NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

INTEGRATED SPACECRAFT DESIGN TOOLS

by

Troy W. Pannebecker

December 1999

Thesis Advisor:	Brij N. Agrawal
Co-Advisor:	Herschel H. Loomis, Jr.

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 1999	**3. REPORT TYPE AND DATES COVERED** MS / Engineer's Thesis

4. TITLE AND SUBTITLE Integrated Spacecraft Design Tools	**5. FUNDING NUMBERS**
6. AUTHOR(S) Pannebecker, Troy W.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

The thesis surveys current software tools to design satellites and develops an integrated spreadsheet-based tool for preliminary spacecraft design. First, several existing and future design tools – both commercially available and company proprietary – are discussed and evaluated. Second, a spreadsheet-based design tool which is generally applicable to any earth-orbiting satellite is developed. Preliminary design of all satellite subsystems is performed on separate sheets of the Excel workbook. Based on user-entered orbital data, propellant and mass budgets are also calculated. The design technique and spreadsheet implementation is presented along with the underlying "first principles" theory and equations.

14. SUBJECT TERMS Spacecraft, Satellites, Design Tools, Concurrent Engineering	**15. NUMBER OF PAGES**
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFI- CATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

Approved for public release; distribution is unlimited

INTEGRATED SPACECRAFT DESIGN TOOLS

Troy W. Pannebecker
Major, United States Air Force
B.S., Pennsylvania State University, 1987

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

and

AERONAUTICAL AND ASTRONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1999

Author: ______________________________
Troy W. Pannebecker

Approved by: B. N. Agrawal
Brij N. Agrawal, Thesis Advisor

____
Herschel H. Loomis, Jr, Co-Advisor

____
Gerald H. Lindsey, Chairman
Department of Aeronautics and Astronautics

ABSTRACT

The thesis surveys current software tools to design satellites and develops an integrated spreadsheet-based tool for preliminary spacecraft design. First, several existing and future design tools - both commercially available and company proprietary - are discussed and evaluated. Second, a spreadsheet-based design tool which is generally applicable to any earth-orbiting satellite is developed. Preliminary design of all satellite subsystems is performed on separate sheets of the Excel workbook. Based on user-entered orbital data, propellant and mass budgets are also calculated. The design technique and spreadsheet implementation is presented along with the underlying "first principles" theory and equations.

DISCLAIMER

The views, opinions, and judgments expressed about the integrated design tools evaluated in the industry survey are solely those of the author and do not reflect the official policy or position of the Naval Postgraduate School, the United States Air Force, the Department of Defense, or the United States Government.

The reader is cautioned that the spacecraft integrated preliminary design tool developed in this thesis may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure the spreadsheet calculations are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ACS	Attitude Control Subsystem
ADCS	Attitude Determination and Control
AGI	Analytical Graphics, Inc.
AI&T	Assembly, Integration, and Test
BER	Bit Error Rate
BOL	Beginning of Life
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CalTech	California Institute of Technology
CAM	Computer Aided Manufacturing
CBS	Cost Breakdown Structure
CDC	Concept Development Center
CEO	Chief Executive Officer
Comm	Communications
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off The Shelf
CPT	Communications Payload Team
CPU	Central Processor Unit
CTek	Computational Technologies, Inc
DBMS	Database Management System
Delta V	Change in the Velocity Vector
DMS	Data Management Subsystem
E_b/N_o	Energy-per-bit to Noise-density Ratio
EIRP	Effective Isotropic Radiated Power
EOL	End of Life
EOPT	Electro-Optical Payload Team
EP	Electric Propulsion
EPS	Electrical Power Subsystem
ESDI	Electrical System Design and Integration
EW	East - West
FCC	Federal Communications Commission
FISH	Fast Interface Shell
FltSatCom	Fleet Communications Satellite
FSW	Flight Software

GaAs	Gallium Arsenide
GEO	Geosynchronous Earth Orbit
GSO	Geostationary Earth Orbit
GST	Ground Segment Team
GTO	Geosynchronous Transfer Orbit
GUI	Graphical User Interface
ICDF	Integrated Concept Design Facility
InP	Indium Phosphide
IR	Infrared
ISE	Intelligent Synthesis Environment
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
MAAT	Mission Area Architecture Team
MEO	Medium Earth Orbit
MMH	Monomethyl Hydrazine
MOI	Moment of Inertia
MS®	Microsoft® Corporation
MSC	Macneal-Schwendler Corporation
MSE	Mission and Systems Engineer
N_2O_4	Nitrogen Tetroxide
NASA	National Aeronautics and Space Administration
NiH_2	Nickel Hydrogen
NPS	Naval Postgraduate School
NS	North - South
OLE	Microsoft® Object, Linking, and Embedding utility
OODBMS	Object Oriented Database Management System
OTV	Orbital Transfer Vehicle
PC	Personal Computer
PDC	Preliminary Design Center
P/L	Payload
POC	Point of Contract
PTC	Parametric Technologies Corporation
RCS	Reaction Control System
RF	RadioFrequency
RO	Requirements Object
RPET	Relational Parameter Exchange Tool
RPM	Revolutions Per Minute

SAT	Systems Architecture Team
SCDT	Spacecraft Design Tool
SCDTrades	Spacecraft Design Trades
SDRC	Structural Dynamics Research Corporation
SEAL	Science and Engineering Analysis Library
SEAS	Scientific and Engineering Application System
Si	Silicon
SMAD	Space Mission Analysis and Design
SMS	Structures and Mechanisms Subsystem
SOAP	Satellite Orbital Analysis Program
SOSAT	System of Systems Architecture Team
SOW	Statement of Work
SST	Space Systems Team
STK®	Satellite Tool Kit®
TCS	Thermal Control Subsystem
Team X	Advanced Projects Design Team at JPL
Telecomm	Telecommunications
TQM	Total Quality Management
TRL	NASA Technical Readiness Levels
TT&C	Telemetry Tracking and Control
U.S.	United States
VIS	Virtual Intelligence Simulator

ACKNOWLEDGEMENTS

While one person usually receives all of the credit, any effort of this scope is really the result of a team. I feel compelled to take a moment to identify and thank a few key individuals who significantly impacted my thesis and made my time at NPS so meaningful.

First and foremost, I want to thank two of the most remarkable leaders I've had the good fortune to meet over my Air Force career, Col. John Cunningham, USAF, and CAPT Tom McKannon, USN. Thank you for having the faith, confidence, and trust in me to select me for this exceptional opportunity, and for your continuing advice and counsel on life and my career.

Many companies and organizations provided me with information on their satellite design tools and on their concurrent engineering process. In particular, I want to identify a few key individuals: Andrew Dowdy and Dr. David Bearden at Aerospace Corporation; Joel Sercel, Bob Oberto, Steve Wall, and Jeff Smith at the Jet Propulsion Laboratory; Julie Heim at TRW; John Collins at Microcosm; Dr. Gary Stanley at Computational Technologies, Inc.; and Ron Mader and Mike Brown at the Naval Research Laboratory. Thank you for taking time out of your busy schedules to share your knowledge, insights, and experience with me.

Dr. Brij Agrawal and Dr. Hersh Loomis were directly responsible for this thesis, its quality, and in fact its very completion. They gave me complete freedom to pursue my studies but were always there to provide that crucial nudge in the right direction when I hit an obstacle or veered off track. Thank you for your patience, support, and guidance.

I also want to thank the NPS staff and faculty. Particularly, I want to thank Dr. Oscar Biblarz, LtCol Mike Mullady, Professor Craig Baldwin, and others in the Aeronautics and Astronautics Department and the Space Systems Academic Group for creating such an exceptional and unique educational environment. To Sandra Days and the rest of the thesis processing staff, I appreciate your open, friendly, and prompt support. At a very busy time, when most students just want to turn in their theses and move on, your positive, uplifting attitude and cooperation were very refreshing and encouraging.

I particularly want to recognize the efforts of CMR Mike McMaster and Maj Butch Jansen. It is clear they truly care about their students and their students' success. Their support, commitment, and personal involvement are overwhelming. Words cannot adequately express the deep sense of debt and gratitude I feel, so I will simply say thank you for all you have done.

Finally, and most significantly, I need to thank my family. My parents instilled in me solid Christian values, a positive work ethic, and respect for myself and others. They molded me into who I am and provide a constant foundation of support, encouragement, and acceptance. To my wife, Stacey, and my children, Courtney and Laura, I also want to say thank you and I'm sorry. Thank you for your patience, tolerance, and understanding. I'm sorry for all the nights apart and weekends I've spent locked away studying. I'm sorry for the times when I was so tired or so stressed out that I was impatient and irritable. You are the most important things in my life.

And to Stacey, what can I say? She was my editor, proofreader, and counselor all rolled into one. She has earned this degree as much as I have, and probably learned as much. You have been by my side every step of the way; I couldn't have done it without you. As much as my parents shaped my childhood, you made me who I am today. You once asked me how I can stay with you, but I ask, "How can I live without you?" You are the best thing that ever happened to me, and I love you.

I. INTRODUCTION

A. BACKGROUND

Since the first U.S. satellite, Explorer I, was launched in January 1958, the ensuing decades saw spacecraft evolve into large, massive, and highly complex systems. As the use of space lost its novelty and became interwoven into everyday life, the requirements placed on spacecraft exploded. U.S. aerospace companies relied on large, sophisticated, and therefore expensive, platforms to meet the increasingly demanding mission requirements. In the big budget era of the 1960's, 70's, and 80's, spacecraft design was driven almost exclusively by performance, with cost and schedule a secondary consideration.

The high degree of system complexity led to specialization and a sequential design approach. Designers limited their expertise to a particular field or an individual subsystem. Subsystems were developed by separate teams, with little communication between groups. Each functional specialty tried to optimize their subsystem, often leading to suboptimization of the spacecraft. Requirements and interfaces between subsystems were captured in a detailed set of documents, where were parsed into increasingly lower levels. As satellites became ever larger and more complex, the documentation set became unwieldy to modify, inhibiting creativity. Because of the sequential process, one designer could not start until the results from another engineer were complete. Trade studies were limited since the first design took so long there was little time left for iteration. Development timelines stretched out and costs soared. Major satellite programs could cost hundreds of millions, or even billions, of dollars. Development schedules of 10 to 15 years were common, with technology frozen early in the design process, ensuring obsolescence even before launch.

To resolve some of these problems, many companies turned to a centralized design process. Instead of sequentially passing a design though the various functional teams, a system engineer gathered contributing information from each group at the same time. New computerized tools were developed to facilitate the accumulation, storage, and manipulation of the design data. The different subsystem designs could now be linked together, improving configuration control and allowing trade studies to be performed quickly. While the centralized approach was an improvement and cut design time from years to months, there were still many problems. With the subsystem experts removed from the integration process, there was an inherent danger of misusing the new tools. The subsystem designer's sense of ownership was lost, breeding fear and mistrust of the system engineer. The designers were concerned that the system engineer only wanted their models and rules-of-thumb, and, once provided, they would be excluded from the rest of the design process. The new design tools were still limited to simple models and algorithms, so much of the design detail was lost compared to a sequential process. (Aguilar, 1998, p. 778-779)

With the end of the cold war, the 1990's saw a dramatic decrease in the government's investment in science and technology. Bloated budgets and open-ended schedules were replaced with tight funding limits and strict schedule constraints. The design focus shifted from strictly performance-driven to a combination of cost, schedule, and performance benefit. For the first time, cost and schedule were not only considered in the design process, but were coequal with performance. The trade space was expanded to include all three factors to ensure the "best value" was achieved, and some programs adopted a design-to-cost philosophy. These changes demanded innovative design, integration, and test methods to improve system performance while lowering the cost and shortening the development time. As a result, the design approach moved from an isolated subsystem view to a system-level orientation and from a sequential to a parallel process. This was the evolution of the concurrent engineering process.

The 1990's witnessed an even more dramatic transformation in the commercial space industry. Until then, space was dominated by the government and military. The major growth in demand for communications and other civilian satellites created a fiercely competitive commercial market as the government contractors expanded their civilian divisions. The intense competition forced companies to contain costs as they rushed to fill the ever-increasing spacecraft orders. Since the buyers were now private companies, they demanded a return on their investment as soon as possible, so schedules were very challenging and tightly controlled. From a design perspective, being the first to bring new satellites and technologies to market reaped tremendous benefits of added orders and higher profits. To be competitive, companies simply could not take 10 years and invest billions of dollars to design a new satellite. The new commercial aerospace companies shortened the development cycle with parallel design at the system level, embracing the concurrent engineering philosophy.

To implement the concurrent engineering process, design centers are turning to a new class of development techniques and computerized tools. The Jet Propulsion Laboratory (JPL) defines concurrent engineering as, “a design methodology where design takes place in a common environment containing all the principal designers, all the design tools and other material necessary for the design to proceed” (Wall, 1996, p. 5). This definition highlights the three key aspects for a successful approach: a dedicated design team, a facility where the team can interact, and a set of collaborative processes and design tools that facilitate rapid design. Most experts emphasize the importance of the last aspect, integrated system models for conceptual design and cost estimation.

B. SCOPE

The efforts of a handful of companies into concurrent engineering have caught on throughout the space industry. Most aerospace companies and government space centers now have efforts to implement a concurrent engineering process and develop the associated computer-aided tools. Since this promises to be the design approach for the foreseeable future, it is imperative that engineering students, especially those in the aerospace field, understand the concurrent engineering process and gain an exposure to the design tools. Toward that end, this thesis explores integrated design tools for conceptual and preliminary design of spacecraft.

The first part of the thesis conducts a survey into current and future commercial and government integrated design tools. While there are a wealth of subsystem tools, there are only a few system-level integrated tools. The individual subsystem tools are not addressed in this thesis. First, there are far too many, even for some subsystems, to be adequately presented in one thesis. Second, the focus of this effort is on system-level, integrated tools. Seven design tools were personally observed by the author and are presented. In addition to a brief description, their advantages and disadvantages are discussed, along with an evaluation of their applicability to the curriculum at the Naval Postgraduate School. For reference, the main points of contact in each company are listed in Appendix A.

The second part of the thesis develops a spreadsheet-based integrated design tool. The design tool enables a user to quickly and accurately determine the key parameters of a satellite design, and is generally applicable to any Earth-orbiting spacecraft. Each subsystem is modeled on a separate sheet, or page. The individual sheets are fully linked, so design trades can be performed by changing a value and observing the effects as they ripple through the design. Chapter III serves as a users guide and acts as a brief tutorial on key aspects of spacecraft design. A printout of each sheet is included in Appendix B, and Appendix C contains a 3.5" diskette with an Excel file of the complete design tool.

II. DESIGN TOOL SURVEY

To reduce the overall cost of satellite programs and shorten the development schedule, many aerospace centers are adopting a concurrent engineering philosophy. A parallel approach to design requires a new methodology which links requirements and ties together the subsystems and their individual tools. Many applications focus on the conceptual and preliminary design phases, which offer the potential for the greatest payoffs. Up to 70% of the life cycle costs of the spacecraft program are determined by decisions made during conceptual design (Aerospace, 1994, p. 1). To support analysis, design, and trade studies over the entire satellite life cycle – mission, spacecraft, and operations – concurrent engineering needs a design process focusing on enabling system-level development using computer-aided engineering techniques. This, in turn, requires a new class of design tools which allow compression of the design phase, integrate the individual subsystems, and enable technical trades during the conceptual design phase.

The goals for developing the new design tools are many and varied. Obviously, a major intent is to decrease the overall spacecraft system cost and shorten the development time. However, this should be achieved while improving spacecraft performance, reliability, and manufacturability. In addition, they should facilitate collaborative development efforts between subsystems and must perform technical trades at the conceptual design phase where the impact is greatest. Finally, because of the rapid pace of technological advances, they should be easily upgraded and allow the addition of new technologies and approaches into the tool.

To gain an understanding of the types and characteristics of integrated design tools, several aerospace design centers were surveyed to see what tools were currently in use, or will be in the near future. A brief description of seven design tools, which were personally observed by the author, are presented, along with a discussion of some of their advantages and disadvantages. Most were developed in-house for internal use and are considered proprietary, but two are commercially available. Since the intent of the survey was to identify potential candidates for incorporation into the space systems and astronautical engineering curricula at the Naval Postgraduate School (NPS), the applicability or adaptability to an academic environment is also evaluated.

The different tools were evaluated against a long list of criteria. Obviously, the model's power, flexibility, and adaptability are important. How easy is it to use, what is the fidelity of the model, and how accurate are the results? Can an initial conceptual design be evolved to lower, more detailed, levels within the same tool? Another important characteristic is the ability to include factors beyond performance and technical design. Does the tool include estimates of cost, schedule, and risk? For concurrent engineering, a good model must also integrate the different aspects of the design process. Does the model allow and even facilitate trade studies and optimization? Are systems engineering functions automatically applied to maintain configuration control and design consistency across subsystem interfaces? Given the pace of technological advances, to remain viable design tools must be upgradable. How easily can new capabilities and features be added? Does the tool provide the ability to add and evaluate new satellite technologies and manufacturing techniques? Finally, bringing the model to NPS and applying it to an academic curriculum will require support from the source organization. Is technical support available, and how cooperative and responsive is the originating developer of the design tool?

The views, opinions, and judgments expressed in the evaluation of the surveyed design tools are solely those of the author. The statements do not reflect the official policy or position of the Naval Postgraduate School, the United States Air Force, the Department of Defense, or the United States Government.

The survey revealed there are essentially two types of tools: those based on a spreadsheet application such as Microsoft® Excel and those which are a stand-alone software program. Spreadsheet-based tools are used in the Concept Development Center (CDC) at the Aerospace Corporation, in the Preliminary Design Center (PDC) at JPL, and in the Integrated Concept Design Facility (ICDF) at TRW. The California Institute of Technology (CalTech), under the direction of Professor Joel Sercel, is developing several design tools which use spreadsheets and other software applications. Lockheed Martin Corporation developed a stand-alone program called the Virtual Intelligence Simulator (VIS). In addition to these proprietary tools, there were two which are commercially available. Microcosm markets the Space Mission Analysis and Design (SMAD) software program, based on the Larson and Wertz (1992) book of the same name. The final tool is GENSAT, developed by Computational Technologies (CTek).

A. SPREADSHEET-BASED TOOLS

Spreadsheets offer many inherent benefits when used as design tools. First of all, modern spreadsheet applications are very powerful and extremely flexible. They can be easily customized for each specific spacecraft design. New formulas, satellite components, and design characteristics can be added in advance based on the initial requirements or "on-the-fly" during the actual design session. For trade studies, multiple cases can easily and rapidly be executed on the same sheet, providing a direct comparison of the results. They are readily available and easy to use. Most people, and especially those with a technical background, are already familiar with how to enter and manipulate data in a spreadsheet. Results can be displayed in tabular form or graphically. But perhaps their most important attribute is connectivity. Not only can sheets within a workbook be linked, but separate workbooks for individual subsystems can also be linked together.

It is important to remember, however, that spreadsheet-based design tools are just that – tools. They assist the engineers in the design process but cannot replace their expertise or creativity. Spreadsheets simply automate tedious calculations and provide templates so important aspects are not overlooked. Because they can be linked, spreadsheets also facilitate and manage the sharing of information between subsystem experts, which is critically important in a concurrent engineering process.

With these important features and limitations in mind, several spreadsheet-based design tools are described below. The Aerospace Corporation and JPL have long-standing programs and are widely recognized as leaders in the field. Their design facilities have been studied and copied by many other organizations. In fact, TRW's ICDF is very similar to these two tools, although it does have some significant differences. As part of their undergraduate design program, CalTech is developing several different design tools which use spreadsheets and other common software applications.

1. Aerospace Corporation Concept Design Center

The Aerospace CDC is a large-scale, distributed spreadsheet-based system to effectively organize cross-discipline conceptual design and analysis capabilities. It provides an interactive real-time environment allowing customers to work more closely with engineering experts. The individual efforts of the subsystem engineers are coordinated by a system engineer to capture the system's internal and external interfaces and represent the life cycle of the entire system. Each subsystem station is linked to the system engineer and to the other subsystems. In addition to technical development, the analysis includes estimates of the cost, schedule, and risk from very early in the design process.

In their CDC user's guide, Aerospace defines the CDC as composed of three key elements.

> The CDC is 1) a team that draws on the breadth of The Aerospace Corporation's engineering expertise; 2) a facility where the Program Office and customer can interact efficiently with the team of expert [sic], and 3) a process for applying innovative design tools to produce quality results quickly. These three elements work together to yield an engineering analysis product with greater detail and consistency, and that is produced in a much shorter amount of time and at less cost than traditional systems engineering studies. (Aerospace, 1998, p. 1)

The Aerospace definition emphasizes the three key aspects for a successful systems engineering approach. In addition to the subsystem designers, the CDC brings together other engineering experts not normally included in the conceptual design phase, such as cost, ground systems, and software. (Aguilar, 1998, p. 779)

The CDC team is an ad-hoc group which meets on an as-needed basis, and includes members from across all of the functional departments in Aerospace. Team members represent the full expertise and capability of their respective departments. They develop and maintain the actual spreadsheet models and databases, keeping "ownership" in the hands of the functional experts. They are also responsible to select and train additional team members as necessary to ensure broad level support. Currently, there are only two or three representatives per subsystem or station, but Aerospace is working to get broad-based exposure among the functional experts to develop department-level support.

There are now a total of five teams, discussed further below, which address different stages of a spacecraft life cycle. The team directly responsible for the actual designing of spacecraft is the Space Systems Team (SST). The SST consists of the following functional areas:

- Propulsion
- Attitude Determination and Control
- Communications
- Command and Data Handling

- Electrical Power
- Thermal
- Structures
- Cost/Risk
- Ground Segment
- Payload Processing
- Astrodynamics
- Software
- Radar Payloads
- Electro-optical Payloads
- Systems Engineering

The systems engineer coordinates the entire effort, organizing the study in cooperation with the customer and planning and conducting group activities. (Aguilar, 1998, p. 779)

The CDC facility provides a central meeting location with all of the resources needed to carry out the design process. The entire team, program office representatives, and customer are all seated around one table. This face-to-face contact facilitates a free and open exchange of ideas and information. The individual subsystems are arranged so those with the most frequent interaction are located next to each other. Overhead projectors can display the output from any computer terminal on a screen to focus the group's attention on a specific issue. A schematic of the Aerospace CDC is shown in Figure 2.1.

The CDC process enables the subsystem experts to prepare their contributions simultaneously and in the presence of the other team members and the customer. To facilitate team interaction, the CDC uses personal computers and spreadsheet software to manage the sharing of information, which supports sound configuration control and helps ensure continuity across the entire design.

- 15 Pentium Workstations
- Dedicated NT Server
- 2 Video Projectors

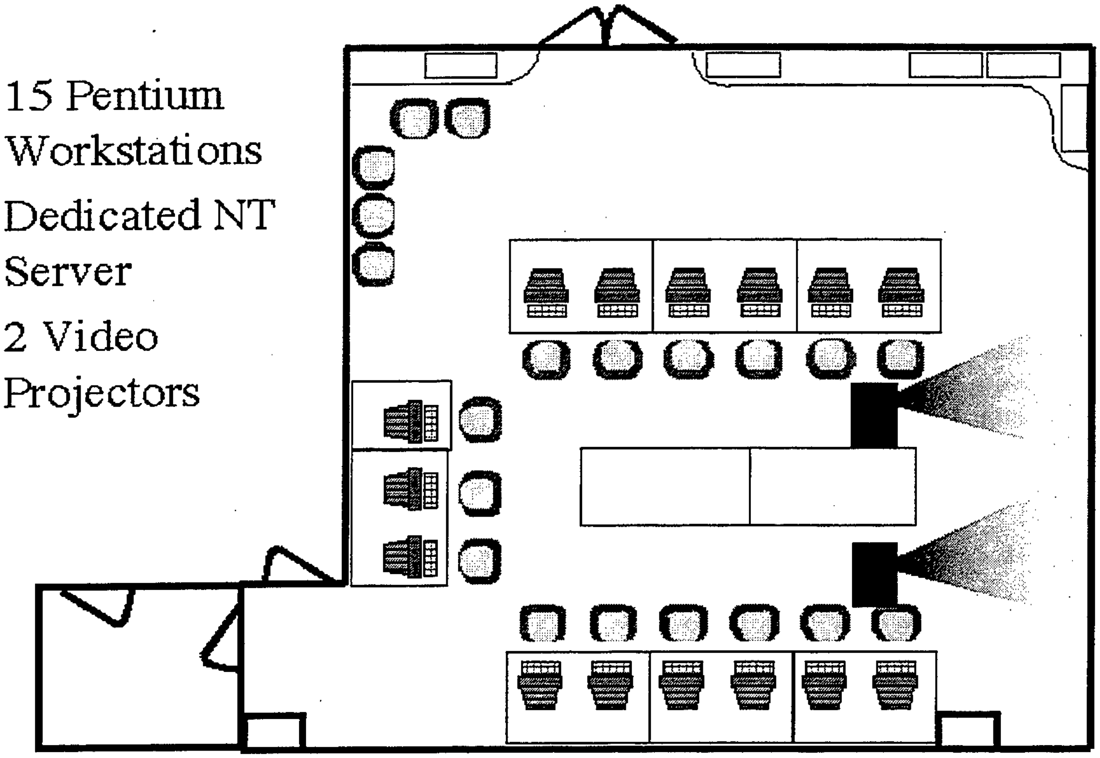


Figure 2.1 CDC Facility Layout (Aerospace, 1998, p. 9)

A typical CDC design study consists of three phases. First is advanced planning. The customer meets with the systems engineer and other necessary team members to determine the scope of the effort and generate a statement of work (SOW). Based on the requirements of the SOW, team members can then adjust the existing subsystem models or develop new ones. The second phase is the design sessions, where the entire design team is assembled, along with the customer, to created the spacecraft design. A typical design takes 2 to 3 sessions of 3 hours each, but as many follow-up sessions can be scheduled as needed. The final phase is the post-session wrap-up. The results of the study are documented and each subsystem model is archived. With the archived models, future studies, which further refine the original study or explore a similar design, can pick up where the first one stopped. Each subsystem designer writes a report documenting the design and discussing key aspects and assumptions. The report also includes any important information not captured in the subsystem models

. At the heart of the CDC process is a set of distributed subsystem models hosted on Microsoft® Excel 5.0 spreadsheet software. Excel provides the capability to link information between spreadsheets, so each individual model can be built separately. Each model contains a minimum of three spreadsheets. The first sheet includes the necessary equations, heuristics, and algorithms to calculate the design parameters. Second is a database of components used in the particular subsystem. The last sheet, and the one that makes the whole process work, is a pre-formatted page to exchange information between subsystem models. Besides these three sheets, the functional experts may establish additional sheets as necessary for their subsystem.

All of the individual subsystem models are linked though a network server, and data is exchanged using the Microsoft® Object, Linking, and Embedding (OLE) utility. Links are created so the output of one model is an input to one or more of the other models. Thus, the configuration is dynamically updated as changes made by one subsystem engineer ripple though the design. All of the models link to the systems engineer, who accumulates the key design features and ensures compatibility across the subsystem interfaces. To simplify the interconnectivity, the input/output parameters are formalized. Each item entering or leaving a model is documented with a name, value, and comment describing where the value is linked. Color codes were developed for defining different variable types and to assist in controlling parameters. (Aerospace, 1994, p. 9)

The subsystems are designed using several different methods. First, and most obviously, calculations are based on "first-principles" analytical relationships. Another common approach applies heuristics and scaling ratios based on historical approximations, estimating the new design parameters from past spacecraft properties. To ensure the design is realistic, many subsystems, especially attitude determination and control (ADCS), communications, and telemetry tracking and control (TT&C), select components from a parts database. For example, propulsion can select existing tanks and thrusters, or they can be sized by analytical or historical relationships. However, since the CDC typically designs at the conceptual level, care must be taken to prevent locking into specific parts or vendors. Design experts can perform detailed off-line simulations using an individual subsystem tool, then feed the results back into the spreadsheet model. With the multi-tasking capabilities of current PC operating systems, this can frequently be done in real-time on the same computer, during the design session (Aguilar, 1998, p. 782). Cost estimates use a "top down approach," establishing parametric relationships from historical cost trends to relate spacecraft costs to physical, technical, and performance parameters

(Bearden, 1998, p. 29-30). Estimates of risk rely on subjective assessments by the subsystem experts, using the (TRL).

With the complexity and sophistication of the CDC design tool, it is important to remember that it is only a tool. The key to the entire approach is still the active involvement of the engineering expert. Aerospace likens the spreadsheet tools to a musical instrument.

> The musician is the one who is determining what notes, rhythm, and tempo are being sounded. The instrument is just the mechanism that the musician uses to present his music. In a similar fashion, the spreadsheet-based design tools facilitate the design process by allowing the design specialists to contribute their experience, expertise, and creativity in a consistent and flexible manner. In fact, the CDC process can be compared to an orchestra where the systems engineer conducts all of the engineers who relate their talents through the spreadsheets. (Aguilar, 1998, p. 781)

The spreadsheets provide the means to capture the results of an inherently creative process.

The CDC has been highly effective for Aerospace. Studies that historically took several months have been cut to weeks, while still producing the same level of detail. Resource expenditures have dropped from 50% to 75%, even after factoring in the costs of establishing and maintaining the CDC. Team members also benefit from their participation in the CDC process. Since they represent their respective functional organizations, they must be knowledgeable on all aspects of their discipline and stay current on new technology. In addition, and perhaps more importantly, involvement in concurrent engineering allows the subsystem engineers to gain an enhanced system-level awareness. (Aguilar, 1998, p. 782)

The CDC has been so effective, Aerospace is expanding the number of teams. The teams form a hierarchical set addressing different aspects of the spacecraft mission design process. Results from one team can be passed to another team, allowing the mission design to be successively refined. Conversely, each team can operate independently by abstracting the lower level details as appropriate. Despite focusing on different levels of the spacecraft mission, each team has approximately the same number of members. All of the teams share the same facility, but their specific spreadsheet models are different. The SST was the first team formed and is the one discussed thus far. Because of the success of the SST, Aerospace formed four other teams. The Systems Architecture Team (SAT) focuses on the system architecture for individual space systems, and performs trades on the constellation, payload performance, and concept of operations. The Ground Segment Team (GST) develops architectures for command and control, data processing and dissemination, software, facilities, personnel, and communications. The Electro-Optical Payload Team (EOPT) creates conceptual designs of electro-optical payloads. Finally, the Launch Vehicle Team designs and evaluates new expendable and reusable launch systems. Aerospace is considering even more teams, including a System of Systems Architecture Team (SOSAT), a Mission Area Architecture Team (MAAT), and another payload team, the Communications Payload Team (CPT).

Clearly, the CDC is a very impressive tool which provides many benefits to the design engineers, however, it does have a few limitations. Since it is a spreadsheet-based tool, it has all of the advantages discussed on page 7 above. It is a powerful and flexible tool, is easy to use, and gives accurate results for a conceptual design. The process addresses not only technical design, but includes cost, schedule, and risk from the earliest stages. However, basing the cost estimates on historical relationships assumes these trends will apply to future costs, which given new technologies and techniques is not necessarily a reasonable assumption.

New capabilities can be readily added, and in fact the subsystem models are normally customized for each specific study. New technologies are easily added and evaluated by simply updating the component databases, analytical relationships, or historical ratios. Keeping "ownership" in the hands of the design experts increases the likelihood the subsystem models will be well developed and maintained. However, it also leads to a lack of consistency in model fidelity. Some of the engineers create extremely detailed models, while other areas are much less vigorous.

The distributed spreadsheets are almost ideally suited to perform trade studies. Linking the individual sheets carries changes in one subsystem model throughout the entire system design. However, the spreadsheets do not perform optimization. Because the sheets are fully linked, they do accommodate "manual" optimization, where the engineers enter successive values to improve the design. Another drawback is the lack of automated systems engineering, which would prevent incompatibilities across interfaces as the design changes. While the subsystem parameters are automatically transferred to other models and some values will recompute, the subsystem engineer must manually ensure changes are consistently adopted throughout all of the subsystem designs.

Finally, Aerospace would probably provide reasonable technical support and assistance if the CDC design tool was incorporated into the NPS curriculum. There are good contacts at the working level, and the engineers are friendly, open, and responsive. However, the Aerospace upper management has expressed some hesitation over sharing the software models. They have had some problems with sharing software in the past, though not with NPS. In addition, the component databases, which ground a design in reality, are considered proprietary and would definitely not be shared even if the spreadsheet models are.

The CDC distributed spreadsheets are not a perfect match to the NPS curriculum, but they could be adapted to enhance the learning process. The best fit is to the group spacecraft design class, although the CDC process differs from the current instructional approach. As noted before, the CDC design tool relies on the engineer's experience and expertise, which the students normally lack. In fact, the point of the class is to provide exposure to the design process. Instead of accomplishing the design in a one week session, the student's efforts are guided by the professor over the entire eleven week term. However, the "CDC session" could be held toward the end of the class, pulling together the knowledge and information gain over the quarter.

Maintaining the models could also be a problem. The actual subsystem models, the analytical relationships and historical ratios, could be updated by the students when they adapt them for their specific design. However, the component databases present a greater challenge. First of all, since the databases would not be provided with the spreadsheet models, they would have to be created. Although the databases are not required to successfully use the subsystem models, they definitely add to the quality of the design. In addition, the design class usually requires selection of specific components. Even after the databases are created, it is unlikely they could be reasonably and accurately maintained by handing the responsibility off from class to class. This responsibility should be assigned to a faculty or staff member. However, the learning process would be diminished if the students could simply select components from the database. A large part of the learning occurs from taking with vendors and manufacturers to find out what parts are commercially available and under development. Again, this problem could be overcome if the databases were withheld until later in the quarter. In addition, information obtained by the students during the industry survey could then be given to the staff maintainer to keep the database current.

2. Jet Propulsion Laboratory Preliminary Design Center

In 1994, JPL reworked the way they develop new space mission concepts. JPL had three goals for the new process: 1) improve the quality of new mission studies with concurrent engineering, 2) develop "generalists" from promising engineers, and 3) create a reusable study process with trained personnel, facilities, procedures, and software and hardware tools (Wall, 1996, p. 2). This led to the creation of the PDC.

The PDC and the associated design process is virtually identical to the CDC at Aerospace. In fact, JPL contracted with the Aerospace Corporation to develop the concurrent engineering methodology. Mimicking their own CDC, Aerospace created the structure of the distributed spreadsheets, developed the information backbone linking the individual models, and helped establish the relationships between the subsystems – what data must be exchanged between which subsystems, the so called N^2 relationships. Aerospace also supported the installation, test, and maintenance of the distributed network and helped JPL develop the process for conducting concurrent design sessions using the distributed spreadsheet framework. However, just as at Aerospace, the functional engineers created the actual subsystem models, giving "ownership" to the JPL experts.

There are only a few minor differences between the PDC and CDC. First, data is exchanged using the MacIntosh Operating System's Publish and Subscribe utility instead of the MS® OLE. Second, JPL's breakdown of functional areas is slightly different. The PDC subsystems are illustrated in Figure 2.2. Lastly, JPL uses a dedicated design team, called the Advanced Projects Design Team but commonly referred to as Team X. Unlike the ad-hoc team at Aerospace, which meets on an as needed bases, Team X exists year round as a pre-assembled team. Use of a standing team allows the members to get to know each other and to learn how to work well together. To prevent the entire team membership from changing at once, members serve staggered, one-year terms.

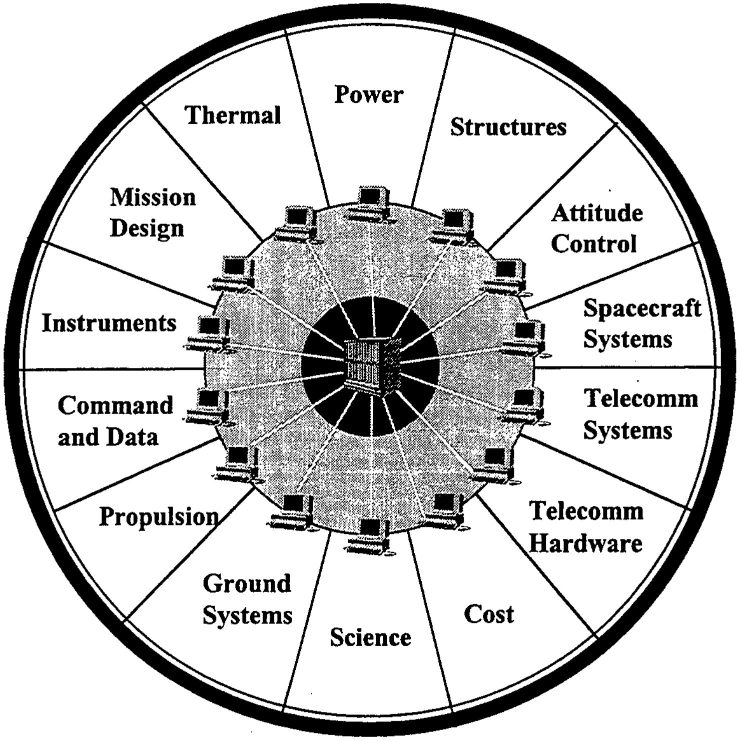


Figure 2.2 PDC Subsystem Breakout

The PDC has proven to be as effective for JPL as the CDC is for Aerospace. JPL estimates the design time was cut by a factor of 10 and costs were cut in half. This was accomplished without decreasing the quality of the design, and in many cases the design is better than before. The PDC design estimates are usually within 30% of the actual mass, power, and cost for programs which continue through production, launch, and operations. The success of the PDC has earned the attention of the National Aeronautics and Space Administration (NASA). Team X was one of only nine JPL process improvement teams to receive the 1997 Total Quality Management (TQM) Redesign Award Medallion. Team X was also presented a NASA Group Achievement Award, one of the highest classes of awards available to JPL employees.

Because they are nearly identical, the PDC shares the same advantages and disadvantages with the CDC. The applicability of the PDC to the NPS curriculum is also the same, with one significant exception in the level of support from JPL. In contrast to Aerospace, which was very open and cooperative, JPL engineers and management was practically non-responsive to all but the simplest inquires. While JPL did provide copies of a few articles and allowed this author to observe a PDC design session, all other discussion and interaction was extremely difficult. Therefore, if NPS decides to obtain a distributed, spreadsheet-based design tool, it should be through the Aerospace Corporation.

Team X and the PDC are only a small part of NASA's efforts to reinvent their internal design process. Called the (ISE), Dan Goldin, the NASA administrator, gives the following description.

> One of the major objectives if ISE is to significantly enhance the rapid creation of innovative affordable products and missions. ISE uses a synergistic combination of leading-edge technologies, including high performance computing, high capacity communications and networking, human-centered computing, knowledge-based engineering, computational intelligence, virtual product development, and product information management. The environment will link scientists, design teams, manufacturers, suppliers, and consultants who participate in the mission synthesis as well as in the creation and operation of the aerospace system. (Goldin, 1998, p. 1)

NASA is developing ISE for their own use at their various design centers, prime contractors, and major vendors. Taking a step toward realizing the goals of linking engineers and design centers, JPL, Aerospace, and TRW recently joined together to design an Earth-orbiting spacecraft. The goal of the demonstration was to show dispersed sites can be linked to accomplish meaningful work.

The three-site collaboration was very communications intensive. The separate design teams interacted through two video-teleconferencing channels. The three local networks were linked and used Microsoft® NetMeeting software, which allowed any site to remotely update a spreadsheet or other file at any other site. To further facilitate discussion, six "meet-me" conferencing lines were established for side discussions. These lines were set up so that any number of people could call in from any site and be linked together. Two lines were always in use: one by the team leader and one by the facilities coordinator. Finally, two sets of individual computers at each site were networked, allowing side work on supporting sheets, graphs, or other documents. Despite some minor glitches in establishing the communications links, the collaborative effort was very successful. The conceptual design was completed in the normal one-week series of sessions. Perhaps more importantly, valuable experience was gained in the logistics of connecting dispersed sites and in conducting joint design sessions with outside organizations.

