
RETURN
END
*

```

```

SUBROUTINE FSTSPLT(RX,A,A1,A2,A3,DV,TIM,IFLG)
*
*
* ABSTRACT: THIS SUBROUTINE ESTIMATES THE ANGLE COMBNATION FOR A
*           3 SPLIT PLANE CHANGE IN A BI-ELLIPTIC TRANSFER THAT
*           REQUIRES THE MINIMUM TOTAL DELTA V. THE ESTIMATES FOR
*           THE MINIMUM TOTAL DELTA V SHOULD BE GOOD (WITHIN 0.5
*           PERCENT) FOR PLANE CHANGE ANGLES LESS THAN OR EQUAL
*           TO 10 DEGREES.
*
* ARGUMENTS:
*
* NAME      TYPE  IN/OUT  COMMENT
* RX        RL    IN      TRANSFER RADIUS
* A         RL    IN      PLANE CHANGE (DEG)
* A1        RL    OUT     PLANE CHANGE AT FIRST BURN (DEG)
* A2        RL    OUT     PLANE CHANGE AT SECOND BURN (DEG)
* A3        RL    OUT     PLANE CHANGE AT THIRD BURN (DEG)
* DV        RL    OUT     TOTAL DELTA V NEEDED FOR TRANSFER (KM/SEC)
* TIM       RL    OUT     TOTAL TIME OF TRANSFER (SEC)
* IFLG     INT    OUT     0 = O.K., 1 = ESTIMATE MAY NOT BE
*                               ACCURATE FOR A>10.
*
*
* VERSION  DATE      NAME  COMMENTS
* 1.00    19MAY88   DCW   ORIGINAL CODE-GENERAL RESEARCH CORP. - LA
* 1.00    7JUN88   SCM   ADDED COMMON STATEMENT, TRANSFER TIME
*
*
* PARAMETER (AMAX =10.0)
* COMMON XMU, PI, R0, RT, P0, PT
*
*
* SET WARNING FLAG FOR LARGE PLANE CHANGES
*
*
* GMU = XMU
* IF (A.GT.AMAX) THEN
*   IFLG = 1
* ELSE
*   IFLG = 0
* END IF
*
*
* CALCULATE TRANSFER TIME
*
*
* A12 = (RX+R0)/2.
* A23 = (RX+RT)/2.
* T12 = PI*SQRT(A12**3/GMU)
* T23 = PI*SQRT(A23**3/GMU)
* TIM = T12 + T23
*
*
* CALCULATE TRANSFER VELOCITIES
*
*
* V1 = SQRT(GMU/R0)
* VT1 = SQRT(2*GMU*RX/(R0*(R0+RX)))
* V2 = SQRT(2*GMU*R0/(RX*(RX+R0)))
* VT2 = SQRT(2*GMU*RT/(RX*(RX+RT)))
* V3 = SQRT(2*GMU*RX/(RT*(RT+RX)))
* VT3 = SQRT(GMU/RT)
*
*
* ESTIMATE THE OPTIMAL SPLIT PLANE CHANGE
*
*
* E1 = ABS(VT1-V1)
* E2 = ABS(VT2-V2)
* E3 = ABS(VT3-V3)

```

```

EE = E3*V1*VT1*V2*VT2+E1*V2*VT2*V3*VT3+E2*V1*VT1*V3*VT3
IF (EE.NE.0.) THEN
  A1 = A*E1*V2*VT2*V3*VT3/EE
  A2 = A*E2*V1*VT1*V3*VT3/EE
ELSE
  A1 = 0.
  A2 = 0.
