PROJECT WISH
The Emerald City
Phase III

P-194

The Ohio State University
Department of Aeronautical and Astronautical Engineering

1991-1992 NASA/USRA University
Advanced Design Program

Space Design

DESIGN TEAM

Professor: Dr. Hayrani Oz
Teaching Assistant: Jim Dunne

Winter and Spring Quarters 1991, 1992

Stan Butchar	George Owens
Tommy George	Mike Perrea
Rob Hellstrom	Paul Semeraro
Tricia Kringen	Phil Thorndike

In addition to the above,
Winter Quarter included:

Ioannis Mikellides	Wei Wei

ABSTRACT

Phase III of Project Wish saw the evolution of the Emerald City (E-City) from a collection of specialized independent analyses and ideas to a working structural design integrated with major support systems and analyses. Emphasis was placed on comparing and contrasting the closed and open cycle gas core nuclear rocket engines to further determine the optimum propulsive system for the E-City. Power and thermal control requirements were then defined and the question of how to meet these requirements was addressed. Software was developed to automate the mission/system/ configuration analysis so changes dictated by various subsystem constraints could be managed efficiently and analyzed interactively. In addition, the liquid hydrogen propellant tank was statically designed for minimum mass and shape optimization using a finite element modeling package called SDRC I-DEAS while spoke and shaft cross-sectional areas were optimized on ASTROS (Automated Structural Optimization System). A structural dynamic analysis also conducted using ASTROS enabled a study of the displacements, accelerations, modes and frequencies of the E-City. Finally, the attitude control system design began with an initial mass moment of inertia analysis and was then designed and optimized using linear quadratic regulator control theory.

EMERALD CITY
CONFIGURATION
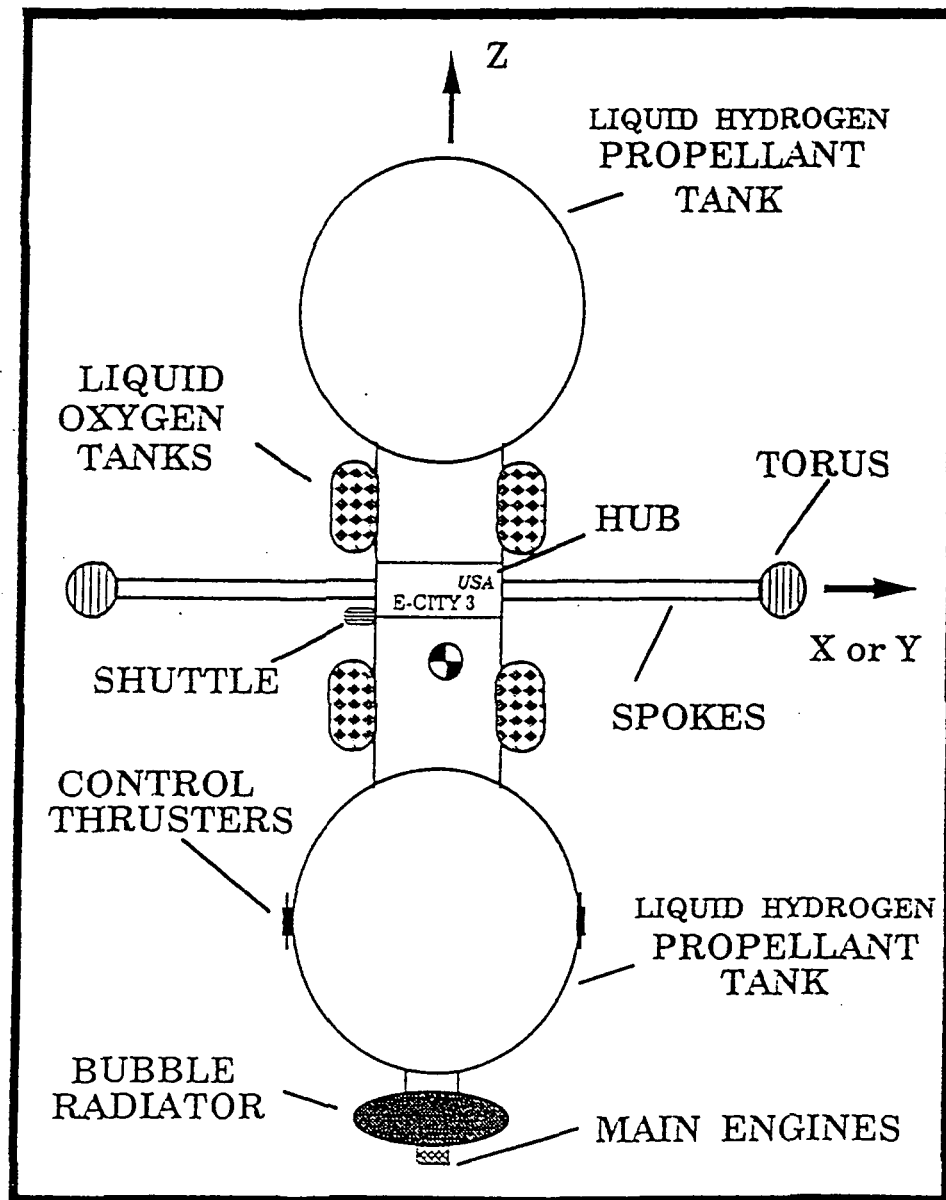
LIQUID HYDROGEN
PROPELLANT
TANK

LIQUID
OXYGEN
TANKS

TORUS

HUB

USA
E-CITY 3

Z

X or Y

SHUTTLE

SPOKES

CONTROL
THRUSTERS

LIQUID HYDROGEN
PROPELLANT
TANK

BUBBLE
RADIATOR

MAIN ENGINES

ACKNOWLEDGEMENTS

FOREWORD

The work presented in this report represents the third and last phase of a 3-year advanced space design project for a "Permanently Manned Autonomous Space Oasis (PEMASO)." The design has evolved over the three years 1989-1992. This year's work was built upon the efforts of the previous two years and addressed more rigorously the propulsion, thermal and power, structural and attitude control system designs. Along with these subdisciplines, other subdisciplines pertinent to the project such as communication systems, life support systems and orbital mechanics were studied during Phase I (1989-90) and Phase II (1990-91). Specifically, orbital mechanics has received extensive attention in Phase II. Phase III (1991-92) revisited the subdisciplines which were decided to be the major design drivers based on the previous years' work and did not address other subdisciplines which in the final analysis would only be minor players for a project of such scope. Indeed, as this report will indicate propulsion, thermal and power, structural and attitude control are the disciplines that govern the enormous orders of magnitude of the design variables consistent with the size of PEMASO. The overall design is far less sensitive to variations in other subdisciplines not covered in Phase III.

Dr. Hayrani Oz, July 1992

TABLE OF CONTENTS

LIST OF TABLES

OSU ADVANCED SPACE DESIGN

- NASA LEWIS
  - Propulsion
  - Power
  - Orbital Mechanics
- IDEAS
  - Static Structure
- DECstations
  - FDL WPAFB ASTROS
    - Structural Optimization
    - Structural Dynamics
  - MATLAB
    - Attitude Control
  - VAX System
    - System Parameters
    - Active Thermal Control
- Macintosh
  - Technical Reports
  - Graphics
- IBM PC
  - Passive Thermal Control
  - Mass Moments of Inertia
  - Graphics

CHAPTER 1

INTRODUCTION AND PROJECT BACKGROUND

1.0 INTRODUCTION

It's the year 2050 and the world breathlessly awaits the unveiling of the Emerald City. The culmination of years of painstaking design, analysis, testing, and construction are about to come to an end. Once thought to be too conceptual in nature, the Emerald City now stands majestically before all, keeping the United States of America the forerunner in space exploration and technology. The last frontier is about to be explored.

Martians and Lunarians are on stand-by as the Emerald City completes it's final systems check and prepares to leave it's position at L-5. The innerplanetary satellite network broadcasts this historic event live. "10 9 8 7 6 5 4 3 2 1...All systems are Go!...We have successful main engine start...E-City 1 Mission to Saturn is off!"

In the year 1992, time was spent rehashing the blueprints, finetuning the details, and checking the results, again and again. Accuracy was of utmost importance. Let us not tarry as we look into the final design specs and tests that were presented by the final design team. Quick, the Emerald City is about to be unveiled...

1.1 PROJECT BACKGROUND

Project WISH, a three year advanced design project at the Ohio State University, began as a possible follow-up to the current Space Exploration Initiative (SEI) program set forth by President Bush and NASA. The design entails a Permanently Manned Autonomous Space Oasis (PEMASO), designated the Emerald City (E-City), with a mission to support colonization and exploration efforts throughout the solar system. Home to 1000 colonists, the E-City must have the capability to re-station itself almost anywhere in the solar system within a transit time of three to five years. Envisioned to become operational in the year 2050 (see Table 1.1), PEMASO must be self-sufficient, requiring no additional resources from Earth. At a nominal orbit of 4 AU's (see Figure 1.1), the E-City will be in an ideal location to mine the asteroids for natural resources as well as to obtain hydrogen from Jupiter's atmosphere.

1.1

Phases I[1] and II[2] of Project Wish established the ground work for Phase III and were conducted during the 1989-1990 and 1990-1991 academic years. Phase I encompassed a general level study of the major systems required for the E-City while Phase II completed a more in-depth study into the disciplines of orbital mechanics, propulsion, attitude control, and human factors. Guidelines were also established for the design of the ship and were used to carry out two particular missions of interest: a Saturn Envelope mission and an Earth-to-Mars mission (see Table 1.2).