3. TRW Integrated Concept Development Facility

The TRW ICDF is similar to the CDC and PDC, however, there are important differences which reflect the difference in application and usage. As a prime contractor, TRW needs more detailed information in terms of the technical design and costs. Results from the ICDF study will form the basis of a competitive proposal, which TRW could become contractually required to produce. Therefore, their spacecraft design and cost estimates must go beyond conceptual into preliminary design and focus more on current designs instead of future concepts. Unlike Aerospace and JPL, who do not want to restrict themselves to current technologies, TRW needs to select specific components and vendors. The ICDF tools provide higher fidelity, are more automated, and are much more dependent on well-defined component databases than either the CDC or PDC.

TRW defines their ICDF as composed of four elements: the environment, the process, databases, and tools. The environment provides an in-place, trained, and co-located "core team" working in a real-time design environment, enabling more iterations and producing high-quality end-to-end solutions. Standard processes ensure discipline, visibility, and ownership by the functional experts and ensures balanced solutions. Databases contain consolidated, validated, constantly-updated, and controlled information on the spacecraft components which are critical to accurate, competitive design solutions. The automated tools and validated, integrated engineering models provide high fidelity concepts and allow an expanded trade space and "what if" exercises, which mitigates risk. TRW's definition, while different in form, does not really differ from the Aerospace definition of the CDC. The ICDF environment encompasses the team and facility elements of the CDC definition. However, TRW expands the CDC process element into three distinct items, emphasizing their greater reliance on component databases and more formalized design tools. (TRW, 1998, p. 5)

The heart of the ICDF process is still a set of distributed subsystem models hosted on MS® Excel. As would be expected due to differences in corporate cultures, the breakout of subsystems is different than at either Aerospace or JPL. The ICDF models are allocated to the following subsystem stations:

- Structures and Mechanisms Subsystem (SMS)
- Thermal Control Subsystem (TCS)
- Attitude Control Subsystem (ACS)
- Propulsion
- Launch Vehicles / Mission
- Assembly, Integration, and Test (AI&T)
- Electrical Power Subsystem (EPS)
- Electrical System Design and Integration (ESDI)
- Data Management Subsystem / Telemetry, Tracking, and Control (DMS / TT&C)
- Flight Software (FSW)
- Mission and Systems Engineer / Payload (MSE / P/L)
- Cost
- Administration / Toolmeister

A schematic of the TRW ICDF is shown in Figure 2.3.

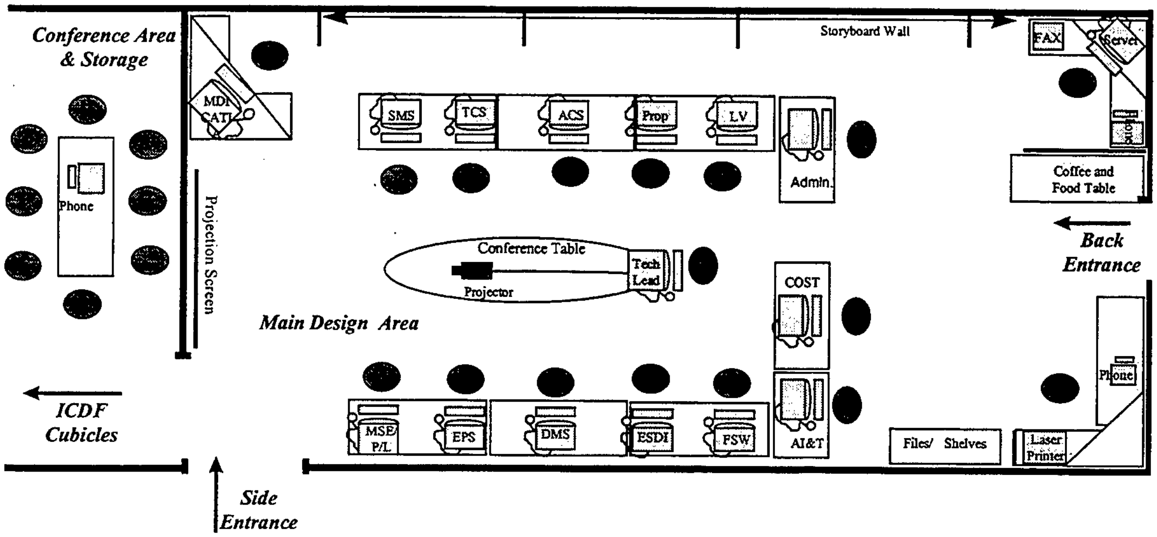


Figure 2.3 ICDF Facility Layout (TRW, 1998, p. 10)

While the subsystem models – the analytical relationships, historical ratios, and configuration summary – are created in an Excel spreadsheet, the component databases are not. Because the databases are more extensive and involved, they were removed from Excel, which has reasonable but limited databasing capabilities, and instead are hosted in MS® Access, a relational database software application. Using Access allowed the creation of an ICDF utility to automatically selected the proper components based on the current design configuration. The Excel models and Access databases are linked together, which is facilitated by the use of a common software suite, MS® Office. TRW also uses Access to perform the data exchange between the subsystems, allowing the transfer to be more automated and formalized than Excel can achieve. In essence, Excel provides the "front end" user interface to the engineers and Access performs the real work of tracking the design iterations and integrating the subsystem models.

TRW views the ICDF design tool as four separate but integrated components. First, the Data Transfer Tool facilitates and controls the electronic transfer of information between subsystems. A subsystem engineer can share updated design data by clicking a simple send/receive data button. To ensure everyone is working on the same iteration, all data is time stamped and new data is highlighted for quick reference. Second, a Standard Components Database consolidates up-to-date technical performance, cost, schedule, and heritage data on all hardware commonly used on TRW spacecraft. Next, the Standard Component Selection Tool interfaces the Data Transfer Tool and the Systems Engineering Tool to find the current spacecraft configuration, then automatically selects the appropriate hardware component to meet the design requirements. Finally, the Systems Engineering Tool gathers the subsystem component selections and mass, power, and cost budgets, and summarizes the bus, payload, and launch vehicle data for verification and feasibility assessment. As a further safeguard to keep the entire team synchronized on the same design iteration, the latest subsystem update times are highlighted. In addition to these four components, the ICDF includes a reference library of spacecraft and launch vehicle data. (TRW, 1998, p. 18-21)

Despite the more automated tools, the success of the ICDF still depends on the subsystem design experts. They create the individual subsystem models in Excel, keeping ownership in the hands of the functional experts, and maintain their database of components. The design expert must verify the automated calculations and ensure the selected components are indeed the best choice for the specific design requirements. Despite the automation, the engineers still have great flexibility. They can override the results of any automatic operation, and can tailor the spacecraft design by updating the models and/or databases during the session. In short, the experience, expertise, and creativity of the human engineer is still the key factor.

The ICDF has been highly effective for TRW. The demonstrated cost savings for preparing a design estimate or proposal are 30% to 40% (TRW, 1998, p. 6). At the same time, design definition has improved while the development schedule was significantly reduced.

Since the differences are primarily in implementation, the ICDF shares most of the same advantages and disadvantages with the CDC, as discussed on page 13. However, there are a few notable exceptions. TRW has improved on the cost estimation technique. Cost data is extracted from the expanded databases along with the component, and are based on current vendor prices. For the in-house assembly, integration, and test costs, the ICDF is linked with the TRW accounting system and uses the same work breakdown structure cost models and cost estimating relationships as the rest of TRW. This ensure the cost estimates are based on the most current direct labor and overhead rates. While the models are just as susceptible to variations in fidelity, in practice TRW has achieved a higher degree of consistency than either the CDC or PDC. Support and cooperation was also very good. Unfortunately, TRW considered the entire ICDF system highly proprietary and will not share it with NPS.

4. CalTech Design Tools

CalTech is developing four different design tools based on spreadsheets and other software applications. The tools are being developed by undergraduate juniors and seniors under the direction and guidance of Professor Joel Sercel, an engineer at JPL and visiting professor at CalTech.

Professor Sercel believes strongly that design success is not a function of the tools, but rather depends on the people. To design a good spacecraft, the engineers must understand the relationships between the subsystems, the so-called N^2 relationships. As Professor Sercel puts it, "what I need from you and what you need from me." Therefore, the first tool he and his students created was the Relational Parameter Exchange Tool (RPET). Developed in the Filemaker software application, RPET is a database establishing the data requirements between the subsystems. As the design progresses, the satellite parameters can be entered into the database, so RPET can also be used to exchange the data. Since Filemaker accepts standard data formats, RPET can act as a bridge to electronically link other design tools together.

The Spacecraft Design Tool (SCDT) and SCDT Wizard provide a simple and user friendly method to create satellite drawings. The SCDT Wizard provides an Excel-based Graphical User Interface (GUI) to accept key configuration parameters, such as the size and shape of the satellite, the number of solar arrays, and the location of major components. The user enters this information by selecting from drop-down menus and clicking on check boxes. The Wizard then translates the entered information into a standard data format and sends it to the SCDT. The SCDT uses MiniCAD 7 to create the spacecraft from a library of component drawings. If a design requires a component not in the library, the user can create one and then save it to the library for future reference. Since the drawing is created in MiniCAD, it is fully exportable to other applications, such as the Satellite Orbital Analysis Program (SOAP).

The last two tools under development are the Spacecraft Design Trades (SCD Trades) and ICETOP. SCD Trades is an Excel-based tool which accepts requirements from the user through a GUI, then generates key design parameters and component sizes from "first-principles" analytical relationships and historical trends. Professor Sercel estimates there are over 400 equations and relationships built into SCD Trades. Since it is fully integrated, trade studies can be performed quickly by changing the input data and observing the effect on the results. Ultimately, the goal is to link the results from the SCD Trades through the SCDT Wizard into the SCDT, which will then automatically create the satellite drawing. Finally, ICETOP is a low thrust trajectory optimization tool. Since it was still under development, there was little information available about its use or features.

While the CalTech tools are certainly scaled down from the other spreadsheet-based tools at Aerospace, JPL, and TRW, they are still powerful and flexible tools providing excellent conceptual level results. All of the tools are simple and easy to use, and can be learned quickly. Since they are developed in readily-available and common software applications, they are easy to maintain and upgrade. However, none of the four tools address cost, schedule, or risk estimates. For most, this was not the point of the tool. In addition, there is nothing preventing these features from being added, especially to SCD Trades and RPET. The CalTech tools also offer one additional advantage. Unlike the distributed spreadsheet tools, they are designed to be used by a single engineer, such as an engineering manager or a student.

All four tools would make an excellent addition to the NPS curriculum. While NPS provides a great foundation across all spacecraft subsystems, each discipline is taught separately. It is not until the capstone group design class where students first begin to deal with the interrelationships between the different functional areas. Introducing RPET, along with a class period or two of discussion, would provide valuable guidance and direction. SCDT and SCDT Wizard would also be invaluable. Both the individual and group design classes required a satellite drawing as part of the final package. Unfortunately, few students have any experience in using Computer Aided Design (CAD) programs, most of which have steep learning curves. This leaves the students with two alternatives: either create the drawings manually in a presentation package such as PowerPoint, or invest the time to learn a CAD package. SCD Trades could be used as a template, to ensure no important design aspects are overlooked, and to track the results as the student progresses through the design. Finally, electric propulsion and low/continuous thrust maneuvers are becoming increasingly common. For the first time, a commercially available standard spacecraft bus, the new Hughes 702 bus, uses electric propulsion. This trend is sure to increase in the future. However, the current NPS curriculum only minimally introduces low thrust propulsion, and then only in the propulsion course and not in orbital dynamics. Obtaining ICETOP would provide a useful tool on this growing topic and provide a focus for future instruction.

B. STAND-ALONE SOFTWARE TOOLS

Spreadsheet-based design tools are easy to use, powerful, and flexible, and afford many advantages and benefits to the respective design centers. However, they do have their limits. Spreadsheets do not handle transcendental equations, and have trouble with iterative relationships without creating "circular references." Despite the links between subsystems, spreadsheet-based tools cannot optimize, although they do facilitate "manual" optimization. They have no simulation capabilities and can not provide an automated systems engineering function. These limitations restrict spreadsheet-based design tools to the conceptual and preliminary design phases.

In contrast, stand-alone design tools have virtually no limitations on their capabilities or features. With modern software engineering techniques, computers can be programmed to do just about anything, as evidenced by the variety and features of the many subsystem applications.

A software program can generate a design to any level of detail and model each subsystem to maintain configuration control and enforce internal interfaces. The design can be optimized based on the input requirements and the design criteria. In fact, there are many commercially available optimization programs which can simply be incorporated into a design tool. Finally, software tools can assess the performance of the design by simulating the interaction of the spacecraft with the external environment.

All of this power comes with a price. Stand-alone design tools usually have a long, difficult, and expensive development process. Because of the software complexity, the programs are normally created and maintained by dedicated programmers instead of the functional design experts. As the complexity grows, it becomes almost impossible to verify and validate the code. Stand-alone tools also tend to be more rigid and take longer to learn. Instead of using a familiar application, the design engineers must learn a new computer program. This problem also grows with the sophistication of the design tool. While flexibility can be programmed in, modifying the underlying code, the actual design tool, is much more difficult, and again is not performed by the subsystem expert.

With the general characteristics in mind, three specific stand-alone design tools are described below. Lockheed Martin developed the VIS to model detailed satellite designs and simulate the operation of spacecraft and entire constellations. SMAD, sold commercially by Microcosm, is an inexpensive, simple, and easy to use windows-based program to bound a conceptual design of a spacecraft. GENSAT is a relatively new program developed by CTek to support the entire design process, from conceptual through detailed design, and simulate the spacecraft performance.

1. Lockheed Martin Virtual Intelligence Simulator

The Virtual Intelligence Simulator (VIS) is a time based simulation environment for modeling intelligence problems. The goal of VIS is to enable virtual design of spacecraft, systems, or even a system of systems from concept to flight. A concept or system can be evolved downward into progressively more detailed designs without the need to reenter the full model; individual components are updated as the design develops. The simulation environment allows the designers to understand the results and performance of a particular design, and gives insight into why the results are the way they are. Numeric results of any parameter or aspect are also captured for off-line, post-simulation analysis and evaluation. VIS gives designers an understanding of how a system will work without the need to build or assemble expensive hardware and software.

VIS is based on an underlying concept that in a real system all of the functions have a physical location, therefore all of the system functions in the simulation must have a physical location. Instead of developing an extensive and complex model unique to each system, VIS serves as a “universal translator,” providing a simulation environment to link together individual pieces or objects. In developing VIS, Lockheed Martin did not want to force new tools on the designers. Instead, they went to their designers and pulled in their existing tools and models. VIS provides the structure to integrate the diverse tools using an object oriented approach.

There are six top level classes of objects in VIS: CONFIG, SYSTEM, EXTERNAL, WORLD, REPORTS, and GRAPHICS. CONFIG objects determine how the VIS backbone is configured and how it operates. Examples include the simulation clock or atmospheric and weather models. SYSTEM objects are all of the “things” that make up the system, such as airplanes, spacecraft, and ground stations. EXTERNAL objects are any non-environmental externals that cause a system response. The best example is targets, such as enemy forces. Collectively, SYSTEM and EXTERNAL objects are called PLATFORMS since they are the most significant part of VIS. PLATFORMS are what the user is really concerned with when modeling a system. WORLD objects, also called jujus, are non-man-made objects outside the system. Jujus are things that can influence the system without being a part of it, like the position of the sun and moon, and are applied identically to all other objects in the system. REPORTS are reporting functions called directly by the VIS backbone, and provide all of the data output from the simulation. Finally, GRAPHICS objects are symbols, graphics, or GUIs called by the VIS backbone. The GRAPHICS objects represent each PLATFORM object so the user can watch the simulation on projection screens.

To track the large number of objects in the system, VIS uses linked lists. Different types or classes of objects are organized into separate lists, which are themselves part of other higher-level lists. Targets can be treated as individual objects with each target represented by a different item in a linked list or as target decks, where an entire set of targets is represented by a single item in the linked list. Target decks may have hundreds of thousands of items.

Representing the deck as one item provides computational efficiency and avoids the linked list overhead without compromising model fidelity.

PLATFORMS are the top-most objects in the system. They represent all of the objects that make up the system, such as people, buildings, computers, tanks, airplanes, or spacecraft. PLATFORMS can also be individual components of larger objects, like reaction wheels, batteries, thrusters, or antennas. All of the PLATFORMS in the system are defined with a common set of functions, which are MOVE, TASKING, BUS, SENSE, COMM, and GRAPHICS. These functions are actually "virtual" functions, or subroutines, that are passed the necessary data to act on the PLATFORM under calculation.

The MOVE function calculates the position, velocity, and acceleration vector in both Earth fixed and inertial coordinates. Computing both sets of coordinates avoids multiple recalculations of the vectors, adding computation efficiency. VIS also includes conversion routines, further streamlining the calculations. The MOVE function can also be a NULL function since, for example, buildings have a location but rarely ever move.

Logic functions are implemented in the TASK and BUS functions. TASK functions are the "brain", or logic functions, representing any decisional logic function in the system. As an example, a human operator simulated by an artificial intelligence simulation would be attached to a PLATFORM as a TASK function. Non-decisional logic, such as control functions, is computed by the BUS function. The BUS function computes the mechanical and physical properties of the PLATFORM, such as attitude, battery state of charge, etc. Since an object can have several logic functions, PLATFORMs use the same linked list approach that VIS uses to track PLATFORMS.

The SENSE and COMM functions provide the information about or collected by each PLATFORM. The SENSE function models the sensors, and determines what sensory information was collected using a set of inheritable bus characteristics. The COMM function models the transmission links, and computes antenna pointing, link equations, bit error rates, etc.

A graphical icon is attached to each PLATFORM by the GRAPHIC function. Since PLATFORMS are so critical to the simulation, they have their own graphics function instead of using the GRAPHICS object. In the event a GRAPHICS function is not specified for a particular PLATFORM, default icons have been hard coded into VIS.

Since VIS only provides a simulation environment, the actual simulation, or instance, does not exist until run time. Each instance is run from a master script file, which sets up the clock, defines the parameters of the simulation, and determines the rules for that run. Using a script file provides traceability from inputs to outputs and allows the same simulation to be run multiple times. Note, though, that multiple runs of the identical script file may generate different results. This is the inherent nature of the VIS simulation and modeling environment and is what allows insight into the real-world performance of the system. As a result, VIS is also very useful in operational analysis.

VIS can run in several different modes. The script file can run in either batch mode or interactive mode. Interactive mode allows the user to influence the course of the simulation and directly supports operational analysis and war gaming exercises. The simulation clock can also run in several different modes, including incremented time (full speed), real time, real time with a minimum time step, and scaled real time. Incremented time is the fastest way to complete an analysis, and is used for performance and utility determinations. In real time mode, the simulation clock calls the Central Processor Unit (CPU) clock to determine the delta time since the last update. Depending on CPU loading, real time mode may have non-uniform time increments. In real time with a minimum time set mode, the simulation clock will insert a pause if the CPU clock has not yet reached the minimum time step. Thus, the simulation clock will advance in increments equal to or greater than the minimum step. In scaled real time, the simulation clock again calls the CPU clock to determine the delta time, then multiplies by the scaling factor. Real time and scaled real time are best suited to operational analysis and operational demonstrations. Individual functions can also be forced to execute at a specific time or at a certain rate. In this case, the function calls the CPU clock directly and does not use the simulation clock.

Script files run in a specific order, or flow, to prevent conflicts and ensure the data integrity of the results. Some calculations are dependent on other information so some functions must naturally occur before others. The simulation clock is updated first, since the time sets the basis for all other calculations. Similarly, the jujus are updated next since they also apply identically to all other objects. This step updates the positions of non-system objects. The first PLATFORM function to be executed is the MOVE function. Calling the MOVE function first prevents causality problems. With the positions of all of the objects updated, the jujus are called again, this time to determine local effects like gravity, perturbations, or weather. Next, all of the logic functions are updated. Since operators could issue commands to the PLATFORMS requiring a response, the TASK functions are updated first followed by the BUS functions. With the locations and properties of all of the objects in the system updated, the SENSE function can determine what sensory information is collected. COMM functions are executed last, since the information must be generated by the other functions before it can be reported.

Once all of the objects are updated and the data generated, the script turns to outputting the data. The GRAPHICS are updated next, to allow the user to observe what the simulation is doing. In batch mode, this step is skipped. Finally, while watching the simulation is great for demonstrations, analysis and evaluation requires numeric results. Therefore, at the end of each time step the REPORT functions are called. Information can be reported in two ways. First, if a system function generates a report in the real system, then the report is generated by that particular PLATFORM. All other information is collected by the REPORT object, which records information not captured as part of the system. For example, consider the power output of a solar array. If the spacecraft has telemetry for the solar array power output, it is recorded as telemetry in the SENSE function and reported thought the COMM function. If not, the output power can be captured with the REPORT object. It can also be reported in both ways, allowing comparison of the telemetry reading with the actual value.

The objectized modeling approach is extremely powerful and affords a high degree of flexibility and adaptability. First, the use of objects makes VIS modular and allows data and information to come from many sources. The model defining each PLATFORM can be directly coded into VIS, a data interface to another software program, a data interface to another instance of VIS, or a data interface to real hardware. Coded models are developed by the responsible design engineer and can be based on parametric equations, heuristics, historical algorithms, or parameters of actual hardware. Second, a system can be modeled only to the depth necessary to satisfy the required complexity and level of fidelity. During concept exploration, a detailed model is unnecessary and would needlessly slow down the simulation. The wealth of detailed information makes data interpretation more difficult. However, for performance analysis simple models are inadequate. The modular approach allows PLATFORM models to evolve as the design is refined and allows the model to be replaced with real hardware. Third, it reduces modeling time and costs. Once a model is developed, it can be stored in a library and reused in future simulations. In many cases, each PLATFORM will have several models of varying fidelity. Finally, integrating individual objects into a backbone structure simplifies verification and validation. It is unnecessary to validate the entire, full-up system, which doesn't exist until simulation run time anyway. Instead, each PLATFORM model is validated individually by the design engineer. The VIS structure is also verified to ensure the interfaces and data flow are working correctly.

The biggest benefit of the object oriented approach is that it allows the simulation to be built up like a real spacecraft or system. If done correctly, VIS does not know if an object is a model or a piece of real hardware. A one-for-one interface should exist between VIS and the model or hardware. As the design is evolved, more models will be replaced with actual hardware. After sufficient development, all of the simulation models will be replaced allowing VIS to be removed from the middle and the remaining hardware could be launched. Since VIS contains all of the interfaces, it would form the ground station. This has never actually been done, but it is in theory possible.

VIS has given Lockheed Martin tremendous benefits. First, VIS is clearly a very powerful, flexible, and adaptable tool. The model can provide limitless fidelity, and can use real flight hardware or software to see the actual response. Almost more importantly, the fidelity of the model is easily varied to suit the needs of the study. Results from the simulation will vary with the level of fidelity, but can be very detailed and highly accurate. The object oriented approach facilitates evolution of the design to progressively lower levels of detail. New spacecraft components and technologies are easily added. A design engineer only needs to create a model and plug it into the VIS backbone. Finally, the VIS structure enforces consistent interfaces and system engineering practices uniformly across the simulation. If components are incompatible or configured incorrectly, it is immediately apparent.

Despite its power, VIS does have a few drawbacks. As the name implies, VIS is more of a simulation tool than a design tool. VIS also does not optimize, but is very good at trade studies. It does not help the subsystem engineers design and develop their subsystems. However, it will help the design team assess the performance and capabilities of the design throughout the development process. The simulation will clearly show the effects due to different components, and changes will ripple throughout the design. VIS does not address factors beyond the design and performance. Simulations provide no insight into cost and schedule, but may help in assessing risk. The VIS backbone, as with any large computer program, was costly to develop and is difficult to upgrade. Lockheed Martin maintains a small team of computer engineers just to maintain and administer VIS. The dedicated staff makes VIS easier to use for the design engineers and frees them to focus on their subsystems and models.

While not a perfect fit, VIS would be very useful to both the space systems engineering and space systems operations curriculums at NPS. In the final design classes, the engineering students would be capable of developing simple component models, teaching them a valuable skill not currently included in the curriculum. Whether the models were developed by the students or selected from an existing library, VIS would allow the students to see the performance of their design, providing invaluable feedback. The space system operations students also complete a final spacecraft design project. In addition, the operations students could use VIS to perform operational analysis and war gaming exercises for existing and proposed spacecraft, constellations, and systems of systems.

Lockheed Martin would be very supportive. They were very open, cooperative, and responsive, and have a working relationship with NPS. They already have a liaison office to other government organizations and have identified a point of contact specifically for NPS. They plan to make VIS available to remote government users, including connectivity into the VIS simulation environment and tech support. Access to VIS would include the object libraries, unlike the component databases for the CDC, PDC, and ICDF. The libraries would be maintained and updated by Lockheed Martin engineers, not by the students or faculty members. However, it would still be advisable to have several faculty members receive training on VIS. This will enable them to train the students, ease the burden on Lockheed Martin, and smooth the entire process. In short, NPS would gain definite advantages with access to VIS while incurring minimal cost or responsibility.

2. Microcosm Space Mission Analysis and Design Software

The Space Mission Analysis and Design (SMAD) software is an inexpensive, PC-based tool to assist in developing preliminary/conceptual spacecraft and mission designs. The software implements many of the design algorithms discussed in the Larson and Wertz textbook of the same name. While the program assumes the user knows how to design spacecraft, the algorithms are necessarily simplified to allow quick estimates for designs, with the associated inaccuracies. Despite the simplifications, SMAD can be used to perform spacecraft sizing, develop a preliminary design, conduct orbit analysis, analyze spacecraft systems, and learn about spacecraft and the spacecraft design process. (SMAD, 1994, p. 1)

In addition to its design capabilities, the SMAD program can also be used to perform parametric or trade studies. To allow the user to see the effects of varying a given input, many parameters can be varied over a range of values. SMAD then calculates a table of the selected variable versus other dependent parameters. The tabular data can also be plotted, allowing the user to see the information graphically. Note, however, that SMAD does not optimize. The user must still select the best value and enter it into the worksheet.

All of the worksheets are user friendly and easy to use, with graphical and textual inputs and outputs. Navigation within and between modules is easily done by clicking on the appropriate button. Individual pages are clearly identified and well organized. Input values are entered directly into data boxes and are checked against an allowable range. While parameters are designated as inputs and outputs, any of the values can be entered and SMAD computes all of the others. If the entered value is within the valid range, any possible calculations are performed and displayed. If not, a message is displayed with information on the error. Additional information can be obtained on all input and most output parameters by simply clicking on the parameter. Of course, all work can be printed or saved, and is stored as a simple ASCII file which can be viewed with any standard text editor.

The SMAD program also includes a good on-line help utility, making it a useful education and training tool. While SMAD assumes a certain level of knowledge on the part of the user, the help utility clearly explains how to perform a design or trade study. Key parameters and variables are defined and their typical values or ranges are identified. Finally, the help utility includes specific references to the appropriate sections of the SMAD textbook.

The SMAD software consists of twelve graphics oriented program modules. Unlike the two administrative modules, the ten technical modules are made up of several worksheets, or pages, which implement the algorithms for a particular subsystem or design facet. The twelve modules are:

- SMAD Index / Menu
- SMAD Help
- Orbits

- Orbit Maneuvering
- Propulsion
- Attitude Control
- Electrical Power
- Thermal
- Structures
- Communications System
- Observation Payloads
- Spacecraft Design Budgets

While designed to be stand-alone, the modules can be used together to perform a complete design or analysis. Many of the modules accept inputs from other pages or compute output for other worksheets. However, while input and computed values are shared within a module, the program does not automatically transfer information between modules. The user must reenter it into each module as necessary. The following descriptions of the SMAD software program are based on the SMAD Users Guide, provided courtesy of Microcosm, Inc. (SMAD, 1994, p. 9-20)

The two administrative modules help the user interact with, navigate through, and understand the technical modules. The SMAD Index / Menu module serves as a "table of contents" or gateway to the other modules. The user clicks on one of ten buttons to access a particular technical module. The SMAD Help utility is a separate, stand-alone module, but it is not accessed directly by the user. Instead, it is called in one of the technical modules by clicking on a parameter or through the menu bar, and is automatically called when an invalid data value is entered. The Help module provides all of the additional information on the input/output parameters, variables, and values.

The Orbits module computes the key orbital parameters and geometries. Calculations are performed on four separate pages: *velocity*, *atmospheric drag*, *viewing geometry*, and *general.* All of the pages assume circular orbits, use Hohmann transfers, and neglect the rotation of the Earth. The module also includes a nice ground track display program, allowing the user to see the orbit projected on a flat Mercator map.

The *velocity* worksheet computes several velocity related parameters based on the orbital altitude. From the altitude, SMAD calculates the orbital radius, the circular orbital velocity in inertial space, the ground track velocity, and the angular velocity relative to the center of the Earth. This page also calculates several velocity changes (delta Vs). First, it computes the delta V to change the orbital altitude by 1 km. Since the module assumes circular orbits, the initial and final orbits are both circular and are connected with a Hohmann transfer. The deorbit delta V is determined as the velocity change to lower the orbit to a 150 km circular orbit. Finally, the plane change delta V is the velocity increment necessary to change the direction of the velocity vector by one degree.

In the *atmospheric drag* worksheet, the user enters the ballistic coefficient of the satellite and selects a mean or maximum atmospheric drag model. With this information, and the previously entered orbital altitude, SMAD computes the scale height. Atmospheric density is calculated using a standard, variable scale height exponential model. This, in turn, allows SMAD to determine the orbital decay rate, the orbit lifetime, and the delta V necessary to maintain the orbital altitude. The calculations assume a linear decay rate, which is a normal simplification but is only valid for small variations from the actual orbital altitude.

The *viewing geometry* worksheet determines several other orbital geometry parameters of potential interest. The user inputs either the satellite and target coordinates or the elevation angle, and the orbital altitude is again passed from the other pages. The page then calculates the Earth angular radius, the Earth central angle, the nadir angle, the maximum angular velocity, the distance to the target, and the maximum time in view (assuming a non-rotating Earth).

The last Orbits worksheet is the *general* page. From the orbital altitude, this page computes the orbit period, the number of revolutions per day, the range to the horizon, the maximum eclipse time, and the nodal spacing. Once the user inputs the inclination, the nodal precession rate is calculated.

The Orbit Maneuvering module determines the mission velocity budget, including orbital transfer, orbit maintenance, and end-of-life disposal. The modules includes four worksheets: *delta velocity budget*, *orbit transfers*, *orbit maintenance*, and *orbit end-of-life*. Note that if the mission uses a transfer vehicle, the orbital transfer calculations apply to it and not to the spacecraft.

The *delta velocity budget* worksheet is simply a summary page. It accepts no direct inputs and performs no calculations. Instead, it merely summarizes the individual velocity changes from the other worksheets. The user cannot change any parameters on this page. All inputs must be made through the other pages.

The *orbit transfer* worksheet computes the delta V for three types of orbit changes. The *Hohmann transfer* page determines the delta V to change between two circular, coplanar orbits. The page reports the total delta V to the *delta velocity budget* page, but the initial and final delta Vs and the transfer time are also displayed. Velocity increments to change the direction of the velocity vector, but not its magnitude, are calculated on the *simple plane change* page. The orbit is again assumed to be circular. The user enters the orbital altitude and the desired angular change, and the page reports the total delta V. Finally, combination maneuvers are computed on the *non-coplanar transfer* page, which calculates the delta V to transfer between two circular, noncoplanar orbits. The user enters the altitude of the initial and final orbits and the angle between them. The user must also specify the percentage of the plane change performed at each velocity change. The page reports the total delta V to the *delta velocity budget* page, and displays the initial and final velocity increments.

Stationkeeping requirements for geosynchronous Earth orbit (GEO) or low Earth orbit (LEO) orbits are calculated in the *orbit maintenance* worksheet. The user first selects the orbit type (GEO or LEO) and enters the orbital altitude and mission duration. If the altitude falls between 33 650 km and 36 000 km, it is considered a GEO orbit. The user must then specify the station longitude. The page calculates the delta V requirements for North-South and East-West stationkeeping. For LEO orbits, the user must enter the spacecraft ballistic coefficient and choose either a mean or maximum atmospheric density model. This information is not transferred from the Orbits module. The page calculates the annual delta V requirements to maintain the desired orbit.

Finally, the velocity increments to dispose of a satellite at the end of its mission are determined in the *orbit end-of-life* worksheet. The user can select between two options: reduce to a 150 km circular orbit to deorbit or transfer to a disposal orbit. For either option, the user enters the initial and final orbital altitudes. The worksheet computes the delta V to perform a Hohmann transfer to the final orbit.

The Propulsion module determines the propulsive requirements for the entire mission, including the spacecraft, orbital transfer vehicle, and launch vehicle. Basic inputs to this module include the spacecraft dry mass, key orbital parameters, and the delta V requirements for attitude control and all orbit changes. Some of the input variables may be calculated in other modules, but the values are not automatically passed into the Propulsion module; the user must reenter them. The module contains four worksheets: *propulsion basics*, *propulsion budget*, *transfer vehicle*, and *launch vehicle selection*.

The *propulsion basics* worksheet computes the total propellant mass consumed for three different uses. The *orbital insertion* page uses the ideal rocket equation to compute the propellant mass necessary to produce the required delta V. Inputs include the delta V required from the spacecraft (not from the transfer vehicle, if any), the propellant specific impulse, the thruster size, and either the initial or final spacecraft mass. With this information, SMAD computes the effective exhaust velocity, the propellant mass flow rate, the total propellant consumed, and either the final or initial spacecraft mass (whichever the user didn't enter). The *orbital maneuvering* page operates just like the *orbital insertion* screen, with the same inputs and outputs. They are separate screens to show their individual contributions to the propellant budget. For consistency, the *orbital insertion* final spacecraft mass should be used as the initial spacecraft mass in the *orbital maneuvering* page. Finally, the attitude control page calculates the total propellant mass consumed based on the total impulse, which is the total attitude control force applied to the spacecraft multiplied by the total time the force is applied. The total impulse can be calculated in the Attitude Control module. If this module is not available, the propellant mass can be estimated as a percentage of the orbital maneuvering propellant mass.

The *propulsion budget* worksheet sums the individual propellant masses and computes the total propellant load. The propellant masses are passed from each screen in the *propulsion basics* worksheet and cannot be changed here. The user can specify a propellant margin to account for residual propellant, growth, or other uncertainties. The user can also select one of three propulsion subsystem types, either solid, liquid, or cold gas. SMAD then calculates the total wet mass of the spacecraft propulsion subsystem.

The *transfer vehicle* worksheet estimates the size and performance of the orbital transfer vehicle (OTV), if one is used. Inputs include the spacecraft wet mass at launch, the delta V for orbital insertion, and the specific impulse and pad mass of the OTV. The spacecraft mass should be equal to the initial spacecraft mass used in the *orbit insertion* page. While the user can change the value on this page, the change is not updated on the other pages. The worksheet computes the burn-out mass of the OTV as a user-specified percentage of the OTV pad mass. The worksheet also computes the total delta V capability of the OTV from the ideal rocket equation. Note that the delta V for orbital insertion is not actually used in the calculations; it is simply displayed next to the OTV delta V capability for comparison.

The *launch vehicle selection* worksheet compares the capability of a selected launch vehicle with the launch requirements. The user must input key orbital parameters, an estimate of the booster adapter mass, and the desired performance margin. Spacecraft dry mass and OTV pad mass are passed from previous pages. As before, the value can be updated on this page, but changes will not be passed back to the other worksheets. The worksheet provides tables and plots of the capabilities for several launch vehicles. Once a launch vehicle is selected, the launch reliability, acceleration loads, and fundamental frequencies are provided for use in the Structures module.

The Attitude Control module determines the magnitude of torques acting on the spacecraft and estimates the size of the attitude control subsystem. The subsystem can use a combination of reaction/momentum wheels, thrusters, and/or magnetic torquers. Calculations are performed in three worksheets: *disturbance torques*, *slewing torques*, and *attitude control system sizing*.

The *disturbance torques* worksheet estimates the magnitude of external torques acting on the spacecraft. This worksheet requires a long list of rather detailed information. Fortunately, the on-line help utility provides good descriptions and typical data values or ranges to assist the user in completing each page. The user must enter the orbital parameters, the sun incidence angle, the maximum off-nadir point requirements (determined by the desired mission), and information on several physical characteristics of the spacecraft, including the moments of inertia, the maximum surface area facing the sun, the surface reflectivity, and the residual dipole. Information on the center of gravity, center of solar pressure, and center of aerodynamic pressure is also required. Normally, the center of gravity is set to zero and the centers of solar and aerodynamic pressures are specified as offsets. Finally, the user must reenter the atmospheric density and spacecraft velocity, which are calculated in other modules. With this extensive set of information, the worksheet determines worst-case estimates for the gravity gradient, solar radiation, magnetic, and aerodynamic torques along with the total external disturbance torque. The worst-case calculations assume a circular polar orbit and a nadir-pointing spacecraft.

Internal, or spacecraft-generated torques are addressed in the *slewing torques* worksheet. This worksheet computes the torque required to rotate a zero-momentum (non-rotating) spacecraft through a given angle in a specified amount of time. Inputs are simply the spacecraft moment of inertia about the spin axis, the slew angle, and the maneuver time. The calculations assume the spacecraft accelerates over half of the time and decelerates over the other half so the spacecraft is not rotating at the end of the maneuver.

With the torque requirements computed, the *attitude control system size* worksheet estimates the size and mass of the entire subsystem. Calculations are performed in three separate pages for reaction/momentum wheels, thrusters, and magnetic torquers. The *wheel sizing* page uses the largest torque, either disturbance or slewing, computed in the previous worksheets to determine the angular momentum requirements. Wheel capacity is then calculated based on conservation of angular momentum. Once the user enters two values for the wheel radius, mass, or angular velocity, the worksheet computes the third. The *thruster sizing* page estimates the thruster size and total propellant mass requirements. The user can select one of three options to size the thrusters: slewing a zero momentum spacecraft, slewing a momentum-biased spacecraft, or momentum dumping. Finally, the *magnetic torquers* page calculates the spacecraft magnetic dipole required to reject the disturbance torques or to slew the spacecraft. This calculation is not worst case since the page assumes the maximum value of the Earth's magnetic field for the specified orbital altitude.

The Electrical Power module computes the size and mass of the electrical power subsystem on the *electrical power source*, *solar array sizing*, and *energy storage* worksheets. Basic inputs include the average and eclipse power requirements, mission orbit parameters, and mission duration. This information is automatically shared between the worksheets.

The *electrical power source* worksheet identifies the power source and provides an estimate of the specific power. The user can select one of four common power systems: solar photovoltaic, solar thermal dynamic, radio-isotope, or nuclear reactor. Solar photovoltaic systems can use either peak power tracking or direct energy transfer power regulation. Power density cannot be entered directly; instead the value can only be specified as low, average, or high. The mass of the electrical power subsystem is then calculated by simply dividing the power density into the average energy requirement.

The solar array is designed in the *solar array sizing* worksheet. The user must enter the sunlit and eclipse power requirements, mission duration, and solar maximum incidence angle. As a guide, the worksheet will display a figure of the daylight and maximum eclipse times for circular orbits as a function of orbit altitude. The solar array degradation must also be specified. Inherent degradation is due to inefficiencies like packing factor, elevated operating temperatures, and shadowing. Annual array degradation is caused by aging, coverglass darkening, thermal stress, and radiation damage. Finally, the user can select from a list of solar cell types, which in turn specifies the cell efficiency. The worksheet computes the solar array area to satisfy the power requirements entered in the previous worksheet, and the beginning of life and end of life power output.

The *electrical storage* worksheet determines the mass and capacity of the primary and secondary batteries. The battery type is selected from a list and the power density can be specified as low, medium, or high. Eclipse duration is passed forward from the *solar array sizing* worksheet. Another key input is the depth of discharge, which depends on the type of battery and the expected number of charge-discharge cycles over the life of the mission. To help select this value, the worksheet displays a figure relating the orbital altitude to the number of discharge cycles and allowable depth of discharge. Finally, the user must enter the battery transmission efficiency, number of batteries, and the bus voltage. The worksheet calculates the mass of the secondary battery along with its capacity in both Watt-hours and Amp-hours. Similarly, the primary battery is sized by the battery type, power requirements, and time of operation.