END IF
A3 = A-A1-A2
*
*   CALCULATE THE RESULTING TOTAL DELTA V
*
  DV = SQRT(V1*V1+VT1*VT1-2.0*V1*VT1*COSD(A1))
&   +SQRT(V2*V2+VT2*VT2-2.0*V2*VT2*COSD(A2))
&   +SQRT(V3*V3+VT3*VT3-2.0*V3*VT3*COSD(A3))
*
  RETURN
END

```

Sample Output Excerpt

**** INPUT DATA ****

S/V POSITION: THETAMIN, THETAMAX, DELTHETA = 0.00 360.00 20.00
 S/C POSITION: PHMIN, PHMAX, DPH = 0.00 360.00 20.00
 INITIAL ALT = 100.00KM FINAL ALT = 1000.00KM
 MIN ALT, MAX ALT, PLANE CHANGE ANGLE (KM, DEG) = 100.00 1000.00 5.00

**** OUTPUT DATA ****

ALP1MIN, ALP1MAX = 16.21 48.12 DEG ALP2MIN, ALP2MAX = 196.21 228.12 DEG

THETA1	THETA2	THETA3	PHI	PHI2	M	RX	TWAIT1	TWAIT2	DVTOT	I1	I2	I3	TTOT
0.00	0.00	0.00	0.00	40.91	10	6684.266	0.00	25945.23	0.8256	0.57	2.39	2.03	5590.44
20.00	80.00	180.00	0.00	215.91	3	6826.14	2306.24	7783.57	0.8259	0.95	2.46	1.59	5678.02
40.00	180.00	180.00	0.00	199.45	3	7288.087	2017.96	7783.57	0.8251	2.09	2.66	0.24	5966.30
60.00	180.00	0.00	0.00	34.91	2	6854.436	1729.68	5189.05	0.8259	1.03	2.47	1.50	5695.54
80.00	180.00	0.00	0.00	18.45	2	7315.912	1441.40	5189.05	0.8250	2.16	2.68	0.17	5983.82
100.00	180.00	180.00	0.00	213.91	1	6882.695	1153.12	2594.52	0.8259	1.10	2.48	1.42	5713.06
120.00	180.00	180.00	0.00	197.45	1	7343.710	864.84	2594.52	0.8250	2.22	2.69	0.09	6001.34
140.00	180.00	0.00	0.00	32.91	0	6910.927	576.56	0.00	0.8260	1.17	2.50	1.33	5730.57
160.00	180.00	0.00	0.00	16.45	0	7371.482	288.28	0.00	0.8257	2.28	2.70	0.02	6018.85
180.00	180.00	0.00	0.00	40.91	10	6684.266	0.00	25945.23	0.8256	0.57	2.39	2.03	5590.44
200.00	0.00	180.00	0.00	215.91	3	6826.147	2306.24	7783.57	0.8259	0.95	2.46	1.59	5678.02
220.00	0.00	180.00	0.00	199.45	3	7288.087	2017.96	7783.57	0.8251	2.09	2.66	0.24	5966.30
240.00	0.00	0.00	0.00	34.91	2	6854.436	1729.68	5189.05	0.8259	1.03	2.47	1.50	5695.54
260.00	0.00	0.00	0.00	18.45	2	7315.912	1441.40	5189.05	0.8250	2.16	2.68	0.17	5983.82
280.00	0.00	180.00	0.00	213.91	1	6882.695	1153.12	2594.52	0.8259	1.10	2.48	1.42	5713.06
300.00	0.00	180.00	0.00	197.45	1	7343.710	864.84	2594.52	0.8250	2.22	2.69	0.09	6001.34
320.00	0.00	0.00	0.00	32.91	0	6910.925	576.56	0.00	0.8260	1.17	2.50	1.33	5730.57
340.00	0.00	0.00	0.00	16.45	0	7371.482	288.28	0.00	0.8257	2.28	2.70	0.02	6018.85
360.00	360.00	0.00	0.00	40.91	10	6684.266	0.00	25945.23	0.8256	0.57	2.39	2.03	5590.44

Appendix C

Sample COSEMS Output File

COSEMS
SUMMARY OUTPUT REPORT

Case File Name: POLAR
COSEMS Version: 5.1

Date: 21-OCT-1991
Page: 1

* AVAILABILITY STATISTICS *

=====
- Constellation Operational Availability -
=====

Year	LOW Availability (%)
1	14.92 +/- 0.07
2	53.62 +/- 0.41
3	82.31 +/- 0.65
4	98.74 +/- 0.59
5	98.93 +/- 0.51
6	97.07 +/- 0.82
7	95.58 +/- 2.26
8	95.60 +/- 2.91
9	95.35 +/- 2.80
10	97.46 +/- 1.57
Average:	82.96 +/- 0.85

COSEMS SUMMARY OUTPUT REPORT

=====

- Cumulative Operational Availabilities -

=====

Avail- ability (%)	LOW	
	% Time Less Than	Availability
5	2.50 +/-	0.00
10	5.00 +/-	0.00
15	5.00 +/-	0.00
20	7.50 +/-	0.00
25	7.50 +/-	0.00
30	10.00 +/-	0.00