1.2 OVERVIEW

Phase III of Project Wish saw the evolution of the E-City from a collection of specialized independent analyses and ideas to a working structural design integrated with major support systems and analyses. Optimization and system integration were key in establishing the final design parameters. Detailed analyses and studies were conducted in propulsion and power, mission/system/configuration design, static and dynamic structures, and attitude control.

Due to the hazardous plume radiation problem and high mass penalty associated with the open- cycle Space Radiating Gas Core Nuclear Rocket Engine (SRGCNR) studied in Phase II of Project WISH, this year's analysis of the propulsion system focused on the closed-cycle Gas Core Nuclear Light Bulb Engine (NLB). Emphasis was placed on comparing and contrasting the NLB with the SRGCNR to further determine the optimum propulsive system for the E-City. Power and thermal control requirements were then defined and the question of how to meet these requirements was addressed. Software was developed to automate the mission/system/configuration design so changes dictated by various subsystem constraints could be managed efficiently and designed interactively. This analysis also studied the hydrodynamic effect of tank rotation and the possibility of a dual-spin station as well as the mass and volume penalties/advantages associated with using additional tanks. In addition, the liquid hydrogen propellant tank was statically designed for minimum mass and shape optimization using SDRC I-DEAS[3] while spoke and shaft cross sectional areas were optimized on ASTROS[4]. A structural dynamic analysis also conducted using ASTROS enabled a study of the natural vibration of the crew quarters and its effect on the

1.2

entire ship design to be completed. Finally, the attitude stability and control system began with an initial mass moment of inertia analysis (MMI) and was then designed using optimal control theory via PRO-MATLAB[5]. The goal was to control the gyroscopic wobble of the station following a disturbance in a manner that would be acceptable to the crew. This included defining and optimizing attitude control parameters such as the propellant mass and control power requirements, number of thrusters, number of thruster clusters, and state and control weighting parameters.

As the configuration and system analyses were optimized, each individual analysis was updated to remain consistent with the latest results and findings. The following chapters represent the specifications and design requirements of the E-City as designated by the final Phase III design team.

Table 1.1: Envisioned Time Line for Project WISH

1990 - President Bush announces Mars Initiative to reach
       the Red Planet in 30 years.

1991 - NASA presents a program to send sensing probes
       throughout the solar system. Projects include the
       return to the inner solar system, exploration of
       the asteroid belt, and further missions to the
       outer planets.

1996 - Space Station Freedom becomes operational.

1998 - Heavy Lift Launch Vehicle makes its maiden
       flight.

2000 - Construction begins on a near-geosynchronous
       Earth orbit space station.

2005 - U.S. returns to the moon.

2010 - Construction begins on the Moon Base.
     - Maiden flight of National Aerospace Plane.

2015 - Moon Base becomes fully operational.

2020 - First manned mission to Mars.

2023 - First living modules constructed on Mars.
     - Construction begins on Reusable Interplanetary
       Ships (R.I.S.) for carrying personnel and cargo.

2028 - Mars Base becomes fully operational.

2030 - R.I.S.'s becomes operational.

2040 - Implementation of Project WISH.

2045 - Unmanned probe sent to Alpha Centari.

2050 - The Emerald City becomes operational.
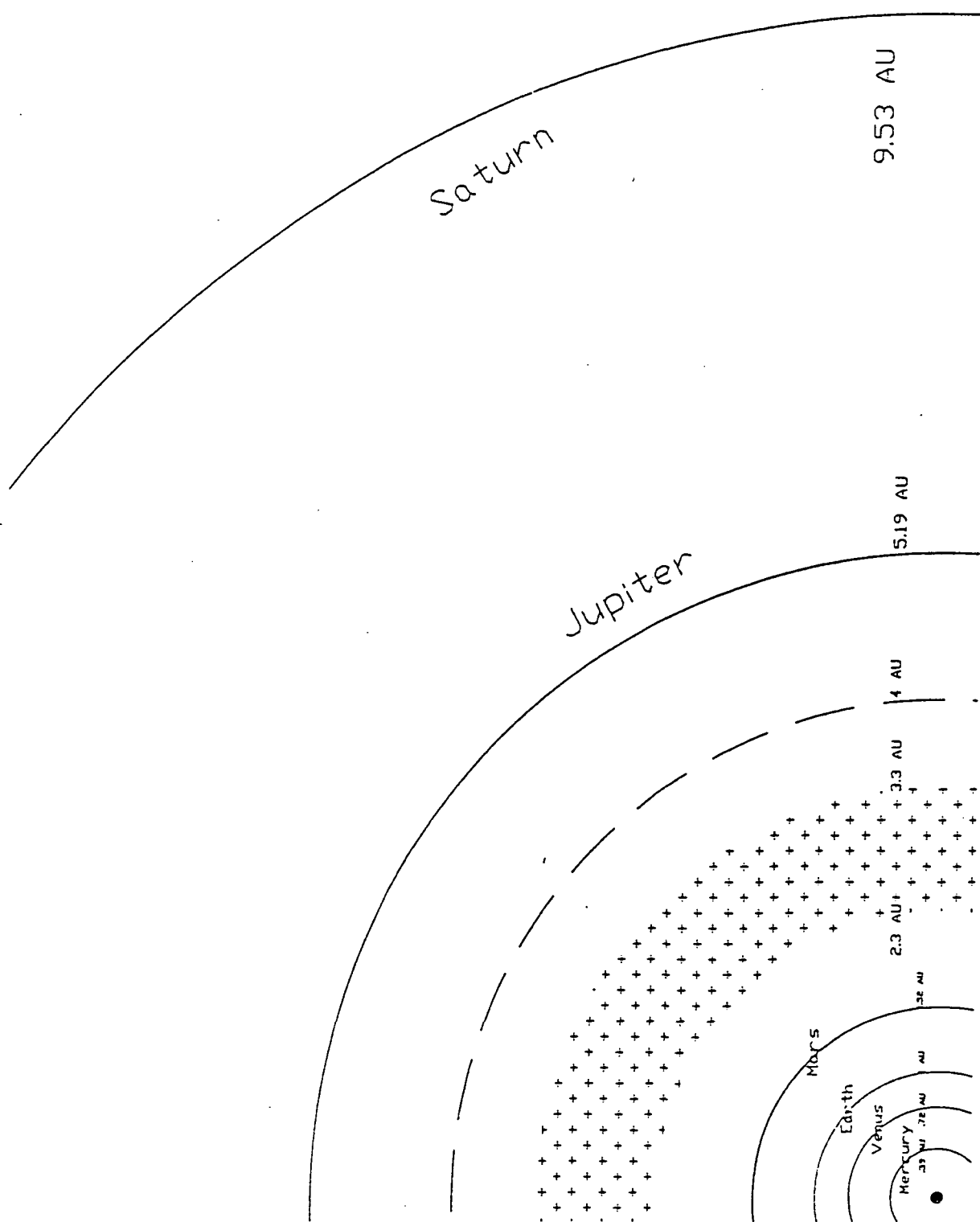
Figure 1.1: Nominal Orbit of the Emerald City

Table 1.2: Summary of Design Variables for Saturn Envelope and Mars Missions

	Saturn Mission	Earth to Mars
People	1000	500
ΔV_{total} (km/s)	50	12.6
ΔV_1 (km/s)	-	5.1
ΔV_2 (km/s)	-	7.5
F (Newtons)	4.4×10^3	4.4×10^3
I_{sp} (seconds)	5000	5000
t_{total} (days)	20	11.53
t_1 (days)	-	5
t_2 (days)	-	6.53
number of engines	172	33
m_{dry} (kg)	1.5×10^3	1×10^3
m_{prop} (kg)	1.111×10^3	7.746×10^4
m_1 (kg)	4.4376×10^7	8.514×10^4
m_2 (kg)	3.457×10^4	2.165×10^3
m_3 (kg)	2.658×10^3	2.947×10^4
m_4 (kg)	4.16×10^3	1.295×10^3
m_3 / m_2	0.083	0.167
m_4 / m_2	0.639	0.228
V_{tot} (m^3)	3.742×10^7	4.151×10^6
r_{hole} (m)	100	50
h_1 (m)	1200	528
h (m)	1270	569
R (m)	700	700
r_{min} (m)	37	26
I_z (kg m^2)	5.6144×10^{14}	3.8061×10^{14}
I_x, I_y (kg m^2)	9.9568×10^{14}	2.1731×10^{14}
r (I_z / I_x)	0.563876	1.75154
max/min g-levels	0.8/0.72	0.8/0.74
n (spin rate, rpm)	0.99	0.99

CHAPTER 2

PROPULSION

2.0 INTRODUCTION

The propulsion system is perhaps the most challenging of all the design aspects of Project WISH. As discussed in the Phase II[2] report, delta-V's ranging from 50 to 100 km/sec is a mission requirement. Achieving such high delta-V's is no easy task, and throughout the three year period of Project WISH, feasibility studies of several conceptual propulsion systems have been performed. The Phase I[1] design team had analyzed chemical, nuclear, and anti-matter rocket engine characteristics before recommending anti-matter as the most probable system. The Phase II design team, reconsidering that the anti-matter engine was too conceptual in nature for the time frame of Project WISH, proposed to use a gas-core nuclear rocket engine. Known specifically as the the space radiated gas-core nuclear rocket (SRGCNR), or open-cycle engine, the Phase II team had hoped that the high specific impulse it generated and projected technological feasibility of this engine would prove satisfactory to the needs of E-City. However, due to hazardous radiation emitted from the exhaust plume and high mass penalty associated with these engines, it was decided to reconsider once more the system used for main propulsion.