The Thermal module computes the maximum and minimum spacecraft temperatures. It also estimates the radiator area and internal heater power required to maintain user-specified temperature limits. Thermal analyses can be performed on three different spacecraft geometries, each on a separate page: *spherical spacecraft*, *flat plate*, or *combined*. The worksheet calculates heat inputs from four sources: solar energy, Earth infrared energy, Earth albedo, and internal power dissipation. The module assumes the spacecraft is isothermal, and only calculates steady state temperatures. However, these are normal simplifications for preliminary thermal analyses.

The *spherical spacecraft* worksheet assumes the spacecraft is an isothermal sphere. Inputs include the orbital altitude, surface area, and the emissivity and solar absorptivity of the surface. The user can select a low, medium, or high value for the surface properties, but cannot enter a specific value. The emitted energy is computed from the Stefan-Boltzmann equation.

With this information, the worksheet can compute the maximum and minimum steady state temperature of the spacecraft. If the user specifies an allowable range, the worksheet estimates the radiator area and internal heater power to maintain the temperature within these bounds. However, since the calculations assume an isothermal spacecraft, the worksheet cannot determine the heating and cooling requirements for individual components.

Temperatures for solar arrays and other panels are computed in the *flat plate* worksheet. Temperatures of the flat plate are computed with a similar approach used for the spherical spacecraft. The orbital altitude is passed from the preceding worksheet. Other inputs are similar to those for the spherical spacecraft, except the user must also identify an incidence angle. The module then assumes the top of the plate is facing the sun at the specified incidence angle and the bottom is facing the Earth. Different surface properties can be entered for the top and bottom of the plate. Since solar panels are typically flat plates, an extra term is added to the energy balance equation to compensate for electrical power generation. Unfortunately, the module only accepts solar cell efficiencies between 7% and 10%, which is low by today's standards. Since solar arrays should be as cold as possible, and since flat panels act like a radiator, heater powers and radiator areas are not calculated.

The *combined* worksheet computes the maximum and minimum steady state temperature for a sphere with a flat plate. The entire structure is still assumed to be isothermal, so the sphere and plate must be the same temperature. Difference surface properties can be entered for the sphere and the top and bottom of the plate. The inputs are the same as for the preceding two pages, and any parameter updates on this page are passed back to the other worksheets. The radiator area and internal heater power required to maintain user-specified temperature limits is also computed and passed back to the previous pages.

The Structures modules calculates a first order estimate of the mass of the structural subsystem necessary to support the spacecraft during launch. The spacecraft is assumed to be cylindrical, with either a monocoque or semi-monocoque structure.. Calculations are based on determining the necessary structural thickness needed to carry the applied launch loads. The thickness is computed on an ordered series of five worksheets, each addressing a different aspect of the design requirements. As each successive page determines the thickness, it is automatically transferred forward to the next worksheet. However, no information is shared backwards. If the passed thickness initially satisfies the design requirements of that page, it should not be decreased or the previous requirements will no longer be met. The five worksheets, in order, are *design properties*, *applied loads*, *rigidity*, *stability*, and *semi-monocoque*. Information on the launch environment, including the axial and lateral accelerations and the fundamental frequencies, can be obtained from the Propulsion module, but must be manually reentered into the Structures module.

The *design properties* worksheet does not actually perform any calculations. Instead, it collects a wealth of information for the other worksheets. The page also summarizes and displays the results from the other worksheets. Since the satellite is assumed to be cylindrical, the user is asked to enter the length and radius of the spacecraft, which is determined by the launch vehicle fairing envelope and the payload dimensions. The structural material is selected from a list, which in turn sets Young's modulus, Poison's ratio, the material density, and the allowable stresses. Lateral and axial launch accelerations and the fundamental frequency for the launch vehicle selected in the Propulsion module are reentered. Finally, the user specifies the desired factor of safety. All of these inputs are shared with the appropriate other worksheets.

The *applied loads*, *rigidity*, and *stability* worksheets form an ordered series of pages that refine the thickness of the structure to ensure it meets all design requirements. The structural thickness is first computed on the *applied loads* worksheet. The lateral and axial launch loads and the spacecraft mass are converted into an equivalent applied load. The worksheet then calculates the cross-sectional area and the resulting cylinder thickness required to support the applied loads. Next, the *rigidity* worksheet computes the cylinder deflections and natural frequencies. These values are then compared to the limits imposed by the launch environment. If the deflections and/or frequencies are outside the limits, the user can enter the design requirement and the page computes a new thickness. Finally, the *stability* worksheet determines the thickness required to prevent buckling by computing the margin of safety. Unlike in the *rigidity* worksheet, if the margin of safety is too low the user cannot simply enter the desired value to recompute a new thickness since the calculations are based on a transcendental equation. The increased thickness must be found from trial and error. Now that the minimum thickness satisfies all of the design requirements, the stability page calculates the mass of the monocoque structure. The *stability* worksheet also allows the use of stringers, either to meet the margin of safety requirements or to decrease the structural mass.

If the user does not want to use a cylindrical, monocoque structure, the *semi-monocoque* worksheets calculates the thickness and mass for a panel structure with stiffeners. As a first estimate, the skin thickness is set to half of the thickness from the *rigidity* worksheet. To specify stiffeners, the user can select from a list of 4, 8, 12, or 16 stiffeners. The worksheet then computes the area and mass of the skin, the mass of the stringers, the total structural mass, and the design margin.

The Communications System module calculates the link margin for a given communications system design. It consists of a single worksheet of parameters used to design a link budget. Information is organized into three basic groups for the transmitter properties, receiver properties, and performance requirements. Unlike the other modules, the parameters are not designated as inputs or outputs. Once enough information is entered to calculate a particular parameter, it is computed and displayed. Calculations assume parabolic antennas. The module only completes one link design at a time. If the spacecraft uses multiple uplinks, downlinks, and crosslinks, the worksheet must be completed several times.

A link design normally begins by specifying the carrier frequency, which is usually not a design parameter. Instead, it is determined by mission requirements or external constraints. The mission may demand certain data rates, require atmospheric penetration, or necessitate compatibility with other elements of an overall system. Examples of external constraints include Federal Communications Commission (FCC) frequency allocations or the background radiofrequency (RF) environment.

To complete the link budget, the worksheet needs information on the transmitter properties. The user must enter either the transmit antenna diameter or the transmit beam width. Once one is entered, the worksheet computes the other. Transmitter power, in Watts or decibels (dB), is based on the data rate or the capabilities of the spacecraft. The accuracy of the attitude control system determines the point errors of the transmit antenna. The remaining parameters, which can be entered or computed, include the transmitter line losses, the antenna gain, and the effective isotropic radiated power (EIRP).

Receiver properties are similar to those for the transmitter. The user again enters either the antenna diameter or beam width, and the worksheet computes the other one. System noise temperature is a function of several individual noise contributions from inside the receiver or external sources. The receive antenna losses, pointing offset, and gain complete the data set on the receiver properties.

Performance requirements dictate the data rates and bit error rates for a given orbit. The orbital altitude and elevation angle determine the propagation path length, which in turn sets the free space loss. Other losses include atmospheric attenuation and polarization losses. Mission or ground system requirements usually specify a given bit error rate (BER). The energy-per-bit to noise-density ratio (E_b/N_o) is a function of the bit error rate and the modulation scheme. The worksheet presents a graph of BER as a function of noise density and allows the user to select the required noise density directly from the figure. Finally, the worksheet accepts information on the implementation losses and determines the carrier-signal to noise-density ratio.

With all of the above information, the Communications System module calculates the link margin. If the link margin is too low or excessively high, the user can change one or more input parameters and see the recalculated margin. The problem can also be worked "backwards" by entering the desired link margin, then progressing back through the worksheet to the input parameters.

The Observation Payloads module determines the mission requirements placed on the payload and estimates the payload's size and mass. The module consists of three worksheets: *electromagnetic spectrum*, *payload parameters*, and *payload sizing*. Calculations rely on information on the orbital parameters from the Orbits module. Outputs are provided for several other sheets. The physical dimensions and mass of the payload are used in both the Propulsion and Structures modules.

The *electromagnetic spectrum* worksheet estimates the best frequency and sensitivity to detect a given target. After entering the temperature of the target, the peak wavelength is computed from Wein's Law. The Stefan-Boltzmann law calculates the total radiant emmitance, which determines the required sensitivity of the sensor. Finally, the spectral irradiance is found with Plank's Law.

Preliminary sizing of an optical or infrared (IR) payload is provided by the *payload parameters* worksheet. The primary constraints on the payload size are the distance to the target, found from the orbital altitude and elevation angle from the Orbits module, and the wavelength, calculated in the previous worksheet. Coverage and resolution are determined by the focal length, object plane radius, and aperture diameter. Focal length is directly proportional to the image plane radius, which is constrained by the numerical aperture and by physical limitations on how small a detector can be made. At the other extreme, the overall size of the spacecraft limits how large the image plane can be. The object plane radius determines the coverage capabilities. The radius should be as large as possible, providing the best coverage, but is limited by the focal length, mission coverage requirements, optics numerical aperture, and the resolution capability of the detector. The main parameters computed on the worksheet include the focal length and the angular resolution. As calculated, the angular resolution represents the diffraction limits of the optics and is not necessarily achievable due to atmospheric distortion.

The *payload sizing* worksheet estimates the physical parameters and computes the data rate generated by the payload. The data rate, in bits per second, is found by multiplying the bits per pixel, pixels per image, and the images per second. The size of the detector sets the number of pixels, and the image rate is dictated by mission requirements. The size of the payload is computed by scaling from a similar, existing system. The user enters the aperture diameter, linear dimensions, mass, and power requirements of an existing payload. Using the required aperture of the new sensor to scale these values, the worksheet determines the payload linear dimensions, area, volume, mass, and power requirements.

The Spacecraft Design Budgets module estimates several spacecraft level parameters along with the subsystem and spacecraft reliabilities. Three worksheets, *spacecraft sizing*, *reliability budgets*, and *reliability basics*, convert the payload characteristics into the requirements for the spacecraft and estimate system reliability. The user can enter previously known information or can compute the needed values from the other modules. However, this module essentially recomputes parameters found in other modules, which use more detailed calculations.

The *spacecraft sizing* worksheet converts several payload parameters into sizing estimates for the spacecraft. The payload mass is converted to the spacecraft dry mass by applying a user-selected scaling factor ranging between 2 and 7. Total spacecraft mass, or wet mass, is found by adding the dry mass to the total propellant mass. After the user enters the total velocity change requirements and specific impulse of the propulsion system, the orbital maneuvering propellant mass is calculated from the ideal rocket equation. The attitude control propellant mass and propellant margin are found by scaling the orbital maneuvering propellant mass. To find the spacecraft volume, the user selects from typical values of spacecraft density, which is divided into the wet mass. By assuming the spacecraft is a cube, the worksheet calculates the linear dimensions and cross-sectional area. With the mass and dimensions determined, the moments of inertia are computed. Finally, the total spacecraft power requirements are computed by applying another scaling factor to the payload power.

Reliability information is estimated in the *reliability budget* and *reliability basics* worksheets. The *reliability budget* worksheet calculates the reliability of the total spacecraft system. The worksheet assumes that a single failure in any subsystem results in failure of the entire system. Reliabilities entered for each individual subsystem are simply multiplied together. The individual subsystem reliabilities are found in the *reliability basics* worksheets. The user must enter the failure rate, operating time, and number of components in each subsystem. In addition, the user can select either series or parallel components and either similar or different components. The worksheet then estimates the subsystem reliabilities.

Combined, the 10 technical modules address nearly every significant aspect of a preliminary spacecraft design. The modules are simple and easy to use. The help utility is quite good, providing definitions, explanations, and typical values on all parameters. All input values are automatically error checked against typical or allowable ranges. While these features make the SMAD software a good training and education tool, the level of instruction and design is most appropriate for the undergraduate level or the non-technical / non-aerospace professional.

The fidelity of the SMAD design model is limited. Many of the calculations are over-simplified, yielding results that only bound a spacecraft design. The assumptions also limit the applicability of the SMAD design tool. All of the calculations assume circular orbits, and in some cases only LEO or GEO orbits. Even worse, assumptions are inconsistently applied across the modules. Sometimes the spacecraft is assumed to be a sphere, sometimes a cylinder, and at other times a cube. Scaling factors are frequently used, further limiting the model fidelity. Cost, schedule, and risk are not addressed in any module, however, the software does perform a limited reliability analysis for the launch vehicle and spacecraft.

The SMAD design tool is a stand-alone software program and provides no flexibility or adaptability to the user. In many cases, parameters are entered by selecting either mean-maximum or low-medium-high instead of entering a numerical value. In other cases, the user selects a component from a list, which then specifies values for the associated parameters. As a result, SMAD cannot evolve a preliminary design to lower, more detailed levels. The use of hard-coded lists prevents new technologies from being incorporated into the SMAD software. The user cannot even circumvent this limitation by entering values for the new component or technology. The executable code is inaccessible to the user, so the software itself cannot be upgraded to add new features or capabilities without buying a new version or contracting with Microcosm. Currently, no upgrades are available and none are under development.

The use of stand-alone modules means the different aspects of the design process are not integrated throughout the model. Systems engineering principles and configuration control are not enforced across the subsystems. The same values must be repeatedly entered into separate modules, which becomes tedious and introduces the possibility of entry errors. Even if the intended value is entered, the user could inadvertently input different values into the separate modules, making the design inconsistent and lowering the accuracy of the results. The entire design is never summarized or displayed. The Design Budget module appears to do this, but then develops new, higher-level and more course estimates for many parameters that are calculated in more detail in the individual modules.

Several modules support trade studies and parametric analyses. Tables and graphs are easily created showing how some parameters change as another value is varied. However, the trades studies are not linked into the design model. The user must reenter the essential results of the analysis. This also places the burden of tracking dependencies on the user. If a dependency is forgotten or a value not updated, the model will become inconsistent and accuracy will suffer.

Because of its simplicity and technical level, the SMAD software package is not well suited to the NPS curriculum. The program better matches undergraduate studies. Microcosm was very supportive and responsive to inquiries. They provided materials and manuals on the SMAD software and several engineers met with the author. Technical support may still be a problem, though, since Microcosm no longer maintains a standing software team. The program was developed several years ago, and there are no plans for any updates. Despite this support, and while the book is excellent, the software design tool is far to simplified and limited. The Space Systems curriculums are much more extensive and detailed. The program would be essentially useless during the final design projects, which require a considerable amount of design detail and integration. In short, although it is inexpensive and easy to use, the SMAD software program is not recommended for NPS.

3. Computational Technologies GENSAT

Founded in November 1993, Computational Technologies, Inc. (CTek) was established to introduce practical, object-oriented technologies into mainstream space engineering companies. CTek's goal was to streamline complex engineering projects by reducing costs, simplifying complexity, and improving productivity. To establish the capabilities their new system, called GENSAT, CTek relied on the endorsement of well-known professional scientists and engineers in the satellite and space field. Perhaps one of the best known consultant is the former CTek Chief Executive Officer (CEO) and Chairman of the Board, Dr. Marshall Kaplan. Dr. Kaplan is the author of several widely-used textbooks on satellite design, particularly in the area of dynamics and control. CTek has also forged partnerships with several dominant, state-of-of-the-art engineering software companies and spacecraft manufacturers.

GENSAT is a general-purpose systems engineering software environment supporting all phases of spacecraft design and manufacture. Program requirements are directly captured and fused with software tools and expert rules, creating a single, unified project model. The project model is composed of a hierarchically organized collection of "engineering objects" stored in an object-oriented database. An intuitive, object-oriented graphical user interface allows the project model to be rapidly customized to meet the needs of a particular program. In addition to representing the design, the project model serves as a "live" virtual prototype, providing extensive modeling, simulation, and optimization capabilities.

Representing a new type of software systems technology, GENSAT uses a truly open architecture to transparently integrate essentially any engineering software tool. Instead of replacing an engineer's existing tools, GENSAT enhances and extends the functionality of the tools and engineering methods. Legacy software codes written by the engineer, commercially available space applications, and utilities written within GENSAT can all be seamlessly incorporated into one powerful, fully-integrated design tool.

GENSAT provides a collaborative engineering and project management environment to manage the complexity of the project development process. Based on a client/server computing architecture, GENSAT can be easily integrated with an in-house computer network. Applications, tools, and databases can be stored locally or distributed across numerous platforms, and can even be accessed remotely via the Internet. Teams of engineers can work individually or jointly at the component, subsystem, or system level. The project model can be continuously evolved to iteratively create any level of complexity or detail. The use of versioning allows multiple levels of detail to exist simultaneously. Versioning also permits engineers to work on pieces of the overall design or even on entirely different versions of the complete model in a controlled and reproducible manner. As individual analyses and trade studies are performed, GENSAT automatically tracks the project design history and enforces constraints across the entire project model, maintaining consistency and ensuring feasibility.

A fully integrated system environment is created through a core object oriented software framework that links together distinct design applications with a high-level interface. As depicted in Figure 2.4, the GENSAT architecture consists of three components: an object oriented graphical user interface, any number of individual software applications, and a support framework system called the Scientific and Engineering Application System, or SEAS. While the applications are an essential part of GENSAT and are fully integrated into the system, they are not considered pieces of the SEAS framework. The following descriptions of the GENSAT system and the SEAS architecture are derived from the GENSAT Technical Overview, provided courtesy of Computational Technologies, Inc. (CTek, 1998, p. 1-14)

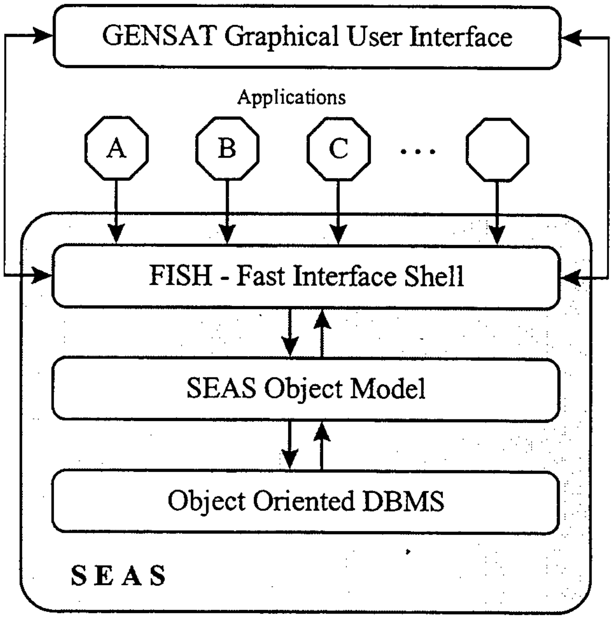


Figure 2.4 GENSAT System Architecture (CTek, 1998, p. 3)

A high-level, object oriented graphical user interface (GUI) provides access to the various applications and project models. Because of the size and hierarchical structure of the project models, the use of a GUI simplifies navigation through the objects and applications. Projects, project versions, and objects can be quickly and easily selected from pre-existing lists. Simply clicking on an object displays its state or performs an operation. The GUI can also be used to perform the various engineering or systems operations. While GENSAT scripts and functions can be created or edited directly in SEAS through a command window, most users will primarily work with a particular project version using only the GUI.

Individual design tools and applications are an important part of the overall GENSAT system. The distinct tools are integrated as needed to effectively form a single interacting application, yielding far greater power than the combined capabilities of each individual tool. The open architecture and object oriented framework allows essentially any application to be transparently incorporated into the GENSAT structure. The tools can cover a wide range of capabilities, including commercially available computer aided design (CAD), computer aided engineering (CAE), or computer aided manufacturing (CAM) programs, productivity tools such as Framemaker or Excel, or even proprietary legacy codes written in FORTRAN, C, or C++. GENSAT comes with a dozen pre-integrated applications providing capabilities across all aspects of the design process, including satellite analysis, finite element analysis, general mathematical analysis, graphical visualization, text processing, data management, inter-process communication, and engineering design optimization.

CTek has formed strategic alliances with several engineering software companies, allowing them to integrated the corresponding applications directly into the GENSAT environment. All of the following software tools are pre-installed in the standard version of GENSAT.

Maple®, by Waterloo Maple, Inc., is an interactive mathematical environment that manipulates symbolic algebraic expressions, and includes a built-in library of over 2,700 engineering and scientific functions (Waterloo Maple, 1999). The MathWorks, Inc. markets MATLAB®, a powerful matrix processor that combines numerical computation, advanced graphics and visualization, and a high-level programming language in one application. Simulink®, also by The MathWorks, Inc., is built on top of MATLAB®, and provides and interactive tool to assemble graphical block diagrams to model, simulate, and analyze dynamic systems

(MathWorks, 1999). Orbital analysis and 3-D visualization of the satellites and other space-related objects is performed by the Satellite Tool Kit (STK®), developed by Analytical Graphics, Inc. (AGI) (AGI, 1999). AutoCAD® is a computer aided design tool developed by Autodesk, Inc. VMA Engineering is a leader in structural analysis and design optimization. They market GENESIS®, which optimizes the structure by converging a series of finite element analyses (VMA Engineering, 1999).

To provide enhanced capabilities, even more applications are provided for an additional fee. For example, the following three optional tools are pre-integrated into GENSAT. If detailed structural design and optimization is necessary, users can order MSC.Nastran® by Macneal-Schwendler Corporation (MSC) (MSC, 1999). Parametric Technologies Corporation (PTC) provides mechanical design and simulation with their Pro/ENGINEER® software application (PTC, 1999). To gain an "manufacturability" viewpoint early in the design process, I-DEAS® by Structural Dynamics Research Corporation (SDRC) develops digital master design project models with CAD/CAM/CAE technology (SDRC, 1999). The optional applications can either enhance or replace the standard tools, as determined by the needs of the user.

The Science and Engineering Application System (SEAS) forms the core framework technology for GENSAT. SEAS both integrates diverse design applications and manages the complexity of the design process, providing a host of benefits to the design engineer. With the broad range of applications, tools, and legacy data integrated directly into its framework, SEAS possesses advanced modeling, simulation, and optimization capabilities. Incorporating existing applications and second-generation codes allows designers to work in their traditional environment, freeing them from the need to learn a sophisticated new software program. The distributed processing environment facilitates concurrent engineering and multidisciplinary analysis and design. Project models are dynamically extensible and highly flexible. The models capture all of the system requirements, constraints, and expert rules imposed on the design, and accommodate design modifications and new project requirements made at any stage of the project lifecycle. SEAS manages the complexity of an extensive and detailed project model through the use of abstraction and encapsulation.

SEAS is a fully CORBA compliant object oriented software technology and is an open system in many regards. The Common Object Request Broker Architecture, or CORBA is a data exchange protocol introduced by the Object Management Group (OMG) to allow any application to communicate with any other. The OMG is a consortium of companies formed to establish industry guidelines and detailed specifications providing a common framework for application development. For more information on CORBA, access the OMG website or type CORBA into any standard Internet search tool. (OMG, 1999).

CORBA compliance provides a number of benefits to the SEAS environment. First and most obviously, users may integrate any kind of application or legacy code directly into the architecture. In fact, a number of powerful applications are pre-integrated or sold as options. Second, in addition to applications, the object database provides the facility to integrate legacy data from almost any other database, such as Oracle or SyBase. Next, SEAS provides the tools to create specialized applications within the GENSAT environment. Users can follow any methodology they wish to create the data model. Finally, the methods for integrating the applications and legacy data are freely distributed to all users. The only proprietary information on GENSAT and SEAS is the design and code for the interface software language and how it is integrated with the commercial database.

The SEAS framework combines three fundamental components, as depicted in Figure 2.4. The functional capabilities and open system architecture are achieved by combining an interactive, object oriented software language, called FISH for Fast Interface SHell, with an embedded, commercial object oriented database management system (OODBMS). The third, and key, component is the SEAS object oriented project model. The basic GENSAT system also includes a generic template model, called GenStar.

The Fast Interface Shell, or FISH, is a high level interactive object oriented software language providing a single, consistent user interface to all of the various distinct design applications. In essence, FISH is the "glue" that holds the framework together, and fills three primary roles in the SEAS environment. First, it is used to create the common project model. Second, it facilitates the integration of the distinct applications and legacy codes into GENSAT. Third, it provides a single, interactive language for executing operations in the integrated system. These operations can be functions supplied within the individual applications, available within the embedded OODBMS, provided by the FISH language, or new functions written in FISH that combine the functionality of a number of applications.

All distinct applications and legacy codes communicate with each other and the GENSAT system through FISH. Traditional "controllers" merely transfer data or control the flow of execution between applications. Most integrated applications are interconnected by binary associations, where A talks to B, B talks to C, and C talks to A. Unfortunately, the complexity of the binary arrangement grows exponentially with the number of individual applications. Due to the large number of integrated applications, this arrangement was not adequate for GENSAT. Instead of using a binary approach, FISH provides a single common interface integrating all of the software tools and the GENSAT system and manages the complexity of data and data sharing. Programs written in the FISH language, called scripts, "wrap" each application or legacy code. The script does more then simply act as a "data translator"; it integrates and extends the functionality available from the applications and the OODBMS.

The power of FISH is realized through the use of its many utility functions. FISH combines in a single language a high level mathematical and logical application language, most of the functions and capabilities of a complete programming language, and an engineering database language. The intrinsic functions include Unix system and database functions and several hundred Science and Engineering Analysis Library (SEAL) mathematical and logical functions. An extensive library of objects and functions for creating, modifying, and destroying objects and databases is also included. Finally, FISH has utility functions for manipulating FISH data and functions for designing customized graphical interfaces and other GENSAT applications.

FISH performs all of the programming, database, and integration functions to build and support system applications. Incorporating the intrinsic data types into arbitrary objects and applications provides an almost unlimited capability for representing engineering data, enabling the creation of arbitrarily complex models. These operations can be applied anywhere in the project model, from the local component or subsystem to the global system level. On-line documentation provides definitions and formats not only for the intrinsic functions, but can be extended to include the user-created functions.

A distributed Object Oriented Database Management System (OODBMS) transparently supports FISH. While FISH wraps each application, it is the OODBMS that performs all of the data management tasks. The database captures all of the relationships between the different entities in the project model, and assembles composite objects on demand from data resident in distinct applications or located in different legacy databases. The OODBMS also automates the configuration management procedures, allowing concurrent engineering activities to be performed across engineering disciplines and within different phases of the design cycle. It can track and manage numerous versions of the project model in several phases of the design cycle by creating a persistently stored dynamic object model.

All of the operations of the GENSAT system are triggered by the OODBMS. Through the FISH interface, the database provides the ability to customize and extend the functionality of the integrated system beyond the combined individual functions of the applications. As a fully integrated system, the OODBMS can perform queries or simulations of the engineering system that were previously impossible. New operations created in the integrated system can be tested without the need to recompile and debug code or test each change. Integrability concerns, like file exchange, object attribute changes, or remote application execution, occur transparently. Best of all, the database never needs to be directly used to provide these capabilities. All database operations can be defined via the GUI interface and the project model. This frees the user from the need to learn an entire new software product or database system, allowing rapid early benefits.

While the previous components provide the design, analysis, and optimization capabilities, the key component of the GENSAT system is an object oriented project model. The project model is the reason the other components exist, and it ties them all together. The model is created under the GUI with the integrated applications through FISH, and is stored in the OODBMS. All of the higher level operations on the project model can be performed with the GUI. The remainder of lower level operations are performed using FISH directly (the usual case) or can be written in the user's language of choice and then integrated with FISH.

System requirements are directly captured in expert rules. The expert rules impose constraints and restrictions on how operations are performed and on the value objects can have. GENSAT performs the rules and operations optimally. This means that only the rules or operations that are absolutely necessary to provide the needed information are performed. Object values are stored and are only recomputed if it is necessary to update another value. The object values and the aggregate project and project version are stored persistently in the OODBMS.

The project model is highly dynamic, and can be evolved or refined by executing operations to perform trades studies or multidisciplinary analysis and optimization. In addition to modifying existing objects, users can define new objects, create new relationships between objects, and impose or modify system requirements. The OODBMS tracks the changes and automatically checks for impacts on the rest of the model. A configuration management system within GENSAT allows these activities to be performed concurrently among several multidisciplinary teams.

The project model is organized into a hierarchical collection of objects. The highest level in the hierarchy is the "class", which represents a number of objects with the same structure. Each particular object is called an "instance" of the class. A class is decomposed into aggregate subclasses, called "attributes." For example, a space system might be divided into mission, spacecraft, support systems, and management classes. The spacecraft class could contain objects for the orbital geometry, bus, payload, mass properties, and performance. Each subsystem would be an attribute of the bus object. This decomposition continues until all of the essential information and characteristics of the system being designed are captured in the model.

There are many different types of attributes and attribute values. Attributes can be prescribed or derived. Prescribed attributes have their values entered or set interactively by the user. Values for derived attributes are computed by formulas or expert rules. Attribute values can be one of two types, either user-defined or intrinsic. User-defined values specify another class name while intrinsic values have data types used within FISH, such as NUMBER, STRING, MATRIX, or LIST. An attribute has a value, or is said to be in a given "state," when there is a particular instance of the attribute. For example, an attribute with an intrinsic value of type NUMBER would have a numerical value specified. The lowest level class in the hierarchy has attributes that are all of the intrinsic type. Many classes have attributes that correspond to operations performed on instances of those classes. These classes are used to view operations, display the state of an object, or represent data corresponding to an object state such as a geometric model, schematic, or table of values. If different objects share common attributes, they are organized into "base classes."

As the project model is evolved, more and more attribute values are defined. An "instantiated" class means all of the object instances have values set for the prescribed attributes. A "project instance" has a specific instantiation throughout the model; in other words, all of the prescribed attributes in every class have specific values.

Each attribute has a corresponding Requirements Object (RO). ROs capture the definition and use of the associated object in a data dictionary. They also identify the kinds of constraints or rules related to the attribute. Most importantly, ROs provide the interface to the rules or constraints applied to the object's attribute. ROs form the "glue," or object associations, between the different classes in the project model. This allows the user to modify the constraints and many of the rules used to compute attribute values without evolving to a new project version.

To ground the design in reality and provide a high degree of detail, lists of real hardware or components are stored in Catalog Objects. Most classes that represent physical entities, like satellites or ground stations, include a Catalog Object. Continuing with the space systems example from above, the spacecraft class could contain a Catalog Object with attributes for batteries, solar cells, reaction wheels, structural members, sensors, etc. In addition to providing great fidelity, data on real components allows the user to evaluate "make or buy" decisions and allows comparison of the commercial off the shelf (COTS) option with the customized or optimized version of the same component designed in GENSAT.

Another important object in the project model is the BUDGET class. The BUDGET class is an excellent example of the power and flexibility of the GENSAT system. BUDGET objects have used defined attributes created in FISH using its intrinsic data types. The attributes consist of labeled, two-dimensional arrays resembling spreadsheets. They allow users to assemble many different kinds of budgets, such as weight, mass, propellant, cost, etc. However, BUDGET objects are far more powerful than a static or even dynamic list of elements. When the value of a BUDGET attribute is modified, it causes the system to execute any set of operations including analysis or optimization computations at any level of depth necessary to update the attributes of interest. Combined with the parametric analysis functions available within GENSAT, BUDGET objects are a perfect utility for performing system-level parametric trade studies. Of course, the other features of FISH also apply to the BUDGET class, so constraints are automatically checked and all operations are performed optimally.

Most classes also include STRUCTURE attributes, since nearly all solid components have a structural aspect. The STRUCTURE attributes provide a tremendous amount of depth and sophistication to the GENSAT system. Any object in the project model that could be subject to structural analysis includes a "StructureView" operation. By changing an attribute value, the user can change the structural size or material properties and observe the impacts on the static, dynamic, and vibration responses. STRUCTURE objects contain detail down to individual beams and plates, enabling designers to model cross-sections and material properties. Finite element analyses can be automatically performed on any assembly or subassembly of components selected from throughout the project model, providing very accurate weight, strength, stiffness, and frequency information. Combined with the parametric analysis capabilities, the size, material, structural configuration, or section properties of individual components can be varied in any combination to minimize the mass while maintaining structural integrity of the member. Optimization can also be performed at the system level. The user specifies which beams and plates are considered design variables, and GENSAT will minimize the weight while still enforcing all stress, deflection, and frequency constraints.

In addition to the technical design capabilities, the project model also improves the cost modeling and estimation process. Traditional cost models are based on crude initial estimates obtained from parametric scaling relations, such as dollars per kilogram, or use advanced statistical estimation tools. Current sophisticated cost studies rely on a Cost Breakdown Structure (CBS) coupled with the parametric statistical tools. The project model is a highly detailed CBS that provides cost estimates or actual, hard costs that can be accumulated from any level of depth. Since the GENSAT system completely integrates all data sources, the cost model can be tied directly into a company's financial accounting system. The analysis tools and statistical estimation functions provided by the GENSAT environment can be used to run powerful queries and simulations, facilitating the comparison of outputs from competing cost models. These capabilities allow the user to estimate the life cycle costs of the system and develop cost budgets for all phases of the design process, from conceptual, preliminary, and detailed design through integration, test, and manufacturing. Trade studies and optimization functions can identify important system factors that reduce costs by isolating key drivers of cost or other performance measures. Most importantly, the user can observe the impact that changes made locally at the component or subsystem level have on the overall project.

GenStar is a general-purpose conceptual project model built using SEAS. The project model is the critical component in the GENSAT system, yet is complex and time consuming to construct. Therefore, GenStar is provided as a template that can be rapidly customized to meet the needs of any space program and serves as a training model demonstrating how to build a project system. It is built in a generic fashion, with an open class structure, generic class definitions, and general-purpose rules. GenStar is based on Loral's GlobalStar Communications Satellite System, an actual 48-satellite LEO constellation program.

Given its power and complexity, it is not surprising that the GENSAT system is expensive, but it can provide dramatic benefits. GENSAT costs about a hundred thousand dollars for a single server with five seats. That cost provides the SEAS environment, GUI, and several pre-integrated software design applications. Beyond the standard set, additional optional applications are also pre-integrated, but further increase the price. However, despite the costs, GENSAT offers the potential of significant savings. CTek estimates GENSAT will reduce project costs and shorten the design cycle by 20% to 40%, while improving product quality, reliability, and manufacturability (CTek website, 1999). On an actual, five year $200 million spacecraft engineering project, Dr. Ronald Dotson, a Spacecraft Engineer for Lockheed-Martin Corporation, estimates GENSAT shaved 17% to 28% off the development time and saved between 37 and 59 million dollars (CTek website, 1998). Clearly, for multi-hundred million dollar programs, the advantages provided by GENSAT can outweigh the costs.

GENSAT is an extremely impressive and powerful system with a wealth of features and capabilities. By its very nature, the open software architecture is highly flexible and adaptable. Objects can be formed into arbitrarily complex structures to create detailed project models of limitless fidelity, producing extremely accurate results. The complexity and power of the GENSAT system make it somewhat difficult to use. The interactive GUI, object oriented FISH language, and automated database functions help ease the burden on the design engineer. In addition, the existing applications and legacy tools are seamlessly incorporated into the SEAS structure, allowing the user to continue working in a familiar environment. However, to take full advantage of GENSAT's capabilities, users still need an extensive amount of training and experience.

The GENSAT environment fully integrates the different aspects and phases of the design process. Using GenStar, the project design can be quickly evolved not only over the different design phases, but can be extended into test, manufacturing, and deployment. Multiple project versions can exist simultaneously, facilitating concurrent engineering. The project model addresses factors beyond the design and performance, including cost, schedule, and risk. With the object oriented approach, virtually any factor of interest can be incorporated into the model as an attribute. GENSAT capabilities apply equally to all objects, providing extensive cost estimation and modeling capabilities. In addition, the cost model can be linked directly to the financial accounting system, ensuring cost data is current.

The core capabilities of the GENSAT environment are its analysis, simulation, and optimization operations. Trade studies are easily performed, with the effects of any changes clearly demonstrated by applying them uniformly and consistently across the model. A configuration manager automatically enforces constraints, ensuring design consistency and feasibility. Beyond simply capturing the design, the project model provides a powerful simulation capability giving the user insight into the response and performance of the system. Optimization can be performed locally on individual components or subsystems, or globally across the entire project model.

GENSAT is inherently upgradable. FISH was designed to integrate distinct applications and tools into one coherent system. Tools are routinely incorporated as needed to perform an analysis or operation. Advanced technologies can be added in many ways. For example, new components can be added as a Catalog Object, defined as an object with appropriate attribute values, or modeled as part of the overall project model.

GENSAT will be an excellent addition to the NPS Space Systems program. From its inception, GENSAT was defined and developed by garnering the support of professionals in the space industry. CTek is very open and responsive to inquires from NPS representatives, and is eager to work with companies and academic institutions. Recently, CTek and NPS have formed a partnership, providing eight seats in the GENSAT system to the school. As a state-of-the-art integrated design tool, it will place NPS at the forefront of this relatively new and important design technology. Since it is marketed as a commercial product, CTek provides 24-hour support for GENSAT.

Access to GENSAT will greatly benefit the students. The various applications could be folded into the appropriate class and taught throughout the two and a half year program. The modeling, analysis, and trade study capabilities are a perfect match to both the individual and group design projects. Students could see how well their designs would perform through the simulation capabilities, providing invaluable feedback. The operations curriculum could use GENSAT to design system architectures or to perform operational analyses. However, as with the other surveyed tools, faculty members should first learn how to use the GENSAT system. A faculty member should also be responsible for maintaining the Catalog Objects, although the students could obtain the information as part of their industry surveys during the final design project.

C. SUMMARY

While the various companies and organizations followed different approaches, all of the resulting design tools, except for the SMAD software, provided an amazingly common set of features and capabilities. First and foremost, the design centers emphasized the importance of the design engineer and subsystem experts in the development process. Ownership of the subsystem design, the satellite model, and the design tool itself were kept in the hands of the engineer. Instead of replacing designers, the tools augmented their role by providing a software environment, computational facilities, and automation of information and configuration management. To facilitate trade studies and collaboration between engineering disciplines, the tools were fully integrated, propagating changes throughout the design. Each tools provided some form of automated systems engineering to ensure interface compatibility and design feasibility. The all targeted the conceptual and preliminary design phase, and most could be applied across the full spectrum of design and into manufacturing and deployment. Besides the technical design and performance, the tools addressed cost, schedule, risk, and reliability of the spacecraft. Related systems, such as the ground station or launch vehicle could also be included. With the rapid advancement of space technology, the tools were very flexible and adaptable. New space technologies were readily incorporated and each tool was easily modified or upgraded. Finally, every tool provided the capability to include real components in the design.

Regardless of the approach, all of the design tools have been tremendously successful and provided a wealth of benefits. All of the companies reported dramatic cost and schedule savings while improving the quality, detail, and fidelity of the design. Cost savings were typically in the range of 30% to 50%. Schedule savings were even more dramatic, with reductions of up to 80% or even 90%. The design tools slashed development schedules from months to weeks or even days. The stand-alone tools also provided one key benefit over those of the distributed spreadsheets – insight into the performance of the design. The stand-alone tools included powerful optimization and simulation capabilities not possible in a spreadsheet.

The survey revealed a number of important characteristics of a good spacecraft design tool. First of all, it must be user friendly. No matter how powerful the tool may be, it is still useless if the engineers will not use it. It should be fully integrated and provide connectivity between each subsystem design in the spacecraft. To facilitate trade studies, essential data should be automatically transferred between the individual components and subsystems. The tools should incorporate some form of automated systems management to impose interface constraints. It should also be possible to evolve the design, refining it to progressively lower levels of detail. To accommodate the explosive rate of change in space technology, the tools must be flexible and adaptable, with new satellite components and model features easily incorporated. Finally and most importantly, the ultimate decision authority must reside with the design engineer or subsystem expert. The user must have the ability to modify or override the automatic results from the design tool. To the maximum extent possible, these essential characteristics were incorporated into the design tool developed as part of this thesis and presented in the following chapter.

III. SPACECRAFT INTEGRATED PRELIMINARY DESIGN TOOL

The preceding survey revealed a wide variety of integrated design tools. Since they were developed by large aerospace companies, the design tools are aimed at enabling concurrent engineering and collaborative design by a development team. Only one tool, the SMAD software, is intended for a single person, such as an engineering manager or student, to use for a quick preliminary design or trade study. The SMAD software's simplicity and lack of integration and data sharing makes it less than suitable to bound a prospective satellite project. On the other hand, the other tools are overly complex for these purposes. The conclusion was that no integrated design tool currently exists for a single person to use to perform a top-level trade study or preliminary design. Therefore, the aim of this thesis is to develop one.