35	10.00 +/-	0.00
40	12.50 +/-	0.00
45	12.50 +/-	0.00
50	15.00 +/-	0.00
55	15.00 +/-	0.00
60	17.50 +/-	0.00
65	17.87 +/-	0.43
70	20.25 +/-	0.36
75	23.37 +/-	1.33
80	26.38 +/-	2.13
85	27.75 +/-	3.20
90	32.88 +/-	4.64
95	42.00 +/-	5.60
100	100.00 +/-	0.00

COSEMS SUMMARY OUTPUT REPORT

=====

= Ring Operational Availability =

=====

LOW

Year	Ring	1	Ring	2	Ring	3
1	44.91 +/-	0.06	0.00 +/-	0.00	29.67 +/-	0.40
2	69.10 +/-	1.20	59.74 +/-	0.11	58.51 +/-	2.20
3	99.52 +/-	0.61	79.34 +/-	0.74	98.88 +/-	1.86
4	97.61 +/-	2.27	98.71 +/-	2.10	98.18 +/-	2.41
5	98.37 +/-	1.48	99.67 +/-	0.18	99.05 +/-	0.92
6	97.04 +/-	1.80	96.86 +/-	2.06	95.24 +/-	2.92
7	96.50 +/-	2.32	94.00 +/-	5.71	96.28 +/-	2.60
8	95.17 +/-	4.23	96.88 +/-	2.70	94.17 +/-	4.73
9	96.31 +/-	2.53	95.83 +/-	3.41	96.12 +/-	2.89
10	97.08 +/-	2.91	98.46 +/-	1.24	99.11 +/-	0.67
Average:	89.16 +/-	1.09	81.95 +/-	0.97	86.52 +/-	0.83
Year	Ring	4	Ring	5	Ring	6
1	0.00 +/-	0.00	14.95 +/-	0.03	0.00 +/-	0.00
2	44.81 +/-	0.07	59.67 +/-	0.38	29.89 +/-	0.05
3	67.29 +/-	2.19	89.45 +/-	0.55	59.39 +/-	0.99
4	99.59 +/-	0.39	98.66 +/-	1.55	99.72 +/-	0.09
5	98.46 +/-	2.09	98.73 +/-	1.40	99.31 +/-	0.72
6	99.84 +/-	0.08	94.85 +/-	2.44	98.59 +/-	1.68
7	94.75 +/-	3.79	97.05 +/-	2.13	94.91 +/-	2.69
8	95.46 +/-	4.01	95.56 +/-	2.53	96.38 +/-	2.75
9	93.97 +/-	5.37	96.49 +/-	2.29	93.40 +/-	5.19
10	96.50 +/-	3.66	98.33 +/-	1.24	95.29 +/-	4.63
Average:	79.07 +/-	1.41	84.37 +/-	0.70	76.69 +/-	1.32

COSEMS SUMMARY OUTPUT REPORT

 * CRITICAL SATELLITE FAILURES *

Year	Cause				Total	
	Hardware Failure		Consumable Depletion			
1	0.05 +/-	0.10	0.00 +/-	0.00	0.05 +/-	0.10
2	0.30 +/-	0.22	0.00 +/-	0.00	0.30 +/-	0.22
3	0.95 +/-	0.47	0.00 +/-	0.00	0.95 +/-	0.47
4	1.85 +/-	0.61	0.00 +/-	0.00	1.85 +/-	0.61
5	2.15 +/-	0.61	0.00 +/-	0.00	2.15 +/-	0.61
6	2.20 +/-	0.64	1.50 +/-	1.01	3.70 +/-	1.30
7	1.00 +/-	0.57	1.90 +/-	1.40	2.90 +/-	1.58
8	2.00 +/-	0.64	1.15 +/-	0.98	3.15 +/-	1.12
9	2.10 +/-	0.68	0.30 +/-	0.38	2.40 +/-	0.70
10	1.80 +/-	0.54	0.00 +/-	0.00	1.80 +/-	0.54
Total:	14.40 +/-	1.61	4.85 +/-	3.16	19.25 +/-	3.43

Year	Occurrences of Excessive Down Time				Total	
	Hardware Failure		Consumable Depletion			
1	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
4	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
5	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
6	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
7	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
8	0.05 +/-	0.10	0.30 +/-	0.63	0.35 +/-	0.63
9	0.10 +/-	0.14	0.40 +/-	0.60	0.50 +/-	0.69
10	0.00 +/-	0.00	0.10 +/-	0.14	0.10 +/-	0.14
Total:	0.15 +/-	0.17	0.80 +/-	1.30	0.95 +/-	1.40

COSEMS SUMMARY OUTPUT REPORT

Year	LOW Operational Failures				Total	
	ORU Replacement		Consumable Replenishment			
1	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
4	0.10 +/-	0.14	0.00 +/-	0.00	0.10 +/-	0.14
5	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
6	0.45 +/-	0.32	1.10 +/-	0.57	1.55 +/-	0.56
7	0.45 +/-	0.32	0.80 +/-	0.42	1.25 +/-	0.48