The SRGCNR engine, referred to earlier as the open-cycle engine, was not the only propulsion system studied by the Phase II design team. Another system, known as the closed-cycle nuclear rocket engine, was also studied by the Phase II team. For reasons that will be explained in more detail later in the chapter, the closed-cycle engine, also known as the nuclear light bulb (NLB) engine, showed desirable characteristics, but was not considered an acceptable propulsion system for Project WISH in Phase II.

This year, the NLB was considered again, and a more detailed feasibility study of the engine was performed. Using the previous Project WISH reports and information provided by the NASA Lewis Research Center, it was possible to obtain results for the NLB similar to that of the Phase II analysis for the open-cycle engine. By direct comparison, the NLB engine proved to be a more desirable system than the open-cycle engine.

2.1

It is the purpose of this chapter to give a brief comparison of the two engines, which will lead to the feasibility analysis of the NLB engine. From the results, it will be shown that the NLB engine satisfies the conditions of projected available technology and performance requirements for E-City.

2.1 THE OPEN-CYCLE AND NUCLEAR LIGHT BULB ENGINES

2.1.1 The Open-Cycle (SRGCNR) Engine

Figure 2.1 shows the conceptual sketch of the open-cycle engine[6]. It consists of a pressure shell, a moderator, a nozzle, a turbopump, and an external radiator. The open-cycle engine operates by transmitting thermal radiation, which is generated by fissioning uranium, to hydrogen propellant, which is then expelled out the nozzle at extremely high speeds. The advantages to this type of system are the high values of specific impulse that can be obtained. For the conceptual engine shown in Figure 2.1, the values of specific impulse can range from 2000 to 8000 seconds.
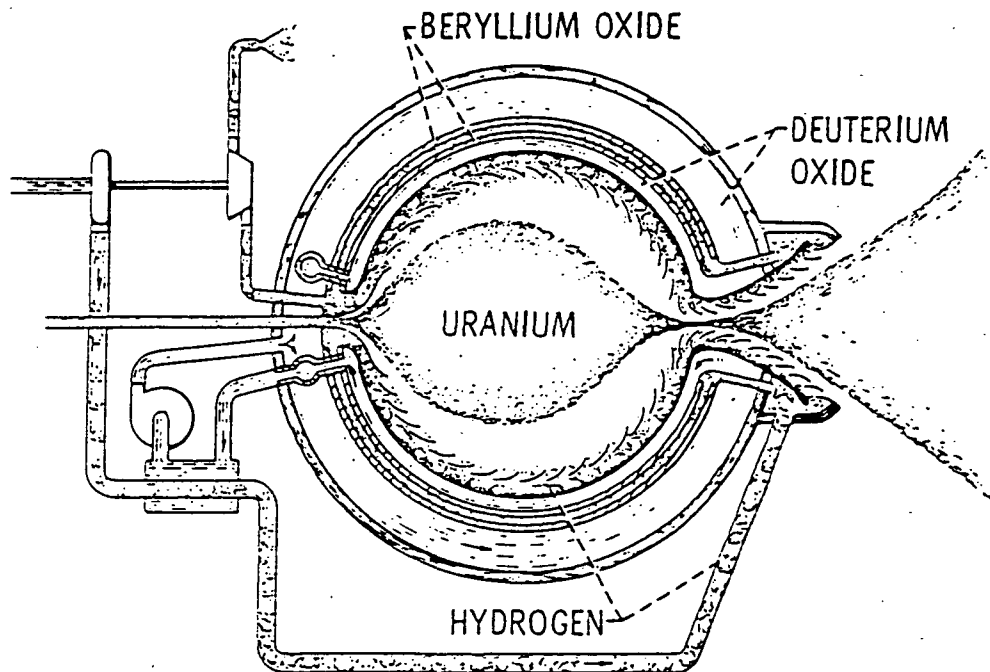
Figure 2.1: Conceptual Sketch of the SRGCNR Engine.

2.2

However, the SRGCNR has two detrimental effects, both of which are related. Because the gaseous uranium core is in contact with the hydrogen propellant, there is a loss of uranium out of the cavity through the nozzle. The Phase II design team had estimated that approximately 2 metric tons of uranium will be lost per engine per day of powered flight time. The second drawback is that the exhaust plume from this type of engine contains large amounts of radiation which results in excessive shielding required to protect the crew on board E-City.

2.1.2 The Closed-Cycle (NLB) Engine

The principle of operation of the NLB is similar to that of the SRGCNR engine, except that the gaseous uranium is enclosed in some type of internally cooled, transparent structure. In this way, the propellant does not come into contact with the uranium core. Shown in Figure 2.2,
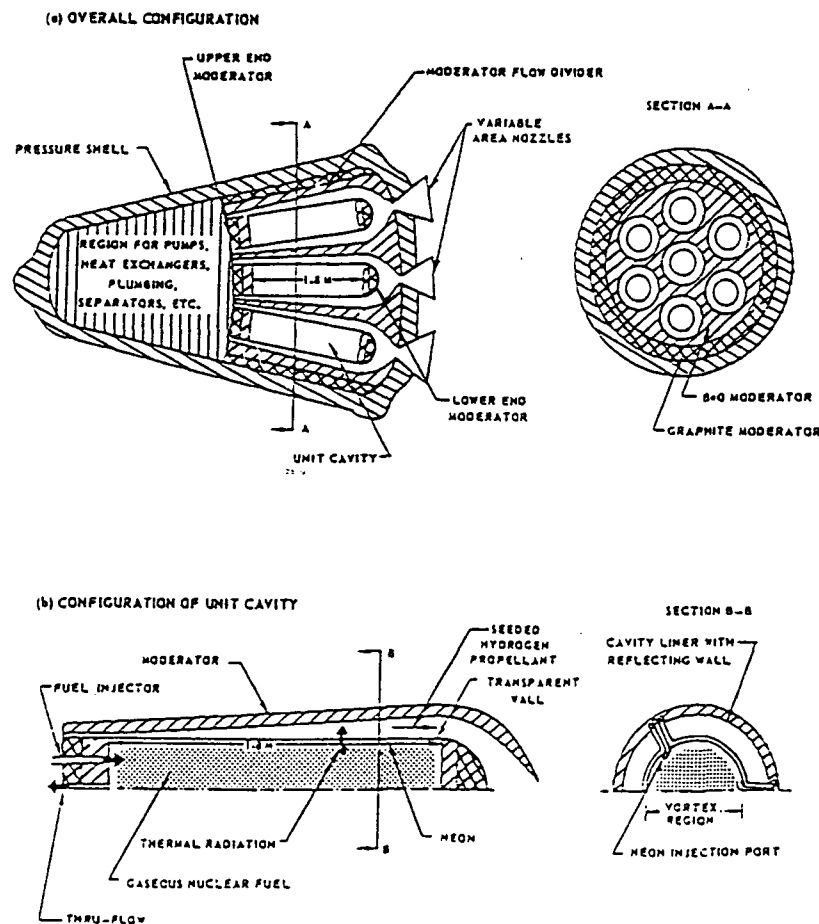
Figure 2.2: Thrust Chamber, and Basic Engine Configuration for NLB.

2.3

the gaseous uranium is surrounded by neon or some other noble gas spinning in a vortex, which provides for temperature attenuation and prevents the uranium from coming into contact with the transparent structure. The thermal radiation generated by the fissioning uranium is transmitted through the transparent structure to a seeded hydrogen propellant. The seeding is made of microparticles of tungsten to help absorb the radiative energy to ensure that it is transferred to the propellant and not the outer walls. As in the case of the open-cycle engine, this propellant is also expelled out of the nozzle at very high speeds[7].