Even though it is developed for a single user, the new design tool should incorporate as many of the essential characteristics revealed by the survey. It should be very user friendly and easy to use, and address all subsystems and aspects of a spacecraft design. It should be fully integrated and automatically transfer data between subsystems, yet enforce interface constraints. Flexibility and adaptability should be designed into the new tool. Finally, the lone user must have complete control over the calculations and results.

The first step was selecting the appropriate software vehicle. Stand-alone tools are very powerful, but unless they are based on a complex open-systems environment, they lack adaptability. They are also difficult to maintain and upgrade. If the design tool code is publicly available, only programmers experienced in the selected computer language would be able to easily modify or update the code. On the other hand, spreadsheets are widely available and widely used. Most students, engineers, and technical managers are already familiar with them. Therefore, spreadsheets were selected as the best software medium. This immediately provided the inherent benefits discussed on page 7.

The developed design tool is very user friendly and easy to use. A user can open the spreadsheet and immediately begin customizing a spacecraft design through a clear and intuitive graphical interface. In fact, the interface is the spreadsheet itself, which is a familiar and comfortable environment to most technical and non-technical professionals. Color-coded input and output cells are coherently organized and clearly labeled with titles and units. Input cells are light orange, while output cells are light blue. To protect against inadvertent overwrite of calculation cells, and to maintain the integrity of the equations, the worksheet is locked. Data can only be entered into designated input cells. However, the sheets were locked without a password, so they can be unlocked if necessary.

A spacecraft design requires an extensive amount of rather detailed information. The design tool therefore provides a number of utilities and features to assist the user in selecting appropriate values and finding or correcting erroneous entries. All input values are checked to ensure they are valid and numeric. For example, a spacecraft component cannot have a negative mass. Error messages are displayed notifying the user of the mistake and providing insight into the source. Any non-numeric inputs are simply ignored and do not generate error messages. Many inputs are also checked to make sure they fall with typical or reasonable bounds. For example, solar cells do not provide an efficiency of 50%. If the value exceeds a normal range, warning messages are displayed to alert the user the entered information many not be correct or suitable. As values are entered, all other effected values are recomputed and displayed. The effect of any change is immediately apparent. Calculations stop for error flags, but proceed with warnings. Finally, to ease the information burden on the user, default values are automatically provided for all entries. The user can quickly complete a design by entering only a few values, confident that all of the other data is typical and representative of the current state of technology.

Individual pages perform the calculations for a particular subsystem or design aspect. In all, there are seven sheets: General, Orbit, Delta V, Propellant, EPS, Thermal, and Mass. The individual sheets are fully integrated, with all necessary data automatically transferred. Data dependencies are discussed at the beginning of the subsection on each page. If erroneous information is passed, the corresponding flags would be out of sight on the originating page. Therefore, each page also flags if it receives bad data, and points to the offending sheet. Constraints and interfaces are enforced by only permitting shared data to be modified on the originating page and not on any subsequent, receiving pages. This protects design integrity and prevents the design from getting "out of sync." All key results or transferred information that are computed by the spreadsheet have a corresponding input cell that overrides the calculated value. This guarantees decision authority remains in the hands of the user, where it belongs. If the user does not want to consider a particular aspect of the design, it can be zeroed out. Since the design tool automatically recomputes effected values and fully shares all necessary data, trade studies can be performed easily. The user can change one value, then observe the impacts elsewhere within that sheet or on subsequent pages. Unfortunately, spreadsheets are not good for automatic optimization. However, the ease of performing trade studies can be used as a form of "manual" optimization.

Spreadsheets are inherently flexible and adaptable, and these attributes are passed on to the design tool. Equations can be modified or updated by simply unlocking the worksheet and entering a new expression. New calculations can also be added and linked in to the rest of the design or displayed off on the side. All equations and calculations, the "design code" of the spreadsheet, are placed at the bottom of each page and, while out of the way, are readily accessible to the user. In addition, a methodology was specifically adopted to make the code as flexible and easily updated as possible. A consistent form and flow of calculations was followed to the maximum extent possible. Calculation cells are labeled and reasonably organized. Default values are entered in labeled cells instead of hard-coded into the equations. Therefore, they can be quickly updated as the state of technology progresses.

In addition to easing the information burden on the user, the default values form a complete spacecraft design. The default values were selected to demonstrate the features and capabilities of the design tool and represent the current state of satellite technology, but they also illustrate a sample design and the design process. Collectively, the default values design a geostationary communications satellite. The spacecraft has a 7 year design life and is integrated in a 2 m cubic bus, like the Hughes 601-series standard buses. It was launched from Cape Kennedy into a geosynchronous transfer orbit (GTO) with a 5.3 hour time of flight. An apogee kick motor places 1,010 kg of the 2,500 kg separation mass in the final geosynchronous orbit. The transfer orbit is spin stabilized at 45 revolutions per minute (rpm), but once on-orbit the attitude control system provides three-axis stabilization. The satellite is deployed to the 210.3° East longitude station, providing the worst case estimates for stationkeeping power while still providing coverage of the mainland United States. The propellent budget allows for a single repositioning of 180° in 30 days. Attitude control and orbit maintenance are provided with a bipropellant reaction control system, using a 50%-50% mixture, by volume, of monomethyl hydrazine (MMH) and nitrogen tetroxide (N_2O_4). The 110.2 kg payload draws 1,000 W over the entire orbit, both sunlit and eclipse. Housekeeping required another 130 W. A partially regulated dual power bus provides the required power with Gallium Arsenide (GaAs) solar cells with an efficiency of 18%. The cells are grouped into two single-gimbled wings of two 1.7 m x 2.0 m panels, for a total array area of 13.6 m^2. Energy during the eclipse period is provided by two 84-cell cell batteries with a 14.3 Amp-hr capacity. The payload transmits half of the 1,000 W into the communications channel. An augmented passive thermal control system maintains the temperature between 10° C and 40° C by rejecting the waste heat through a 4 m^2 radiator coated with optical solar reflector material. This proposed design still has a 10.9% mass margin, a reasonable value for preliminary design.

The design tool was developed in Microsoft® Excel 97, a widely-available spreadsheet program. To fully use the design tool, the user must load an Excel add-in that is provided with Excel but not normally pre-installed. The Delta V sheet uses modified Bessel functions, which are part of the Analysis ToolPak. To install the add-in, select Tools on the menu bar, and choose Add-Ins.... In the Add-Ins... window, check the box next to the Analysis ToolPak, and click the OK button to accept and install the add-in. Installation can be verified by clicking on the function button on the tool bar, selecting All under the Function category, and then scrolling down the Function name window to the "b"s to find the Bessel functions.

To be fair, the design tool should be compared to the same criteria used to evaluate the surveyed tools. As previously discussed, it is very flexible, adaptable, and upgradable by virtue of the spreadsheet environment and by design. New features and calculations can be added easily. Since default values are not hard-coded, new spacecraft technologies are readily incorporated. For preliminary design, the design tool is reasonably powerful with good fidelity. The tool was designed to be generally applicable, with limiting assumptions minimized. Although calculations are based on first principles and heuristics, the results are quite accurate. The individual sheets are fully integrated and automatically share all necessary data. The tool is great for trade studies, but does not optimize. It provides a limited ability to track and evolve the design by entering refined numeric values. However, the refined values must be computed in separate applications. Factors beyond the technical design, like cost, schedule, or risk, are not addressed but could be incorporated. This design tool is an excellent compliment to the NPS curriculum. The ability to perform rapid trade studies will greatly benefit the final individual design project. In addition, the tool's inherent flexibility will allow the students to quickly mold the tool into whatever they need.

The remainder of this chapter provides a brief tutorial on spacecraft design and serves as a Users Manual. All input and output parameters are listed and explained, along with typical ranges of values. The chapter also documents all of the essential equations used in the design tool and defines the associated variables.

A. GENERAL

The General sheet accepts information on the overall spacecraft parameters, namely the design life, separation mass, size, and shape. From this information, default moments of inertia (MOIs) are computed assuming a homogeneous, uniform mass distribution. A printout of the sheet is provided in Appendix B on page 143.

The design life is entered in years and can be any number greater than zero. Satellites placed in GEO are typically designed to last between 10 and 15 years, but frequently continue operating for well over 20 years. Because of the increased eclipse and thermal cycles, LEO satellites have a much shorter life and are usually designed for less than 5 years. The design life is also strongly influenced by the specific orbital altitude due to the ambient radiation levels. Satellites which pass though the van Allen radiation belts have much shorter lives. This is the primary reason why LEO orbits are kept below an altitude of 1,000 km. GEO is, of course, well outside the van Allen belts, allowing the longer design life. The default value is 7 years, which splits the difference between the nominal LEO and GEO time.

The separation mass can also be any number greater than zero. It is highly mission dependent. GEO satellites tend to be larger and more massive than LEO satellites and can reach as much as 3,000 to 4,000 kg. On the opposite end of the spectrum, some scientific and specialized satellites have masses of less than 100 kg. Technological advances in miniaturization and packaging are reducing the mass of all spacecraft while increasing capability. As an outgrowth of this technology, in the last few years there have been some serious studies and preliminary designs for micro- and nanosatellites, targeting masses of as little as 20 kg while still performing a useful mission. The separation mass is also strongly influenced by the launch vehicle throw-weight capability. Launch costs usually represent a large portion of the total budget for most satellite programs. To reduce the costs, programs use as small a launch vehicle as possible. Frequently, the launch vehicle is selected in advance and the satellite is designed to fit within its capabilities. This places a limit on the separation mass.

The user can select one of three spacecraft shapes: spherical, circular cylinder, or rectangular cylinder. Note that cubes are merely a special case of rectangular cylinders. Tumbling spacecraft, those with no attitude control, are typically spherical, both because of mass properties and so the shape itself has no preferred direction. Most spin stabilized spacecraft are circular cylinders, while rectangular satellites are normally 3-axis stabilized. Most other spacecraft shapes can be reasonably approximated by one of these three basic shapes.

Entries for the spacecraft size depend on the selected shape. For spherical satellites, the user only enters one value for the diameter. The diameter plus the height must be entered for circular cylinders. The spreadsheet assumes the axis of revolution is the spacecraft z-axis. Finally, for rectangular cylinders, the length, width, and height must be specified for the x-, y-, and z-axes, respectively. The labels for the entry lines change accordingly for the selected shape, and any extra entry lines are blanked. For example, if the user selects "spherical," the bottom two entry cells are blanked. Any inputs into the unnecessary lines are ignored. Spacecraft are normally sized to be just large enough to contain the required components. Empty spaces require a larger structure, increasing its mass. The size of the spacecraft is also limited by the launch vehicle shroud. As discussed above, the launch vehicle is frequently selected in advance and the satellite is designed to fit. To help prevent unreasonable or unrealistic dimensions, a warning is displayed if an entered diameter, length, or width exceeds 4.6 m or if an entered height exceeds 18.5 m. If the dimensions are greater than these values, the satellite is too large for virtually every current launch vehicle.

To help the user enter appropriate values, a table identifying launch vehicle capabilities is included at the bottom of the sheet. The table lists the shroud size and throw-weight to LEO, GEO, and geosynchronous transfer orbit (GTO) for many common launch vehicles.

From this information, default MOIs about the shape centroid are computed. The calculations assume the separation mass is uniformly distributed over the spacecraft volume. For a sphere, the MOI is given by:

$$\mathrm{MOI} = \frac{2}{5}\mathrm{mR}^2 \tag{3.1}$$

For circular cylinders, the MOI about the spin axis (z-axis) is:

$$\mathrm{MOI} = \frac{1}{2}\mathrm{mR}^2. \tag{3.2}$$

The MOI for the other two axes (x- and y-axes) is:

$$\mathrm{MOI} = \frac{m}{12}\left(3R^2 + H^2\right). \tag{3.3}$$

Finally, the MOIs for a rectangular cylinder is found from:

$$\mathrm{MOI} = \frac{m}{12}\left(A^2 + B^2\right). \tag{3.4}$$

In the above MOI equations, m is the spacecraft mass, R is the radius, H is the height, and A and B are the length, width, or height, as appropriate. For example, to compute the MOI for the x-axis, A and B would be the width and height. The default MOIs are based on the separation mass. However, actual MOIs should be calculated based on the spacecraft mass at the time of interest. Therefore, the user may want to compute MOIs for the dry mass or the on-orbit mass. (Larson and Wertz, 1992, p. 452)

B. ORBIT

The Orbit sheet accepts inputs on the initial orbit into which the spacecraft is launched and on the final mission orbit. If necessary, an intermediate transfer orbit between those two is automatically computed. The sheet calculates the orbit period or transfer orbit time of flight and the inclination for sun synchronous orbits. For GEO and geostationary Earth orbits (GSO), the longitude station is also entered. A printout of the sheet is included on page 144 of Appendix B.

If the user intends for the launch vehicle to place the spacecraft into the final mission orbit, the "direct insertion" button at the top of the sheet should be selected. Only data on the final orbit is necessary and the initial and intermediate orbits are blanked out. On the other hand, if the satellite is first placed into an initial parking orbit or is launched into a transfer orbit the user should choose the "not direct insertion" button. Since it is very common to launch into parking orbits or GTO, the latter option is the default. In this case, information on both the initial and final orbits is required.

To specify the initial or final orbits, the user enters the inclination and one of three pairs of numbers which determine the orbit size and shape: 1) the semi-major axis and eccentricity, 2) the perigee and apogee altitude, or 3) the perigee and apogee radii. Entering any one pair uniquely determines the rest, which are computed and displayed. See Vallado, pages 130-132, or Agrawal, pages 64-67, for development of the equations. If values are entered into more than one pair, they are checked for consistency with error messages given as appropriate. For example, if a perigee and apogee altitude of 1,000 km is entered, giving a circular low Earth orbit, the user cannot enter an eccentricity of 0.2 or an apogee radius of 6,878 km; this data set is inconsistent. Default values for the orbits are based on launch from Kennedy Space Center into a circular LEO parking orbit transferring to a final GSO mission orbit.

For GEO and GSO orbits, the user must identify the longitude of the subsatellite point. The longitude station is necessary to calculate the longitudinal (East-West) stationkeeping requirements on the Delta V sheet. As will be discussed further in the description of the delta V calculations, the drift acceleration varies around the GEO ring depending on the satellite's angular distance from one of the stable longitude positions at 75.3° E or 255.3° E (104.7° W). To provide a conservative estimate, the default value assumes the worst case longitude station over the United States, at 210.3° E (149.7° W). If the final orbit is not GEO or GSO, the longitude station entry line is blanked.

To be sun-synchronous, the final orbit's nodal precession rate must match the Earth's rotation rate about the sun. The Earth completes one revolution in 365.25 days, giving a rotation rate of 0.9865°/day. Note that the rotation rate is positive, so the orbit must be retrograde, in other words have an inclination greater than 90°. Since the nodal precession rate is a function of the inclination and semi-major axis, an orbit with a given radius must have a specific inclination to be sun synchronous. This inclination is found from (NPS, p. 6):

$$\cos(i) = \frac{2\,\dot{\Omega}\,a^{7/2}\,(1-e^2)^2}{3\,R_e^2\,J_2\,\sqrt{\mu}} \qquad (3.5)$$

where $\dot{\Omega} = 1.9910\text{x}10^{-7}$ rad/sec (0.9865°/day) is the nodal regression rate, $J_2 = 1.08263\text{x}10^{-3}$ is harmonic of the Earth's budge, $R_e = 6{,}378.1363$ km is the radius of the Earth, and a, e, and i have their usual meanings of semi-major axis, eccentricity, and inclination, respectively. To help the user, the sun synchronous inclination is calculated and displayed for LEO orbits, i.e. orbit with less than 2,000 km altitude, immediately below the entry line. However, this value is not automatically used in subsequent calculations. If a sun synchronous orbit is wanted, the user must enter the computed value into the sheet as the final orbit's inclination.

With the initial and final orbits correctly specified, the transfer orbit is automatically determined, if necessary. The apogee radii of the initial and final orbits are compared. If they are the same, then the initial orbit is the transfer orbit and the intermediate orbit is blanked out. If they differ, the sheet automatically calculates the intermediate orbit necessary to implement a standard Hohmann transfer. The perigee radius and apogee radius are set equal to those for the initial and final orbits, respectively. The remaining orbital parameters are then computed from these two values. The remaining question for the transfer orbit is where to perform any inclination changes.

Inclination changes require a large velocity change compared to a simple coplanar Hohmann transfer. Even if the maneuvers are combined, the inclination change still drives the required velocity change. Therefore, to keep the total velocity change as small as possible it is important to determine the best inclination change for each maneuver. An excellent algorithm for estimating the optimum inclination changes is developed in Vallado, pages 306-310:

$$\Delta i_{\text{initial}} = s\,\Delta i \qquad (3.6a)$$

$$\Delta i_{\text{final}} = (1-s)\Delta i \qquad (3.6b)$$

$$s = \frac{1}{\Delta i}\tan^{-1}\left[\frac{\sin(\Delta i)}{(r_{\text{final}} / r_{\text{initial}})^{3/2} + \cos(\Delta i)}\right] \qquad (3.6c)$$

where Δi is the total inclination change, $\Delta i_{\text{initial}}$ is the inclination change at the first maneuver, Δi_{final} is the inclination change at the second maneuver, s is a scaling term, r_{initial} is the radius of the initial orbit, and r_{final} is the radius of the final orbit. The spreadsheet estimates the best inclination changes at each maneuver, which are automatically carried forward to subsequent calculations unless the user overrides it. For example, if the user enters a zero for the percent change at the first maneuver, the entire inclination change is performed at apogee.

In addition to the consistency checks, the sheet performs error and validity checks on entered values. For example, the sheet checks to ensure the semi-major axis is greater than the radius of the Earth, the perigee altitude is greater than zero, the apogee altitude or radius is greater than perigee, and eccentricity is between $0 \le e < 1$ for Earth orbits. The longitude station must be between 0° and 360° if given in East longitudes or between 0° and 180° West. If the final orbit perigee altitude is below 200 km, resulting in very high drag and therefore a short mission life, a warning is given to make sure the user really intended to enter that value. This is particularly useful when the perigee altitude is not explicitly entered but is computed from the entered values for the semi-major axis and eccentricity.

For an additional verification of the entered information, flags identifying the orbit and inclination type are provided. Orbits are identified as LEO, medium Earth orbit (MEO), GEO, GSO, Supersync, or Molniya. An orbit is a LEO if the semi-major axis is less than 2,000 km altitude. If the semi-major axis is between 42,100 and 42,220 km, it is considered to be GEO. A circular (eccentricity less than 0.001) GEO with an inclination less than 0.1° is GSO. An orbit that falls between LEO and GEO is identified as MEO while any orbit beyond GEO/GSO is Supersync. The specialized Molniya orbit has very specific parameters. It has a period of 12 siderial hours with a semi-major axis of 26,610 km, an eccentricity between 0.70 and 0.75, and has a 63.4° inclination. Inclinations are identified as either polar or retrograde. A polar orbit has an inclination between 88° and 90°. Any inclination over 90° is flagged as retrograde. The standard, prograde orbit is not specifically identified or flagged.

The inclination of the initial orbit is limited by the launch site. From simple geometry, the minimum inclination a particular launch site can achieve is determined by the site's latitude. Range constraints can impose further limits on the achievable inclinations. As an aid in determining the inclination of the initial orbit, a table of the maximum and minimum inclinations for various launch sites around the world is given at the bottom of the Orbit sheet. The table is included in the spreadsheet printout in Appendix B. Data for this table was derived from Larson and Wertz, pages 680-681 and Vallado, pages 296-297.

An interesting observation can be made from the table. Note that to achieve the minimum inclination, the launch must be due East. However, if range constraints prohibit a 90° launch azimuth, then the minimum inclination cannot be achieved. Several sites have this limitation. To launch into polar orbits, the azimuth range must include either 0° (due North) or 180° (due South). Retrograde orbits require launch azimuths between 180° and 360°. For example, at a latitude of 34.6°, Vandenberg can launch into azimuths between 147° and 201°. Since the azimuth range does not include 90°, Vandenberg cannot launch into orbits inclined 34.6°. It can launch into polar and retrograde orbits since the azimuth range includes 180° to 201°. As a result of the range constraints, the minimum prograde inclination which can be achieved from Vandenberg is limited to 63.36°, permitting launches into Molniya orbits.

C. DELTA V

The Delta V sheet determines the total velocity change necessary over the mission duration, including orbital maneuver from the initial to the final orbit, stationkeeping, repositioning, and disposal at the end of the satellite's useful life. The propellant mass to control the spin rate during orbit transfer and to orient the thrust vector for perigee and/or apogee motor firing is also calculated. Stationkeeping requirements include corrections for inclination (North-South) and longitude (East-West) drift in GEO and for drag in LEO. To dispose of the satellite, the user can choose to either deorbit the satellite or boost it into a disposal orbit. The sheet is display on page 147 of Appendix B.

An additional delta V can be entered directly, either to meet unique mission demands or correct for perturbations not addressed in the sheet. Velocity requirements to adjust for the unsymmetrical Earth and third-body effects are only computed for GEO. For satellites in LEO and MEO, these perturbations are typically uncorrected and can require very large delta Vs, which would distort the velocity requirement and thereby the propellant budget. However, if the user needs to compensate for these perturbations the necessary equations are provided at the end of this section.

Information from the General, Orbit, Propulsion, and EPS sheets is used in the computations. In one equation or another, all of the General sheet's data is used. The Orbit information is used extensively throughout most of the velocity calculations. The spacecraft on-orbit mass is imported from the Propellant sheet and is used in the atmospheric drag calculations. Atmospheric drag also used the solar array area from EPS. Errors in these sheets may cause subsequent errors in the Delta V sheet.

1. Additional Delta V

The Additional Delta V is a simple, direct number entry. The only restriction is that the entered value must be greater than or equal to zero. It provides the means to increase the total delta V for any and all requirements not computed elsewhere in the sheet.

Other velocity changes may arise from several sources. Unique mission requirements may dictate extra maneuvering, rendezvous, or space debris avoidance. Military satellites might have the need to evade anti-satellite weapons. Another possibility is additional stationkeeping requirements to compensate for other perturbations. Third body effects from the sun, moon, and other heavenly bodies and accelerations due to gravity variations caused by the nonuniform Earth, the so called J_2 effects, are not included in the spreadsheet calculations for satellites in LEO and MEO. In these orbital altitudes, these preturbations can drive very large velocity requirements but are typically left uncorrected. If the user needs to account for them, the necessary equations are included at the end of this section. The Additional Delta V entry line gives the flexibility to include these items in the preliminary design.

2. Orbital Transfer

The Orbital Transfer section calculates the delta V to maneuver the spacecraft into the final mission orbit with either a standard Hohmann-like transfer or a low-thrust spiral transfer using an electric propulsion (EP) system. Note that it is unnecessary to enter zeros into one of the two transfer methods. This sheet only calculates the change in velocity for the two transfer types. The method used in the design estimate is then selected on the Propellant sheet by choosing the appropriate propulsion subsystem. For example, to use the EP spiral transfer, the user does not need to enter zeros into the first and second maneuver delta V's. Simply go to the Propellant sheet and select the "EP" button on the "First Maneuver" line.

The Hohmann-like maneuver transfers the satellite from perigee of the initial parking orbit to the final orbit apogee. If the apogees of the initial and final orbits are equal, the initial orbit is the transfer orbit. In this case, only one maneuver is necessary and the second delta V is zero. Of course, for a direct insertion neither maneuver is necessary so both delta Vs are zero. The calculated values assume combined maneuvers, changing the size, shape and inclination. It further assumes that perigee and apogee, and thereby the maneuvers, occur at nodal crossings (i.e. the argument of perigee is 0° or 180°). Equations to calculate orbital velocities and velocity changes can be found in virtually every orbital mechanics or spacecraft textbook. A good development is provided by Larson and Wertz, pages 144 - 151, or Vallado, pages 275 - 281 and 306 - 308. The Delta V sheet finds the velocity increment for the combined maneuver from:

$$\Delta v = \sqrt{v_2^{\,2} + v_1^{\,2} - 2\, v_1 v_2 \cos(\Delta i)} \qquad (3.7)$$

where v_1 is the satellite velocity just prior to the maneuver, v_2 is the new velocity just after the maneuver, and Δi is the inclination change for that maneuver.

For the spiral transfer, the EP system produces a continuous, low-level thrust to gradually change the orbit size, shape and inclination. Since it does not use two discrete, impulsive maneuvers, the velocity change cannot be computed using the Hohmann method. Instead, the delta V is simply estimated by the difference between the velocity of the two orbits (Larson and Wertz, 1992, p. 147):

$$\Delta v = \left| v_2 - v_1 \right| \tag{3.8}$$

where v_1 and v_2 have the same meanings as above. For elliptic orbits, where the velocity varies throughout the orbit, this estimate is based on the mean orbital velocity. Note that because of the low thrust level, spiral transfers take many revolutions over days, weeks, or even months. The transfer time of flight cited in the Orbit sheet only applies to the Hohmann transfer.

The default values are computed from the orbital parameters entered in the Orbit sheet. Therefore, the user should enter different numbers with caution. The defaults are the actual delta Vs necessary to perform the maneuvers. To prevent accidental entry, warning messages appear if the entered number differs from the computed values. The only other error message for the Orbital Transfer data is if the user enters a negative delta V in any of the entry lines.

3. Reorientation and Spin Control during Transfer

Frequently, spacecraft are spin stabilized during the transfer orbit even if they will be 3-axis stabilized once on station in the final mission orbit. This section allows the user to select the stabilization method during the transfer orbit independently from the attitude control method. If the user elects to spin stabilize the spacecraft during the transfer orbit, the requirements for three maneuvers are computed: initial spin up, reorientation of the thrust vector, and spin down for final orientation into the mission attitude. If 3-axis stabilization is chosen, the final spin rate must be zero. If the user enters a non-zero spin rate, a warning will be provided. Very little torque is needed to reorient a non-spinning spacecraft, so the spin control and reorientation requirements are zero. In addition to the maneuvers, the requirement for nutation or attitude control is estimated for either stabilization method. Of course, if Direct Insertion is selected in the Orbit sheet, there are no requirements on the spacecraft during the transfer orbit. In this case, this section is unnecessary and the entries are blanked.

Pure torques are used to control the spin rate and to reorient the spin axis. This section assumes the propulsion subsystem is used to impart the torques. Since no delta V is imparted to the spacecraft, the propellant mass is calculated directly. If the torques will be generated by reaction wheels, momentum wheels, or magnetic torque rods, the user must enter zeros for the propellant masses, either on the three separate lines or with a single entry in the Total Propellant Mass line. The same effect can be achieved by choosing 3-axis stabilization, which also zeros all of the propellant masses.

Computing the propellant mass for any of the three maneuvers requires the same information about the spacecraft and propulsion subsystem. For the spacecraft, the default values are based on the information entered in the General sheet. The calculations assume the spacecraft is spinning about the yaw, or z, axis, so the default MOI is the MOI_Z value. For the spherical and circular cylinder shapes, the default moment arm is simply the spacecraft radius. For rectangular spacecraft, the thrusters are assumed to be in the corners, giving the maximum possible moment arm. If the user enters a value greater than the radius or diagonal length, a message is displayed warning that the moment arm is larger than the size of the spacecraft. However, the entered value is accepted and used.

For the propulsion subsystem, the calculations depend on the number of thrusters and the thruster force, efficiency, and specific impulse. While the number of thrusters can be any positive integer, use of only one thruster will impart a net delta V to the spacecraft in addition to the torque. The net delta V will act as a disturbance, and must be corrected. In this case, an extra delta V should be entered in the Additional Delta V entry line. As used here, efficiency does not refer to the internal efficiency of the thruster; internal efficiency is part of the specific impulse. Instead, it represents the decrease in effective force due to misalignment of the thrust axis. To generate the maximum torque, the thruster force should act perpendicular to the spin axis. However, thrusters are typically canted between 5° and 10° from the perpendicular to prevent plume impingement on the spacecraft. The efficiency is found from the cosine of the cant angle. Finally, default values for the thruster force and specific impulse are based on bipropellant reaction control jets. For further guidance in selecting appropriate values, the user can refer to the table at the bottom of the Propellant sheet which lists normal ranges of specific impulse and thrust force for several types of propulsion systems. With this information, the torque is easily found as the product of the number of thrusters, thruster force, efficiency, and the moment arm.

a. Spin Up

To spin stabilize during the transfer orbit, spacecraft usually must be "spun up" to a higher rotation rate. Typical spin rates are between 40 and 60 rpm. Below 40 rpm the spin rate is too low to effectively stabilize the spacecraft. Above 60 rpm, centripetal accelerations cause excessive stress on the spacecraft structure. However, at separation, launch vehicles typically provide an angular velocity of only 5 to 10 rpm.

To compute the propellant mass, the change in angular momentum is first found for the difference in the spin rates:

$$\Delta H = I\,\Delta\omega = I\left(\omega_f - \omega_i\right) \qquad (3.9)$$

where ω_i is the spin rate at separation, ω_f is desired final spin rate, and I is the MOI about the spin axis. The propellant mass is found from the thruster firing time, which equals the time to spin up. The quantities are computed from:

$$\Delta T = \frac{I\,\Delta\omega}{\eta\, n\, F\, r} \qquad (3.10)$$

$$m_p = \frac{n\, F\, \Delta T}{I_{sp}\, g_o} \qquad (3.11)$$

where ΔT is the time to spin up (in seconds), m_p is the propellant mass, η is the efficiency, n is the number of thrusters, F is the thruster force, r is the moment arm, I_{sp} is the specific impulse, and g_o is the acceleration of gravity at the Earth's surface. Note that these calculations can also be used to compute the propellant mass to spin down the spacecraft for 3-axis stabilization if the launch vehicle imparts a spin rate at separation.

b. Reorientation

During orbital transfer, the spacecraft must be reoriented to align the thrust direction of the perigee and apogee kick motors with the required direction of the delta V. The orientation of a spinning spacecraft stays fixed in inertial space unless a moment is applied. Therefore, the propulsion subsystem must exert a torque, requiring propellant. If the spacecraft is not spin stabilized, its orientation can be changed easily and requires negligible propellant mass.

The first step in computing the propellant mass is determining the reorientation angle. For coplanar maneuvers, i.e. when there is no inclination change between the initial and final orbits, the velocity vector and direction of the delta V are aligned and no reorientation is necessary. For combined maneuvers, i.e. when the initial and final orbits have different inclinations, the reorientation angle is found from the magnitudes of the orbital velocity and delta V. Figure 3.1 depicts the geometry of the velocity vectors for the first and second maneuvers. For the first maneuver, the reorientation angle, $\Delta\theta_1$, can be found from the law of sines:

$$\frac{\sin(\Delta i_1)}{\Delta v_1} = \frac{\sin(180 - \Delta\theta_1)}{v_t}. \qquad (3.12)$$

Typically, the spacecraft is allowed to remain in the new orientation and is not realigned with the new velocity vector, which would require additional propellant mass. For the second maneuver, if necessary, the angle between velocity vector and the delta V is again found from the law of sines:

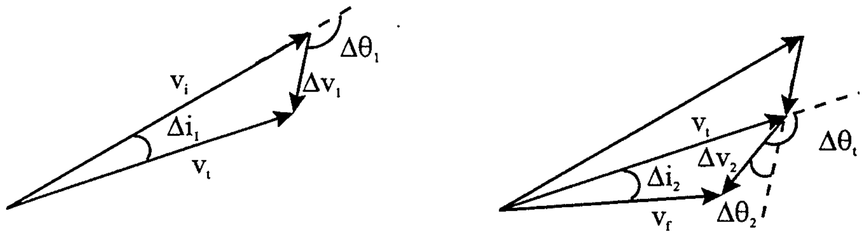
$$\frac{\sin(\Delta i_2)}{\Delta v_2} = \frac{\sin(180 - \Delta\theta_t)}{v_f}. \qquad (3.13)$$

However, the spacecraft is already at an angle to the velocity vector as a result of the first reorientation. Therefore, the second maneuver's net reorientation angle is only:

$$\Delta\theta_2 = \Delta\theta_t - \Delta\theta_1 + \Delta i_1 \qquad (3.14)$$

The Angle for the First Maneuver and Angle for the Second Maneuver are $\Delta\theta_1$ and $\Delta\theta_2$, respectively. These calculations assume the spacecraft is not realigned with the velocity vector between maneuver. If the user wants the spacecraft aligned with the velocity vector during the transfer, enter $\Delta\theta_t$, computed from Eq 3.13, as the Angle for the Second Maneuver. In addition, realigning the spacecraft will require about as much fuel as the first reorientation, so the First Maneuver Propellant Mass should be doubled.

There are two methods to reorient a spinning spacecraft: reorient while spinning or spin down to zero. The two methods will require different fuel masses. For smaller angles, reorienting the spinning spacecraft requires less propellant mass. For larger angles, it takes less fuel to spin down to zero, reorient, and then spin back up. The angle where these two methods trade off depends on the spin rate and the MOI about the spin axis.



a. First Maneuver

b. Second Maneuver

Figure 3.1 Vector Geometry of the Reorientation Angles

To calculate the reorientation propellant mass, the change in angular momentum must first be determined.

$$\Delta H = I\,\omega\,\Delta\theta \qquad (3.15)$$

For the spin down method, the propellant mass is again found from Eq 3.10 and Eq 3.11. In this case, the change in spin rate, Δω, equals the Spin Rate - Final value since the spacecraft is spun down to zero. However, the propellant mass computed from Eq 3.11 is only half the required amount since the satellite must be spun back up. The maximum change in angular momentum is:

$$\Delta H_{max} = 2\,I\,\Delta\omega \qquad (3.16)$$

To reorient the spacecraft while spinning, the propellant mass is again found from the thruster firing time. The firing time is still the change in angular momentum divided by the applied torque, however the form of the equation is slightly different than in Eq 3.10. For reorientation, the firing time is found from:

$$\Delta T = \frac{I\,\omega\,\Delta\theta}{\eta\,n\,F\,r} \qquad (3.17)$$

The propellant mass is then found from the firing time by using Eq 3.11 as before. Note that if the change in angular momentum is greater than the value from Eq 3.16, then the spin down method is used. The spreadsheet performs the calculations for both methods and automatically selects the smaller propellant mass.

c. Final Reorientation / Spin Down

Once the satellite reaches the final orbit, it must be placed in the proper orientation to perform its mission. Typically, this requires the yaw, or z, axis to be pointed to nadir, but other orientations are common especially for duel spin spacecraft and satellites in LEO orbits. In any case, it is unlikely the spacecraft will be in the proper orientation following the transfer orbit and apogee maneuver. Therefore, a final reorientation must be performed. The spreadsheet calculations assume the spin down method is used.

Finding the propellant mass to spin down for final reorientation uses exactly the same procedure as spin up. The change in angular momentum is given by Eq 3.9 and the propellant mass from Eqs 3.10 and 3.11. The only difference from the spin up calculations is the change in spin rate equals the starting spin rate, which is the Spin Rate - Final value.

d. Attitude / Nutation Control

The final propellant mass estimate in this section is for attitude or nutation control. Since they have no gyroscopic stiffness, 3-axis stabilized spacecraft can have their orientation, or attitude, easily changed by perturbations. The propulsion subsystem is usually used to control the spacecraft attitude. Spinning spacecraft face a different problem. The spacecraft is in a stable state only if it is spinning about the major MOI axis (Kaplan, 1976, p. 62-64), which is rarely the case. Over the transfer orbit, the spacecraft will tend to diverge from its initial spin axis toward the major MOI axis, causing the nutation angle to increase. The propulsion is usually used to control spacecraft nutation. The propellant mass required for attitude or nutation control depend on many unknown factors and are extremely difficult to compute. Therefore, the default values are based on historical amounts. For 3-axis stabilization, the default mass is 18 kg. Spin stabilization provides gyroscopic stiffness, and thereby some inherent stabilization, so the default mass is only 15 kg. These default values are typical for GTO, but are overestimates if the final orbit is LEO. Since the propellant mass depends on the period of time the propulsion subsystem provides attitude control, it is proportional to the transfer orbit time of flight. Thus, the default mass can be scaled by the ratio of the actual orbital transfer time to the nominal GTO time of flight of 320.2 minutes.

4. Repositioning

Frequently, satellites are repositioned within the orbital plane to adjust the coverage. Repositioning is accomplished by changing the orbital altitude. This changes the orbit's period, causing the spacecraft to drift relative to its original orbital position.

Only three inputs are needed to find the necessary propellant mass: the number of times the user wants to reposition the satellite over its mission life, the angle the satellite is moved relative to its original orbital position, and the time to accomplish the repositioning. The number of repositioning can be any positive integer. The repositioning angle must be between 0° and 180°; to reposition the satellite more than 180°, simply go the other direction. Finally, the repositioning time can have any positive value and is expressed in days. Larger angles and faster repositionings require larger changes in the orbital altitude and therefore more propellant mass.

The orbital drift rate, Δn, is determined by the repositioning angle and time. The velocity requirements are found from the drift rate from:

$$\Delta v = -\frac{\sqrt{a}\sqrt{\mu}}{3\,v}\Delta n \qquad (3.18)$$

where a is the semi-major axis, μ is the Earth's gravitational constant (= 398,600.5 km^2/sec^3), and v is the orbital velocity. In elliptical orbits, the velocity varies throughout the orbit. Therefore, the delta V requirement is estimated from the mean orbital velocity, i.e. the circular velocity, given by:

$$v = \sqrt{\frac{\mu}{a}}. \qquad (3.19)$$

Substituting Eq 3.19 into Eq 3.18 gives the equation actually used in the spreadsheet:

$$\Delta v = -\frac{a}{3}\Delta n. \qquad (3.20)$$

Default values are based on a single repositioning of the satellite through the maximum angle of 180° in one month.

5. End-of-Life Disposal

At the end of the mission life, the satellite should be removed from its orbit to prevent clutter and debris from filling the useful orbits. End-of-life disposal can be accomplished in two ways: boost the spacecraft into a disposal orbit or lower the perigee into the Earth's atmosphere to cause deorbit. Satellites in GEO are virtually always boosted up into slightly supersync disposal orbit. The GEO disposal orbit must be high-enough to prevent collisions with active satellites while they are repositioned and is typically 500 to 1,000 km above the GEO radius. LEO satellites are normally deorbited over a small number of revolutions.

The user selects the disposal method by entering one of two possible values into the lines under the desired disposal type. Note that the two disposal methods are mutually exclusive; the satellite cannot be placed in a disposal orbit and deorbited. Entering values under both methods will prompt an error message.

To specify a disposal orbit, the user can enter either the change in semi-major axis or enter its new value directly. If both values are entered, they must be consistent or an error message will be generated. For example, if a satellite in GEO is to be boosted up by 500 km, the semi-major axis cannot also be specified as 41,000 km; these values are inconsistent. The disposal orbit can be higher or lower than the original orbit, corresponding to a positive or negative change in semi-major axis. However, the disposal orbit must still be above the surface of the Earth. The delta V is approximated by the difference between the mean, or circular, orbital velocities as in Eq 3.8. Using the mean velocities allows either or both orbits to be elliptical. In addition, the calculations assume the inclination is not changed. Changing inclination requires large delta Vs, needlessly increasing the required propellant mass.

To deorbit the satellite, perigee is dropped into the atmosphere where drag then causes the orbit to decay until the spacecraft falls back to Earth. The user can enter either the new perigee altitude or perigee radius. Again, if both values are entered they must be consistent. Perigee radius must equal the perigee altitude plus the radius of the Earth or an error message is given. Perigee is typically dropped to between 50 and 150 km altitude. While allowable, it is unnecessary to reduce the perigee altitude below 50 km and requires excessive propellant mass. Above an altitude of 200 km, drag is too low to deorbit the spacecraft within a reasonable time. If the new perigee altitude is outside this range, from 50 to 200 km, a warning message is given to alert the user the specified values could be improved. The delta V is then calculated from half of a standard coplanar Hohmann transfer. Since the satellite is being removed from orbit, there is no point in changing the inclination, which, especially at low altitudes requires a large amount of fuel. In addition, it is unnecessary to perform the second Hohmann maneuver. Atmospheric drag will take care of it without requiring any additional propellant mass.