8	0.40 +/-	0.28	0.30 +/-	0.22	0.70 +/-	0.34
9	0.10 +/-	0.21	0.15 +/-	0.17	0.25 +/-	0.26
10	0.15 +/-	0.17	0.05 +/-	0.10	0.20 +/-	0.19
Total:	1.65 +/-	0.61	2.40 +/-	0.92	4.05 +/-	0.98

 - Hardware Failure (By Subsystem) -

Year	LOW -- STRUCTURE & THERMAL --			
	First Element		Second Element	
1	0.05 +/-	0.10	0.00 +/-	0.00
2	0.15 +/-	0.17	0.00 +/-	0.00
3	0.50 +/-	0.39	0.00 +/-	0.00
4	0.35 +/-	0.38	0.00 +/-	0.00
5	0.70 +/-	0.22	0.00 +/-	0.00
6	0.65 +/-	0.38	0.00 +/-	0.00
7	0.40 +/-	0.32	0.00 +/-	0.00
8	0.50 +/-	0.24	0.00 +/-	0.00
9	0.30 +/-	0.22	0.00 +/-	0.00
10	0.40 +/-	0.24	0.00 +/-	0.00
Total:	4.00 +/-	0.82	0.00 +/-	0.00

COSEMS SUMMARY OUTPUT REPORT

LOW
-- ELECTRICAL POWER --

Year	First Element		Second Element	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.10 +/-	0.14	0.00 +/-	0.00
3	0.10 +/-	0.14	0.00 +/-	0.00
4	0.65 +/-	0.44	0.00 +/-	0.00
5	0.55 +/-	0.32	0.00 +/-	0.00
6	0.75 +/-	0.43	0.00 +/-	0.00
7	0.30 +/-	0.27	0.00 +/-	0.00
8	0.75 +/-	0.40	0.00 +/-	0.00
9	0.75 +/-	0.40	0.00 +/-	0.00
10	0.70 +/-	0.38	0.00 +/-	0.00
Total:	4.65 +/-	1.16	0.00 +/-	0.00

-- GUIDANCE & CONTROL --

Year	First Element		Second Element	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.05 +/-	0.10	0.00 +/-	0.00
3	0.30 +/-	0.22	0.00 +/-	0.00
4	0.75 +/-	0.37	0.00 +/-	0.00
5	0.85 +/-	0.46	0.00 +/-	0.00
6	0.65 +/-	0.38	0.00 +/-	0.00
7	0.30 +/-	0.22	0.00 +/-	0.00
8	0.55 +/-	0.39	0.00 +/-	0.00
9	0.90 +/-	0.48	0.00 +/-	0.00
10	0.45 +/-	0.28	0.00 +/-	0.00
Total:	4.80 +/-	0.92	0.00 +/-	0.00

COSEMS SUMMARY OUTPUT REPORT

LOW
-- PAYLOAD --

Year	First Element		Second Element	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.05 +/-	0.10	0.00 +/-	0.00
4	0.10 +/-	0.14	0.00 +/-	0.00
5	0.05 +/-	0.10	0.00 +/-	0.00
6	0.15 +/-	0.17	0.00 +/-	0.00
7	0.00 +/-	0.00	0.00 +/-	0.00
8	0.20 +/-	0.19	0.00 +/-	0.00
9	0.15 +/-	0.17	0.00 +/-	0.00
10	0.25 +/-	0.21	0.00 +/-	0.00
Total:	0.95 +/-	0.47	0.00 +/-	0.00

COSEMS SUMMARY OUTPUT REPORT

 * CRITICAL SASS FAILURES *

LOW

Year	STV/FTS		OSCRS	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.00 +/-	0.00	0.00 +/-	0.00
5	0.05 +/-	0.10	0.00 +/-	0.00
6	0.25 +/-	0.26	0.20 +/-	0.24
7	0.25 +/-	0.26	0.25 +/-	0.26
8	0.10 +/-	0.14	0.20 +/-	0.19
9	0.05 +/-	0.10	0.00 +/-	0.00
10	0.05 +/-	0.10	0.05 +/-	0.10
Total:	0.75 +/-	0.50	0.70 +/-	0.40

COSEMS SUMMARY OUTPUT REPORT

 * NUMBER OF MISSIONS *

LOW

-- Satellite Replacement Due To Excessive Down Time --						
Year	Hardware Failure		Consumable Depletion		Total	
1	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
4	0.20 +/-	0.19	0.00 +/-	0.00	0.20 +/-	0.19
5	0.15 +/-	0.17	0.00 +/-	0.00	0.15 +/-	0.17
6	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
7	0.05 +/-	0.10	0.00 +/-	0.00	0.05 +/-	0.10
8	0.05 +/-	0.10	0.40 +/-	0.84	0.45 +/-	0.84
9	0.00 +/-	0.00	0.35 +/-	0.51	0.35 +/-	0.51
10	0.10 +/-	0.14	0.15 +/-	0.23	0.25 +/-	0.34
Total:	0.55 +/-	0.36	0.90 +/-	1.50	1.45 +/-	1.77
-- Satellite Replacement Due To Operational Failures --						
Year	Failed ORU Replacement		Failed Consumable Replenishment		Total	