Figure 2.2 is a representative sketch of one chamber making up the NLB engine. The complete engine, shown in Figure 2.3,

ENGINE POWER · 4600 MW
OPERATING PRESSURE - 500 ATM
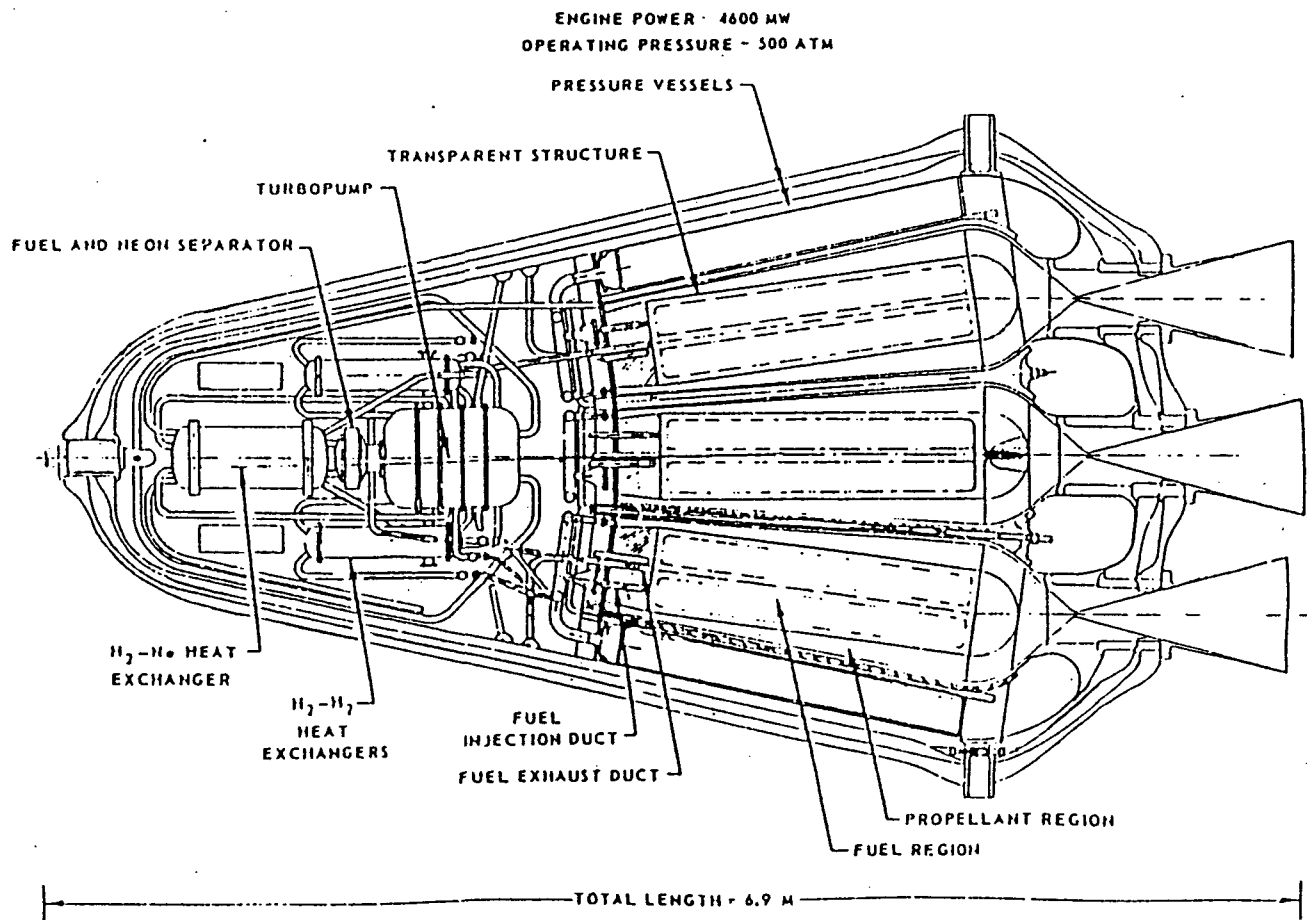
Figure 2.3: Side View of Complete Reference Nuclear Light Bulb Engine.

consists of seven of these chambers. The multiple transparent structures result in a higher overall

2.4

surface area of transmitted radiation, which leads to a higher energy transmission to the propellant. This, in turn, leads to a higher thrust output of the engine compared to an engine of the same size using the same volume of gaseous uranium and employing only one large chamber.

The turbopumps, moderator, and heat exchangers are used to keep the engine from overheating. They form a series of closed-cycle loops that recirculate neon, uranium, and hydrogen within the engine. It is through these cycles, and the moderator surrounding the chambers, that the need for an external radiator is eliminated.

There are many advantages to this engine. First, there is no loss of uranium. Second, because the core does not come into contact with the propellant, the exhaust plume does not contain harmful radiation, and so there is no need for external shielding. Because of the closed-cycle control systems, the engine is throttleable[8]. This feature is very useful during startup and shutdown of the main engines (which will be discussed in detail in Chapter 6), and in application to attitude and control of E-City. The thrust level for the NLB is significantly higher than the open-cycle engine. However, the specific impulse of a given NLB engine is usually half that of an open-cycle engine of comparable size.

2.1.3 Evaluation of the NLB and Open-Cycle Engines

In order to justify the study of the NLB Engine, it is important to understand the governing parameters that led the Phase II team to choose the open-cycle engine over the NLB. Table 2.1 is a comparison of thrust, engine weight, and specific impulse values between three open-cycle and three NLB engines. This table comes directly from the Phase II report, and represents the existing data last year's design team used for their analysis. From the values, it can be seen that the open-cycle has almost twice the specific impulse of the NLB, while the NLB is capable of producing a thrust level that is approximately an order of magnitude greater than the open-cycle engine. In terms of mass, the NLB is characteristically heavier than the SRGCNR, on account of the turbopumps, moderator, and heat exchangers.

From the theoretical investigation given in detail within the Phase II report, two very important equations arose from the analysis. The equation for the propellant mass ratio (eq. 2.1), and the expression for the number of engines needed for a given mission (eq. 2.2) show that for a

2.5

given velocity requirement; the specific impulse is a major governing factor in optimizing the propellant mass ratio and number of engines for the E-City. In both cases, the higher the specific impulse, the lower the propellant mass ratio and number of engines required for a given mission. It can also be seen that an increase in thrust will not reduce the propellant mass ratio, and so it would seem that the thrust of the engine is not as important as the specific impulse. For these reasons, the Phase II team concluded that the engine capable of producing the highest specific impulse was the only suitable propulsion system for the ship. And so, the open-cycle engine was determined as satisfying the design specifications for the E-City.

Table 2.1: Comparison of I_{sp}, Thrust, and Engine Mass for the NLB and SRGCNR Engines.

Engine Type	Isp (seconds)	Thrust (Newtons)	Engine Mass (kg)
SRGCNR	2400	22,240	36,280
SRGCNR	5500	177,900	101,440
SRGCNR	6000	444,750	213,350
NLB	1780	133,370	14,050
NLB	2355	1,334,200	34,475
NLB	2635	4,002,800	385,500

$$\frac{m_p}{m_o} = 1 - e^{-\frac{\Delta v}{I_{sp} g}} \tag{2.1}$$

$$n = \frac{m_{dry} I_{sp} g}{F_i t_{pr}} (e^{\frac{\Delta v}{I_{sp} g}} - 1) \tag{2.2}$$

2.6

Utilizing this open-cycle engine, the Phase II team calculated that approximately 172 SRGCNR engines would be needed for a mission to Saturn! An Earth-to-Mars mission would require at least 33 engines. This resulted in a tremendous total engine mass, along with the extra mass required for radiation shielding and extra uranium to account for the loss from each engine during burn time. Still, because of its very high specific impulse, the disadvantages of the SRGCNR engine were temporarily overlooked. It was hoped by the Phase II team that they could be remedied at some later time.

This year, deciding to avoid the plume radiation problem altogether, and noting very high thrust values of NLB engines despite lower values of specific impulse, the NLB was studied further. It was thought that the high thrust-to-weight ratios for these engines could more than offset the effect of its lower specific impulse in comparison to the SRGCNR engines.

The thrust-to-weight ratio is a measure of how much more mass, along with its own, that a propulsive device can accelerate to a particular velocity. A higher thrust-to-weight ratio means that for a given payload, the fewer the number of engines that are needed to accelerate that mass to a certain velocity. Directly related to this ratio is the thrust. Upon closer inspection of equation 2.2, for a given payload mass, change in velocity, and specific impulse, the number of engines decreases as the thrust is increased. Although the propellant mass ratio is not affected by the thrust, a decrease in the number of engines means a decrease in the total mass of the ship, m_0. Because the propellant mass ratio stays the same, this means that the overall mass of required propellant will decrease as well. Table 2.2 lists the thrust, engine mass, and thrust-to-weight ratios for the open-cycle and closed-cycle engines presented in Table 2.1. The thrust, and thrust-to-weight ratios of the nuclear light bulb engines are higher than those of the SRGCNR engines. Looking again to equation 2.2, it can be qualitatively seen that an increase in the thrust by a factor of ten reduces the required number of engines by the same amount. Because the specific impulse for an NLB engine is almost half that of an SRGCNR, this helps to reduce the number of engines, although it is not easily seen by inspection.

2.7

Table 2.2: Thrust, Engine Mass, and Thrust-to-Weight Ratio for the NLB and SRGCNR Engines.

Engine Type	Thrust (Newtons)	Engine Mass (kg)	T/W Ratio
SRGCNR	22,240	36,280	.06
SRGCNR	177,900	101,440	.18
SRGCNR	444,750	213,350	.21
NLB	133,370	14,050	.97
NLB	1,334,200	34,475	3.95
NLB	4,002,800	385,500	1.06

From the qualitative analysis, and from the desire to eliminate the radiative exhaust plume, it did indeed seem that further study of the nuclear light bulb engine was warranted. From a quantitative analysis, using the values listed in Table 2.1, the reduction in the number of engines was dramatic, and the impact of this on other mission design parameters are listed in Chapter 3.

2.2 ANALYSIS

2.2.1 Goals

As stated earlier, it was the goal of this year's team to use the existing material on the NLB, such as NASA reports, text, and the previous two Project WISH reports to gain more knowledge on how the nuclear light bulb operated, and what it would take to generate the same type of results that the Phase II team did for the open-cycle engine. It was hoped that a FORTRAN program could be written that would yield accurate values for NLB propulsion characteristics within the

thrust ranges needed for E-City. In this way, a direct comparison between the two systems would determine which one was most effective in satisfying the design requirements of Project WISH.