6. GEO North - South Stationkeeping

The gravitational influence of the sun and moon cause the inclination of geosynchronous satellites to drift from the desired angle. The sun causes the inclination to drift by 0.269°/yr. The drift rate caused by the moon varies between 0.478°/yr and 0.674°/yr depending on the angle between the lunar orbit and the satellite's orbit. Therefore, the combined drift rate falls between 0.747°/yr and 0.943°/yr. The average drift rate is 0.8475°/yr. (Agrawal, 1986, p. 87)

As the inclination of the orbital plane changes, the satellite will appear to have north-south oscillations which will grow in amplitude unless counteracted by firing thrusters. If left uncorrected, the inclination will vary between ±15° over a period of about 108 years. For satellites with low inclinations, as most GEO satellites are, the inclination will increase by about 1° per year for the first 10 years, then slowly reaches 15° over the next 17 years. After that, the direction of the drift reverses until the orbit is inclined 15° in the opposite direction (Vallado, 1997, p. 748). To successfully perform the mission, the satellite's inclination variation is usually limited to less than 3° for mobile communications satellites, such as FltSatCom, and to 0.1° for fixed GEO communications satellites due to beamwidths and antenna patterns.

To correct for the inclination drift, the satellite thruster must be fired. The inclination is allowed to drift until the allowable limit is reached, at which point a noncoplanar maneuver is performed to restore the inclination. To maximize the time between maneuvers and minimize the required propellant mass, the orbit plane is not just set back to the original inclination but is actually changed to the allowable limit in the opposite direction. In other words, the inclination is changed from the positive limit to the negative limit. For this type of maneuver, the velocity increment is given by:

$$\Delta v = 2 \, v \sin(i_L) \qquad (3.21)$$

where i_L is the allowable inclination limit. The time between maneuvers is:

$$T = \frac{2 \, i_L}{DR} \qquad (3.22)$$

where DR is the inclination drift rate. The number of maneuvers is then found by dividing the time between maneuvers into the design life of the satellite.

To perform the calculations, the user specifies values for the Allowed Inclination Variation and the Inclination Drift Rate averaged over the design life. The allowed inclination variation is dictated by mission requirements and must be greater than zero. A 0° inclination limit would require continuous thrust and very high propellant mass. As a note, if the limit is greater than 15°, the satellite will naturally remain within the allowable inclination without corrections. As stated above, GEO communications satellites typically have an inclination limit of 0.1°, which is the default value. Inclination drift rates for any given year can be found in tables of astronomical data (see Agrawal 1986, p. 78). As discussed above, the drift rate must be between 0.747°/yr and 0.943°/yr or an error message is provided. The default value uses the average drift rate of 0.8475°/yr.

With this information, the spreadsheet calculates the delta V per maneuver, the time between maneuvers, and the number of maneuvers of the design life. The mean orbital velocity is used in Eq 3.21 to find delta V. Note the number of maneuvers is rounded down since the next one would occur after the end of the design life. Finally, the Total NS Delta V is the product of the total number of maneuver with the delta V each maneuver requires. Of course, if the mission orbit is not GEO these calculations do not apply and the entries are blanked.

7. GEO East - West Stationkeeping

Small irregularities in the Earth's mass distribution cause a longitudinal drift acceleration on geosynchronous satellites. The equatorial cross section of the Earth is not circular, it has a small bulge. As a result, the gravitational attraction is not directly toward the center of the Earth, but is actually directed toward the bulge. This creates a component of force acting along or opposite the satellite's velocity. For satellites in LEO, this effect averages out over several revolutions. However, a satellite in GEO maintains the same relative position to the mass asymmetries so the effects accumulate.

The equatorial bulge creates longitudinal accelerations directed toward two points which are almost, but not exactly, opposite each other. These two accelerations balance at four points, two stable and two unstable, where the gravitational acceleration is truly radial and the longitudinal acceleration is zero. A satellite placed at either of the stable points, located at 75° E and 252° E (108° W), will naturally remain there without corrections. If the satellite is placed at the unstable points, it will drift to the nearest stable point. However, since the acceleration acts toward the stable point, the drift rate will continue to increase until the satellite passes through the stable point, reversing the direction of acceleration. As a result, the satellite will oscillate about the stable point. An excellent description of this effect is provided in Gordon and Morgan, 1993, pages 71-76. Similarly, if a satellite is located between the stable and unstable points, it will again oscillate about the stable longitude. The amplitude of the oscillation will equal the difference between the longitude of the original location and the nearest stable longitude. For example, if the satellite is located 30° from a stable longitude, say at 105° E, it will oscillate between 45° E and 105° E. Since satellites are placed approximately every 3° E around the equatorial ring, the longitudinal station must be maintained within reasonable limits.

When the allowable limit is reached, an orbital maneuver must be performed to stop the drift and return the satellite to its assigned longitude station. Therefore, the maneuver not only stops the spacecraft, but imparts a drift rate in the opposite direction. If done correctly, the longitudinal acceleration will just cancel the drift rate as the satellite reaches the opposite limit. At this point, the drift again reverses and heads back to the first limit. If successful, the thrusters are fired only at one of the longitude limits.

To calculate the velocity increment, the user must specify the allowable longitude variation. The limit is determined by mission needs and to prevent the satellite from drifting into adjacent longitude stations. Since satellites are stationed every 3°, a warning message is provided if the entered limit is larger than this value. For most GEO communications satellites, the beamwidth and antenna pattern limits the allowable longitudinal drift to 0.1°, which is the default.

With all of the needed information specified, the east-west (EW) stationkeeping requirement can be calculated. A complete development of the following equations is provided in Agrawal, 1986, pages 83-84 and 88-91. The longitudinal acceleration due to the ellipticity of the Earth's equator is given by:

$$\ddot{\lambda} = -0.00168 \sin 2(\lambda - \lambda_s) \qquad (3.23)$$

where λ is the longitude station and λ_S is the nearest stable longitude. For GEO longitude station specified in the Orbit sheet, the spreadsheet automatically selects the nearest stable point. The selected longitude is displayed to allow the user to double check the calculations, if desired. As noted above, if the satellite is stationed at one of the equilibrium points, the longitudinal drift acceleration is zero. To avoid a mathematical singularity, drift acceleration is "clipped," with a minimum value of 0.000001°/day^2. This is a reasonable approximation since, in a practical sense, there will still be a small velocity requirement even if the satellite is located exactly at a stable longitude.

The time between stationkeeping maneuvers is:

$$T = 4\left(\frac{\Delta\lambda}{|\ddot{\lambda}|}\right)^{1/2} \qquad (3.24)$$

where $\Delta\lambda$ is the allowable longitude limit. Dividing this result into the mission life gives the number of maneuvers. Just as for NS stationkeeping, the number of maneuvers is again rounded down since the next one would occur after the end of the mission life.

The required delta V per year, in m/sec, is:

$$\Delta v|_{year} = 1{,}032.95\,|\ddot{\lambda}| \qquad (3.25a)$$

$$= 1.74 \sin 2(\lambda - \lambda_s)\,. \qquad (3.25b)$$

The Total EW Delta V is then simply the product of the annual delta V and the mission life. As an interesting side note, from Eqs 3.25 it is clear the annual delta V requirement is not dependent on the allowable longitudinal variation. The limit only determines the number of maneuvers and the time between them. As before, if the mission orbit is not GEO these calculations do not apply and the entries are blanked.

8. Atmospheric Drag

The primary nongravitational force which acts on satellites in LEO is atmospheric drag. Drag always opposes the direction of motion and decreases the spacecraft's orbital energy. This causes the orbit to get smaller, lowering the satellite further into the atmosphere which increases the drag. However, as the orbital radius gets smaller, the velocity actually increases. This is one of the interesting paradoxes of orbital mechanics; drag increases the spacecraft velocity. Unless the energy is restored, the satellite will eventually reenter the atmosphere and fall back to Earth. To increase the energy, the satellite's thrusters must be fired. Unfortunately, all of the available equations which compute the required delta V directly only apply to circular orbits. Since the spreadsheet is intended to be a general design tool, applicable to any Earth orbit: circular or elliptic, a different approach is needed.

Drag causes variations in most of the orbital elements. Fortunately, most of the variations are periodic and average out over several revolutions. The main secular effects are in the semi-major axis and eccentricity of the orbit. Larson and Wertz provide an equation for the change in semi-major axis per revolution which is valid for any orbit, so this is a good starting point. This equation is:

$$\Delta a_{rev} = -2\pi(C_D A / m)a^2\rho_p \exp(-c)[I_o + 2eI_1] \qquad (3.26)$$

with

$$c \equiv ae / H \qquad (3.27)$$

where C_D is the coefficient of drag, A is the satellite's cross-sectional area, m is the satellite's mass, ρ_p is the atmospheric density at perigee, H is the density scale height, and a and e have their usual definitions of semi-major axis and eccentricity. The I_i are Modified Bessel Functions of order i and argument c. Values for I_i can be found in most standard mathematical tables or from the Excel add-in function BESSELI. (Larson and Wertz, 1992, p. 143)

The change in semi-major axis can be related to a delta V by considering the orbital energy. From basic orbital mechanics, the energy of an orbit is:

$$\varepsilon = \frac{v^2}{2} - \frac{\mu}{a} = -\frac{\mu}{2a}. \qquad (3.28)$$

Changes in the semi-major axis cause changes in the orbital velocity. If the changes are assumed to be small, i.e. the orbit is corrected frequently, the change in energy can be approximated as:

$$\Delta\varepsilon = \left[\frac{(v+\Delta v)^2}{2} - \frac{\mu}{(a+\Delta a)}\right] - \left[\frac{v^2}{2} - \frac{\mu}{a}\right] = \left[-\frac{\mu}{2(a+\Delta a)}\right] - \left[-\frac{\mu}{2a}\right]. \qquad (3.29)$$

Since the changes are assumed to be small, Δa << a and Δv << v. Applying a binomial expansion:

$$(a+\Delta a)^{-1} = a^{-1}\left(1+\frac{\Delta a}{a}\right)^{-1} \approx a^{-1}\left(1-\frac{\Delta a}{a}\right) \qquad (3.30a)$$

and

$$(v+\Delta v)^2 = v^2\left(1+\frac{\Delta v}{v}\right)^2 \approx v^2\left(1+2\frac{\Delta v}{v}\right). \qquad (3.30b)$$

Substituting Eqs 3.30 into Eq 3.29 gives:

$$\Delta\varepsilon = \left[\frac{v^2+2v\Delta v}{2} - \frac{\mu}{a}\left(1-\frac{\Delta a}{a}\right)\right] - \left[\frac{v^2}{2} - \frac{\mu}{a}\right] = \left[-\frac{\mu}{2a}\left(1-\frac{\Delta a}{a}\right)\right] - \left[-\frac{\mu}{2a}\right]. \qquad (3.31)$$

Canceling terms and simplifying gives:

$$\Delta v_{rev} = -\frac{\mu}{2va^2}\Delta a_{rev}. \qquad (3.32)$$

Since this equation is for changes over a complete revolution, the mean velocity can be used without loss of generality. Substituting the expression for the mean velocity, $v = (\mu/a)^{1/2}$, into Eq 3.32, the expression simplifies to:

$$\Delta v_{rev} = -\frac{1}{2}\sqrt{\frac{\mu}{a^3}}\,\Delta a_{rev}\,. \qquad (3.33)$$

However, to compute Δa_{rev}, the atmospheric density at perigee must still be determined.

Atmospheric density is approximated by a piecewise exponential model. The density at a particular height is found from:

$$\rho_p = \rho_o \exp\left(-\frac{r_p - r_o}{H}\right) \qquad (3.34)$$

where ρ_p is the atmospheric density at perigee, r_p is the perigee radius, r_o is radius of the base of the appropriate region, ρ_o is reference density at r_o, and H is the scale height for the region. The model assumes a spherically symmetric distribution of particles, with the density decaying exponentially within each region. It provides a valid approximation to an altitude of 1,500 km, but above that drag is negligible. The spreadsheet automatically calculates the density at perigee of the final orbit specified in the Orbits sheet. The radius of perigee is used to select the appropriate altitude region. The reference density and scale height are then found from a lookup table and used in Eq 3.34. (Vallado, 1997, p. 502-510)

To compute the delta V for drag makeup, the user must enter the spacecraft's cross sectional area, mass, and coefficient of drag. The cross sectional area is the frontal area in the direction of flight. The default is the sum of the area of the spacecraft body plus the solar array area. The spacecraft body area is determined from the size and shape specified in the General sheet. For spherical spacecraft, it is simply πr^2. For cylindrical spacecraft, the spreadsheet assumes the yaw, or z, axis is nadir pointing and computes the area of the xy face. Note that if the default area is used and work is done in the EPS sheet after completing these calculations, the drag delta V might change. The smaller the spacecraft mass, the larger the delta V requirement will be for a given orbital altitude. The default value is the mass placed in the final orbit, in other words the separation mass less the orbital transfer propellant and motor casings, is any. This value is transferred from the Propellant sheet. The coefficient of drag is a dimensionless value which expresses how susceptible the spacecraft is to drag effects and depends on the satellite configuration. By definition, the coefficient of drag must be greater than zero. Negative entries will generate an error message. Spheres have a drag coefficient of one. Most satellites have a drag coefficient between 2 and 4, with 2.2 a reasonable average (Larson and Wertz, 1992, p. 143 and 207). In an effort to prevent errors, if the entered value is outside the normal range of 2 to 4, a warning message will be displayed. However, the entered value is accepted and used in subsequent calculations.

The ballistic coefficient, defined as $m/(C_DA)$ is another measure of the satellite's susceptibility to drag effects. Low values mean drag has a large effect on the spacecraft. Since this is a common and important parameter, especially for LEO spacecraft, it is computed and displayed for convenience. The user cannot directly enter a value for the ballistic coefficient. It is completely specified by the spacecraft area, mass, and coefficient of drag.

The last step is to put these values together to compute the velocity requirement. First, the change in the semi-major axis per revolution is found using Eq 3.26. Note that to perform these calculations, the Analysis ToolPak Excel Add-In, which contains the Bessel functions, must be installed. From this value, the change in orbital velocity per revolution is computed from Eq 3.33. The number of revolutions per day is determined by dividing the orbit period into one day. Multiplying the preceding two quantities gives the delta V per day. Multiplying by 365.25 gives the delta V per year, which is displayed. Finally, the total velocity requirement is found by multiplying the annual velocity increment by the design life.

9. Other Perturbations

There are several other orbital perturbations in addition to those addressed in the spreadsheet. The most notable disturbances are due to solar radiation pressure, third body effects, and the nonspherical Earth. These perturbations were omitted since they are either negligible, in the case of solar pressure, or are normally left uncorrected. In many applications, the satellite is simply allowed to drift. Its relative position to the constellation does not change since the entire constellation drifts in the same manner. In addition, correcting for the third body effects and nonspherical Earth takes a very large velocity requirement, on the order of several hundred kilometers per second each year. This would take more propellant then the satellite could carry. Including these effects in the spreadsheet calculations would seriously distort the true requirements. Finally, some specialty orbits capitalize on the effects of these perturbations, so performing orbital corrections would destroy their unique characteristics. For example, sun synchronous orbits are only possible because the Earth is nonspherical.

Under very specific circumstances and for short mission durations, correcting for these effects may be important. Therefore, a brief discussion and the necessary equations are provided below. If the user needs to correct for these perturbations, the delta V can be calculated by hand and entered into the Additional Delta V line. For a detailed discussion on general perturbation techniques, see Vallado, 1997, pages 578 to 621. For a more brief discussion, Larson and Wertz, 1992, provide a good summary on pages 139 to 143.

Solar radiation pressure causes periodic variations in all of the orbital elements. The effects are greatest on spacecraft with low ballistic coefficients, those with low mass and large illuminated areas. Calculating these effects is rather complex and depends on several factors which are difficult to estimate, including the illuminated area and reflectivity, the attitude with respect to the sun, the solar flux arriving at the satellite's position, and the overall solar activity. The calculations are further complicated if the satellite passes through eclipse. Even if these parameters can be reasonably estimated, the solar pressure effects are usually small except for spacecraft with low mass and large surface areas.

The gravitational attraction of the Sun and Moon cause periodic variations in all of the orbital elements. The only secular variations are in the argument of perigee, ω, and the right ascension of the ascending node (RAAN), Ω. While the J_2 effects dominate in LEO, the third body effects are important for higher altitude orbits. In fact, the third body perturbations are the source of the GEO EW stationkeeping requirement, which was specifically addressed in the spreadsheet. The time rate of change due to the Sun and Moon are:

$$\dot{\Omega} = -\frac{3\mu_3(2+3e^2)\left[2-3\sin^2(i_3)\right]}{16r_3^3n\sqrt{1-e^2}}\cos(i) \tag{3.35}$$

$$\dot{\omega} = \frac{3\mu_3\left[2-3\sin^2(i_3)\right]}{16r_3^3n\sqrt{1-e^2}}\left\{e^2+4-5\sin^2(i)\right\} \tag{3.36}$$

where μ is the gravitational parameter for the Earth, i and e the inclination and eccentricity of the satellite orbit, μ_3 is the gravitational parameter for the third body, and i_3 is the inclination of the third body relative to the Earth. Orbital parameters for the Sun and Moon are listed in Table 3-1. The mean motion, n is:

$$n = \sqrt{\frac{\mu}{a^3}} \qquad (3.37)$$

where a is the semi-major axis of the satellite orbit. Eqs 3.31 and 3.32 assume the third body is in a circular orbit. A quick check of the eccentricities in Table 3-1 shows this is a very reasonable assumption. (Vallado, 1997, p. 611-617)

Table 3-1. Third Body Orbital Parameters (Vallado, 1997, p. 615)

		Sun	Moon
Mean Distance	r_3	149,598,023 km	384,400 km
Inclination	i_3	0.0°	5.1°
Eccentricity	e_3	0.0	0.05
Gravitation Parameter	μ_3	$1.327\text{x}10^{11}$	$4.9\text{x}10^{3}$ km^3/sec^2

The nonspherical Earth also causes periodic variations in all of the orbital elements, but just like for the third body effects, the only secular variations are in the argument of perigee and RAAN. The derivation of the orbital parameters assumed the Earth is perfectly spherical with a homogeneous mass distribution. While close, in reality neither of these assumptions is true. The Earth is slightly oblate with a bulge at the equator. Even if the Earth were spherical, mountains, oceans, mineral deposits, and the different types of rocks cause variations in the mass distribution. For satellites in GEO and below, the Earth's asymmetries is the dominant perturbation until drag takes over for very low altitude orbits. Beyond GEO, the Sun and Moon have the largest effect. For GEO satellites, the unsymmetrical Earth causes the EW drift, which was specifically addressed in the spreadsheet. For any orbit, the time rate of change in the argument of perigee and RAAN due to the nonspherical Earth is:

$$\dot{\Omega} = -1.5n\, J_2(R_e / a)^2 \cos(i)\,(1-e^2)^{-2} \qquad (3.38)$$

$$\dot{\omega} = 0.75n\, J_2(R_e / a)^2[4-5\sin^2(i)](1-e^2)^{-2} \qquad (3.39)$$

where $J_2 = 1.08263\text{x}10^{-3}$ is the Earth's second zonal harmonic, R_e = 6,378.1363 km is the radius of the Earth, and a, e, and i have their usual meanings of semi-major axis, eccentricity, and inclination, respectively. The mean motion, n, is again given by Eq 3.37. Eq 3.38 shows the

Earth's asymmetries have no effect on polar orbits since cos(90°) = 0. The inclination of the Molniya orbit is derived from Eq 3.39. The highly elliptic Molniya orbits are used to provide coverage of a particular hemisphere, usually the northern, so it is important to maintain the argument of perigee. Setting $\dot{\omega} = 0$ in Eq 3.39 gives inclinations of 63.4° or 116.6°. (Larson and Wertz, 1992, p. 140-142)

To enter them into the preliminary design, the third body and J_2 perturbations must be translated into a velocity increment which can then be entered into the Additional Delta V line. Variations to the argument of perigee and RAAN represent changes in the orientation of the orbital plane. The orbit's orientation is corrected with a standard non-coplanar maneuver, so the required delta V is found from:

$$\Delta v = 2v\sin\left(\frac{\theta}{2}\right) \qquad (3.40)$$

where v is the satellite velocity at the point of the maneuver and θ is the angle through which the orbit is change. For the argument of perigee, θ is the daily drift angle, which is simply found from the time rate of change. For RAAN θ depends on the orbit inclination and is found from:

$$\theta = \cos^2(i) + \sin^2(i)\cos(\Delta\Omega) \qquad (3.41)$$

where i is the inclination and ΔΩ is the daily drift in RAAN. Eq 3.40 only applies to circular orbits, however for elliptical orbits, the delta V can be reasonably approximated using the mean orbital velocity.

The delta V requirements to correct for the third body and J_2 effects can be very large. First, for LEO orbits the drift rates in the argument of perigee and RAAN can each be as much as 5°-10° per day. Second, plane change maneuvers intrinsically require large delta Vs. This is the primary reason, combined with the fact the orbital corrections are rarely performed, why the third body and nonspherical Earth perturbations were not included in the spreadsheet.

D. PROPELLANT

The Propellant sheet determines the propellant budget by converting the required delta Vs into the corresponding propellant mass necessary to perform the maneuvers. The propellant masses are then allocated to fuel tanks and the margin and residual are applied. The individual masses are totaled and the spacecraft dry mass is found. For reference, the sheet is included in Appendix B on page 150.

1. Propellant Mass Calculations

The delta V requirements are transferred from the Delta V sheet and cannot be adjusted in this sheet. The velocity increments can be modified only in the Delta V sheet. In addition to the delta Vs, the reorientation/spin control requirement is transferred in directly as a propellant mass. It cannot be adjusted here, but can be modified only in the originating Delta V sheet. Finally, propellent for attitude control is estimated as a percentage of the other orbital maneuvering masses.

The first step is to select the type of propulsion subsystem performing each maneuver: monopropellant, bipropellant, electric propulsion (EP), cold gas, or a solid motor. However, each propulsion subsystem is not applicable to every maneuver. For example, solid motors are only an option for orbital transfers. Once ignited, solid propellants burn to completion and cannot be restarted. The orbital transfer delta Vs are each applied in a single maneuver, but the other maneuvers are performed repeatedly over the life of the spacecraft. For these maneuvers, the propulsion subsystem must have a restart capability, precluding the use of solid motors. Electric propulsion is not an option for the second orbital transfer maneuver. While having very high specific impulses, EP provides very low thrust levels. To perform an orbital transfer, thrust is continuously applied over a long period of time until the desired orbit is achieved. In essence, it is one long maneuver so there is no second maneuver. Therefore, EP is not a choice for the second maneuver and the line is blanked if EP is selected for the first maneuver. The sheet assumes an integrated bipropellant propulsion subsystem is used for all maneuvers.

Next, the specific impulse and efficiency of the propulsion subsystem selected for each maneuver is identified. As used here, efficiency is not the internal efficiency of the thruster, which actually figures into the specific impulse. Instead, it is the efficiency of thruster alignment. To prevent plume impingement on the spacecraft, thrusters are frequently canted from the optimum direction. This misalignment causes a decrease in the effective thrust. The efficiency is simply the cosine of the cant angle, and is entered as a decimal fraction between zero and one. Entered values for the specific impulse must be greater than zero but have no upper limit. EP systems can have very large specific impulses, although even than it rarely exceeds 8,000 sec.

Default values depend on the propulsion subsystem and maneuver. Thrusters are usually operated in different modes for the various maneuvers. For example, a bipropellant subsystem might perform a single, longer burn to reposition the satellite, but might fire a series of short pulses for stationkeeping. The mode of operation changes the specific impulse. In addition, thruster locations vary for the various maneuvers. Some locations are more likely to cause plume impingement so the thruster is more likely to be canted. Therefore, the efficiency also depends on the maneuver. To help the user select appropriate values, a table listing the specific impulse and thrust level for many propulsion subsystem types is included at the bottom of the sheet (Larson and Wertz, 1992, p. 644-645).

With this information, the propellant mass and net spacecraft mass after each maneuver can be found from the required delta V increments. The propellant mass needed to perform each maneuver is computed from the ideal rocket equation:

$$m_p = m_i\left(1 - e^{-\Delta v / (I_{sp} g_o)}\right) \tag{3.42}$$

where m_p is the propellant mass for the maneuver, m_i is the initial spacecraft mass before the maneuver, Δv is the change in velocity imparted to the vehicle, I_{sp} is the specific impulse, and $g_o = 9.81$ m/sec^2 is the gravitational attraction at the Earth's surface. In Eq 3.42, the initial mass is the mass of the spacecraft after the preceding maneuvers. In addition to the delta Vs, the propellant mass for reorientation/spin control during transfer is transferred in from the Delta V sheet. The changes in mass are then subtracted from the initial value, which starts at the separation mass from the General sheet.

Attitude control requirements are computed directly as propellant masses instead of from a delta V. Thrusters are typically used in pairs to impart a pure torque on the spacecraft, which is used for control during delta V maneuvers, for spin stabilization and maneuvering, to counter disturbance torques, and for attitude maneuvers or slewing. There are many methods for providing attitude control of the spacecraft. Many, such as passive gravity gradients, magnetic torque rods, or reaction wheels, require little or no propellant. However, gravity gradient and magnetic systems do not provide high levels of control torque. In addition, reaction wheel systems must be periodically desaturated to cancel the accumulation of non-cyclic disturbances. Therefore, most attitude control subsystems use thrusters to at least some extent. Thruster can provide large control torques, and also provide control of the spacecraft's translational velocity.

Propellant mass for attitude control is difficult to estimate accurately, since the amount is highly dependent on mission requirements, such as pointing accuracy and slewing, and on the magnitude of cyclic and aperiodic disturbances. Preliminary estimates are typically based on historical averages, either from the spacecraft mass and orbit or as a percentage of propellent mass for the other orbital maneuvers. Typically, attitude control propellant mass is approximately 10% of the other maneuvering masses, and is the bases if the default value.

The orbital transfer maneuvers are frequently provided by large, dedicated rocket engines which are external to the satellite itself. The most obvious example is solid motors, but liquid mono- or bipropellant systems can also be bolt-on systems. Once the maneuver is performed and the propellant is expended, the empty motor casing represents dead weight. Therefore, it is normally jettisoned to reduce the spacecraft mass for subsequent maneuvers, which in turn decreases the propellant mass required to perform a given delta V. The spreadsheet provides lines to subtract the mass of separated motor casings after the first and second maneuvers. The default values depend on the selected propulsion subsystem. Solid kick motors typically have mass fractions from 88% to 95% with an average of 93% (Larson and Wertz, 1992, p. 651). Therefore, if the orbital transfer maneuvers are provided by a solid motor, the default motor casing mass is 7% of the propellant mass for that maneuver. For any other type, the spacecraft is assumed to have an integrated propulsion subsystem so the motor is not separated. If the separated motor casing mass is entered, it must of course be greater than or equal to zero; the casing cannot have negative mass. As discussed before, if the orbital transfers are performed with EP the second maneuver is irrelevant so any entered casing mass is ignored.

In addition to the mass for each maneuver, propellant budgets typically include provisions for margin and residual propellant. A margin of 10% is customarily applied for growth in the delta V requirements, off-nominal operation of the thrusters, or other unforeseen circumstances. Some propellant always remains in the tanks and fuel lines, so the propellant budget must include an amount for the residual. The margin and residual are computed individually for each tank in the Propellant Mass Allocation to Tanks section at the bottom of the sheet. The totals for all of the tanks are then brought back up and added into the propellant budget.

The individual propellant mass requirements for each maneuver are then totaled along with the margin and residual. This sum is displayed at the bottom of the propellant budget, in the

TOTAL PROPELLANT MASS line, and is the total propellant mass required over the life of the mission. Note that the last computed mass in the column is also the spacecraft dry mass, since it is the separation mass less the propellant.

If at any point the spacecraft mass becomes negative, the calculations stop and an error message is issued. This happens whenever the propellant mass needed to perform a given delta V exceeds the remaining spacecraft mass. This problem can be solved in two ways. First, the separation mass can be increased in the General sheet. Second, the propellant mass requirements can be reduced. This can be done by changing the type of propulsion subsystem, increasing the specific impulse and/or efficiency, or decreasing the propellant margin and residual. Another option is to use a more powerful launch vehicle to eliminate the second maneuver or even achieve a direct insertion. If none of these options is acceptable, then the spacecraft cannot carry sufficient propellant and the flight profile is not feasible.

2. Propellant Mass Allocation to Tanks

Having computed the propellant requirements for each maneuver, the masses are transferred to the bottom section of the sheet where they are allocated to a fuel tank. For reference, the propulsion subsystem type is also echoed from above. The allocation assumes each type of propulsion subsystem uses a different propellant. However, it also assumes every use of a given propulsion type uses the same propellant. If a monopropellant system is chosen for three different maneuvers, the sheet assumes the same propellant, hydrazine for example, is used each time. The allocation is simply performed by putting the propellant masses into a different tank for each type of propulsion subsystem. Solid motors are the exception. Once ignited, solid motors burn to completion and cannot be used for more than one maneuver. If solid propulsion subsystems are selected for both the first and second maneuvers, the two masses are allocated to different tanks. The other complication to this simple allocation scheme is the bipropellant subsystem type.

Bipropellant propulsion systems combine a fuel and an oxidizer, so the maneuver's propellant mass cannot be allocated to a single tank. Instead, it must be decomposed into the mass of the fuel and the mass of the oxidizer, with each placed in a separate tank. If both mono- and bipropellant propulsion subsystems are selected, the allocation assumes the bipropellant fuel is also the monopropellant and places them in the same tank.

The mixture ratio is the ratio at which the fuel and oxidizer are combined, and is defined as the ratio of the oxidizer mass flow rate to the fuel mass flow rate. Since both must be supplied to the thruster over the entire firing time, this also equals the ratio of the masses. Therefore, the mixture ratio is given by:

$$r = \frac{m_o}{m_f} \qquad (3.43)$$

where r is the mixture ratio, m_o is the oxidizer mass for the maneuver, and m_f is the fuel mass. Turning this equation around, the individual masses are found from:

$$m_o = \frac{r\,m}{r+1} \qquad (3.44)$$

$$m_f = \frac{m}{r+1} \qquad (3.45)$$

where $m = m_o + m_f$ is the propellant mass for the maneuver. If a mixture ratio is entered, it obviously must be greater than zero or an error is generated. In addition, the mixture ratio for any oxidizer/fuel pair is rarely less than 1 or greater than 8, so a warning message is issued if the entered value is outside this range. The default value is based on a nitrogen tetroxide (N_2O_4) / monomethylhydrazine (MMH) combination. A mixture ratio of 1.64 is commonly used since it results in two tanks of equal size (Larson and Wertz, 1992, p. 659). If a bipropellant propulsion subsystem is not selected, the mixture ratio is blanked and any entered value is ignored. To help the user select an appropriate value, a table listing the mixture ratios for several bipropellant combinations is included at the bottom of the sheet. This table was extracted from data throughout Sutton, 1992.

The propellant mass for any one propulsion subsystem type can be allocated to several different tanks. As noted above, the allocation assumes that each time a particular propulsion type is selected it uses the same propellant so the masses are placed in the same tank. However, this is just the default allocation. The sheet allows for multiple, independent subsystems even for a given type of propulsion. For example, if two independent cold gas propulsion subsystems are used, the propellant masses must be allocated to two different tanks. To do this, just enter a zero in the default tank and reenter the propellant mass in a different tank. The spreadsheet allows for up to 10 tanks, which should be more than enough for all but the most exotic mission profiles. The same process can be used if the mono- and bipropellant subsystems don't use the same fuel.

Just zero the bipropellant fuel from the monopropellant tank and reenter it in its own tank. To help avoid errors, if the allocated fuel masses for a maneuver don't sum to the computed mass, a warning message is issued alerting the user the values may have been reentered incorrectly.

The tanks can be named to help keep track of the different propellants. The initial tank names, i.e. EP, Cold Gas, or Solid 1, are only for convenience. The spreadsheet treats all of the tanks the same, except for the two solid tanks, so any tank can be used for any propellant. The margin and residual are applied differently to the two solid propellant tanks, as discussed below.

Once all of the propellant masses are allocated to tanks, the margin and residual can be applied. They can be specified as either a fixed amount or as a fraction of the propellant mass in the tank. If both are entered, the fixed mass supersedes the fraction. The user can enter the margin and residual as a global value or individually for each tank. The global value is applied to each tank which contains at least some propellant. However, the global value is not applied to the tanks for the solid propellants. Solid motors have very little residual propellant and typically do not include a margin since they must burn to completion. Entering a value under an individual tank will override the global value, but only for that tank. This allows the user to apply a value to all of the tanks without having to enter it 10 times, yet provides for individual exceptions for selected tanks. The default margin is 10% with a 5% default residual. To avoid excessive propellant masses, a warning message is issued if any tank exceeds a margin of 30% and/or a 10% residual.

An example will help illustrate this process. A spacecraft has a solid perigee kick motor but uses an integrated bipropellant engine for the apogee maneuver. On-orbit, a monopropellant propulsion subsystem provides most of the maneuvers with a cold gas system for fine control. The mono- and bipropellant systems do not share a common fuel. Such a propulsion subsystem is overly complex and is not commonly used, however, it will demonstrate the application of margin and residual. Propellant will be allocated to five tanks: two for the bipropellants and one each for the solid, monopropellant, and cold gas systems. For a 15% margin in all of the tanks, the value of 0.15 is entered under the For Each Tank heading. For the five tanks with propellant masses, four will adjust to the entered global margin but the margin is not applied to the solid propellant mass. The other five tanks contain no propellant so the global margin is not applied. Since cold gas systems have very low specific impulses and are therefore more sensitive to the mass, a 20% margin is specified by entering 0.20 under Tank 4, Cold Gas. The 20% margin will only be applied to Tank 4. Since the margin is not applied to the solid motors, Tank 5, Solid 1, still has a zero margin. To account for the small variation in solid propellants and motor performance, a 5 kg margin is specified by entering a 5 into the Margin Mass line. The margin mass is applied only to Tank 5, and would supersede the global fraction, although as noted above the global values are not applied to the solid tanks. Values for the residual fraction and mass work in exactly the same way as the margin.

Finally, the propellant mass in each tank is found by adding the margin and residual to the allocated mass. The margins and residuals are also summed across all tanks, which is then passed back up to the Propellant Mass Calculations section. These values are added to the propellant required for each maneuver to find the Total Propellant Mass requirement.

E. EPS

The EPS sheet works through the operations to size the battery capacity and design the solar array. The sheet is provided on page 155 of Appendix B. The calculations assume the array is an èxternal, flat panel which follows the sun in at least one axis. For orbit inclinations over 25°, the array is further assumed to be double-gimbaled to track the sun, reducing the incidence angle to zero. The battery capacity is sized to power the satellite during eclipse and to provide supplemental power during illuminated operation if the array is sized for the average power. Many factors are included in the analysis, such as: voltage and power drops in the array, battery, and power bus(es); battery and solar cell efficiency; depth of discharge limitations; array operating temperature; radiation degradation; sun incidence angle; and variations in the solar intensity.

Information from the General, Orbit, and Propellant sheets is used in the EPS analysis. The design life is a critical factor in sizing the EPS subsystem and is imported from the General sheet along with the spacecraft size. The Orbit sheet provides the inclination and semi-major axis. To determine the electric propulsion power requirements, the Propellant sheet is checked to see if EP is selected. If so, its specific impulse is transferred in to use in estimating the default EP power requirement.

1. EPS System Data

Before the battery and solar array can be sized, the basic characteristics of the power subsystem must be defined. As the first step, the user enters information on the voltage regulation type, bus redundancy, and eclipse and noneclipse operating voltage.

The voltage regulation type is selected from a drop-down list. The bus voltage can be left completely unregulated and allowed to float with the output voltage of the array or battery, partially regulated to remain within certain limits, or fully regulated to a specific value. In an unregulated bus, the voltage is regulated separately at each component. The bus voltage equals the output voltage of the array or battery. The maximum voltage is determined by the cold solar array as it emerges from eclipse. If the array output voltage drops, either due to the sun incidence angle or entry into eclipse, the bus voltage also drops until the battery discharge control set point is reached. At this point, the battery is switched directly to the power bus, which usually causes a step increase in the bus voltage. As the battery discharges, the voltage will again drop. For successful operation, the individual components must be able to handle the voltage swings and step increases of the unregulated bus. In a partially regulated bus, the output of the solar array output is regulated within allowable limits by shunt regulators. However, during eclipse the battery output is unregulated so the bus voltage will again vary as the battery discharges. Finally, in a fully regulated bus, the voltage is regulated during both eclipse and the illuminated portion of the orbit. The array is again regulated by shunt regulators, but the battery output is now controlled by a discharge regulator. Unfortunately, the discharge regulator introduces additional losses, decreasing the battery efficiency and increasing the thermal dissipation. (Agrawal, 1986, p. 367-368)

A central question in the reliability of the spacecraft is whether to connect all of the components to a single bus or split them between two, redundant buses. Obviously, in a single bus configuration, all of the mission and housekeeping equipment, including redundant and backup units, are connected to the same power bus. To protect against single point failures in the power subsystem, a dual-bus system divides the components between two, independent buses. The loads are balanced between the two buses to maintain equal battery depth of discharge. With dual buses, the satellite can continue at least partial operations even if one power system completely fails. No single failure in the power subsystem or in the equipment can affect more than half of the spacecraft system. But the advantages of a dual-bus system come at a price;

redundant buses introduce additional mass and complexity into the design. (Agrawal, 1986, p. 367)

The last characteristic of the power subsystem is the bus voltage. The user must specify the operating voltage during illuminated and eclipse portions of the orbit. For a fully regulated bus, these values must be the same or an error message is displayed. In a partially regulated bus, the noneclipse voltage dictates the regulated output of the array. Since the battery voltage is unregulated, the eclipse voltage is merely the design point for the battery and also represents the battery discharge set point. If an unregulated bus is selected, both voltages are only used as design points to determine the number of battery cells and solar cells connected in series. The user can specify any value for the operating voltages, provided they are greater than zero. Over the past two decades there has been a steady trend toward higher bus voltages. In the 1960s, bus voltages were fairly low, with values between 20 and 30 volts common. With improvements in semiconductors and electronics, bus voltages in the 1970s and early 1980s were on the order of 40 to 50 volts. Today's high-power spacecraft, some with powers well over 10,000 W, have bus voltages of 100 to 140 volts to reduce the current and thereby the resistance losses.

2. Orbit Data

For the user's convenience, this section displays the period of the final orbit and the maximum duration of eclipse. These values cannot be changed here since they are completely determined by the final orbit. In fact, both values are estimated solely from the semi-major axis, which is entered in the Orbit sheet. The orbital period is calculated here in exactly the same manner as in the Orbit sheet, where it is also displayed.

The eclipse duration depends on the beta angle, which is the angle between the orbital plane and the sun's rays. The beta angle is a function of the orbit's inclination and RAAN, and can be computed from:

$$\sin(\beta) = \sin(\varepsilon)\cos(i)\sin(\theta_{VE}) - \cos(\varepsilon)\sin(i)\sin(\theta_{VE}) + \sin(i)\sin(\Omega)\cos(\theta_{VE}) \tag{3.46}$$

where i is the orbital inclination, Ω is the orbital RAAN, θ_{VE} is the angular separation of the Earth from vernal equinox in the Earth's orbit about the sun, and $\varepsilon = 23.44°$ is the obliquity of the ecliptic. If the satellite is at a radius less than the radius of the Earth divided by the sine of the beta angle, $r < R_e/\sin(\beta)$, it is in eclipse. The eclipse duration is found from the fraction of the orbit which passes through the Earth's shadow. The angle subtended by the Earth's shadow is calculated as:

$$\theta_e = 2\cos^{-1}\left(\frac{\sqrt{1-(R_e/r)^2}}{\cos(\beta)}\right) \qquad (3.47)$$

where R_e is the Earth's radius and r is the actual radius of the satellite. In precise calculations, this angle depends on the actual orbit radius, however it can be reasonably approximated by the mean orbital radius, which is also the semi-major axis. The eclipse duration is then found from:

$$T_e = \frac{P\,\theta_e}{2\pi} \qquad (3.48)$$

where P is the orbital period. From Eq 3.47, it is clear the maximum eclipse duration occurs when $\cos(\beta) = 1$, so it is unnecessary to compute the beta angle. For the maximum eclipse duration case, with $\cos(\beta) = 1$, Eq 3.47 simplifies to:

$$\theta_e = 2\sin^{-1}\left(\frac{R_e}{r}\right) \qquad (3.49)$$

For an excellent development of these equations, see Agrawal, 1986, p. 99 - 101.