1	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
4	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
5	0.10 +/-	0.14	0.00 +/-	0.00	0.10 +/-	0.14
6	0.20 +/-	0.19	0.65 +/-	0.38	0.85 +/-	0.41
7	0.40 +/-	0.32	0.75 +/-	0.43	1.15 +/-	0.53
8	0.20 +/-	0.19	0.40 +/-	0.28	0.60 +/-	0.35
9	0.15 +/-	0.17	0.25 +/-	0.21	0.40 +/-	0.28
10	0.20 +/-	0.24	0.05 +/-	0.10	0.25 +/-	0.26
Total:	1.25 +/-	0.52	2.10 +/-	0.87	3.35 +/-	0.96

COSEMS SUMMARY OUTPUT REPORT

LOW
-- Other Satellite Replacement --

Year	Hardware Failure		Consumable Depletion		Total	
1	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
4	0.20 +/-	0.19	0.00 +/-	0.00	0.20 +/-	0.19
5	0.65 +/-	0.41	0.00 +/-	0.00	0.65 +/-	0.41
6	0.50 +/-	0.39	0.00 +/-	0.00	0.50 +/-	0.39
7	0.30 +/-	0.27	0.00 +/-	0.00	0.30 +/-	0.27
8	0.25 +/-	0.21	0.00 +/-	0.00	0.25 +/-	0.21
9	0.40 +/-	0.24	0.00 +/-	0.00	0.40 +/-	0.24
10	0.55 +/-	0.32	0.00 +/-	0.00	0.55 +/-	0.32
Total:	2.85 +/-	0.76	0.00 +/-	0.00	2.85 +/-	0.76

-- SASS Replacement --

Year	STV/FTS		OSCRS		Total	
1	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
4	0.00 +/-	0.00	0.00 +/-	0.00	0.00 +/-	0.00
5	0.05 +/-	0.10	0.00 +/-	0.00	0.05 +/-	0.10
6	0.25 +/-	0.26	0.20 +/-	0.24	0.45 +/-	0.32
7	0.20 +/-	0.24	0.15 +/-	0.17	0.35 +/-	0.38
8	0.10 +/-	0.14	0.25 +/-	0.21	0.35 +/-	0.23
9	0.05 +/-	0.10	0.05 +/-	0.10	0.10 +/-	0.14
10	0.05 +/-	0.10	0.05 +/-	0.10	0.10 +/-	0.14
Total:	0.70 +/-	0.48	0.70 +/-	0.40	1.40 +/-	0.72

COSEMS SUMMARY OUTPUT REPORT

Year	LOW	
	On-Orbit Support	From Ground
1	0.00 +/-	0.00
2	0.00 +/-	0.00
3	0.00 +/-	0.00
4	0.35 +/-	0.27
5	0.25 +/-	0.21
6	4.30 +/-	0.34
7	4.00 +/-	0.53
8	4.65 +/-	0.95
9	1.90 +/-	0.59
10	1.85 +/-	0.68
Total:	17.30 +/-	1.21

 * RESOURCES CONSUMED *

=====

= Number of Successful and Unsuccessful Launches (Site) =

=====

Year	WTR			
	(ATLAS II	AS)	(ATLAS II DOD/CENT)	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.50 +/-	0.32	5.85 +/-	0.17
5	1.00 +/-	0.55	0.65 +/-	0.46
6	1.40 +/-	0.49	0.25 +/-	0.26
7	1.60 +/-	0.58	0.20 +/-	0.24
8	1.40 +/-	0.88	0.15 +/-	0.17
9	1.30 +/-	0.63	0.10 +/-	0.21
10	1.15 +/-	0.65	0.05 +/-	0.10
Total:	8.35 +/-	1.80	7.25 +/-	0.57

COSEMS SUMMARY OUTPUT REPORT

Year			WTR	
	(ATLAS J/STAR 20)		(DELTA 7920)	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.05 +/-	0.10	0.00 +/-	0.00
5	0.00 +/-	0.00	0.00 +/-	0.00
6	0.15 +/-	0.17	4.05 +/-	0.32
7	0.40 +/-	0.35	4.05 +/-	0.36
8	0.45 +/-	0.32	3.95 +/-	0.47
9	0.70 +/-	0.38	0.75 +/-	0.66
10	0.65 +/-	0.38	0.45 +/-	0.39
Total:	2.40 +/-	0.82	13.25 +/-	0.73
Year	PEGASUS	-	SCOUT	-
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.15 +/-	0.17	0.05 +/-	0.10
5	0.20 +/-	0.19	0.00 +/-	0.00
6	0.45 +/-	0.28	0.05 +/-	0.10
7	0.20 +/-	0.19	0.00 +/-	0.00
8	0.70 +/-	0.34	0.10 +/-	0.14
9	0.75 +/-	0.40	0.25 +/-	0.30
10	0.75 +/-	0.30	0.05 +/-	0.10
Total:	3.20 +/-	0.75	0.50 +/-	0.36

COSEMS SUMMARY OUTPUT REPORT

 =====
 - Number of Launch Failures -
 =====

LOW

-- Satellite Replacement Due To Excessive Down Time --				