2.2.2 Data

The information obtained that proved to be the most vital for the analysis was the data provided by Dr. Stan Borowski of the NASA-Lewis Research Center. This information included performance parameters of seven different NLB engines provided by the research done by Thomas Latham of the United Technologies Research Center[9,10]. Shown in Table 2.3, important parameters such as thrust, specific impulse, radiative temperature, engine mass, thrust-to-weight ratio, and propellant flow for each engine are given.

Table 2.3: Data for Seven NLB Engines.

NLB Performance Characteristics *						
Reactor Power	Radiating Temperature	Specific Impulse	Chamber Pressure	Hydrogen Flow	Thrust	Thrust-to-Weight Ratio
(MWth)	(Kelvin)	(Seconds)	(Atm.)	(kg/sec)	(kN)	
730	5000	1120	310	9.0	98	0.38
2500	7000	1570	430	16.0	245	0.85
4500	8333	1870	500	22.5	472	1.3
10,000	10,000	2150	630	37.2	784	2.3
22,000	12,000	2500	790	59.8	1470	4.0
51,000	15,000	2700	1000	119.0	3136	6.0
160,000	20,000	3100	1300	309.5	9408	6.9

* Provided by T. Latham, c/o Dr. S. Borowski, NASA Lewis Research Center

**Engine Thrust vs. Uranium Temperature
for the Nuclear Light Bulb Engine.
Source: T. Latham, UTRC, c/o S. Borowski, NASA, LeRC**
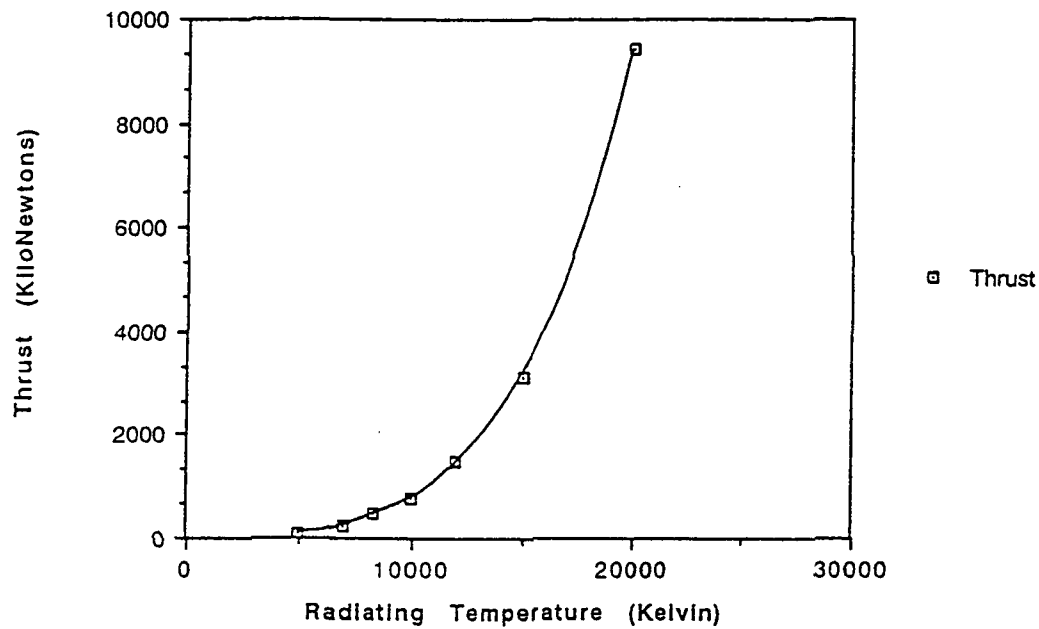
Figure 2.4: Engine Thrust vs. Radiating Temperature

**Specific Impulse vs. Uranium Temperature
for the NLB Engine.
Source: T. Latham, UTRC, c/o S. Borowski, NASA, LeRC**
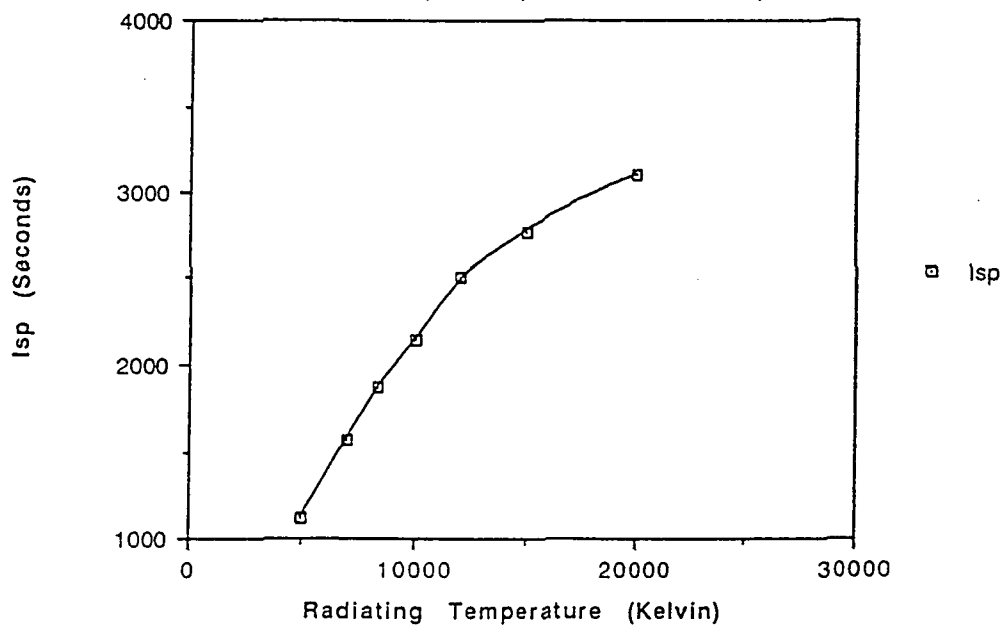
Figure 2.5: Specific Impulse vs. Radiating Temperature.

2.10

Hydrogen Propellant Flow Rate vs. Uranium Temperature
for the Nuclear Light Bulb Engine.
Source: T. Latham, UTRC, c/o S.Borowski, NASA, LeRC
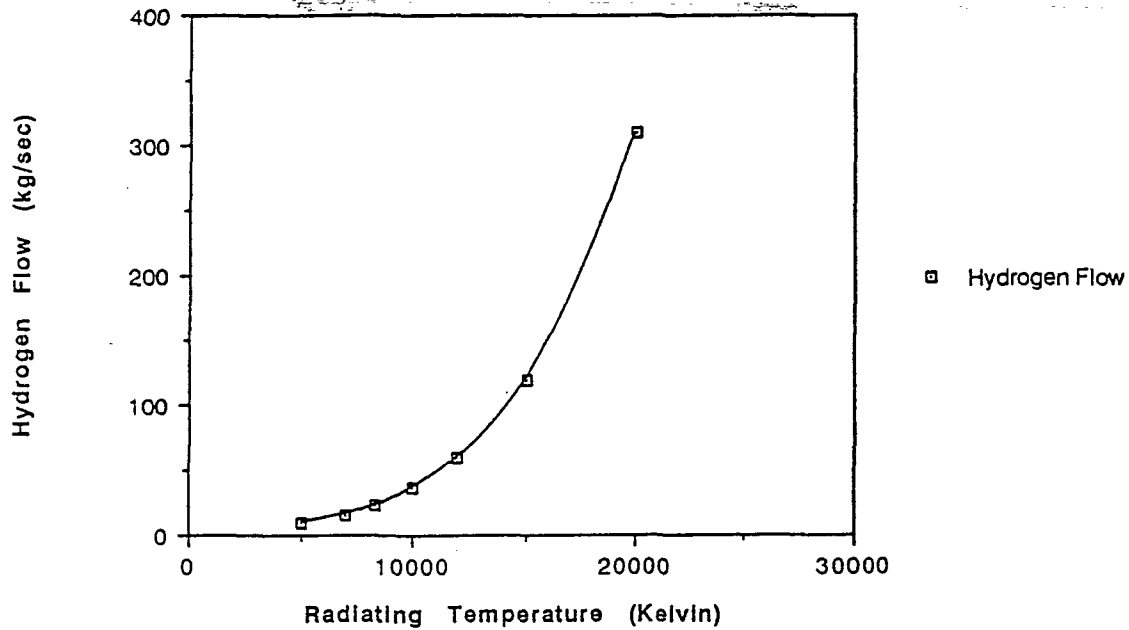
Figure 2.6: Hydrogen Propellant Flow Rate vs. Radiating Temperature.