3. Power Requirements

More than any other factor, the power requirements will drive the size and design of the power subsystem. To make estimation easier, improving the accuracy and fidelity of the design, the total power requirement is divided into several main functions: payload, housekeeping, electric propulsion, and thermal control (heater power). Entered powers must have a non-negative value or an error message is displayed.

Designers typically try to minimize the power consumption during eclipse to reduce the size and mass of the batteries. Nonessential components are powered down or turned off completely. Frequently, even the payload power consumption is reduced during eclipse. For example, a communications satellite may have all of the transmitters and receivers turned on during the illuminated portion of the orbit, but during eclipse only some of them might be active. To incorporate this into the spreadsheet, each function has separate entries for eclipse and noneclipse power requirements.

A satellite exists to support the payload, so the payload power requirement is a key parameter. It includes the power consumption for all mission components, including any sensors or instruments, computer processors, and memory and data storage units. It should also include the power for mission communications, even though housekeeping may also include some communications power. Using a communications satellite as an example, again, since the payload is a communications system, its power requirement should be entered as the payload requirement. However, the satellite state-of-health is normally transmitted over separate communication links and would be a part of the housekeeping power.

For many satellites, the payload does not operate continuously. Even when the payload is not operating, it still usually has a power requirement. When not actively processing data, many computer processors are powered at a lower level instead of being turned off completely, and any volatile memory must be powered continuously or the contents are lost. However, their power consumption is higher during active input/output than when the memory contents are merely being held. Therefore, the payload power requirements are entered as two values, the operating power and the standby power. Considering eclipse, the payload requirements are actually entered as four separate values. The default operating power was arbitrarily set at 1,000 W. The default standby power is 10% of the operating power. Since the default duty cycle is 100%, as discussed below, these values are the same for both eclipse and noneclipse.

The housekeeping requirement provides the power to maintain control and monitor the status of the satellite. It includes command and data handling, attitude control, and any other power requirement except EP and thermal, which are entered separately. Housekeeping is a continuous but small load and usually does not vary from eclipse to noneclipse. The default value is 13% of the payload operating power, which was derived from historical averages.

If electric propulsion is used on-orbit, its power consumption must be included in the power subsystem design. If the user has already selected a specific thruster, its rated power requirement, as identified by the manufacturer, can be entered. If not, the EP power can be estimated as:

$$P = \frac{F\, I_{sp}\, g_o}{2\eta} \qquad (3.50)$$

where F is the thruster force, I_{sp} is its specific impulse, η is the thruster efficiency, and g_o is the gravitational attraction at the Earth's surface.

The EPS sheet checks to see if "EP" is selected on the Propellant sheet for any of the on-orbit maneuvers: station keeping, repositioning, end of life (EOL) disposal, additional delta V, or attitude control. If EP is selected for one or more of these maneuvers, the largest specific impulse is used in Eq 3.50 to estimate the default power requirement. The transfer maneuver is not included here since it does not drive the design, for two reasons. First, the payload is normally off during transfer, so the full array power is available for EP. Second, since the power subsystem is sized to meet the requirements at EOL, there is excess power available at beginning of life (BOL). If no maneuvers are performed with EP, these lines are blanked and any input is ignored. Note, also, that the solar array typically does not provide the full instantaneous EP power requirement. Typically, the EP thrusters are only on for a short period each day, so the total energy consumption is relatively low. Therefore, they are normally powered from the batteries, which are then recharged over the remaining sunlit portion of the orbit. This minimizes the impact on the solar array and battery mass.

The EP duty cycle is the average EP operating time per orbit. It is entered as a percentage and must be between 0 and 100%. While orbital transfers with EP use continuous thrust, once on orbit the EP system usually only operates on the order of 10 to 15 minutes per day. This gives a duty cycle of approximately 1%, which is the default value.

Because EP thrusters have a high power consumption, they usually are not operated during eclipse. In addition, orbital maneuvers are frequently performed when the payload is not operating, if possible. In some cases it cannot be avoided, such as when the payload operates continuously. However, some payloads cannot successfully perform their mission during maneuvers. For example, even the low accelerations from an EP thruster would destroy the phase histories vital to synthetic aperture radars. Or, the payload might also have a high power consumption. Combining their power requirements might drive the power subsystem design to extremes. Therefore, it is best to perform the maneuver during the illuminated portion of the orbit when the payload is not operating.

The spreadsheet takes this into account by comparing the EP maneuver time to the payload operating time. The orbit is divided into four, prioritized, time segments with the EP power requirement applied to only those segments necessary to achieve the EP duty cycle. The EP power is first allocated to the noneclipse time when the payload is not operating. Next is the noneclipse, operating period. If for some reason, the thruster must operate over more than the illuminated portion of the orbit, the power requirement is added to the eclipse, non-operating period, and finally to the eclipse, operating time.

The TCS power is primarily the heater power necessary to maintain the minimum spacecraft temperature. It is computed on the Thermal sheet and cannot be changed here. Instead, it is better to go to the Thermal sheet and change the estimate there. As with the other power loads, the TCS power is computed for operating and non-operating periods during both the eclipse and noneclipse portions of the orbit. The noneclipse, operating state usually defines the thermal control worst-case "hot" condition. However, since this case is defined with the maximum solar intensity. Over the rest of the orbit, when the solar flux is less intense, even the worst-case hot condition can still required some heater power. The non-operating, eclipse period is normally the worst-case "cold" condition and typically represents the largest TCS power requirement.

Since the power requirements discussed above are only estimates, it is customary to add some margin to preliminary designs. The Contingency Load Fraction takes into account the uncertainty in the equipment loads, while the Array Margin accounts for the uncertainty in the radiation degradation and other power prediction factors. These two values are entered as percentages, and must be between zero and 100% or an error message is displayed. The defaults use a 5% contingency load fraction and a 10% array margin, fairly common values for preliminary designs. (Agrawal, 1986, p. 366)

Now that the power requirements and margins are identified, the next step is to enter some information on how and when the payload is operational. The Payload Duty Cycle is the fraction of the orbital period the payload is on and operating. Since operations are sometimes curtailed during eclipse, the Eclipse Operating Fraction is entered separately. Both values are entered as a percentage and must be between 0 and 100%. However, note that they are dependent. For example, if the payload operates continuously, it has a 100% duty cycle. In this case, the eclipse operating fraction must also be 100%. Since other cases might not be this obvious, if the entered eclipse operating fraction is insufficient, the spreadsheet will compute and display the minimum fraction necessary to achieve the duty cycle along with a warning message.

Entering the duty cycle and eclipse operating fraction separately allows the user complete control and flexibility in defining how and when the payload operates. An example will help demonstrate this. Consider a space-based radar system in an 800 km altitude orbit. The period is approximately 100 minutes, with a maximum eclipse duration of about 35 minutes. Suppose mission requirements dictate four, 10 minute operating periods per orbit. This gives a 40% duty cycle. For maximum flexibility, the design might call for three of the operating periods to occur during eclipse, which is the most that will fit within the eclipse duration. This results in an eclipse operation fraction of 30/35 = 85.7%. On the other hand, if the operating periods are more evenly distributed over the orbit, only one will fall within the eclipse and the fraction drops to 28.6%. If the designers want to minimize the battery mass, and mission requirements allow it, there might not be any operating times planned for the eclipse period. Note that this is possible since the total operating time of 40 minutes will easily fit within the 65 minute illuminated portion of the orbit. However, if the operating period is 20 minutes, the payload must operate during eclipse or the duty cycle can not be achieve. For this case, the duty cycle is 80%. If the payload does not operate during eclipse, the maximum achievable duty cycle is only 65%. To achieve an 80% duty cycle, the eclipse operating fraction must be at least 42.9%, or 15 minutes.

If the payload has a duty cycle less than 100%, in other words it does not operate continuously, the EPS subsystem can average the power requirement over the entire orbit. Instead of producing the full, instantaneous, payload power, the array need only generate the average power. When the payload is off, all of the power generated by the array is stored in the battery. Since the array is sized to the average power, while operating the payload draws more power then the array provides. The battery must supply the difference. For example, suppose the space-based radar discussed in the preceding paragraph draws 1,000 W when operating. One option is to size the array for the full, instantaneous power requirement of 1,000 W. With power averaging, the array size can be reduced to provide only 615 W. The remaining 385 W needed to operate the payload comes from the battery. While power averaging allows the array to be smaller and lighter, it increases the number of battery charge-discharge cycles making the battery larger and heavier. The spreadsheet automatically takes these factors into account, as discussed further below, so this tradeoff can be made quickly and easily.

4. Power Analysis

The Power Analysis section takes the power requirements and allocates them to the four operating modes: noneclipse operating, noneclipse non-operating, eclipse operating, and eclipse non-operating. Allocating the power requirements allows the total energy requirement per orbit to be found, which determines the size of the battery. The total energy requirement, along with the battery charging power, also sets the array output power. The analysis is completely determined from the data entered in the Power Requirements so no values can be entered or modified. It is only shown to give the user insight into the calculations instead of performing the computations "behind the scenes" and simply displaying results.

The payload, housekeeping, and thermal power requirements are simply copied down into the appropriate mode. Since EP systems typically draw very high power for only a short time, the EP duty cycle is applied to its power requirement before allocation. If the instantaneous EP power were used, it would force the solar array to be excessively large. The individual power requirements are summed, and the contingency load fraction is applied to find the total instantaneous power requirement.

The time for each mode is found from the payload duty cycle and eclipse operating fraction. The eclipse operating time is found first and is the lesser of the eclipse duration or the total operating time. The eclipse non-operating time is then the remaining portion of the eclipse period. The noneclipse operating time equals the difference between the total and eclipse operating times, up to the duration of the illuminated portion of the orbit. Note, however, that the sum of the eclipse and noneclipse operating time might not sum to the full duty cycle. This occurs if the duty cycle and eclipse operating fraction are not consistent. Finally, the noneclipse non-operating time is simply the remainder of the orbital period.

The instantaneous power is then converted to an energy requirement by multiplying by the time span in each mode. The energy requirement is used to find the power output of the array and the capacity of the battery. The two noneclipse energies are summed to give the array requirement, which is averaged over the illuminated portion of the orbit to find the average power requirement. Note that if the duty cycle is 100% or if power averaging is not used, the average power equals the instantaneous noneclipse operating power. The array power requirement does not directly include the eclipse power; this factors in though the battery charging power.

The battery capacity is the sum of the two eclipse energies plus the supplemental energy needed for power averaging. The noneclipse supplement is the amount of energy the battery must supply during the illuminated portion of the orbit, and equals the difference between the instantaneous operating power and the average array power multiplied by the noneclipse operating time. The total battery capacity is then passed down to the Battery Sizing section, where the charging power is computed.

The average power requirement is added to the battery charging power and the array margin is applied, giving the total power requirement for the array. This value is passed down to the Solar Array Design section to find the size of the solar array.

5. Battery Sizing and Charging Power

Most spacecraft require rechargeable batteries to provide electrical power during launch and eclipse. To ease the power conditioning requirements, it is desirable to have the discharge voltage remain nearly constant until all of the capacity is nearly discharged. This is especially important in partially regulated and unregulated buses. Several factors limit the useful life of a rechargeable battery, including the operating temperature, depth of discharge, and excessive overcharge. Most batteries prefer to operate cool, typically between -5°C and 25°C. If the battery gets too hot, it can chemically degrade the electrolytes. On the other hand, if the battery gets too cold, it retards the chemical process and the battery output drops. Therefore, most batteries have their own radiator to limit the upper temperature, with heaters to prevent them from getting too cold. Repeated cycling to a deep depth of discharge also degrades the battery, but they can tolerate a much larger number of shallow discharges. Therefore, the number of deep discharge cycles is a key design parameter. If the battery is excessively overcharged, it causes the electrolyte to chemically disassociate. While the charge voltage remains constant over most of the charge period, the voltage rises rapidly as the battery reaches a fully charged state. Therefore, overcharge can be limited by controlling the maximum charge voltage. For an excellent discussion of batteries and the process to design them, refer to Agrawal, 1986, p. 347 - 376.

To begin the battery design, the energy storage requirement is copied down from the Power Analysis. To provide full flexibility and give positive control to the user, the user can override the default by entering a different value. However, it should only be changed with considerable forethought and caution. The computed value is the required amount based on the entered power requirements. If a smaller value is entered, the battery will have insufficient capacity to meet the satellite's power requirements. At best this will impact mission operations, and it could lead to the loss of the satellite if the battery becomes fully discharged. On the other side, if a larger value is entered, the battery will have excess capacity. While this is not a serious problem, it adds needless mass to an already heavy component. To prevent an inadvertent change, a warning message is displayed if the entered value does not equal the computed capacity.

The EOL minimum cell voltage determines the number of individual cells which must be linked in series to provide the required bus voltage. As the battery ages, its discharge voltage gradually decreases. To ensure the power requirements are meet over the life of the spacecraft, the batteries are sized by their EOL capability. The EOL voltage is a characteristic of the battery type since it can only drop so far before the cell fails completely. The entered voltage can have any positive value, although 1 volt is fairly common and is the default.

The depth of discharge is perhaps the most important parameter determining the useful life of the battery. A battery can only provide a certain number of charge-discharge cycles for a given depth of discharge. The deeper the battery is discharged, the fewer the number of cycles it can provide. Shallow discharges have a much smaller impact on the battery life. In designing the batteries, the allowable depth of discharge is determined by estimating the number of deep discharge cycles over the design life.

The number of charge-discharge cycles is determined by the number of eclipse periods and the use of power averaging. During eclipse, the battery provides all of the spacecraft power, requiring a deep discharge. Therefore, the minimum number of charge-discharge cycles equals the number of times the satellite passes though eclipse. The eclipse criteria is discussed in the Orbit Data section above. To estimate the number of eclipse cycles, the spreadsheet computes the beta angle and finds the eclipse radius. If the semi-major axis is less than the eclipse radius, the satellite will pass through eclipse on that day. The number of days in eclipse are counted and then multiplied by the number of orbits per day. However, if power averaging is used, the battery will also discharge even if the satellite is not passing through eclipse on that day. To determine if the number of cycles must be adjusted, the depth of discharge for supplemental power is compared to the battery storage capacity. If it is less than 20%, the effect on the battery life will be negligible compared to the eclipse discharge cycles. If, however, the supplemental power requires over half of the battery capacity, the battery effectively has a deep discharge on every orbit. As a result, the number of charge-discharge cycles is increased to equal the total number of orbits. In between these two levels, the extra discharges have a more moderate effect on the battery. The number of cycles is estimated by the average of the number of eclipse cycles and the number of orbits.

The relationship between the number of charge-discharge cycles and the allowable depth of discharge is a fundamental characteristic of the battery. For most batteries, the number of cycles is logarithmically related to the depth of discharge. This means that on a semi-log scale, the graph is nearly linear, so the number of cycles can be converted to a depth of discharge by a simple linear equation. Since nickel hydrogen (NiH_2)batteries are the current industry standard, they were selected as the default battery type. Milden, 1991 presents a graph representative of NiH_2 cells, which was used to find a slope and intercept. To compute the depth of discharge, the spreadsheet substitutes the log of the estimated number of charge-discharge cycles in the equation for a line. The estimate is valid in the range from 1,000 to over 100,000 cycles. From a practical standpoint, however, the depth of discharge was limited to a minimum of 10% and a maximum of 85%.

The next step is to enter some information about the maximum charging voltage per cell, the charger voltage drop, and the charging efficiency. As discussed above, overcharging can damage the battery but can be avoided by limiting the maximum charging voltage. The battery charger is not perfectly efficient and causes a small voltage drop from the applied charge voltage. The battery also has internal losses and dissipates some of the applied energy. These voltages can have any value greater than zero, but all are usually small. The default values for the maximum cell charging voltage and the charger voltage drop are 1.5 and 1.75 volts, respectively, which is typical for NiH_2 batteries. The efficiency is entered as a fraction and must be between 0 and 1. Space-quality batteries usually have efficiencies on the order of 0.90, so this is the default.

To provide redundancy, batteries normally include extra cells. They are far too massive to include a complete spare battery. In addition, batteries rarely fail as a complete unit; instead individual cells fail to an open circuit state. The number of redundant cells can be any positive integer, but is usually limited to only a few cells. The default value gives one extra cell for every five years of design life. The failed cells introduce a voltage drop during both charge and discharge. The number of failed cells and their associated voltage drops must be accounted for when determining the number of cells in series necessary to achieve the desired bus voltage. The voltage drops must be greater than or equal to zero, but are reasonably small. The default voltage drops are 2.5 volts during charging and 1 volt during discharge, which is again fairly typical for NiH_2 batteries.

Battery charge and discharge rates are expressed in multiples of the rated capacity, C, in ampere-hours. A battery discharging at the C rate will expend its rated capacity in one hour. At a rate of 2C, it will take a half hour to discharge the battery. If the charging rate is too low, it can damage the battery. In addition, the charge rate must be sufficiently high to ensure the battery is fully recharged over the illuminated portion of the orbit. Obviously, the higher the charge rate, the faster the battery will return to full charge, but this also means the charging power is higher which in turn requires a larger solar array. The charging power will be minimized if recharging occurs over the entire illuminated portion of the orbit. As it turns out, since the charging rate is expressed as a fraction of the rated capacity, it is actually independent of the capacity. It is found from:

$$r_C = \frac{V_{DB}\ DOD}{t\ \eta\ V_{BC}} \qquad (3.51)$$

where V_{DB} is the eclipse bus voltage, V_{BC} is the maximum charging voltage, DOD is the depth of discharge, η is the charging efficiency, and t is the recharging time, which equals the noneclipse time. The default value is the minimum charge rate which will still fully recharge the battery. However, to prevent damage to the battery, the minimum recharge rate is 0.02, which is a typical lower limit for NiH_2 batteries. Since the default value is the lower limit for the recharge rate, if the user enters a value it must be larger than the default value.

The number of cells in series is determined from the minimum EOL cell voltage and the required eclipse bus voltage. Factoring in the number of failed cells and the associated voltage drops, the number of cells in series is found as:

$$N = \left(\frac{V_{DB} + N_F V_{DD}}{V_D}\right) + N_F \qquad (3.52)$$

where N_F is the number of failed (or redundant) cells, V_{DD} is the bypass diode voltage drop during discharge, V_D is the EOL minimum cell voltage, and V_{DB} is the eclipse bus voltage. Any fractional value is rounded up to the next largest integer. The number of cells cannot be entered by the user since it is completely determined by previous information.

With the number of cells determined, the sheet calculates actual values for the eclipse voltage, the maximum charging voltage, and the required boost voltage for recharging. Since the number of cells might be rounded, the actual eclipse voltage may be slightly different from the specified value, and is given by:

$$V_{DB} = (N - N_F)V_D - N_F V_{DD} \qquad (3.53)$$

where the variables have the same meanings as before. While a battery must be recharged at a higher voltage then the nominal bus voltage, it must be limited to prevent overcharge. The maximum charging voltage is specified by:

$$V_{BC} = (N - N_F)V_{MC} - N_F V_{DC} \qquad (3.54)$$

where V_{MC} is the maximum charging voltage per cell and V_{DC} is the bypass diode voltage drop during charging. The additional charging voltage is provided by a small charge array on the solar panel. The boost voltage it must provide is the difference between the actual bus voltage and the maximum charging voltage, plus any charger voltage drop:

$$V_{CA} = V_{BC} - V_{DB} + V_{CD} \qquad (3.55)$$

where V_{CD} is the charger voltage drop. These voltages are completely determined by previous values and are presented for the user's information.

The capacity of a battery is determined by how long it can provide a given current and is expressed in units of ampere-hours. The battery capacity is computed from:

$$C = \frac{E}{V_{DB}\ DOD} \qquad (3.56)$$

where E is the energy storage requirement. If a dual power bus is selected, then each battery stores half of the required energy and the battery capacity is half the EPS system capacity.

Finally, the battery charging power and recharging time are computed and displayed. The charging power equals the charging current multiplied by the maximum charging voltage:

$$P_{charge} = r_C\, C\, V_{BC} \qquad (3.57)$$

where r_C is the charging rate, C is the battery capacity, and V_{BC} is the maximum charging voltage, as before. The charging time is computed from:

$$t_{recharge} = \frac{E}{P_{charge}\eta} \qquad (3.58)$$

where E is the energy storage requirement and η is the charging efficiency, as before. For a dual power bus, the spreadsheet automatically checks to see if the batteries can be charged sequentially or if they must be charged simultaneously. If, for the specified charging rate, the recharge time is less than half of the noneclipse duration, the spreadsheet assumes the batteries are charged in series. As a result, the EPS recharge time is double that for a single battery, but the charging powers are equal. On the other hand, if the batteries take more than half the orbit to recharge, then they must be recharged at the same time. In this case, the battery and EPS recharge times are the same, but the required charging power will be twice the battery level. The EPS charging power is then passed back up to the Power Analysis section and added into the array power requirement.

6. Solar Cell Data

Before the solar arrays can be designed, the type of solar cell must be selected. The user is given three choices: Silicon (Si), Gallium Arsenide (GaAs), or Indium Phosphide (InP). Since the different cell types have different characteristics, selecting a type will adjust the default values. However, this is really the only affect. Choosing a cell type does not effect the basic calculations. If the user enters solar cell data, the selected cell type is actually irrelevant. For an excellent discussion of solar cells, see Agrawal, 1986, p. 325 - 347.

To ensure realistic values, the defaults were adopted from real cells. The Si and GaAs default values are based on cells produced by Applied Solar Energy Corporation. The InP defaults are for a cell under development by Aerotec Microelectronics. While the default values are based on specific cells, they are representative of the entire class of cells of the same type. The default values for all of the parameters are listed in Table 3-2.

Silicon cells have been extensively used on-orbit. They are readily available and inexpensive, and have a lower density than the other cell types. However, they have the lowest efficiency of the three types, usually between 12% and 14%, and are the most susceptible to radiation degradation. As a result, despite their lower mass, solar arrays made with Si cells are larger and heavier than if the other types are used, but usually cost slightly less.

Table 3.2. Solar Cell Parameters

Parameter	Units	Si	GaAs	InP
Manufacturer		Applied Solar Energy Corporation	Applied Solar Energy Corporation	Aerotec Microelectronics
Cell Size	cm	1 x 1	1 x 1	1 x 1
Mass	g	0.05	0.11	0.0925
Efficiency	%	13	18	16.5
Current - Max Power	Amp	37.0	28.5	31.5
Voltage - Max Power	V	0.50	0.87	0.72
Reference Solar Intensity	W / m^2	1,353	1,353	1,353
Reference Temperature	°C	28	28	28
Temp Coefficient - Current	% / °C	0	0.056	0.068
Temp Coefficient - Voltage	% / °C	-0.443	-0.232	-0.224
Coverglass Thickness	mils	6	3	3
Wiring Loss	V	0.005	0.005	0.005

Gallium Arsenide cells have also been used on orbit, though certainly not as much as Si cells. They have the highest efficiency of the three types and have good radiation tolerance. They are about 50% more expensive than Si cells, although as they are used more the price is dropping.

Finally, Indium Phosphide cells are relative newcomers and have not been used on-orbit in operational spacecraft, although some demonstration cells have been tested. Their efficiency falls between Si and GaAs cells, at about 16.5%. The attractive feature of InP cells is that they are virtually immune to radiation degradation. Radiation doses which degrade GaAs cells by 20%, and Si cells by 30%, decrease the output of InP cells by a mere 5%. For long duration spacecraft or those in orbits which pass though the heart of the van Allen radiation belts, InP cells may prove to be the best choice. They may also prove useful for military satellites which must be hardened to survive nuclear explosions.

To design the array, the physical, electrical, and thermal properties of the solar cells must be identified. Physical properties include the cell size and mass. The size is entered in centimeters and must be greater than zero. The default size is 1 cm x 1 cm for all cell types. The mass is entered in grams, and also must be greater than zero. Default values vary by cell type and are listed in Table 3-2.

For protection from the ambient radiation environment, solar cells are usually shielded with coverglass. Coverglass is reasonably effective at shielding out the larger, more massive protons, but has little effect on the electron radiation. Selection of a thickness depends on the orbital altitude, which determines the radiation environment, and the mission duration. Obviously, thicker coverglass provides more shielding so the radiation degradation is less, allowing the array to be smaller. But the coverglass itself adds mass, and after a certain thickness provides little additional shielding. Therefore, the thickness is usually optimized to provide the lowest overall array mass. Since Si cells are the most susceptible to radiation effects, they normally have more shielding then GaAs or InP cells. The default value for Si cells is 6 mils; only 3 mils is used for the other types. In designing the array, the user is encouraged to try different thicknesses to see the effect on the radiation degradation, array size, and array mass.

The electrical properties of a solar cell include the efficiency, the current and voltage output, and the wiring loss. The efficiency is a measure of the amount of incidence solar energy that is converted to electrical power and is a basic characteristic of the cell type. Si cells have efficiencies between 12% and 14%, GaAs cells range from 18% to 19%, and InP cells have efficiencies on the order of 16.5%. There is a considerable amount of effort into research to increase cell efficiencies, so these values are gradually increasing. For example, in the 1960's, Si cells had efficiencies of 8% to 10%. The next few years promises to see large increases in cell efficiencies. Researchers have recently developed dual junction cells, which effectively stack two solar cells on top of each other, with efficiencies of 25% to 28%. They are now trying to extend the technique to triple and quadruple junction cells, which has the potential to increase efficiencies to the mid 30% range.

Another basic characteristic of solar cells is the relationship between the output current and voltage. As the load on the power subsystem changes, the current and voltage, and therefore the power, also changes. To minimize the size of the solar array, cells are normally operated at or near the so-call "max power" point. The current is also highly dependent on the incident solar energy. Because the Earth's orbit is slightly elliptic, the solar intensity varies cyclically over the year, from a low of 1,309 W/m^2 to a peak of 1,399 W/m^2. Manufactures usually specify the cell output for the mean solar intensity of 1,353 W/m^2.

The wiring loss represents how well the individual solar cells are linked together on the panel. While the loss per cell is quite small, it should still be taken into account because the number of cells is normally large. Any entered value must be greater than or equal to zero. The default value is 0.005 volts and is independent of cell type.

The output of a solar cell depends on its operating temperature. In fact, a change in cell temperature will change the fundamental relationship between the current and voltage. An increase in the operating temperature will slightly increase the current, but it will cause a significant decrease in voltage. The output power characteristics for a solar cell are usually obtained at temperatures between 25°C and 28°C. However, since cells are rarely at this temperature on orbit, it is important to include temperature effects in the array design. This requires specifying not only the current and voltage temperature coefficients, but the reference temperature as well. The reference temperature is entered in degrees Celsius, and obviously must be above absolute zero, or -273.15°C. However, it is almost always around normal room temperature. To prevent inadvertent errors, a warning message is displayed if the entered temperature is outside the range from 15°C and 35°C.

7. Radiation Degradation

The single largest factor limiting the life of the solar array is radiation damage. Since the array must produce the required power at EOL, the radiation environment also has a large influence in sizing the array. Radiation affects both the current and voltage output. Sources of radiation include trapped electrons and protons and energetic protons from solar flares. The van Allen radiation belts are formed by electrons and protons trapped in the Earth's magnetic field. Very energetic protons are emitted from the sun in solar flare eruptions. While the intensity of the trapped radiation is highly altitude dependent, solar flare protons are considered altitude independent.

The space environment includes a wide range of electron and proton energies. To describe the combined effects, the space radiation environment is typically related to an equivalent 1 MeV fluence. Over the past 30 years, NASA has accumulated data on the space radiation environment, and published tables listing the annual equivalent fluence due to both electrons and protons for various altitudes and inclinations (NASA, 1982, p. 6-19 - 6-52). Damage to the solar cells can then be described as the amount of degradation resulting from the equivalent 1 MeV dose. Manufactures can test this under laboratory conditions by exposing the solar cells to given doses of 1 MeV electrons and measuring the degradation in output. This data is then included in the cell's technical data package, usually in graphical form.

The spreadsheet approximates the equivalent 1 MeV fluences for both trapped electrons and trapped protons from look-up tables. The NASA tables were typed into the spreadsheet for various altitudes from the surface of the Earth to beyond GEO and for inclinations from 0° to 90° in 10° increments. The fluences in these two tables assume no shielding, i.e. no coverglass. The inclination and altitude of the final orbit, transferred from the Orbit sheet, is used to enter the tables and extract the four values surrounding the actual orbit values. The equivalent fluences are then estimated by interpolating between these four values, first in altitude and then for the inclination.

The effects of the coverglass are taken into account by applying a knock-down factor to the fluences found above. The NASA tables list not only the unshielded fluences, but include the radiation levels for a wide range of coverglass thicknesses. The data from these tables were used to derive the knock-down factors. Since coverglass has a minimal influence on the electron fluences, the effects could be reasonably approximated by dividing the altitude into two regions. Separate knock-down factors for a given glass thickness were develop for each region. However, protons can be effectively shielded, depending on their energy. Since proton energies significantly vary by altitude, especially depending on whether they are inside or outside the van Allen belts, the coverglass effectiveness was also highly altitude dependent. Therefore, knock-down factors for the protons had to be derived for five separate altitude regions. All of the knock-down factors were entered in another look-up table. The spreadsheet uses the entered coverglass thickness to extract the two knock-down factors for the thicknesses just above and below the actual value. The reduction for the actual coverglass thickness is then found by interpolation. The annual equivalent 1 MeV fluence is found by multiplying the unshielded fluence with the knock-down factor.

The intensity of solar flare protons is independent of altitude, but can be shielded, at least partially, by coverglass. The NASA tables included the annual equivalent 1 MeV fluence due to solar flare protons for a wide range of coverglass thicknesses. This data was entered into yet another look-up table. The spreadsheet uses the entered glass thickness to interpolate the annual equivalent fluence from the data in the look-up table.

The individual fluences for the three radiation sources are summed to find the annual equivalent 1 MeV fluence. The total radiation dose the spacecraft will receive is then found by simply multiplying the annual dose with the design life. To design the arrays, the total radiation dose must be converted to a corresponding degradation in the solar cells.

The output of a solar cell degrades exponentially with the radiation dose. On a logarithmic scale, the plot is nearly, though not quite, linear. As noted above, the technical data from the manufacturer includes a graph of the output current and voltage versus the 1 MeV electron fluences. The graphs for the same three default cells were used to estimate a separate slope and intercept for the current and voltage. Fortunately, the linear approximation is reasonably accurate for the radiation fluences of interest, namely from 10^{13} e-/cm^2 to 10^{17} e-/cm^2. Below a fluence of 10^{13} e-/cm^2, the degradation is negligible for all three cells, and few satellites will receive doses above 10^{17} e-/cm^2. The solar cell degradations were then found by substituting the log of the total fluence into the linear equation.

8. Solar Array Design

Since the output of individual solar cells is so low, they must be joined together in an array to provide the required power. The arrays can either be mounted on the body of the spacecraft or arranged in deployable panels. Body-mounted arrays minimize the overall array mass since the substructure is also the outer skin of the spacecraft. However, the size of the spacecraft limits the size of the array, and thereby the power. In addition, some of the cells will be mounted on the side of the spacecraft facing away from the sun, so only a fraction of the cells produce energy at any time. Deployable panels require their own substructure, but they can be made as large as necessary. They can also track the sun, eliminating the incidence angle losses. For an excellent discussion of solar arrays, see Agrawal, 1986, p. 342 - 347 and p. 376 - 380.

This section assumes the array is a deployed panel. After computing the operating temperature, the array is designed by determining the number of cells connected in series and the number of parallel strings. The size of the charge array is also determined. Finally, the number of panels per wing is entered and the physical size of each panel is calculated.

The power from an array varies over each orbit and throughout the year due to three factors. First, the Earth's orbit about the sun is slightly elliptic, causing a small variation in the solar intensity. Second, the Earth's equatorial plane is inclined to the orbital plane. This causes a variation in the incidence angle between the solar array surface normal and the Sun's rays. Finally, radiation degradation causes the array output to decrease over the design life. The effects of radiation were addressed in the Radiation Degradation section.

The first inputs to this section are the solar intensity and the incidence angle. Over the course of a year, the solar intensity varies from a minimum of 1,309 W/m^2 to a maximum of 1,399 W/m^2, and has a mean of 1,353 W/m^2. Since the array must provide the required power over the entire year, it is normally designed for the worst case condition. Note, however, that the worst case condition depends on both the solar intensity and incidence angle and does not occur at aphelion, where the solar intensity is minimum. Instead, it occurs at summer solstice.

If the array is double-gimbaled, it can track the sun and the incidence angle is always zero. For a single-gimbaled solar array, the incidence angle equals the beta angle. The maximum beta angle equals the orbital inclination plus the obliquity of the ecliptic, which is the angle between the equatorial and ecliptic planes and equals 23.44°. Since the power output varies with the cosine of the incidence angle, arrays are usually double-gimbaled unless the orbit inclination is small. For the default incidence angle, the spreadsheet assumes the array is double-gimbaled if the inclination is 25° or more.

The EOL power requirement is transferred down from the Power Analysis section. Since the value could not be changed there, the user can enter a different power in this section. However, a new power requirement should be entered only with considerable caution and forethought, since the computed value is the actual power needed to meet the identified requirements. If a smaller value is entered, the spacecraft will have insufficient power to perform the mission. If a larger value is used, the array will be larger and more massive than necessary.

The number of array wings can be any positive integer, zero excluded. The total power requirement is equally distributed to each wing. For example, it there are four wings, then each produces a quarter of the total power. Most spacecraft have two wings, one extending from opposite sides of the spacecraft. The default assumes two wings.

Before the array can be designed, its operating temperature must be determined. The output of the solar cells was determined at a reference temperature, however, the on-orbit temperature of the solar array can be very different from the reference. This can have a significant impact on the output power of the array. The operating temperature is determined by the solar absorptance and emittance of the array. By definition, the solar absorptance must be less than one. It must also be greater than the solar cell efficiency, since the cell cannot possibly generate more power than it absorbs. The emittance must be between zero and one, by definition. However, since the two sides of the array are made of different materials, the emittances will probably be different, too. The default values for the solar absorptance and emittances of the front and back are 0.87, 0.82, and 0.85, respectively (Agrawal, 1986, p. 275).

The steady-state operating temperature is found through the heat balance equation, which says the energy in must equal the energy out. Normally, all of the energy is either absorbed or radiated, but a solar array converts part of the absorbed energy to electrical power. Therefore, the solar absorptance must be adjusted to account for this extra energy transfer. The effective solar absorptance is given by:

$$\alpha_{SE} = \alpha_S - F_P \eta \qquad (3.59)$$

where α_{SE} is the effective solar absorptance, α_S is the average solar cell solar absorptance, η is the solar cell efficiency, and F_P is the solar cell packing factor. The packing factor is the ratio of the total active solar cell area to the total substrate area, and represents how efficiently the cells are arranged in the array. Since packing factors are very high, on the order of 99.9%, it is neglected in the calculations. The operating temperature, in Kelvin, is then found from:

$$T_{OP} = \left[\frac{\alpha_{SE}\, S \cos(\alpha_i)}{(\varepsilon_F + \varepsilon_B)\,\sigma} \right] \qquad (3.60)$$

where S is the solar intensity, α_i is the incidence angle, ε_F and ε_B are the emittances of the front and back, and $\sigma = 5.67 \times 10^{-8}$ W/m^2·K^4 is the Stefan-Boltzmann constant. Note that Eq 3.60

computes the operating temperature in Kelvin, which is converted to degrees Celsius by subtracting 273.15.

With the above information, it is now possible to design the main array. The array must not only provide the required power, it must do it at the desired bus voltage. Individual solar cells are linked together in series to achieve the bus voltage. Strings of cells are then connected in parallel to produce the necessary current, and thereby the required power. To ensure sufficient power is generated over the entire life of the spacecraft, the array design is based on the solar cell EOL output. However, the Solar Cell Data section provides information on virgin cells, in other words BOL data.

The output current of the solar array is effected by four main factors: the operating temperature, radiation degradation, the solar intensity and incidence angle, and assembly and other environmental losses. The first three effects were addressed above. The assembly and other environmental losses account the wiring losses in the panel, slip ring, and bus harness, the darkening of the coverglass and adhesive over time, the effects of micrometeoroids and ultraviolet light, and other environmental factors. It is entered as a fraction and must be between zero and one. Since it is a loss, entering a one means there is no decrease, while entering a zero would represent a complete and total degradation. The default value is 0.90, which is representative for today's assembly techniques and for a 10 year design life.

The number of strings connected in parallel is found by comparing the output current from an individual cell, which equals the current for a full string, to the total current needed to achieve the power requirement at the specified bus voltage. The EOL cell current is found by applying the four main losses to the rated BOL output, and is given by:

$$I_{EOL} = I_{MP}\left[1+\delta_I\left(T_{OP}-T_{ref}\right)\right]L_{A/E}L_{RI}\left[\left(S/S_{ref}\right)\cos\left(\alpha_i\right)\right] \qquad (3.61)$$

where I_{MP} is the cell current output at max power, δ_I is the current temperature coefficient, T_{ref} is the reference temperature for the cell parameters, T_{OP} is the solar array operating temperature, $L_{A/E}$ is the assembly and environmental losses, L_{RI} is the radiation degradation in current, S is the actual solar intensity, S_{ref} is the reference solar intensity for the rated cell output, and α_i is the solar incidence angle. The required current per wing is simply the total power requirement per wing divided by the bus voltage, and is found from:

$$I_T = \frac{P}{V_{DI}\ N_{AW}} \tag{3.62}$$

where P is the total power requirement for the array, V_{DI} is the noneclipse bus voltage, and N_{AW} is the number of wings in the array. The number of strings is then easily found by:

$$N_P = \frac{I_T}{I} \tag{3.63}$$

and is round up to the next largest integer.

The output voltage of the solar array is affected by three main factors: the operating temperature, radiation degradation, and the voltage drop from the array to the power distribution bus. Voltage drops are introduced by the slip ring, array wiring harness, and blocking and shunt diodes in the array. Any entered value must be greater than or equal to zero. The default is 2 volts, which is representative of current design and manufacturing methods. Note that, unlike the current, the output voltage is not a function of the incident solar energy. If sufficient solar energy falls on a solar cell, it produces power at a given voltage, which is an intrinsic characteristic of the cell.

The number of cells joined in series is found by comparing the output voltage from an individual cell to the required noneclipse bus voltage. The EOL cell voltage is found by applying the three main losses to the rated BOL output, and is given by:

$$V_{EOL} = \left\{ V_{MP}\left[1 + \delta_V\left(T_{OP} - T_{ref}\right)\right] - \Delta V \right\} L_{RV} \tag{3.64}$$

where V_{MP} is the cell voltage output at max power, δ_V is the voltage temperature coefficient, T_{ref} is the reference temperature for the cell parameters, T_{OP} is the solar array operating temperature, and L_{RV} is the radiation degradation in voltage. The array must not only provide the required bus voltage, but must also overcome the voltage drops from the array to the bus. The number of cells which must be joined in series is:

$$N_S = \frac{V_{DI} + V_{A/B}}{V_{EOL}} \tag{3.65}$$

where V_{DI} is the noneclipse bus voltage, and $V_{A/B}$ is the voltage drop from the array to the bus. Any fractional results are rounded up to the next largest integer.

To allow the user to verify that the array produces the required voltage and power, the actual output for the computed number of cells and strings is displayed. The EOL output can be found directly from the computed values as:

$$\text{EOL Current per Wing} = N_P\, I_{EOL} \qquad (3.66)$$

$$\text{EOL Voltage per Wing} = N_S\, V_{EOL} - V_{A/B} \qquad (3.67)$$

The power is simply the product of the current and voltage. To find the output for the EPS subsystem, the output per wing is multiplied by the number of wings.