Year	Hardware Failure		Consumable Depletion	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.05 +/-	0.10	0.00 +/-	0.00
5	0.00 +/-	0.00	0.00 +/-	0.00
6	0.00 +/-	0.00	0.00 +/-	0.00
7	0.00 +/-	0.00	0.00 +/-	0.00
8	0.00 +/-	0.00	0.00 +/-	0.00
9	0.00 +/-	0.00	0.05 +/-	0.10
10	0.05 +/-	0.10	0.00 +/-	0.00
Total:	0.10 +/-	0.14	0.05 +/-	0.10
-- Satellite Replacement Due To Operational Failures --				
Year	Failed ORU Replacement		Failed Consumable Replenishment	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.00 +/-	0.00	0.00 +/-	0.00
5	0.00 +/-	0.00	0.00 +/-	0.00
6	0.05 +/-	0.10	0.00 +/-	0.00
7	0.05 +/-	0.10	0.05 +/-	0.10
8	0.05 +/-	0.10	0.05 +/-	0.10
9	0.00 +/-	0.00	0.00 +/-	0.00
10	0.05 +/-	0.10	0.00 +/-	0.00
Total:	0.20 +/-	0.24	0.10 +/-	0.14

COSEMS SUMMARY OUTPUT REPORT

LOW				
-- Other Satellite Replacement --				
Year	Hardware Failure		Consumable Depletion	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.05 +/-	0.10	0.00 +/-	0.00
5	0.10 +/-	0.14	0.00 +/-	0.00
6	0.00 +/-	0.00	0.00 +/-	0.00
7	0.00 +/-	0.00	0.00 +/-	0.00
8	0.00 +/-	0.00	0.00 +/-	0.00
9	0.10 +/-	0.14	0.00 +/-	0.00
10	0.00 +/-	0.00	0.00 +/-	0.00
Total:	0.25 +/-	0.26	0.00 +/-	0.00
Year	SASS Replacement			
1	0.00 +/-	0.00		
2	0.00 +/-	0.00		
3	0.00 +/-	0.00		
4	0.00 +/-	0.00		
5	0.00 +/-	0.00		
6	0.00 +/-	0.00		
7	0.10 +/-	0.21		
8	0.15 +/-	0.17		
9	0.05 +/-	0.10		
10	0.00 +/-	0.00		
Total:	0.30 +/-	0.38		

COSEMS SUMMARY OUTPUT REPORT

LOW

Year	On-Orbit Support From Ground		Total Failures	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.50 +/-	0.28	0.60 +/-	0.28
5	0.00 +/-	0.00	0.10 +/-	0.14
6	0.20 +/-	0.19	0.25 +/-	0.21
7	0.25 +/-	0.21	0.45 +/-	0.32
8	0.15 +/-	0.17	0.40 +/-	0.35
9	0.35 +/-	0.27	0.55 +/-	0.39
10	0.05 +/-	0.10	0.15 +/-	0.17
Total:	1.50 +/-	0.58	2.50 +/-	0.78

- Number of Satellites Deployed (Including Spares), Replaced After -
- Initial Deployment, and Lost in Ground Launch Failures -

Year	LOW Satellites	
1	9.00 +/-	0.00
2	12.00 +/-	0.00
3	12.00 +/-	0.00
4	3.50 +/-	0.32
5	1.00 +/-	0.55
6	1.40 +/-	0.49
7	1.60 +/-	0.58
8	1.40 +/-	0.88
9	1.30 +/-	0.63
10	1.15 +/-	0.65
Total:	44.35 +/-	1.80

COSEMS SUMMARY OUTPUT REPORT

 = Number of SASS Units Deployed, Replaced After Initial =
 = Deployment, and Lost in Ground Launch Failures =

Year	STVs		FTSs	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	5.85 +/-	0.17	5.85 +/-	0.17
5	0.65 +/-	0.46	0.65 +/-	0.46
6	0.25 +/-	0.26	0.25 +/-	0.26
7	0.20 +/-	0.24	0.20 +/-	0.24
8	0.15 +/-	0.17	0.15 +/-	0.17
9	0.10 +/-	0.21	0.10 +/-	0.21
10	0.05 +/-	0.10	0.05 +/-	0.10
Total:	7.25 +/-	0.57	7.25 +/-	0.57

Year	FMs	
1	0.00 +/-	0.00
2	0.00 +/-	0.00
3	0.00 +/-	0.00
4	0.00 +/-	0.00
5	0.00 +/-	0.00
6	0.00 +/-	0.00
7	0.00 +/-	0.00
8	0.00 +/-	0.00
9	0.00 +/-	0.00
10	0.00 +/-	0.00
Total:	0.00 +/-	0.00

COSEMS SUMMARY OUTPUT REPORT

Year	OSCRS	COLD/MONO
1	0.00 +/-	0.00
2	0.00 +/-	0.00
3	0.00 +/-	0.00
4	0.00 +/-	0.00
5	0.00 +/-	0.00
6	4.05 +/-	0.32
7	4.05 +/-	0.36
8	3.80 +/-	0.62
9	0.60 +/-	0.47
10	0.40 +/-	0.35
Total:	12.90 +/-	0.40