Thrust-to-Weight Ratio vs. Uranium Temperature
for the Nuclear Light Bulb Engine.
Source: T. Latham, UTRC, c/o S.Borowski, NASA, LeRC
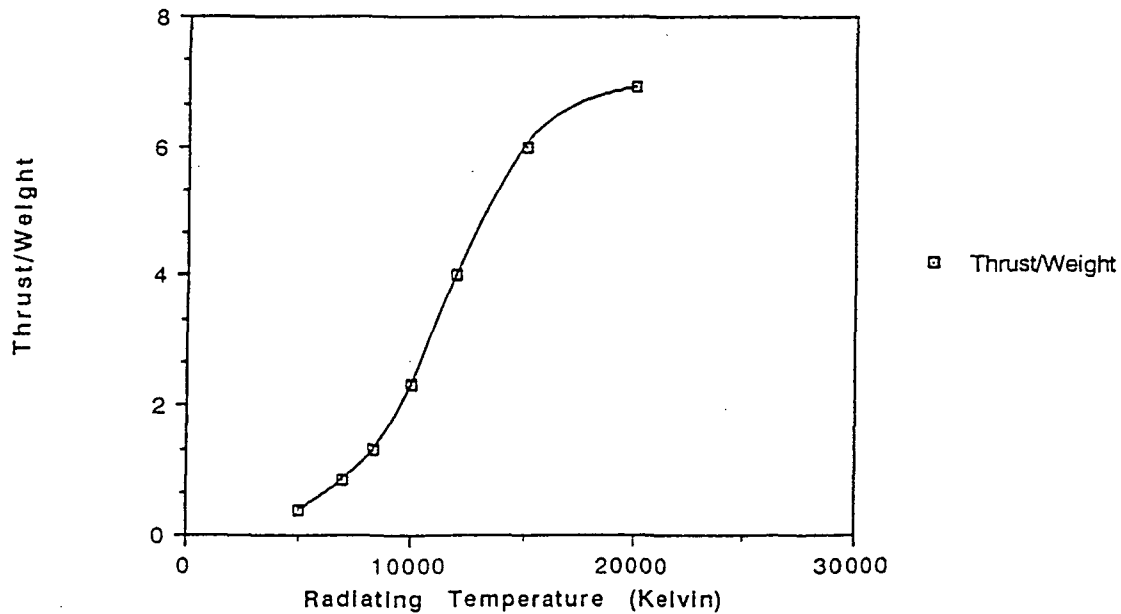
Figure 2.7: Thrust-to-Weight Ratio vs. Radiating Temperature.

Figures 2.4 through 2.7 are some of the graphical representations of the data supplied by Dr. Borowski. Respectively, they are the graphs of engine thrust, specific impulse, hydrogen

2.11

propellant flow rate, and thrust-to-weight ratio versus the uranium radiative temperature for the nuclear light bulb engines. In general, most of the engine parameters follow the same parabolic pattern as shown in the graphs of engine thrust and propellant flow rate as functions of the radiating temperature of the uranium core. However, the specific impulse and the thrust-to-weight ratio are not increasing parabolic functions of the temperature, due to thermodynamic processes occurring within the engine, and the need for an external radiator for NLB engines generating I_{sp}'s greater than 2100 secs[10]. Therefore, the I_{sp} and thrust-to-weight ratio will eventually "level off" to some finite value as radiating temperature is increased.

2.2.3 Method of Analysis

For the purposes of Project WISH, it was desired to find nuclear light bulb characteristics for a number of engines, not necessarily the seven engines given from the available data. In this way, accurate projections could be made for the required technology for the NLB engines.

In terms of the type of analysis performed, there were many ways to analyze the performance characteristics of the NLB. Some methods would include a highly rigorous analysis involving the fluid mechanic and thermodynamic processes within the engine, others would use derived formulas given in a technical report. Concerning the rigorous analysis, it was felt that this method would be most like "re-inventing the wheel". The results of the analysis would be limited by the simplifying assumptions needed to obtain numerical results. It was determined that this type of analysis, although it would be more thorough, would take too much time and effort, especially since it was known that most of the information needed was already available in technical reports. In terms of using derived formulas, although many technical reports existed on the NLB engine, there were very few formulas that were usable. This is in contrast to the Phase II team, who based their analysis on previously derived formulas.

It was assumed that the existing data points were generated by continuous functions. Keeping in mind the thermodynamic processes that can cause discontinuities, it seemed feasible that finding a given engine characteristic as a function of radiating temperature would yield accurate values for the temperature ranges in between the existing data points. In other words, any given engine parameter could be found implicitly for a given value of radiating temperature.

2.12

To generate these functions, a graphics software package for the Macintosh, called CricketGraph, was used. By curve fitting the data, CricketGraph generated the desired functions for all the parameters given in the existing data. Because a parabolic pattern was easiest to obtain an accurate curve fit, all the engine parameters were made a function of the radiating temperature of the uranium. The only exception to this case was finding a function curve for the specific impulse. The most accurate curve fit approximation for this parameter was by making the specific impulse a function of chamber pressure.

Examples of the third order polynomial curve fit approximations are shown in Figures 2.8 through 2.10. The hydrogen flow rate is a representative sample of the majority of engine parameter functions. Figure 2.9 shows the specific impulse as a function of chamber pressure, and Figure 2.10 displays the "two curve fit" of the thrust-to-weight ratio. The R^2 parameter is a measure of the accuracy of the curve fit to the existing data points. $R^2 = 1.000$ represents the highest degree of accuracy.
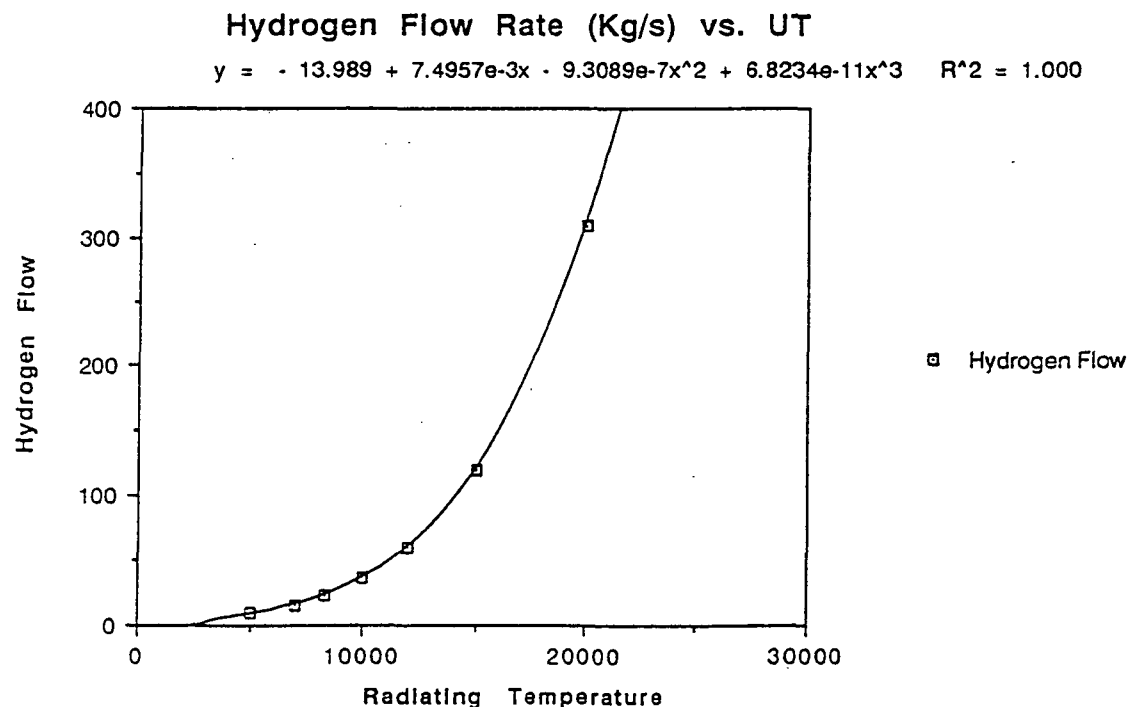
Figure 2.8: Hydrogen Flow Rate Curve Fit.

2.13

Isp vs. Chamber Pressure

y = - 712.47 + 7.5043x - 5.6814e-3x^2 + 1.6652e-6x^3   R^2 = 0.999
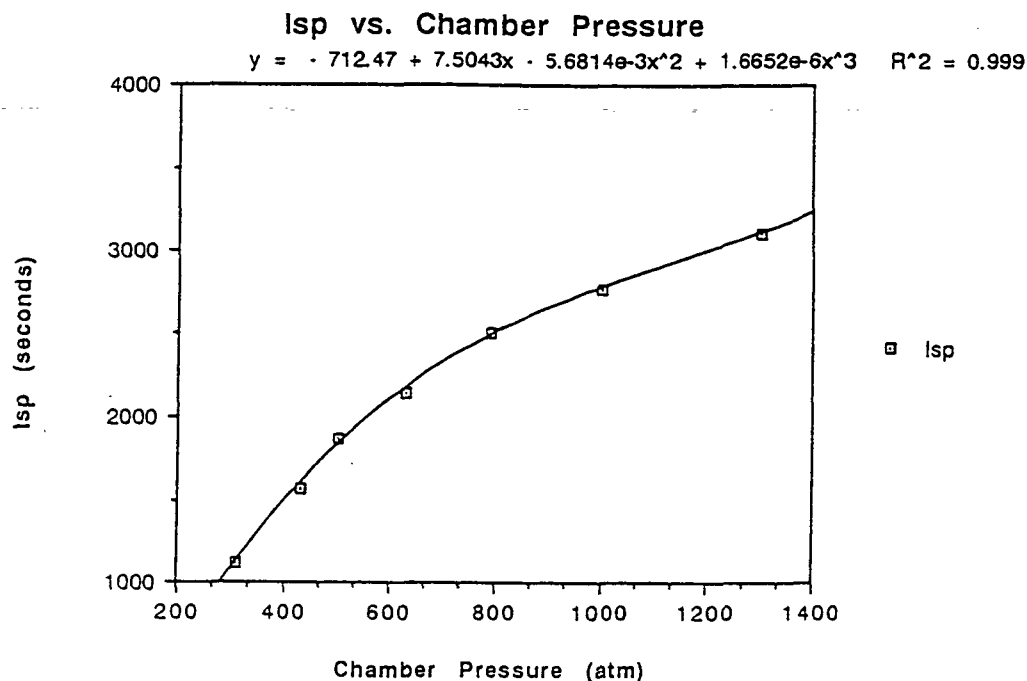
Figure 2.9: I_{sp} as a Function of Chamber Pressure.