It is also common practice to compute the BOL output, which is presented next. To find the BOL output, the losses which occur gradually over the life of the spacecraft must be backed out while keeping the losses present at launch. Therefore, the BOL current and voltage calculations must exclude the radiation degradation. However, the effects of the operating temperature and the array to bus voltage drop remain. The assembly and environmental losses actually fall in-between. The environmental loss occurs over the design life, but assembly losses are present from the beginning. To present a conservative estimate, and because most of the loss is due to assembly, this loss is still applied to the BOL output. Therefore, the BOL array output is given by:

$$\text{BOL Current per Wing} = N_P\, (I_{EOL}/L_{RI}) \qquad (3.68)$$

$$\text{BOL Voltage per Wing} = N_S\, (V_{EOL}/L_{RV}) - V_{A/B} \qquad (3.69)$$

As before, the power is simply the product of the current and voltage and the output for the EPS subsystem is the output per wing multiplied by the number of wings.

Recall that the batteries must be recharged at a voltage above the bus voltage. Therefore, the array must include a small additional panel to provide the boost voltage. Designing the charge array is very analogous to the calculations for the main array, except the boost voltage is used instead of the bus voltage. The number of cells joined in series is found from:

$$N_{SC} = \frac{V_{CA}}{V_{EOL}} \qquad (3.70)$$

where V_{CA} is the boost voltage and V_{EOL} is the cell EOL voltage output. The necessary current for battery charging is found from the battery capacity and the charging rate:

$$I_C = n\, r_C\, C \qquad (3.71)$$

where r_C is the charging rate and C is the battery capacity. The multiplier, n, accounts for whether the batteries are charged sequentially (n=1) or simultaneously (n=2). The number of charging strings is:

$$N_{PC} = \frac{I_C}{I_{EOL}} \qquad (3.72)$$

where I_{EOL} is the cell EOL current output. It is not necessary to have the entire charge array on a dedicated panel or even on the same wing. In fact, if there are two buses, two batteries, and two solar array wings, it is best to split the charge array in half, placing half on each wing. This allows each half-charge array to be dedicated to a particular battery and maintains isolation between the power buses.

The last remaining design feature of the solar array that needs to be determined is its physical size. There are still two free values which must be specified: the number of panels per wing and the length of each panel. The number of panels can be any positive integer, but is usually no more than necessary to accommodate all of the solar cells on reasonably-sized panels. The panel length can have any positive value, but should not be longer than the spacecraft itself. If the panel is too long, it will not fit within the launch vehicle shroud. To find the default values, the spreadsheet uses panels of approximately the same size as the side of the spacecraft. The total number of solar cells on a wing is multiplied by the area of each cell, giving the minimum wing area. The total solar cell area is then divided by the area of the yz face of the spacecraft. The result is rounded up, giving the default number of panels. The default length is the spacecraft height, or z-axis length.

Once these values are identified, the spreadsheet computes the minimum width and panel area that will fit the solar cells. The total solar cell area is divided by the number of panel, giving an approximate panel area. This area must be adjusted, however, since it might rely on partial strings, or even partial cells. The number of complete strings on each panel is found by dividing the approximate panel area by the area for a sting. The result is rounded up to ensure there are no partial strings. The actual panel area is then found by multiplying the number of strings per panel with the area of a string. The panel width is derived from the panel area by dividing by the panel length. The area of each wing and of the entire array is found by simply multiplying the panel area by the number of panels per wing, and then by the number of wings.

9. EPS Mass

The total EPS mass is the sum of the solar array, battery, and power bus masses. These three masses are estimated separately using specific powers and energies. If known, the user can enter the EPS mass directly. Any entered mass must, of course, be greater than zero.

The solar array mass is found from the mass per unit area of the solar cells and the substrate. The individual area masses are then summed and multiplied by the total area of the array. The default solar cell mass per unit area is calculated from their size and mass entered in the Solar Cell Data section. The default substrate area mass is 0.20 g/cm^2, which is a typical value for composite honeycomb. These two values are summed and multiplied by the area of the solar array. If the user enters either or both area masses, they must be greater than zero. On the other hand, the mass of the solar array can be entered directly, and must also be greater than zero.

The battery mass is estimated from its specific energy, which is a measure of the energy storage capacity per unit mass. The specific energy is a characteristic of the battery type and is included with the technical data for the battery. If the user has selected a specific battery, its specific energy can be entered provided it is greater than zero. The default value is 65 W-hrs/kg, a typical value for nickel hydrogen batteries. The battery mass is found by multiplying the specific energy with the stored energy. Note, however, that the stored energy is not the energy storage requirement from the Power Analysis section, since this does not include the battery efficiency, losses, or depth of discharge. Instead, the actual energy stored is the product of the battery capacity, in amp-hours, and the eclipse bus voltage. If the user enters the battery mass directly, it must be greater than zero.

The mass of the power regulation bus is estimated from its specific power, which is the mass per unit of power the bus must distribute. The specific power will depend on the bus type. Regulated buses have more power regulation equipment, so they have higher specific powers. The default values are 9.0 g/W, 9.2 g/W and 21.5 g/W for unregulated, partially regulated, and regulated buses, respectively (Agrawal, 1986, p. 373). Since the bus must be capable of handling the BOL power, the mass is found by multiplying the specific mass by the actual array BOL power output. As before, the user can either enter the specific power and let the spreadsheet compute the mass, or the bus mass can be entered directly. Either value must be greater than zero.

F. THERMAL

The Thermal sheet performs an isothermal, steady-state analysis to estimate the size of the radiator and compute the heater power required to maintain the spacecraft with specified temperature limits. The analysis assumes heat transfer only occurs through the radiator, with all other external surfaces covered with insulation. The solar arrays are excluded from the analysis, since they are thermally isolated from the spacecraft. Note that the operating temperature of the solar arrays is computed in the EPS sheet, where that information is needed. A printout of this sheet is provided on page 162 of Appendix B.

The design of the thermal control system (TCS) is highly dependent on the required temperature limits. Since the satellite is isolated in space, its temperature is determined by simply balancing all heat sources with the heat rejection. This is expressed as:

$$\text{heat stored} = \text{heat in} - \text{heat out} + \text{heat dissipated.} \qquad (3.73)$$

To maintain stable satellite temperatures, with desired bounds, changes in stored heat should be minimized. Therefore, heat out should equal the heat input plus the heat dissipation.

There are many methods for providing thermal control. Passive methods are the simplest and use a combination of thermal coatings and insulation on exterior surfaces. Some equipment requires special cooling, such as batteries and infrared sensor, so they may receive special coatings or be mounted directly on radiators. Other components cannot get too cold and require the use of heaters to supplement the passive coatings. These systems are termed augmented passive or assisted passive systems. High heat dissipation, such as from a large communications satellite, requires an active thermal control system, which uses heat pipes and louvers. By controlling the position of shutters, louvers vary the effective radiator area. The Thermal sheet assumes the spacecraft uses an augmented passive TCS.

The type of attitude stabilization also affects the TCS design. Thermal control for spin stabilized spacecraft is generally simpler than for three axis stabilized spacecraft. The rotation tends to distribute the incident solar intensity even over all surfaces, providing more uniform and stable temperatures. For three-axis stabilized spacecraft, the solar intensity could fall on a single side for protracted periods of time, causing it to get excessively hot. The opposite face receives no solar energy, so it becomes extremely cold. Half an orbit later, the solar orientation of the faces reverses, resulting in wide temperature swings. The TCS must be able to accommodate and smooth out the large temperature gradients. Calculations in the Thermal sheet assume the type of attitude stabilization based on the spacecraft shape specified in the General sheet. Rectangular cylinders are assumed to be three-axis stabilized with radiators on the north and south surfaces. Circular cylinders and spherical shapes are considered spin stabilized with body-mounted radiators.

External energy incident on the spacecraft arises from three primary sources: solar flux, Earth albedo flux, and thermal radiation from the Earth. The thermal analysis only considers the solar energy, since it is by far the dominant term. For very low Earth orbits, the other two terms can cause a small variation in the TCS results. In the event the thermal analysis requires the added fidelity, the equations to compute the incident energy from the albedo flux and infrared radiation are included here. While they cannot be directly added in as external heat sources, they can be factored into the thermal analysis by adjusting the dissipated power.

The Earth reflects a fraction of the incident solar energy back into space as a result of atmospheric scattering and reflection from clouds and Earth surfaces. The albedo flux constant, ϕ_a, is given by:

$$\phi_a = aS \qquad (3.74)$$

where S is the solar flux intensity and a is the albedo coefficient. Ice and snow cover in the high-latitude regions have high reflectances, while the equatorial regions have lower values. As a result, the value of the albedo coefficient varies from 0.1 to 0.8, with a recommended annual mean value of 0.3 ± 0.02. The albedo flux incident on the spacecraft is usually assumed to be constant over the Earth's surface, but the calculation is still complex because it depends on the position of the spacecraft, the orientation of the sun, and the spacecraft altitude. Averaging over the sun's orientation and the surface of a spherical spacecraft, the albedo flux is given by:

$$\phi_A = \frac{Sa}{8}\left(1 - \sqrt{1 - \frac{R_e^2}{\delta^2}}\right) \qquad (3.75)$$

where ϕ_A is the albedo flux at the spacecraft, S is the solar intensity, a is the albedo flux coefficient, R_e is the radius of the Earth, and δ is the radius of the satellite, ie the distance of the satellite from the center of the Earth. (Agrawal, 1986, p. 279)

The Earth absorbs some of the solar energy incident on it. This energy is reemitted as infrared (IR) energy in accordance with the Stefan-Boltzmann law. The mean annual value of the thermal radiation at the Earth's surface is 237 ± 7 W / m^2. Adjusting for space losses, the IR flux at the spacecraft altitude is

$$\phi_T = q_r \frac{R_e^2}{\delta^2} \tag{3.76}$$

where ϕ_T is the IR flux at the spacecraft, $q_r = 237$ W/m^2 is the surface thermal radiation, R_e is the radius of the Earth, and δ is the radius of the satellite as before. (Larson and Wertz, 1992, p. 421)

The Thermal sheet uses information from the General and EPS sheets. The General sheet provides information on the spacecraft shape and dimensions. The amount of dissipated heat the TCS must reject is determined from the satellite power requirements. The payload and housekeeping power is transferred in from the EPS sheet for all four cases of operating and non-operating payload during the eclipse and sunlit periods.

Since all of the heat transfer occurs through the radiator, the first step is to input information on its properties. Solar absorptivity represents the percentage of incident solar energy absorbed by the surface. Infrared emissivity expresses how efficiently the surface radiates infrared energy. The last property is the radiator efficiency, which could be less than one due to surface flaws, view factors, reflections, or other obstructions. The user can enter values for the radiator material or surface coating, but they must be between zero and one. As an added safeguard, a warning is displayed if unusually large (for absorptivity) or small (for emissivity and efficiency) values are entered. Since the TCS must maintain acceptable temperature limits over the entire design life, EOL values should be used. An excellent table listing the BOL and EOL values for absorptivity and emissivity for a wide variety of commonly used materials is given in Agrawal, 1986, page 275. Since the sheet assumes all heat exchange occurs through the radiator, the defaults are typical EOL values for optical solar reflectors (OSR).

The next step is to enter the desired temperature limits. There are only two restrictions on the input values. First, the temperature limits must obviously be above absolute zero Kelvin. Second, the maximum temperature must be higher than the minimum. To minimize TCS mass and heater power requirements, the temperature range should be set as wide as possible while still protecting the performance and reliability of the electronics. Operating temperatures are typically around normal room temperature. Agrawal presents an excellent table on page 266 presenting operating and survival temperature limits for a large selection of common spacecraft components. Another flag is displayed if the entered temperature limits are too far beyond the normal range.

To determine the dissipated power the TCS must reject, the payload and housekeeping power is transferred from the EPS sheet. The power requirements are displayed here for convenience, but cannot be changed in this sheet. If these values need to be adjusted, it must be done in the EPS sheet where they were originally determined. The energy from the payload and housekeeping components can be dissipated in one of two ways: transmitted as useful energy through an antenna or rejected through the radiator as waste heat. In communication satellites, a large portion of the payload power can be transmitted. Besides the payload, energy could also be transmitted in telemetry channels, so the entered value should include all communication links. The power must be greater than zero but less than the total power available. While current amplifiers can achieve efficiencies as high as 70%, it is unusual to have more than half of the payload power transmitted, so a warning is display if the input transmitted power is too large a fraction of the total power. The total power that must be rejected by the TCS is then computed.

To size the radiator, the solar intensity and incidence angle must be specified. As in the EPS section, the solar intensity must be between 1,309 W / m^2 and 1,399 W / m^2 and the incidence angle must be between -90° and 90°. The radiator is normally sized for the worst case hot conditions at EOL. The default values assume the spacecraft is oriented parallel with the equatorial plane, so the worst case angle of incidence is 23.44°. Maximum solar intensity occurs at winter solstice. However, this is only true for the circular and rectangular cylinders. The sun's rays will always be normal to a spherical spacecraft, so the incidence angle is zero. For a sphere, the maximum intensity occurs at perihelion. With all of the terms in the heat balance equation now specified, the radiator area can be computed from:

$$P_{dis} = P_{pay} + P_{hk} - P_{xmit} . \qquad (3.77)$$

$$\varepsilon\sigma T^4 \eta A_r = \alpha_S A S \cos\theta + P_{dis} \qquad (3.78)$$

where A_r is the area of the radiator, A is the exposed area receiving sunlight, ε is the infrared emissivity, σ is the Stefan-Boltzmann constant, T is the radiator temperature in Kelvin, η is the radiator efficiency, α_S is the solar absorptivity, S is the solar intensity, θ is the incidence angle, P_{dis} is the dissipated power, P_{pay} is the payload power, P_{hk} is the housekeeping power, and P_{xmit} is the transmitted power.

With the different spacecraft shapes, the areas in Eq 3.78 must be treated with some care. For cubic spacecraft, ie "box" shaped, the radiator is a flat plate and its area equals the exposed area. For cylindrical spacecraft, the radiator wraps around the entire circumference, but only the side facing the sun receives solar energy. The radiator area varies with the height and the square of the diameter, while the exposed area varies with the height and diameter. In this case, it is best to relate the areas through the height of the radiator. Finally, the sphere could fall into either category, depending on the size of the radiator. If the radiator is sufficiently small so that it fits on the projected area, ie the side facing the sun, then the radiator and exposed areas are equal. If the radiator grows until it wraps onto the back of the sphere, then only the projected area receives any sunlight. If the areas are not adjusted, it would imply the sunlight wraps around the sphere and strikes the back side. The radiator area is found from one of the following equations:

Cubic:

$$A_r = \frac{P_{dis}}{\varepsilon\sigma T^4\eta - \alpha_S S\cos\theta} \tag{3.79a}$$

Cylindrical:

$$h_r = \frac{P_{dis}}{\varepsilon\sigma T^4\eta\pi d - \frac{d}{2}\alpha_S S\cos\theta} \tag{3.79b}$$

Spherical:

$$A_r = \frac{\alpha_S A S\cos\theta + P_{dis}}{\varepsilon\sigma T^4\eta} \tag{3.79c}$$

where h_r is the height of the radiator, d is the spacecraft diameter and the other variables have the same definitions as before. For spherical spacecraft, the Thermal sheet first computes the radiator area by assuming $A_r = A$ in Eq 3.79a. If the computed area exceeds the projected area, the radiator size is recomputed using Eq 3.79c. Finally, for cylindrical satellites, the radiator area is easily found from the height as $A_r = \pi d h_r$.

The radiator is sized to maintain the upper temperature limit during the worst case hot conditions, but the maximum solar intensity is only received over a fraction of the year. The rest of the time, the incident energy can be considerably less than the maximum, making the spacecraft colder. At times during the year, the sun's rays will be parallel to the radiator. Since all heat transfer occurs through the radiator, the spacecraft effectively receives no solar energy. In addition, the spacecraft will encounter eclipse periods and again will receive no sunlight. During these periods, heater power must be applied to ensure critical components do not freeze or become too cold to operate.

To ensure the heater power is adequate to maintain the lower limit, it is computed for the worst case cold condition. The same assumptions on spacecraft orientation used for the hot case also apply here, as does the discussion on the correct areas. For the cylindrical spacecraft, the worst case cold condition occurs during equinox, when the incidence angle goes to 90° from the surface normal. The radiators receive no solar energy. The spherical spacecraft radiators always receive sunlight, except during eclipse, since the rays are always normal to a part of the surface. For the sphere, the solar intensity reaches a minimum at aphelion.

To find the heater power, the minimum temperature must first be calculated. If the minimum temperature is above the lower limit, no extra power is needed. This can happen with spacecraft with high payload power dissipation loads. Essentially, the payload acts like a heater. By rearranging Eq 3.78 to solve for the temperature, the minimum, steady-state temperature is found. Since the area is not factored out of the equation, the same equation applies to all spacecraft shapes:

$$T = \left[\frac{\alpha_S A S \cos\theta + P_{dis}}{\varepsilon \sigma \eta A_r} \right]^{1/4} \qquad (3.80)$$

The temperature found from Eq 3.80 is compared to the lower limit. If the limit is violated, heater power is necessary. To find the required amount of heater power, Eq 3.78 is again rearranged, this time with an extra term:

$$P_{heat} = \varepsilon \sigma T^4 \eta A_r - \alpha_S A S \cos\theta - P_{dis} \qquad (3.81)$$

The computed heater powers are then passed back to the EPS sheet for use in sizing the batteries and solar array.

The mass of the thermal control system is highly dependent on the temperature limits of the spacecraft equipment. It can be accurately estimated only after performing trade studies between the different thermal control designs. However, as a first approximation, the TCS mass can be estimated from the empirical expression:

$$M_T = C_T \times W_D \qquad (3.82)$$

where M_T is the TCS mass, C_T is a scaling coefficient, and W_D is the total of the non-eclipse operating payload power and the housekeeping power (Agrawal, 1986, p. 49). The user can either enter the scaling coefficient and allow the spreadsheet to compute the mass, or the TCS mass can be entered directly. Either value must be greater than zero.

G. MASS

The Mass sheet determines the mass budget, and computes the spacecraft dry mass, the spacecraft bus mass, and a resulting mass margin. It is displayed on page 164 of Appendix B. Except for the subsystem masses that were computed in their respective sheets, all of the calculations are based on parametric or empirical estimates. The user is asked to enter either a scaling coefficient or mass for each subsystem. All entered coefficients must be between zero and one. Obviously, if the subsystem mass is input, it must be greater than zero. The mass budget is highly dependent on information from several other sheets. Since the other sheets cannot be seen while working on the mass, error flags are provided to alert the user when calculation or entry errors occurred on other pages of the design tool. The mass margin is found from the separation mass by subtracting all of the other subsystem or propellant masses. If at any point the required mass exceeds the specified separation mass, an error flag is displayed.

Information from the General, Propellant, EPS, and Thermal sheets is used to develop the mass budget. The General sheet provides the starting point, the separation mass, along with information on the spacecraft shape and design life. As in the Thermal sheet, cubic spacecraft are assumed to be three-axis stabilized while the cylindrical and spherical shapes are considered to be spin stabilized. The Propellant sheet computes all of the propellant masses and motor inert masses. The mass of the EPS and Thermal subsystems are calculated in their respective sheets, and is transferred to the Mass sheet.

Calculations start with the separation mass. The spacecraft dry mass is found by subtracting the expendable propellant masses for both orbit insertion and on-orbit maneuvers. The apogee and perigee motor casing inert masses are subtracted from the dry mass to find the spacecraft bus mass. The bus mass is the dry mass of the spacecraft itself and includes the structure, all subsystems, antennas, solar arrays, other deployables, and the payload and communications system. If no inert casing mass is separated after either orbit insertion maneuver, it is simply set to zero. However, note that there can be expendable propellant masses for the insertion maneuvers without an inert casing mass. This simply means the spacecraft used a unified, non-separable thruster.

The mass of the EPS and Thermal subsystems is computed on their respective sheets and transferred to the mass budget. The values are display here for convenience, but cannot be altered here. If these two masses must be adjusted, the user should make the appropriate changes in the corresponding sheet.

The mass of the propulsion system is found from empirical equations relating the subsystem mass to the design life or total propellant requirements. The propulsion system normally consists of either a reaction control system (RCS) for orbit corrections and attitude control, with separate kick motors for apogee and/or perigee insertion, or a unified system for both on-orbit and insertion maneuvers. The mass of the subsystem varies depending on the types of propellent and thrusters, tank sizes, redundancy, etc. However, empirical relationships provide a good first-order approximation. The equations are different for unified and RCS propulsion systems and for three-axis and spin stabilization. For unified propulsion systems, the mass is found from:

$$M_U = C_U M_{PR} \qquad (3.83)$$

where M_U is the dry mass of a unified system and M_{PR} is the total propellant mass, including apogee and perigee injection requirements. C_U is an empirical coefficient, and equals 0.084 for three-axis stabilization or 0.054 for spin stabilization. For RCS systems, the mass is given by:

Three-axis Stabilized: $$M_{RCS} = \left(0.01 + 0.0115\sqrt{Y}\right)M_{SC} \qquad (3.84a)$$

Spin Stabilized $$M_{RCS} = \left(0.0.006 + 0.007\sqrt{Y}\right)M_{SC} \qquad (3.84b)$$

where M_{RCS} is the mass of the RCS, Y is the design life, and M_{SC} is the beginning of life spacecraft mass. The BOL spacecraft mass is the dry mass plus the propellant for orbit maneuvers, attitude control, and deorbit. It excludes the orbit insertion and reorientation propellant and motor casings. If the user has completed a preliminary propulsion system design, and has selected the tanks, valves, thrusters, and other components, the propulsion subsystem mass can be entered directly. (Agrawal, 1986, p. 44)

The mass of the attitude control subsystem is also found from an empirical equation. It is highly dependent on the type of stabilization, the required attitude control accuracy, component redundancies, and the size and mass of the spacecraft. Some parts of the ACS system are independent of the spacecraft mass, such as the sensors and control computer, while others, like momentum wheels, are highly dependent. Propellant for attitude control is included in the propellant budget, and is not included in the attitude control mass. The total subsystem mass can only be accurately determined after the individual sensors and actuators are selected. However, the following equations provide a good first approximation:

Three-axis Stabilization $$M_{AC} = 65 + 0.022(M_{SC} - 700) \tag{3.85a}$$

Spin Stabilization $$M_{AC} = 31 + 0.027(M_{SC} - 700) \tag{3.85b}$$

where M_{AC} is the mass of the attitude control subsystem and M_{SC} is the spacecraft BOL mass as before. The user can either accept this approximation or can enter an actual attitude control mass directly. (Agrawal, 1986, p. 49)

Several other subsystem masses are estimated by applying an empirical scaling coefficient to an appropriate reference mass. The structural mass is highly dependent on the spacecraft configuration, such as deployables, heavy components, and the skin and face sheet material and thickness. It can be accurately estimated with a preliminary structural analysis only after the spacecraft configuration is finalized. A good first estimate is provided by applying the empirical scaling factor to the spacecraft separation mass. However, the scaling factor is based on an aluminum honeycomb structure. Composite honeycomb panels are increasingly popular and are lighter. The telemetry and control mass is dependent on user requirements and on the complexity of the spacecraft. It typically varies between 2% and 8% of the spacecraft dry mass, with an average of about 4%. The electrical and mechanical integration masses capture the myriad of small components, such as fasteners, washers, adhesives, thermal epoxy, etc, and are computed as a fraction of the spacecraft BOL mass. For each of these subsystems, the user can input a coefficient and allow the worksheet to apply it to the appropriate reference mass. If the particular mass is known, it can be entered directly.

The payload mass includes any remote sensors, receivers, transmitters, communications links, and antennas. It is entirely dependent on mission requirements, and in fact, is often a design requirement. However, it typically falls between 15% and 50% of the spacecraft dry mass (Larson and Wertz, 1992, p. 301). The worksheet uses a default value of 30%. Once again, the user can either enter a new payload fraction or input the payload mass directly.

With the mass of each subsystem set, the remainder of the mass is allocated to margin. The mass margin is found by simply subtracting the individual subsystem, propellant, and inert motor casing masses from the separation mass. If at any time the margin goes negative, in other words, the required mass exceeds the specified separation mass, an error message is displayed.

In this case, the separation mass should be increased or the requirements levied on the spacecraft should be reduced. The margin is also computed as a percentage of the spacecraft dry mass. For a preliminary design, the mass margin should be at least 10%. If the margin percentage drops too low, a warning message is displayed.

IV. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

A. SUMMARY

This thesis conducts an industry survey of spacecraft integrated design tools, and develops a new, spreadsheet-based integrated design tool intended for a single user to quickly develop a preliminary design and conduct trade studies.

The survey revealed there are essentially two different types of design tools in use by commercial space companies: distributed spreadsheet-based tools and stand-alone tools. Spreadsheet based tools offer many inherent benefits. They are widely available and commonly used, yet are very powerful and extremely flexible. They are excellent at performing trade studies, but do not perform optimization or simulation. Stand-alone software tools can be programmed to do almost anything, offering virtually limitless power. However, they tend to be large and complex programs, which take a long time to develop and require a standing team to maintain and upgrade.

Of the seven tools surveyed, only two are available to the public. Microcosm and CTek will gladly sell the commercial applications to anyone. Aerospace, JPL, and especially TRW consider their distributed spread sheets proprietary, and will not provide them to outside organizations. CalTech is willing to share the tools they developed, but only with other academic or governmental organizations.

Four spreadsheet-based tools were surveyed. Three of these, Aerospace, JPL, and TRW, have very similar design tools providing a set of distributed subsystem models. Each subsystem is modeled in separate spreadsheets that are linked together with a network serve, empowering collaborative design and the exchange of information. All three tools were found to be powerful and flexible, yet easy to use. The models include non-technical factors such as cost, schedule, and risk. They are great at trade studies, but do not perform optimization. The forth, CalTech, is developing a set of design tools based on spreadsheets and other common software applications. These tools facilitate the exchange of information between designers, provide a simple user friendly method to model satellites and create representative drawings, and perform low thrust trajectory optimization. The CalTech tools are certainly scaled down from the large distributed spreadsheet tools, but are still quite powerful and flexible. They are based on common software and are easily maintained or upgraded. However, none of the CalTech tools address cost or schedule, although it could be added.

Three different stand-alone tools were reviewed. Lockheed Martin developed VIS to provide a simulation environment for modeling space systems, constellations, and systems of systems to solve intelligence problems. VIS enables virtual design, allowing great insight into the results and performance of a particular design. VIS is extremely powerful, providing virtually limitless fidelity. Despite its complexity, it is still flexible and adaptable. However, VIS is more of a simulation tool than a design tool. Microcosm, Inc. markets the SMAD software program, which implements the equations and design philosophy of the popular Larson and Wertz textbook of the same name. The program includes good descriptions, explanations, and definitions for all of the design parameters. It is very easy to use but is overly simplified. The SMAD software provides a nice, undergraduate-level tutorial on spacecraft design and the design process. Unfortunately, it is not fully integrated, hampering trade studies. The executable code is not user-accessible, so it is not flexible and cannot be upgraded. CTek has recently marketed GENSAT, a general-purpose systems engineering software environment supporting all phases of spacecraft design, manufacture, deployment, and operation. It directly captures program requirements in expert rules and fuses them with powerful modeling, simulation, and optimization tools. GENSAT is extremely impressive and powerful. Its open software architecture is highly flexible and is used to quickly evolve the spacecraft design to very detailed levels. Optimization and simulation functions are directly integrated in the GENSAT environment.

All of the design tools are intended to empower the design engineer. Each one creates a software environment providing computational facilities, automating the exchange of information among members of a design team, and maintaining configuration control. The tools target the preliminary and conceptual design phase, but a few extend beyond design into manufacturing, test, and deployment. To keep pace with the rapidly advancing space technology, all of the design tools except SMAD were very flexible, adaptable, and easily upgraded.

In addition to conducting an industry survey, the thesis developed an integrated software tool for performing a preliminary design. It is intended for a single user such as an engineering manager or student. To capitalize on their inherent benefits, spreadsheets were selected as the best software medium to host the design tool. The tool is very user friendly and easy to use, yet provides a good level of accuracy. Each individual subsystem or aspect of the design is modeled on a separate page of the worksheet. The pages are fully integrated and automatically share all necessary information. This makes trade studies very easy to perform. Default values are provided for all input parameters, and all entered values are checked for validity and reasonableness. The default values provide a sample design of a geostationary communications satellite, demonstrating the features of the design tool and the design process. However, the design tool does not address non-technical factors, such as cost, schedule, or risk.

B. CONCLUSIONS

GENSAT is a perfect match to the NPS curriculum. It can be distributed across the full breadth of the Space Systems curriculum and will benefit both the engineering and operations students. GENSAT provides features, capabilities, and integrated applications that address the entire spectrum of design activities. The individual applications can be incorporated into the instructional material of the corresponding course. By the later quarters, the entire GENSAT system will be covered, so the students can directly apply it to the final design projects. This will improve the quality of the final designs, teach the students about design tools and the design process, and through GENSAT's simulation capabilities provide insight into the success and performance of the design. Because GENSAT is developed through collaboration with space professionals throughout the industry, CTek is eager to work with companies and academic institutions. CTek and NPS have recently reached a teaming agreement, which provides 8 GENSAT seats to NPS. GENSAT is the state-of-the-art integrated design tool, and will place NPS at the forefront of this relatively new but increasingly important design technology.

The CalTech tools are also an excellent match to the Space Systems curriculum. Several of the tools pull together the entire design process in a single, easy-to-use tool. RPET and ICETOP are particularly interesting. RPET addresses the interrelationships of the data requirements between individual subsystems and designers. ICETOP performs optimization of electric propulsion and low thrust trajectories. Both of these topics are currently missing in the Space Systems curriculum. Adopting these programs will fill the gaps and form a basis of instruction.

The distributed spreadsheets from Aerospace, JPL, and TRW would benefit NPS, but are not a perfect match. The best application would be the final group design project. However, that class is not currently structured as a series of collaborative group sessions. In addition, use of the tool relies more heavily on the experience and expertise of the design engineers, which the students normally lack. Finally, the component databases, which ground the design in reality and significantly improve the fidelity, are considered proprietary and would not be provided with the distributed models. However, the curriculum and structure of the final design project could be adjusted to incorporate one of these tools. Using the tools would emphasize the collaborative design process, automate some of the more tedious calculations, and provide important insight into a type of design tool widely used in industry.

VIS would also be useful in the curriculum, but is not a perfect match. Unlike the other tools, which are copied and transferred to the school, NPS would only have access to VIS. This means Lockheed Martin would maintain and upgrade the software. By implementing their designs in VIS, the students would gain invaluable feedback on their success and performance. VIS would be particularly beneficial to the operations curriculum. It would allow them to perform operational analysis and war gaming exercises for existing or proposed satellites, constellations, or systems of systems.

Finally, the SMAD software is not a good match to the NPS course work. While it is simple and easy to use, it is far too simplistic for the depth of detail in the space systems curriculum. The program is better suited to the undergraduate level.

C. RECOMMENDATIONS

As more companies adopt the concurrent engineering process, integrated design tools will become increasingly common. Therefore, the NPS curriculum needs to include instruction and experience on these tools, their application, and their use. GENSAT is the perfect choice. Through the partnering arrangement, CTek has provided access to the GENSAT environment. NPS should maintain, and even strengthen, the relationship with CTek. The various design features and capabilities should be added to the course material throughout the curriculum. To facilitate the instruction, several faculty members should become proficient in GENSAT. A faculty member should also be responsible for developing and maintaining the component database, although the students can assist in this effort. During the final group design project, the students conduct an industry survey of current and projected spacecraft components. This information could be provided to the faculty maintainer.

The new spacecraft integrated design tool, developed as part of this thesis, would benefit the final design projects – both individual and group. Although clearly not as powerful as GENSAT, the design tool will allow the students to perform quick yet accurate preliminary designs and trade studies with the associated computational overhead and complexity. It should be distributed to the students and incorporated into the course content. The design tool can also be advanced and extended, providing research opportunities for a future thesis.

The CalTech tools should also be obtained, even in addition to GENSAT. These tools can be used by individual students, and address two of the few topics missing from the curriculum. They would allow the students to conduct preliminary designs and trade studies without having to deal with the overhead and complexity of the GENSAT system. CalTech has already expressed a willingness to provide the tools. They have offered to come to NPS to provide an introduction, give a tutorial, and discuss design tools and their development. NPS should accept their offer.

With GENSAT, access to the large distributed spreadsheets adds little benefit. Unlike GENSAT, which can be applied throughout the curriculum, the distributed spreadsheets would only benefit the final group design. Acquiring the distributed models would provide insight and experience into a different type of design tool that is widely used. This would provide experience with multiple types of tools, which would enhance the education. However, they would really add little capability, and do not match the current format of the design class. Even if the tools were obtained, they would not include the component databases, which are an important part of the tool. Therefore, these tools are not recommended. If NPS chooses to pursue one of these design tools, it should do so through Aerospace. They were the most open, responsive, and supportive of the three organizations, and they do not consider the entire model proprietary.

The SMAD software program is too simplistic for the NPS instructional level. It is not a good match, and is not recommended.

There are several opportunities for future research. First, NPS should continue to pulse the aerospace industry to stay abreast of advances in concurrent engineering and integrated design tools. Dialogs should be maintained with Aerospace, JPL, and TRW to learn about any updates and future plans for their distributed models. NPS should also track the NASA ISE effort for impacts and advances to the design process and distributed, virtual collaborative design. For the spacecraft integrated design tool, future work can begin where this thesis ends.

Additional sheets can be created for other subsystems, such as ADCS and structures. Other design factors, such as cost, schedule, risk, reliability, etc, can be incorporated into the model. Finally, a database of spacecraft components could be developed and fused with the design tool.

APPENDIX A. POINTS OF CONTACT

Information on the concurrent engineering process and the design tools were provided by specific points of contact (POC) within each company. For assistance in obtaining additional information on the surveyed tools, the POCs are identified below.

The Aerospace Corporation	Stephen Presley	(310) 336-2448
2350 E. El Segundo Boulevard	stephen.p.presley@aero.org	
El Segundo, CA 90245-4691	Andrew Dawdy	(310) 336-6134
http://www.aero.org	andrew.dawdy@aero.org	
	Joseph Aguilar	(310) 336-2179
	joseph.aguilar@aero.org	
	David Bearden, Ph.D	(310) 336-5852
	david.bearden@aero.org	
Computational Technologies, Inc.	David Russel, Ph.D.	(408) 556-9130
2797 Park Avenue	DavidR@ctek.com	
Suite 102	Gary Stanley, Ph.D.	(408) 556-9130
Santa Clara, CA 95050	GaryS@ctek.com	
http://www.ctek.com	James Woolley, Ph.D.	(408) 556-9130
	jimw@ctek.com	
Jet Propulsion Laboratory	Bob Oberto	(818) 354-5608
California Institute of Technology	robert.e.oberto@jpl.nasa.gov	
4800 Oak Grove Drive	Steve Wall	(818) 354-7424
Pasadena, CA 91109-8099	Jeff Smith	(818) 354-1064
http://www.pdc.jpl.nasa.gov		
- California Institute of Technology	Joel Sercel	(818) 354-4044
	sercel@earthlink.net	
Lockheed Martin	Henry Miller	(408) 742-4049
P.O. Box 3504, B156	Mike Sorah	(408) 742-8124
Sunnyvale, CA 94088-3504		

Microcosm, Inc. 2377 Crenshaw Boulevard, Suite 350 Torrance, CA 90501 http://www.smad.com	John Collins	(310) 320-0555
Naval Research Laboratory Spacecraft Engineering Department Code 8200 4555 Overlook Avenue Washington D.C. 20375-5355	Mike Brown	(202) 767-2851
TRW One Space Park Redondo Beach, CA 90278	Julie Heim julie.heim@trw.com	(310) 501-9041

APPENDIX B. PRINTOUT OF DESIGN TOOL WORKSHEETS

General Sheet

	Input		Calculated	
Design Life		yrs	7.0000	yrs
Separation Mass		kg	2,500.0000	kg
- see table below				
Spacecraft Shape	Rectangular Cylinder ▼			
Spacecraft Size - see table below				
Length (x)		m	2.0000	m
Width (y)		m	2.0000	m
Height (z)		m	2.0000	m
Moments of Inertia				
MOI_x			1,666.6667	kg m²
MOI_y			1,666.6667	kg m²
MOI_z			1,666.6667	kg m²

Fairing Sizes and Throw Weights for Various Launch Vehicles

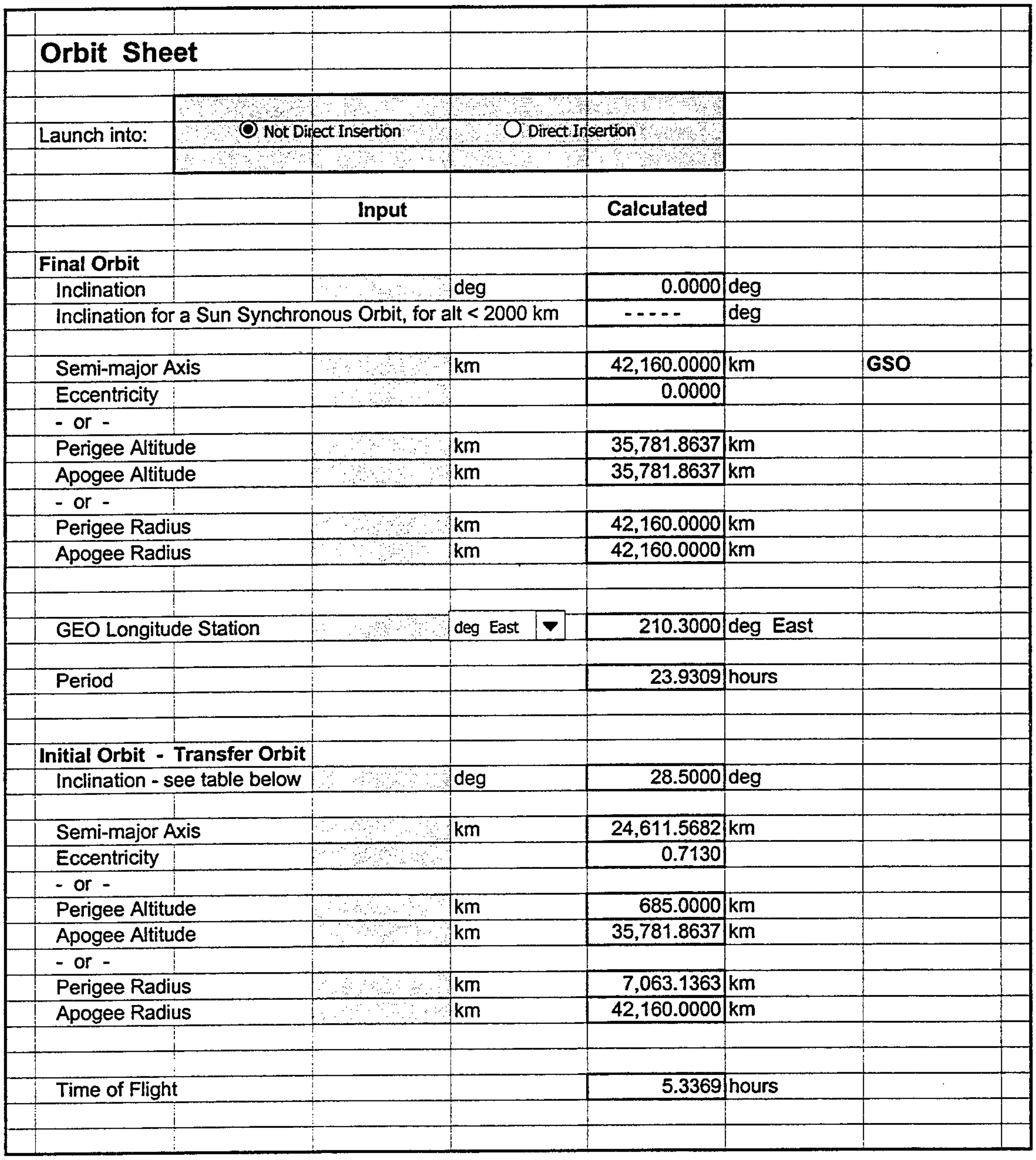
[Source: Agrawal (1986, p24-31), Larson and Wertz (1992, p674-675), and Sutton (1992, p15, 146-147)]