 = Number of ORUS Deployed and Resupplied to SBSPS, Replaced on =
 = Satellites Directly from the Ground and Lost in Ground Launch =
 = Failures =

LOW
 -- ELECTRICAL POWER --

Year	First Element		Second Element	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.45 +/-	0.44	0.00 +/-	0.00
5	0.50 +/-	0.42	0.00 +/-	0.00
6	4.10 +/-	0.69	0.00 +/-	0.00
7	4.15 +/-	0.89	0.00 +/-	0.00
8	3.50 +/-	1.06	0.00 +/-	0.00
9	1.80 +/-	1.04	0.00 +/-	0.00
10	1.20 +/-	0.85	0.00 +/-	0.00
Total:	15.70 +/-	2.03	0.00 +/-	0.00

COSEMS SUMMARY OUTPUT REPORT

LOW
-- GUIDANCE & CONTROL --

Year	First Element		Second Element	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.15 +/-	0.23	0.00 +/-	0.00
5	0.05 +/-	0.10	0.00 +/-	0.00
6	4.35 +/-	0.99	0.00 +/-	0.00
7	4.15 +/-	1.09	0.00 +/-	0.00
8	3.70 +/-	1.11	0.00 +/-	0.00
9	1.55 +/-	0.80	0.00 +/-	0.00
10	1.30 +/-	0.75	0.00 +/-	0.00
Total:	15.25 +/-	2.12	0.00 +/-	0.00

-- PAYLOAD --

Year	First Element		Second Element	
1	0.00 +/-	0.00	0.00 +/-	0.00
2	0.00 +/-	0.00	0.00 +/-	0.00
3	0.00 +/-	0.00	0.00 +/-	0.00
4	0.40 +/-	0.63	0.00 +/-	0.00
5	0.05 +/-	0.10	0.00 +/-	0.00
6	1.40 +/-	0.65	0.00 +/-	0.00
7	0.90 +/-	0.40	0.00 +/-	0.00
8	1.20 +/-	0.85	0.00 +/-	0.00
9	0.55 +/-	0.69	0.00 +/-	0.00
10	0.75 +/-	0.62	0.00 +/-	0.00
Total:	5.25 +/-	1.83	0.00 +/-	0.00

COSEMS SUMMARY OUTPUT REPORT

```

=====
- Total ORUS Deployed, Resupplied, Replaced and Lost in Ground -
- Launch Failures for All Carrier Platform Subsystems and -
- Payload Elements -
=====

```

Year	LOW ORUs		
1	0.00 +/-		0.00
2	0.00 +/-		0.00
3	0.00 +/-		0.00
4	1.00 +/-		0.82
5	0.60 +/-		0.53
6	9.85 +/-		1.58
7	9.20 +/-		1.66
8	8.40 +/-		2.37
9	3.90 +/-		1.37
10	3.25 +/-		1.31

Total: 36.20 +/- 4.42

```

=====
- Satellite Consumables Deployed and Lost in Ground Launch -
- Failures for On-Orbit Support from Ground (Integer Multiple of -
- OSCRS Capacity) -
=====

```

Year	OSCRS COLD/MONO	LOW	
1	0.00 +/-		0.00
2	0.00 +/-		0.00
3	0.00 +/-		0.00
4	0.00 +/-		0.00
5	0.00 +/-		0.00
6	4.05 +/-		0.32
7	4.05 +/-		0.36
8	3.80 +/-		0.62
9	0.60 +/-		0.47
10	0.40 +/-		0.35

Total: 12.90 +/- 0.40

COSEMS SUMMARY OUTPUT REPORT

 - Satellite Consumables of All Types Deployed and Lost in Ground -
 - Launch Failures for On-Orbit Support from Ground -

Year	LOW Weight (kg)	
1	0.00 +/-	0.00
2	0.00 +/-	0.00
3	0.00 +/-	0.00
4	0.00 +/-	0.00
5	0.00 +/-	0.00
6	6993.78 +/-	554.68
7	6993.78 +/-	613.54
8	6562.07 +/-	1068.33
9	1036.12 +/-	803.92
10	690.74 +/-	609.32
Total:	22276.50 +/-	688.77

 - STV Consumables of All Types Deployed, Replaced after -
 - Initial Deployment, and Lost in Ground Launch Failures -

Year	Weight (kg)	
1	0.00 +/-	0.00
2	0.00 +/-	0.00
3	0.00 +/-	0.00
4	8775.00 +/-	257.18
5	975.00 +/-	693.65
6	375.00 +/-	386.19
7	300.00 +/-	367.26
8	225.00 +/-	257.18
9	150.00 +/-	313.95
10	75.00 +/-	156.97
Total:	10875.00 +/-	848.40

Bibliography

1. Bain, Rodney & Weisel, William. Class Notes for Astro 6.36, Advanced Astrodynamics. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, Winter 1991.