Thrust-to-Weight Ratio vs. Uranium Temperature
"Two-Curve" Approximation

y = - 1.3609e-2 + 9.0717e-5x - 1.8375e-8x^2 + 3.2260e-12x^3   R^2 = 1.000

y = - 14.950 + 2.3092e-3x - 6.0833e-8x^2   R^2 = 1.000
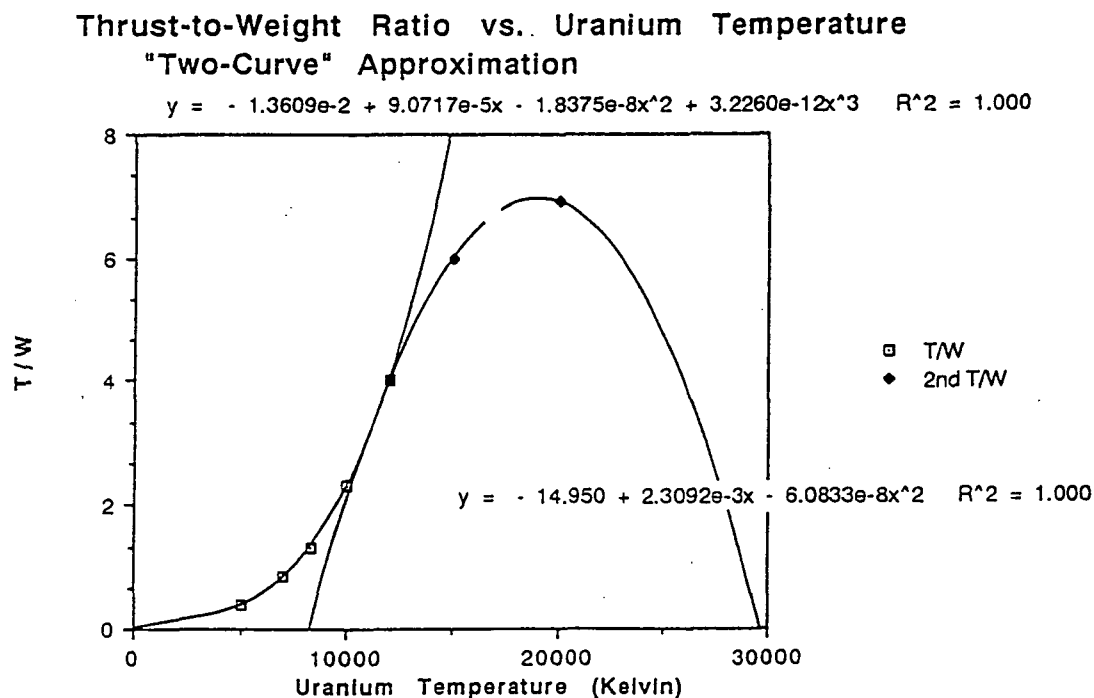
Figure 2.10: "Two-Curve Fit" Approximation for T / W Ratio.

The functions generated from CricketGraph were then used in FORTRAN programs expressly written for the purposes of Project WISH. The purpose of the program NLBDATA.FOR

2.14

was to calculate engine parameters for the uranium radiating temperature range of 5,000 to 20,000 Kelvin in increments of 500 K. Another program, NLB.FOR, generated nuclear light bulb engine characteristics for an engine possessing a thrust level specified by the user. The output for these programs are shown in Figures 2.11 and 2.12. The actual programs can be found in Appendix A at the end of this report.

Table 2.4: Comparison of NLB Engine Parameters at 10,000 K.

Engine Parameter	NASA	NLBDATA	% Deviation
Thrust (KN)	784	771	1.66
Isp (Seconds)	2150	2177	1.26
Mass (kg)	34,747	34,448	.86
Hydrogen Flow (kg/s)	37.2	36.11	2.93
T/W Ratio	2.3	2.28	.87
Reactor Power (MWth)	10,000	10,050	.5
Specific Power (kW/kg)	287.79	291.75	1.38
Chamber Pressure (atm)	630	630.3	.05
Exit Temperature (K)	8,000	7,973	.34
Exit Velocity (m/s)	21,091.5	21,356.4	1.26

Table 2.5: Comparison of NLB Engine Parameters at 20,000 K.

Engine Parameter	NASA	NLBDATA	% Deviation
Thrust (KN)	9,408	9,402	.064
Isp (Seconds)	3,100	3,095	.16
Mass (kg)	138,988	138,887	.07
Hydrogen Flow (kg/s)	309.5	309.4	.032
T/W Ratio	6.9	6.9	0.0
Reactor Power (MWth)	160,000	159,903	.06
Specific Power (kW/kg)	1,151	1,151	.013
Chamber Pressure (atm)	1,300	1,299	.077
Exit Temperature (K)	16,000	16,002	.012
Exit Velocity (m/s)	30,411	30,362	.12

2.15

2.3 RESULTS AND CONCLUSIONS

In order to check the accuracy of NLBDATA.FOR results, the generated engine parameters were checked with existing data from NASA for the radiating temperatures of 10,000 and 20,000 K. The percent deviations between the known and computer generated values were found. This gave an idea as to the accuracy of engine parameter values that the program was generating in between the known points. The results are listed in Tables 2.4 and 2.5 for the temperatures of 10,000 and 20,000 K, respectively. It can be seen that the average percent deviation at the value of 10,000 K is 1.11%, and the average error is 0.06% for 20,000 K.

The source of the deviation stems mainly from the approximation scheme used by CricketGraph. For normal applications, the number of significant figures for the coefficients of the polynomial functions are sufficient. However, because of the large values associated with this analysis, more significant figures for the coefficients were needed.

As mentioned previously, there were other types of analysis that would have yielded results through considerable thermodynamic and fluid mechanics analysis. However, our approach proved to yield accurate values, and it did so within an acceptable time frame.

As expected, the NLB engine far exceeded the open-cycle engine in terms of performance requirements and projected available technology for E-City. Use of the NLB means that the radiation problem is eliminated. The fact that the number of engines required for a given mission is almost ten times less than an open-cycle system points to a further reduction in total engine mass. Based on the results of the analysis, the NLB engine is the recommended propulsive system for E-City.

CHAPTER 3

MISSION, SYSTEM, AND CONFIGURATION DESIGN

3.0 INTRODUCTION

The final design Phase of the E-City was divided into four concurrent tasks: Propulsion system design consisted of a re-evaluation of the NLB engine to eliminate the plume radiation problem associated with the SGRCNR resulting in the selection of the closed-cycle NLB engine as discussed in Chapter 2. Mission profile, Configuration, and System Parameters were to be automated to facilitate the determination of optimum parameters in conjunction with the other three tasks. Static and Dynamic Structural Design was performed using the evolving design parameters and optimized for minimum mass using Automated Structural Optimization System (ASTROS)[4], provided by the Air Force Wright Aeronautical Laboratory, and Structural Dynamics Research Corporation's I-DEAS[3] finite element software (Chapter 4). Design of the Automatic Control System was performed using PRO-MATLAB[5] software to find the optimum control law and thruster configuration to minimize control power requirements as well as propellant consumption and thrust levels required (Chapter 5).

3.1 MISSION PROFILE

The Saturn envelope mission profile addressed in the Phase II report [2] was chosen as the baseline because it has the most demanding performance requirements of all the feasible missions. It requires a delta-V of 50 km/sec for transfer from the nominal orbit of 4 AU. See Figure 1.1.

The primary objective of the structural design was to minimize the dry mass while fulfilling essential performance parameters. The driving factor determining the overall mass is the amount of propellant required for the mission. Equation 3.1 shows the relation between the dry mass and the propellant mass.

$$m_p = m_{dry} \left[\exp \left(\frac{\Delta V}{I_{sp} \, g} \right) - 1 \right] \qquad (3.1)$$

The value of m_p / m_{dry} was 4.19 using a delta-V of 50,000 m/s and an I_{sp} of 3095 sec, which

dramatically shows the impact of adding mass.

3.2 SYSTEM PARAMETERS

The computation of system parameters (masses, dimensions, forces, etc.) was automated so that the effect of changes in the configuration could be analyzed interactively. The 30 design variables used are those that affect the dry mass of the E-City. The program ECITY.FOR is included in Appendix B, and the input variables are listed in Table 3.1.

Table 3.1: Input Variables for Program ECITY.FOR

Mission Parameters	Design Variables	Structural Variables
Population	Volume per Person	Material Density
ΔV	Mass per Person	Torus
	Artificial Gravity	Tanks
	Atmospheric Pressure	Spokes
	Mass per Engine	Shaft
	Specific Impulse	Shield Shell
	Thrust Duration	Working Stress
	Number of Tanks	Torus
	Propellant Pressure	Tanks
	LH_2 Density	Spokes
	Subsystem Masses	Shaft
	Power Generation	Shield Shell
	Payload	
	Communications	
	Heat Exchanger	
	Control Thruster	
	# of Thrusters	
	Control LH_2	
	Control LO_2	

3.3 E-CITY CONFIGURATION

The configuration of the E-City is the result of integrating the requirements dictated by a low mass, structurally sound design, good controllability, and minimum stress on the inhabitants. The inhabited and rotating torus section was found to be the most efficient geometry for the living space. It's dimensions were determined by human factors considerations studied in Phase II[2], Chapter 4.