	Maximum Throw Weight			Maximum Fairing Envelopes	
Launch System	LEO [1]	GTO [2]	GEO [3]	Diameter	Length
	kg	kg	kg	m	m
Space Shuttle	24,400	5,900	2,360	4.6	18.3
Atlas	8,390	3,490	1,050	4.2	9.7
Delta II	5,045	1,820	910	2.8	5.7
Titan II	2,150	-----	-----	2.8	9.0
Titan III	14,400	5,000	1,360	3.6	16.0
Titan IV	21,645	6,350	4,540	4.5	26.0
Pegasus	445	125	-----	1.2	1.9
Scout	525	110	-----	1.2	1.7
Taurus	1,450	375	-----	1.2	3.6
Ariane 4	18,000	6,800	2,478	3.6	12.4
Proton	20,000	5,500	2,200	4.1	7.5
Energia	90,000		18,000	5.5	37.0
Zenit 2	13,740	4,300	4,100	3.3	9.0

[1] LEO: 28.5° inclination circular orbit at 185 km altitude

[2] GTO: Geosynchronous Transfer Orbit

[3] GEO: Geosynchronous Earth Orbit



Orbit Sheet

Launch into: ● Not Direct Insertion ○ Direct Insertion

	Input		Calculated		
Final Orbit					
Inclination		deg	0.0000	deg	
Inclination for a Sun Synchronous Orbit, for alt < 2000 km			-----	deg	
Semi-major Axis		km	42,160.0000	km	GSO
Eccentricity			0.0000		
- or -					
Perigee Altitude		km	35,781.8637	km	
Apogee Altitude		km	35,781.8637	km	
- or -					
Perigee Radius		km	42,160.0000	km	
Apogee Radius		km	42,160.0000	km	
GEO Longitude Station		deg East ▼	210.3000	deg East	
Period			23.9309	hours	
Initial Orbit - Transfer Orbit					
Inclination - see table below		deg	28.5000	deg	
Semi-major Axis		km	24,611.5682	km	
Eccentricity			0.7130		
- or -					
Perigee Altitude		km	685.0000	km	
Apogee Altitude		km	35,781.8637	km	
- or -					
Perigee Radius		km	7,063.1363	km	
Apogee Radius		km	42,160.0000	km	
Time of Flight			5.3369	hours	

Orbit Sheet

Intermediate Orbit - Not Necessary

Hohmann Transfer from Perigee of Initial Orbit to Apogee of Final Orbit

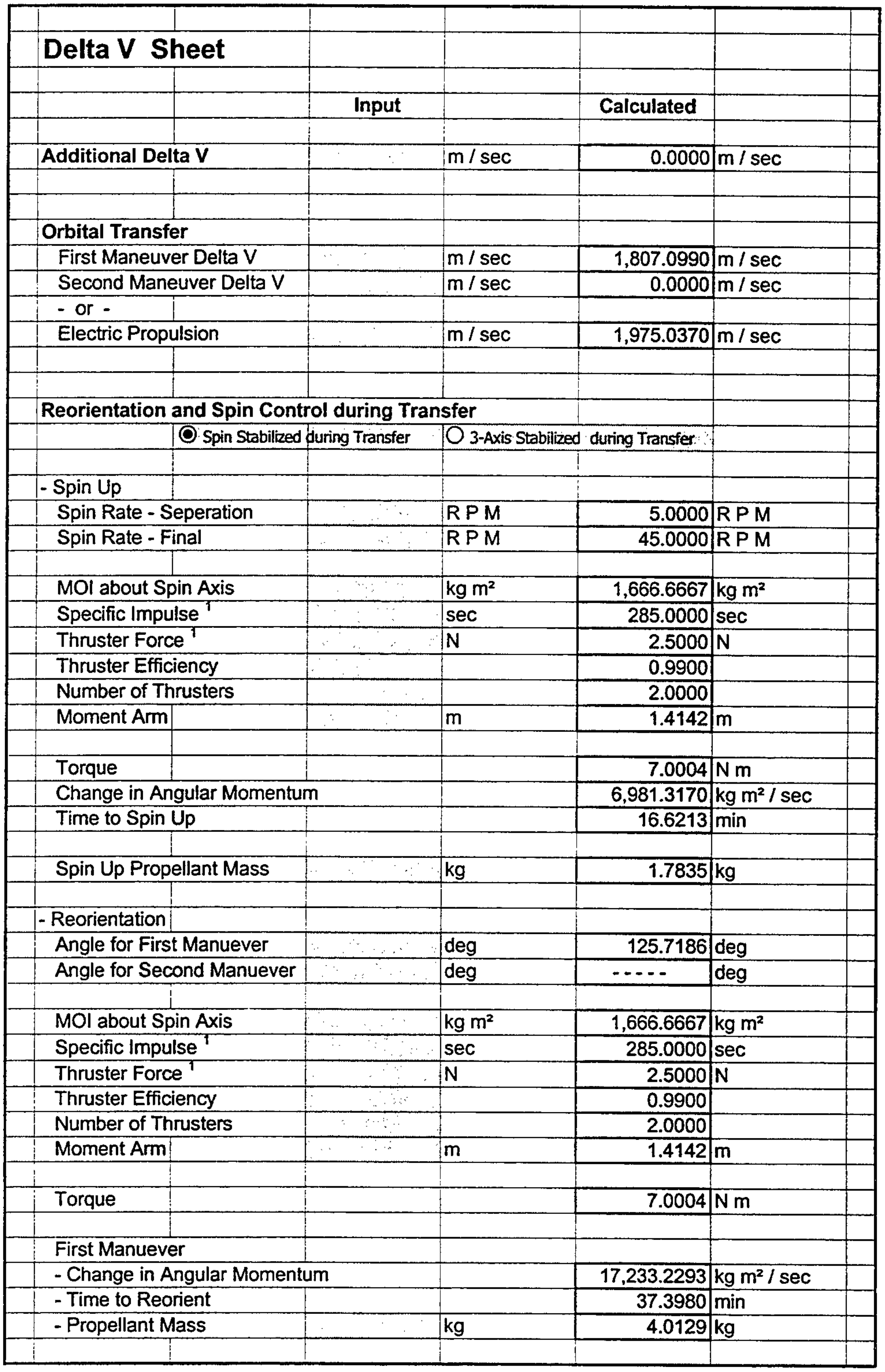
Inclination Change			
- at First Maneuver	%	-----	%
- at Second Maneuve		-----	%
Inclination		-----	
Semi-major Axis		-----	km
Eccentricity		-----	
- or -			
Altitude of First Maneuver		-----	km
Altitude of Second Maneuver		-----	km
- or -			
Radius of First Maneuver		-----	km
Radius of Second Maneuver		-----	km
Time of Flight		-----	

Orbit Sheet

Inclination Limits for Various Launch Sites

[Source: Larson and Wertz (1992, p680-681) and Vallado (1997, p296-297)]

				Launch Azimuth		Inclination	
Launch Site	Country	Lat	Long	Min	Max	Max	Min
		deg North	deg East	deg	deg	deg	deg
Cape Kennedy	USA	28.5000	279.4500	37.0000	112.0000	58.0698	28.5000
Vandenburg	USA	34.6000	239.4000	147.0000	201.0000	107.1567	63.3646
Wallops	USA	37.8500	284.5333	30.0000	125.0000	66.7459	37.8500
Kourou	Arianspace	5.2000	307.2000	340.0000	100.0000	109.9142	5.2000
Plesetsk	Russia	62.8000	40.6000	330.0000	90.0000	103.2117	62.8000
Kapustin Yar	Russia	48.4000	45.8000	350.0000	90.0000	96.6203	48.4000
Tyuratam (Baikonur)	Russia	45.6000	63.4000	340.0000	90.0000	103.8452	45.6000
San Marco	Italy	-2.9333	40.2000	50.0000	150.0000	60.0433	40.2000
Sriharikota	India	13.7000	80.2500	100.0000	290.0000	155.9173	16.9048
Shuang-Ch'Eng-Tzu	China	40.4167	99.8333	350.0000	120.0000	97.5971	40.4167
Xichang	China	28.2500	102.2000	94.0000	105.0000	31.6930	28.5087
Tai-yuan	China	37.7667	112.5000	90.0000	190.0000	97.8900	37.7667
Kagoshima	Japan	31.2333	131.0833	20.0000	150.0000	72.9954	31.2333
Woomera	Australia / US	-30.9500	136.5000	350.0000	15.0000	98.5645	77.1754
Yavne	Israel	31.5167	34.4500	350.0000	120.0000	98.5130	31.5167



Delta V Sheet

	Input		Calculated	
Additional Delta V		m / sec	0.0000	m / sec
Orbital Transfer				
First Maneuver Delta V		m / sec	1,807.0990	m / sec
Second Maneuver Delta V		m / sec	0.0000	m / sec
- or -				
Electric Propulsion		m / sec	1,975.0370	m / sec
Reorientation and Spin Control during Transfer				
● Spin Stabilized during Transfer		○ 3-Axis Stabilized during Transfer		
- Spin Up				
Spin Rate - Seperation		R P M	5.0000	R P M
Spin Rate - Final		R P M	45.0000	R P M
MOI about Spin Axis		kg m²	1,666.6667	kg m²
Specific Impulse [1]		sec	285.0000	sec
Thruster Force [1]		N	2.5000	N
Thruster Efficiency			0.9900	
Number of Thrusters			2.0000	
Moment Arm		m	1.4142	m
Torque			7.0004	N m
Change in Angular Momentum			6,981.3170	kg m² / sec
Time to Spin Up			16.6213	min
Spin Up Propellant Mass		kg	1.7835	kg
- Reorientation				
Angle for First Manuever		deg	125.7186	deg
Angle for Second Manuever		deg	-----	deg
MOI about Spin Axis		kg m²	1,666.6667	kg m²
Specific Impulse [1]		sec	285.0000	sec
Thruster Force [1]		N	2.5000	N
Thruster Efficiency			0.9900	
Number of Thrusters			2.0000	
Moment Arm		m	1.4142	m
Torque			7.0004	N m
First Manuever				
- Change in Angular Momentum			17,233.2293	kg m² / sec
- Time to Reorient			37.3980	min
- Propellant Mass		kg	4.0129	kg

Delta V Sheet

Second Manuever				
- Change in Angular Momentum			-----	kg m² / sec
- Time to Reorient			-----	sec
- Propellant Mass		kg	0.0000	kg
Reorientation Propellant Mass		kg	4.0129	kg
- Final Reorient / Spin Down				
MOI about Spin Axis		kg m²	1,666.6667	kg m²
Specific Impulse [1]		sec	285.0000	sec
Thruster Force [1]		N	2.5000	N
Thruster Efficiency			0.9900	
Number of Thrusters			2.0000	
Moment Arm		m	1.4142	m
Torque			7.0004	N m
Change in Angular Momentum			7,853.9816	kg m² / sec
Time to Spin Down			18.6990	min
Spin Down Propellant Mass		kg	2.0064	kg
- Attitude / Nutation Control		kg	15.0000	kg
- Total Propellant Mass		kg	22.8028	kg
[1] See table at the bottom of the Propellant sheet				
Repositioning				
Number of Repositionings			1.0000	
Repositioning Angle		deg	180.0000	deg
Repositioning Time		days	30.0000	days
Delta V per Reposition			34.0662	m / sec
Repositioning Delta V		m / sec	34.0662	m / sec
End-of-Life Disposal				
- Disposal Orbit				
Increase Semi-Major Axis by:		km	500.0000	km
Change Semi-Major Axis to:		km	42,660.0000	km
- or -				
- Deorbit				
New Perigee Altitude		km		km
New Perigee Radius		km		km
EOL Delta V		m / sec	18.0724	m / sec

Delta V Sheet

GEO North - South Stationkeeping			
Allowed Inclination Variation	deg	0.1000	deg
Inclination Drift Rate	deg / yr	0.8475	deg / yr
Delta V per Maneuver		10.7331	m / sec
Average Time Between Maneuvers		86.1947	days
Number of Maneuvers		29.0000	
Total NS Delta V	m / sec	311.2603	m / sec
GEO East - West Stationkeeping			
Allowed Longitude Variation	deg	0.1000	deg
Nearest Stable Longitude		255.3000	deg
Delta V per Year		1.7350	m / sec per yr
Average Time Between Maneuvers		30.8607	days
Number of Maneuvers		82.0000	
Total EW Delta V	m / sec	12.1450	m / sec
Atmospheric Drag	**- Drag Negligible**		
Spacecraft Area	m²	17.6192	m²
Spacecraft Mass	kg	1,214.9386	kg
Coefficient of Drag		2.2000	
Ballistic Coefficient		31.3434	kg / m²
Delta V per Year		0.0000	m / sec per year
Total Delta V	m / sec	0.0000	m / sec

Propellant Sheet

Propellant Mass Calculations

Maneuver	Mono	Bi	EP	Gas	Solid	Delta V	Specific Impulse [1]	Efficiency	Delta Mass	Mass
						m / sec	sec		kg	kg
Seperation Mass										2,500.0000
Reorient / Spin Control	○	●	○	○					22.8028	2,477.1972
First Manuever	○	○	○	○	●	1,807.0990	285.0000	1.0000	1,179.2586	1,297.9386
- Less Motor Casing									83.0000	1,214.9386
Second Manuever	○	●		○	○	0.0000	300.0000	1.0000	0.0000	1,214.9386
- Less Motor Casing									0.0000	1,214.9386
Station Keeping	○	●	○	○		323.4053	285.0000	0.9500	139.6973	1,075.2413
Repositioning	○	●	○	○		34.0662	300.0000	1.0000	12.3745	1,062.8667
EOL Disposal	○	●	○	○		18.0724	300.0000	1.0000	6.5069	1,056.3599
Additional Delta V	○	●	○	○		0.0000	285.0000	0.9500	0.0000	1,056.3599
Attitude Control	○	●	○	○					15.8579	1,040.5020
Margin									19.7239	1,020.7781
Residual									10.8482	1,009.9299
TOTAL PROPELLANT MASS									1,407.0701	

Propellant Sheet

Propellant Mass Allocation to Tanks

							Tank:	1	2
Maneuver	Mono	Bi	EP	Gas	Solid	Delta Mass	Mixture Ratio[2]	Mono / Biprop Fuel	Biprop Oxidizer
								kg	kg
Reorient / Spin Control		X				22.8028	1.6400	8.6374	14.1654
First Manuever					X	1,179.2586	- - - - -		
Second Manuever		X				0.0000	1.6400	0.0000	0.0000
Station Keeping		X				139.6973	1.6400	52.9156	86.7816
Repositioning		X				12.3745	1.6400	4.6873	7.6872
EOL Disposal		X				6.5069	1.6400	2.4647	4.0421
Additional Delta V		X				0.0000	1.6400	0.0000	0.0000
Attitude Control		X				15.8579	1.6400	6.0068	9.8511
SUBTOTAL								74.7119	122.5275
							For Each Tank		
Margin Fraction							0.1000	0.1000	0.1000
Margin Mass								7.4712	12.2527
Residual Fraction							0.0500	0.0500	0.0500
Residual Mass								4.1092	6.7390
TOTAL								86.2922	141.5192

Propellant Sheet

Propellant Mass Allocatio

Maneuver	3 EP kg	4 Cold Gas kg	5 Solid 1 kg	6 Solid 2 kg
Reorient / Spin Control				
First Manuever			1,179.2586	
Second Manuever				
Station Keeping				
Repositioning				
EOL Disposal				
Additional Delta V				
Attitude Control				
SUBTOTAL	0.0000	0.0000	1,179.2586	0.0000
Margin Fraction	0.1000	0.1000	0.0000	0.0000
Margin Mass	0.0000	0.0000	0.0000	0.0000
Residual Fraction	0.0500	0.0500	0.0000	0.0000
Residual Mass	0.0000	0.0000	0.0000	0.0000
TOTAL	0.0000	0.0000	1,179.2586	0.0000

Propellant Sheet

Propellant Mass Allocatio

Maneuver	7	8	9	10
	kg	kg	kg	kg
Reorient / Spin Control				
First Manuever				
Second Manuever				
Station Keeping				
Repositioning				
EOL Disposal				
Additional Delta V				
Attitude Control				
SUBTOTAL	0.0000	0.0000	0.0000	0.0000
Margin Fraction	0.1000	0.1000	0.1000	0.1000
Margin Mass	0.0000	0.0000	0.0000	0.0000
Residual Fraction	0.0500	0.0500	0.0500	0.0500
Residual Mass	0.0000	0.0000	0.0000	0.0000
TOTAL	0.0000	0.0000	0.0000	0.0000

Propellant Sheet

[1] **Specific Impulse and Thrust Ranges for Spacecraft Propulsion Systems**

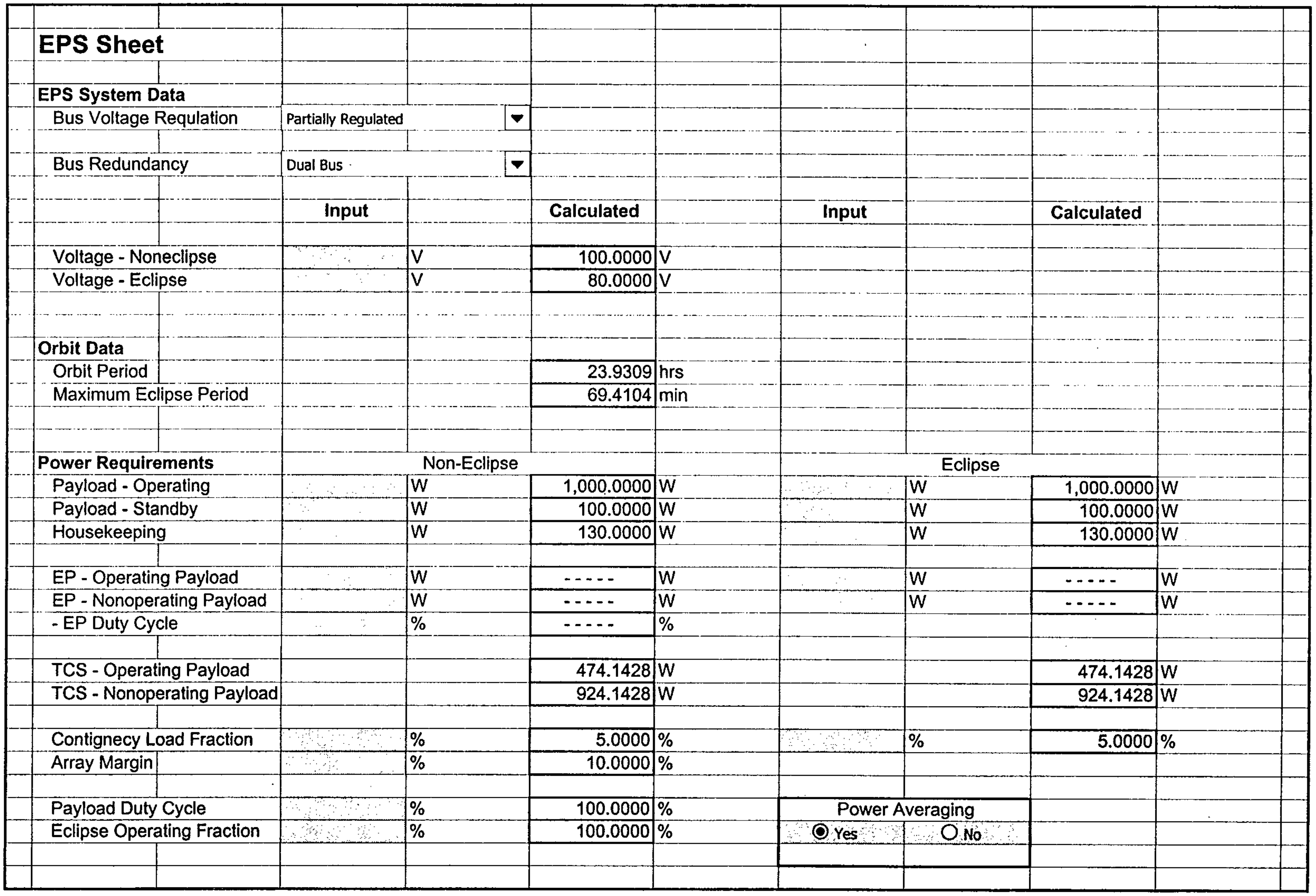
[Source: Larson and Wertz (1992, p644-645), and Sutton (1992, p567)]

Propulsion System Type	Isp Range	Thrust Range	Thruster Efficiency Range	Specific Power
	sec	N	%	W / mN
Cold Gas	50 - 75	0.05 - 200		
Solid	280 - 300	50 - 5,000,000		
Monopropellant	150 - 225	0.05 - 0.5		
Bipropellant	250 - 450	5 - 5,000,000		
Electrothermal				
- Resistojet	150 - 700	0.005 - 0.5	65 - 90	0.5 - 6
- Arcjet	450 - 1,500	0.05 - 5	30 - 50	2 - 3
Electrostatic				
- Ion	1,500 - 6,000	0.000005 - 0.1	60 - 80	10 - 70
- Colloid	1,200	0.000005 - 0.05		
Electromagnetic				
- Magnetoplasmadynamic	1,000 - 8,000	25 - 200	30 - 50	- 100
- Pulsed Plasma	1,000 - 2,000	0.000050 - 0.010	20 - 30	10 - 50
- Pulsed Inductive	2,500	2 - 200		

[2] **Mixture Ratios for Various Bipropellant Combinations**

[Source: Sutton (1992, Ch 6-10)]

Oxidizer	Fuel	Mixture Ratio Range	Isp Range
N_2O_4	MMH	1.6000 - 2.0000	265.0000 - 315.0000
N_2O_4	UDMH	2.6000 - 2.7500	310.0000 - 320.0000
N_2O_4	50% N2H4 / 50% UDMH	1.9000 - 2.0000	290.0000 - 320.0000
LOX	LH2	4.8000 - 6.0000	425.0000 - 450.0000
LOX	Kerosene	2.5000 - 2.7000	320.0000 - 350.0000
LOX	RP-1	2.2000 - 2.4000	290.0000 - 310.0000
Nitric Acid	Aniline	2.7500	218.0000
Nitric Acid	RP-1	4.1000 - 4.8000	255.0000 - 265.0000
Nitric Acid	50% N2H4 / 50% UDMH	1.7300 - 2.2000	270.0000 - 280.0000



EPS Sheet

EPS System Data								
Bus Voltage Requlation	Partially Regulated ▼							
Bus Redundancy	Dual Bus ▼							
	Input		**Calculated**		**Input**		**Calculated**	
Voltage - Noneclipse		V	100.0000	V				
Voltage - Eclipse		V	80.0000	V				
Orbit Data								
Orbit Period			23.9309	hrs				
Maximum Eclipse Period			69.4104	min				
Power Requirements	Non-Eclipse				Eclipse			
Payload - Operating		W	1,000.0000	W		W	1,000.0000	W
Payload - Standby		W	100.0000	W		W	100.0000	W
Housekeeping		W	130.0000	W		W	130.0000	W
EP - Operating Payload		W	-----	W		W	-----	W
EP - Nonoperating Payload		W	-----	W		W	-----	W
- EP Duty Cycle		%	-----	%				
TCS - Operating Payload			474.1428	W			474.1428	W
TCS - Nonoperating Payload			924.1428	W			924.1428	W
Contignecy Load Fraction		%	5.0000	%		%	5.0000	%
Array Margin		%	10.0000	%				
Payload Duty Cycle		%	100.0000	%	Power Averaging			
Eclipse Operating Fraction		%	100.0000	%	● Yes	○ No		

EPS Sheet

Power Analysis			Non Eclipse				Eclipse		
		Operating		Non-Operating		Operating		Non-Operating	
Payload		1,000.0000	W	100.0000	W	1,000.0000	W	100.0000	W
Housekeeping		130.0000	W	130.0000	W	130.0000	W	130.0000	W
Electric Propulsion		0.0000	W	0.0000	W	0.0000	W	0.0000	W
Thermal Control		474.1428	W	924.1428	W	474.1428	W	924.1428	W
SUBTOTAL		1,604.1428	W	1,154.1428	W	1,604.1428	W	1,154.1428	W
Contignecy	0.0500	80.2071	W	57.7071	0.0500	80.2071	W	57.7071	W
Instantaneous Power		1,684.3500	W	1,211.8500	W	1,684.3500	W	1,211.8500	W
Time Span		1,366.4447	min	0.0000	min	69.4104	min	0.0000	min
Energy Requirement		38.3595	kW hr	0.0000	W hr	1,948.5238	W hr	0.0000	W hr
			Array				Battery		
Total Energy Requirement			38.3595	kW hr			1,948.5238	W hr	
Noneclipse Supplement							0.0000	W hr	
Battery Storage Requirement							1,948.5238	W hr	
Average Power Requirement			1,684.3500	W					
Battery Charging Power			95.0654	W					
Array Margin		0.1000	177.9415	W					
Array Power Requirement			1,957.3569	W					

EPS Sheet

Battery Size and Charging Power				
Energy Storage Requirement		W hr	1,948.5238	W hr
- Battery Characteristics				
EOL Minimum Cell Voltage		V	1.0000	V
Charge-Discharge Cycles				
- per Year			101.0000	
- Total			707.0000	
Depth of Discharge		%	85.0000	%
Max Charge Voltage per Cell		V	1.5000	V
Charger Voltage Drop		V	1.7500	V
Charging Efficiency			0.9000	
Redundant Cells per Battery			2.0000	
Bypass Diode Voltage Drop				
- Charging		V	2.5000	V
- Discharging		V	1.0000	V
Charging Rate			0.0259	
	per Battery		for EPS	
Number of Cells	84.0000			
Actual Voltage - Eclipse			80.0000	V
Maximum Charge Voltage			128.0000	V
Boost Voltage for Charging			29.7500	V
Capacity	14.3274	Amp hr	28.6548	Amp hr
Recharging Time	22.7741	hrs	22.7741	hrs
Battery Charging Power	47.5327	W	95.0654	W

EPS Sheet

Solar Cell Data				
Type	Gallium Arsenide ▼			
Size				
- Length		cm	1.0000	cm
- Width		cm	1.0000	cm
Mass		g	0.1100	g
Coverglass Thickness		mils	3.0000	mils
Efficiency		%	18.0000	%
Reference Solar Intensity		W / m²	1,353.0000	W / m²
Current - Max Power		mAmp	28.5000	mAmp
Voltage - Max Power		V	0.8700	V
Wiring Loss		V	0.0050	V
Temperature Coefficients				
- Reference Temperature		deg C	28.0000	deg C
- Current		% / deg C	0.0560	% / deg C
- Voltage		% / deg C	-0.2320	% / deg C
Radiation Degradation				
Annual 1MeV Equivilant Fluence				
- Electrons		e / cm²	3.4238E+13	e / cm²
- Protons		e / cm²	2.6706E+12	e / cm²
- Solar Flares		e / cm²	1.1000E+14	e / cm²
Annual Total		e / cm²	1.4691E+14	e / cm²
1MeV Fluence over Mission		e / cm²	1.0284E+15	e / cm²
Solar Cell Degradation				
- Current			0.8490	
- Voltage			0.8990	

EPS Sheet

Solar Array Design				
Solar Intensity - see Table		W / m²	1,311.0000	W / m²
Solar Incidence Angle		deg	23.4400	deg
EOL Power Requirement		W	1,957.3569	W
Number of Array Wings			2.0000	
Operating Temperature				
- Solar Absorptance			0.8700	
- Emittance, Front Side			0.8200	
- Emittance, Back Side			0.8500	
- Effecitve Solar Absorptance			0.6900	
- Operating Temperature		deg C	32.8257	deg C
Main Array Design				
- Assembly / Enviorn Losses			0.9000	
- Bus / Array Voltage Drop		V	2.0000	V
- EOL Cell Current			19.4119	mAmp
- Required Current per Wing			9.7868	Amp
- Number of String (in Parallel) per Wing			505.0000	
- EOL Cell Voltage			0.7689	V
- Cells per String (in Series)			133.0000	
- Acutal Power Output	per Wing		for EPS	
EOL Current	9.8030	Amp	19.6060	Amp
EOL Voltage	100.2609	V	100.2609	V
EOL Power	982.8564	W	1,965.7127	W
BOL Current	11.5465	Amp	23.0930	Amp
BOL Voltage	111.7496	V	111.7496	V
BOL Power	1,290.3181	W	2,580.6361	W

EPS Sheet

Charge Array						
- Cells in Series			39.0000			
- Strings in Parallel			39.0000			
Size of Array						
- Panels per Wing			2.0000			
- Length of Panel		m	2.0000	m		
- Width of Panel			1.7024	m		
	of Panel		of Wing		of Array	
- Areas	3.4048	m²	6.8096	m²	13.6192	m²
EPS Mass						
Solar Array						
- Solar Cell Mass / Area		g / cm²	0.1100	g / cm²		
- Substrate Mass / Area		g / cm²	0.2000	g / cm²		
- Solar Array Mass		kg	42.2195	kg		
Batteries						
- Specific Energy		W hr / kg	65.0000	W hr / kg		
- Battery Mass		kg	35.2674	kg		
Power Bus						
- Specific Power		g / W	9.2000	g / W		
- Bus Mass		kg	23.7419	kg		
EPS Mass		kg	101.2288	kg		

EPS Sheet

Solar Intensity and Angle between Ecliptic and Equatorial Plane

[Source: Agrawal (1986, p348)

Date	Solar Intensity		Inclinaton Angle	Effective Intensity	
	W / m²		deg	W / m²	
Average	1,353.0000			1,353.0000	
January 3 (perihelion)	1,399.0000	<--max	-23.1100	1,286.7345	
Febuary 1	1,393.0000		-17.2100	1,330.6308	
March 1	1,378.0000		-7.7300	1,365.4780	<-- max
April 1	1,355.0000		-4.2300	1,351.3090	
May 1	1,332.0000		14.9500	1,286.9136	
June 1	1,316.0000		22.0000	1,220.1740	
July 4 (aphelion)	1,309.0000	<-- min	23.1300	1,203.7773	
August 1	1,313.0000		18.1300	1,247.8134	
September 1	1,329.0000		8.4300	1,314.6411	
October 1	1,350.0000		-3.0300	1,348.1127	
November 1	1,374.0000		-14.3000	1,331.4276	
December 1	1,392.0000		21.7500	1,292.9029	
March 21 (Vernal Equinox)	1,362.0000		0.0000	1,362.0000	
June 21 (Summer Solstice)	1,311.0000		23.4400	1,202.8125	<-- min
September 23 (Autumnal Equin	1,345.0000		0.0000	1,345.0000	
December 22 (Winter Solstice)	1,397.0000		-23.4400	1,281.7156	

Thermal Sheet

	Input		Calculated		Input		Calculated	
Radiator Properties								
Solar Absorptivity			0.2000					
Infrared Emissivity			0.8000					
Efficiency			0.9500					
Temperature Limits								
Maximum		deg C	40.0000	deg C				
Minimum		deg C	10.0000	deg C				
Power Dissipation	Non-Eclipse				Eclipse			
Payload - Operating			1,000.0000	W			1,000.0000	W
Payload - Standby			100.0000	W			100.0000	W
Housekeeping			130.0000	W			130.0000	W
Transmitted - Operating		W	500.0000	W		W	500.0000	W
Transmitted - Standby		W	50.0000	W		W	50.0000	W
Total Dissipated - Operating			630.0000	W			630.0000	W
Total Dissipated - Standby			180.0000	W			180.0000	W
Radiator Size - Worst Cast Hot								
Solar Intensity[1]		W / m²	1,397.0000	W / m²				
Solar Incidence Angle		deg	23.4400	deg				
Radiator Size		m²	3.9862	m²				
Radiator Height			-----	m				

Thermal Sheet

Heater Power - Worst Case Cold							
Solar Intensity[1]		W / m²	1,345.0000	W / m²			
Solar Incidence Angle		deg	90.0000	deg			
Operating							
		Non-Eclipse				Eclipse	
- Total Dissipated Power			630.0000	W		630.0000	W
- Steady State Cold Temp			-27.0592	deg C		-27.0592	deg C
- Required Heater Power		W	474.1428	W	W	474.1428	W
Non-Operating							
		Non-Eclipse				Eclipse	
- Total Dissipated Power			180.0000	W		180.0000	W
- Steady State Cold Temp			-93.2304	deg C		-93.2304	deg C
- Required Heater Power		W	924.1428	W	W	924.1428	W

[1] See table at the bottom of the EPS sheet

Thermal System Mass				
Specific Power		kg / W	0.0400	kg / W
Thermal System Mass		kg	45.2000	kg

Mass Sheet

	Input		Calculated	Delta Mass		Mass	
Seperation Mass						2,500.0000	kg
Apogee Maneuver Expendable				1,179.2586	kg	1,320.7414	kg
Perigee Maneuver Expendable				0.0000	kg	1,320.7414	kg
Propellant/pressurant				227.8114	kg	1,092.9299	kg
Dry Mass of Spacecraft						1,092.9299	
Apogee Motor Inert				83.0000	kg	1,009.9299	kg
Perigee Motor Inert				0.0000	kg	1,009.9299	kg
Spacecraft Bus Mass						1,009.9299	kg
Propulsion		kg		50.0371	kg	959.8928	kg
Structures							
Coefficient			0.0870				
Mass		kg		217.5000	kg	742.3928	kg
- Mechanical Integration							
Coefficient			0.0140				
Mass		kg		17.3284	kg	725.0644	kg

Mass Sheet								
EPS					101.2288	kg	623.8357	kg
- Electrical Integration								
Coefficient			0.0390					
Mass		kg			48.2719	kg	575.5637	kg
Thermal					45.2000	kg	530.3637	kg
Attitude Control		kg			76.8303	kg	453.5334	kg
Telemetry & Control								
Coefficient			0.0400					
Mass		kg			40.3972	kg	413.1362	kg
Payload								
Payload Fraction			0.3000					
Mass					302.9790	kg	110.1573	kg
Mass Margin							110.1573	kg
Percentage of Spacecraft Dry Mass							10.9074	%

APPENDIX C. ELECTRONIC COPY OF DESIGN TOOL

This appendix contains a 3.5" diskette with a copy of the submitted and approved version of the Spacecraft Integrated Preliminary Design Tool developed as part of this thesis. The design tool was created in Microsoft® Excel 97.

The reader is cautioned that the spacecraft integrated preliminary design tool developed in this thesis and provided on the diskette may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure the spreadsheet calculations are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

To protect the integrity of the "program", the equations and expressions in the cells, the worksheets are locked without a password. This also prevents the inadvertent overwrite of calculation cells. Entries will only be accepted into certain specific, unlocked cells.

To fully execute the design tool, and in particular the calculations in the Delta V sheet, the BESSELI function must be installed in Excel. The Bessel functions are included in Analysis ToolPak Excel Add-In. To verify the Bessel functions are installed, click on the Function button in the toolbar, under Function category select all, then scroll down through the Function name window to the "b"s to find the BESSELI function. To install the add-in, click on the Tools option on the menu bar, select Add-Ins…, check the box by Analysis ToolPak, and click on OK button.

If this copy of the thesis does not include the diskette, a copy of the design tool can be obtained from the author or through the Aeronautics and Astronautics department at NPS.

LIST OF REFERENCES

Aerospace Corporation, Aerospace Report TOR-95(5409)-2, *Balancing Performance and Cost for Cost-Effective Satellite Systems Design Using an Integrated Cost Engineering Model*, by K.D. Bell and L.A. Hsu, prepared for Space and Missile Systems Center, Air Force Material Command under Contract No. F04701-93-C-0094, 18 December 1994.

Aerospace Corporation Engineering and Technology Group, Aerospace Report ATR-98(8227)-1, *Concept Design Center (CDC) User's Guide*, by A.B. Dawdy, 25 August 1998.

Aerotech Microelectronics, "Indium Phosphide Solar Cells: Space Power of the Future" handout in NPS course EO 3205 Space Power, Monterey, California, February 1998.

Agrawal, B.N., *Design of Geosynchronous Spacecraft*, Prentice-Hall, Inc., 1986.

Aguilar, Joseph A., Dawdy, Andrew B., and Law, Glenn W., "The Aerospace Corporation's Concept Design Center," *Proceedings of the Eighth Annual International Symposium of the International Council on Systems Engineering (INCOSE)*, January 1998.

Analytical Graphics, Inc., "Products." [http://www.stk.com/products/prod1.cfm?cfid=247429&cftoken=40193084], December 1999.

Applied Solar Energy Corporation, "Space Products," handout in NPS course EO 3205 Space Power, Monterey, California, February 1998.

Bearden, David A., *A Methodology for Technology-Insertion Analysis Balancing Benefit, Cost, and Risk*, Doctoral Dissertation Proposal, Department of Aerospace Engineering, University of Southern California, Los Angeles, California, August 1998.

Computational Technologies, Inc., "First In Space Systems Engineering." [http://www.ctek.com/news/ads.html]. November 1999.

Computational Technologies, Inc., "GENSAT Results." [http://www.ctek.com/gensatCaseStudy.html]. July 1998.

Computational Technologies, Inc., *GENSAT Technical Overview*, 1998.

Goldin, Daniel S., Venneri, Samuel L., and Noor, Ahmed K., "New Frontiers in Design Synthesis", paper presented at the 49th International Astronautical Congress, Melbourne, Australia, 28 September 1998 - 2 October 1998.

Gordon, G.D. and Morgan, W.L., *Principles of Communications Satellites*, John Wiley & Sons, Inc., 1993.

Kaplan, M.H., *Modern Spacecraft Dynamics and Control*, John Wiley & Sons, Inc. 1976.

Larson, W.J. and Wertz, J.R., *Space Mission Analysis and Design*, 2d ed., Microcosm, Inc., 1992.

Macneal-Schwendler Corporation, "Software Products." [http://www.mechsolutions.com/products/nastran/index.html]. December 1999.

Milden, Martin J., "Five-Year Update: Nickel Hydrogen Industry Survey," *IEEE AES Systems Magazine*, p. 14-16, November 1991.

National Aeronautics and Space Administration Jet Propulsion Laboratory, JPL Publication 82-69, *Solar Cell Radiation Handbook*, 3rd edition, by H.Y. Tada, J.R. Carter, Jr., B.E. Anspaugh, and R.G. Downing, 1 November 1982.

Naval Postgraduate School, *Practical Orbitology*, Summer 1997.

Object Management Group, "What Is CORBA?." [http://www.omg.org/corba/whatiscorba.html]. December 1999.

Parametric Technologies, Inc., "Pro/ENGINEER Solutions." [http://www.ptc.com/products/proe/index.htm]. December 1999.

Space Mission Analysis and Design Users Guide, KB Sciences, 1994.

Structural Dynamics Research Corporation, "Introducing I-DEAS Master Series." [http://www.sdrc.com/ideas/intro/]. December 1999.

Sutton, George P., *Rocket Propulsion Elements, An Introduction to the Engineering of Rockets*, 6th ed, John Wiley and Sons, Inc., 1992.

TRW, "Integrated Concept Development Facility (ICDF) Introduction and Status", presentation at TRW, El Segundo, California, October 1998.

Vallado, D.A., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, 1997.

VMA Engineering, "GENESIS Fact Sheet." [http://www.vma.com/genesis_fact.html]. December 1999.

Wall, Steve, *JPL's New Concept Development Process*, Jet Propulsion Laboratory, Advanced Projects Development Office, October 1996.

Waterloo Maple, Inc., "Maple V Release 5.1." [http://www.maplesoft.com/products/Maple/maplevr5.html]. December 1999.

Weinberg, I. and others, "Radiation and Temperature Effects in Gallium Arsenide, Indium Phosphide, and Silicon Solar Cells," handout in NPS course EO 3205 Space Power, Monterey, California, February 1998.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center.......... 2
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library.......... 2
Naval Postgraduate School
411 Dyer Road
Monterey, CA 93943-5002

3. Chairman, Code AA 1
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943

4. Professor Brij N. Agrawal, Code AA/Ag 2
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943

5. Professor Herschel H. Loomis, Jr, Code EC/Lm.......... 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943

6. CAPT Tom McKannon, USN.......... 1
National Reconnaissance Office
14675 Lee Road
Chantilly, VA 20151-1715

7. Professor Oscar Biblarz, Code AA/Bi.......... 1
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943

8. Professor Norm Sorensen, Code SP/So 1
Space Systems Academic Group
Naval Postgraduate School
Monterey, CA 3943

9. LT Julie LaPointe, USN.. 1
National Reconnaissance Office
14675 Lee Road
Chantilly, VA 20151-1715

10. Maj Troy W. Pannebecker, USAF.. 3
14407 Red House Drive
Centreville, VA 20120-2808