2. Baker, Robert M. L., Jr. *Astrodynamics: Applications and Advanced Topics*. New York: Academic Press, 1967.
3. Bate, Mueller, & White. *Fundamentals of Astrodynamics*. New York: Dover Publications, 1971.
4. Chandrasekaran, Usha and Sheppard, Sallie. "Implementation and Analysis of Random Variate Generators in Ada," *Journal of Pascal, Ada, & Modula-2*, Vol. 5, No. 4: 27-39 (1986).
5. *Comprehensive Operational Support Evaluation Model for Space (COSEMS) Design Document Version 5.1*, 25 February 1991. Contract F04701-88-D-0028. Los Angeles CA: PRC Inc., February 1991.
6. *Comprehensive Operational Support Evaluation Model for Space (COSEMS) User's Guide Version 5.1*, 25 February 1991. Contract F04701-88-D-0028. Los Angeles CA: PRC Inc., February 1991.
7. COSEMS Source Code. Computer Program. El Segundo CA: PRC Inc., 1991.
8. Draper, R. F. and Gavin, T. R. "Reliability in Long-Life Missions," *AIAA 7th Annual Meeting and Technical Display*. AIAA Paper 70-1247. Houston, Texas (19-22 October 1970).
9. Ely, Neal M. "COSEMS: A Dynamic Simulation Model for Assessing Alternative Maintenance Concepts for the Strategic Defense System," *Proceedings of the Society of Logistics Engineers 24th Annual Symposium*, San Francisco, California, 143-150 (August 1989).
10. Hecht, Herbert, and Hecht, Myron. *Reliability Prediction for Spacecraft*, RADC-TR-85-229. Rome Air Development Center. December 1985.
11. Janz, Ron et al. "COSEMS-A Study Simulation of Space Logistics," *Proceedings of the 2nd AIAA/SOLE Space Logistics Symposium*. AIAA Paper 88-4710. Costa Mesa, California (3-5 October 1988).
12. Janz, Ron and Ely, Neal. "COSEMS: A Dynamic Simulation of Space Logistics," *Proceedings of the AIAA/SOLE 2nd Space Logistics Conference*. Costa Mesa, California (3-5 October 1988).
13. Janz, Ron, et al. "Dynamically Simulating Space Logistics," *Defense Computing*: 34-37 (November/December 1989).
14. Janz, Ron and Mann, Ken. "Application of COSEMS: An Assessment of Constellation Age," *Proceedings of the 3rd AIAA/SOLE Space Logistics Symposium*. Huntsville, Alabama (30 Apr - 2 May 1990).

15. Kapur, K. C., and Lamberson, L. R. *Reliability in Engineering Design*. New York: John Wiley & Sons, 1977.
16. Knuth, Donald E. *The Art of Computer Programming, Volume 2*. Reading: Addison-Wesley Publishing Co., 1971.
17. Law, Averill M. and Kelton, W. David. *Simulation Modeling and Analysis, 2nd*. New York: McGraw-Hill, Inc, 1991.
18. Ling, A. H. K. "Mathematical Modeling," *Reliability and Maintainability of Electronic Systems*, 103–129, edited by J. E. Arsenault and J. A. Roberts. Rockville MD: Computer Science Press, Inc, 1980.
19. Marec, Jean-Pierre. *Optimal Space Trajectories*. Amsterdam: Elsevier Scientific Publishing Company, 1979.
20. Mendenhall, William, et al. *Mathematical Statistics with Applications*. Boston: PWS-Kent Publishing Company, 1990.
21. Morrison, S. C. "Bi-Elliptic Transfer Orbit User's Guide," General Research Company (28 September 1988).
22. Pritchard, Wilbur L. and Sculli, Joseph A. *Satellite Communication: Systems Engineering*. Englewood Cliffs: Prentice-Hall, 1986.
23. Pritsker, A. Alan B. *Introduction to Simulation and SLAM II*. New York: Halsted Press, 1986.
24. Ravindran, Phillips, and Solberg. *Operations Research: Principles and Practice, 2nd Edition*. New York: John Wiley & Sons, 1979.
25. Sargent, Robert G. "An Overview of Verification and Validation of Simulation Models," *Proceedings of the 1987 Winter Simulation Conference*, edited by A. Thesen, H. Grant, and W.D. Kelton. 33–39. Atlanta, Georgia.
26. *Stat/Library*. Houston: IMSL, Inc., 1989.
27. Thomson, William Tyrell. *Introduction to Space Dynamics*. New York: Dover Publications, 1986.
28. Winer, B. J. *Statistical Principles in Experimental Design*. New York: McGraw-Hill Book Co., Inc, 1962.