By using the closed-cycle nuclear engine (NLB), the radiation shield surrounding the torus was minimized to that which was required to protect the inhabitants from cosmic and solar radiation. It consists of 14 meters of liquid hydrogen contained in a separated pressure vessel, with a vacuum between it and the inhabited torus for enhanced thermal insulation.

It has been assumed by the previous design teams on Project WISH that the propellant would occupy a roughly cylindrical space whose long axis is aligned with the spin axis. The question of whether the propellant tanks should spin or not has not been addressed up to this point. It was understood that for this analysis, the results could be scaled to fit larger or smaller tanks without going through the whole optimization process; just analyze the stress concentrations and verify the scale factors.

The geometry and location for the tank has a great impact on the long term attitude stability and control of the E-City as it is a major contributor to mass moments of inertia (See Chapter 5). The mass moment of inertia was plotted against the aspect ratio (AR) of the enclosed LH_2, where AR is the ratio of height to diameter. It was assumed that the thin wall of the tank did not make a significant contribution in comparison to the other components.
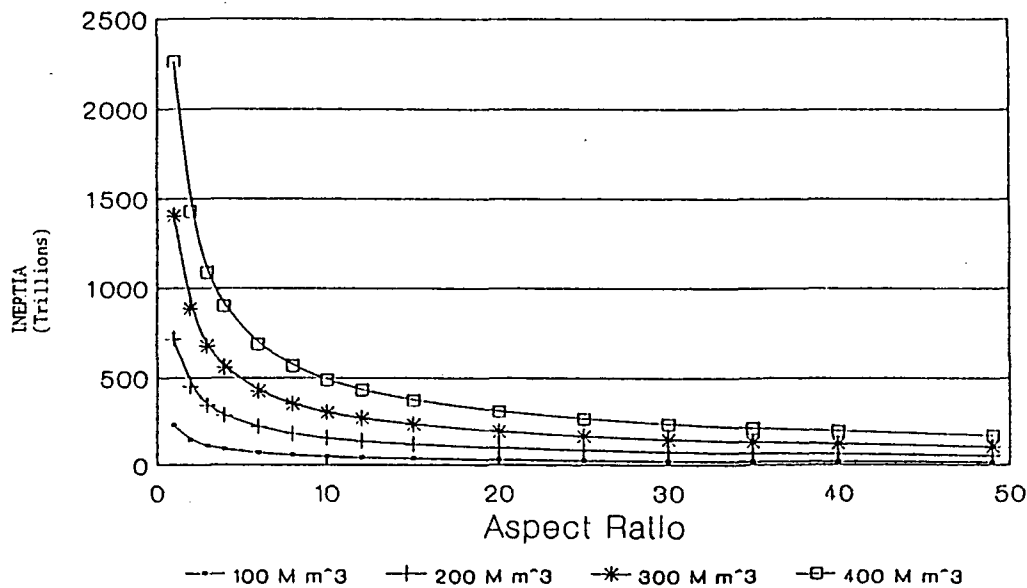
Figure 3.1: Mass Moments of Inertia vs. Aspect Ratio

3.3

From Figure 3.1, AR=6 was chosen as the initial starting point because a small move in either direction would provide a significant change in the mass moment of inertia. It was decided to perform the analysis on one large tank because it is the simplest configuration and the results could easily be scaled for smaller tanks. The initial configuration of the tank was chosen to be cylindrical with spherical end caps. A review of the literature showed that spherical end caps would allow a minimum thickness design of the end caps. The stress in a sphere is given by eq. 3.2 while eq. 3.3a and 3.3b are the stresses for a cylinder.

$$\sigma = pr/2t \hspace{4cm} (3.2)$$
$$\sigma = pr/2t \hspace{1.5cm} \text{(Longitudinal)} \hspace{2cm} (3.3a)$$
$$\sigma = pr/t \hspace{1.5cm} \text{(Circumferential or Hoop)} \hspace{1cm} (3.3b)$$

The program NUMTANK.FOR (Appendix B) shows that for a given volume the spherical tank surface area is lower than that of the cylindrical tank.

3.4 LOADS ON THE TANK

The three loads on the tank are internal pressure, external thrust, and rotation (initially). Internal pressure is required to keep the hydrogen in the liquid state. The working pressure was set at 0.2 atmospheres (atm.), which corresponds to the vapor pressure of liquid hydrogen (LH_2) at 16 Kelvin (-433 degrees F). Using a lower pressure allows the tank wall to be thinner for the same working stress. An aluminum alloy (12% silicon, 0.5% magnesium) was chosen by previous design teams for it's strength properties from NASA SP-413[11]. It has a yield stress of 248 MPa and a working stress of 165 MPa using a safety factor of 1.5. Thrusting imparts an acceleration to the fluid in the tank which will cause a hydrostatic pressure gradient, just as if it were in a gravitational field. A longitudinal acceleration of 0.012 m/s^2 was found using Newton's second law with a dry mass of 11.18 billion kg and a thrust of 136 million Newtons. The pressure gradient can be determined for any height of propellant using the program TANKPRESS.FOR (Appendix B). For cylindrical tanks with spherical ends, assuming the end cap has the same thickness as the cylinder, the pressure due to thrusting will cause a stress concentration at the

3.4

junction of the end caps and cylinder, where the pressure is 21691 Pa. Table 3.2 shows the expected cylindrical wall thickness based on a working stress of 165 MPa and a pressure of 21705 Pa, using eq. 3.3b.

Table 3.2: Minimum Wall Thickness for Different Tank Geometries

AR	Radius (m)	Wall Thickness (m)
6	156.7	.0206
5	167.2	.0220
4	181.2	.0238
3	201.4	.0265
2	235.6	.0310

The maximum pressure due to thrusting was verified to occur when the tank was full. As the propellant was used and the acceleration increased, the pressure at the bottom of the tank decreased for all tank sizes.

It was determined that the propellant tank(s) should not rotate in order to reduce tank mass and reduce the hydrodynamic complexity. A rotating tank would cause the propellant to exert a pressure on the sides of the tank due to "centrifugal force". The program ROTPRESS.FOR (Appendix B) was developed to analyze the side wall pressures due to tank rotation. The worst case condition was evaluated where the tank was completely full (no gas space). Table 3.3 shows the minimum necessary wall thickness for the case where the hydrostatic forces are added to the 0.2 atm. pressure required to keep the hydrogen liquid. Note that the stresses due to the tank wall body forces are ignored in this analysis.

Table 3.3: Minimum Wall Thickness in a Rotating, Pressurized Tank

AR	Radius (m)	Wall Thickness (m)
6	156.7	.0296
5	167.2	.0330
4	181.2	.0378
3	201.4	.0457
2	235.6	.0617

3.5

A decrease in wall thickness of between 30% for the long tank and 99% for the short tank is possible by eliminating the tank rotation, see Table 3.2.

The hydrodynamic effect of tank rotation is a potentially more serious problem, and is caused by a radial acceleration (caused by rotation) that is up to 100 times greater than the longitudinal acceleration (from thrust). At some point the tank will effectively "run out of gas" because the propellant is forced away from the main feed orifice (drain) and against the wall, see Figure 3.2. Table 3.4 lists the radial acceleration at the tank wall. Recall that the longitudinal acceleration is .012 m/s^2.
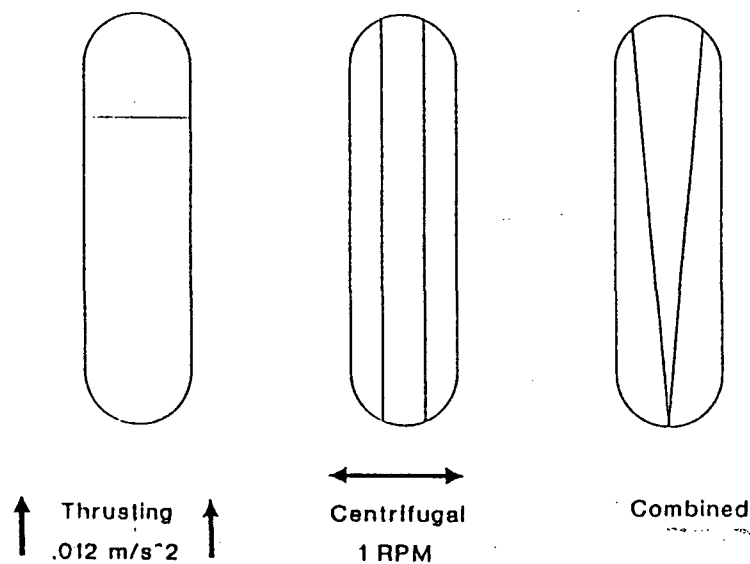
Figure 3.2: Hydrodynamic behavior.

Table 3.4: Radial Acceleration for Different Tanks

AR	Radius (m)	Radial Acceleration (m/s^2)
6	156.7	1.72
5	167.2	1.83
4	181.2	1.99
3	201.4	2.21
2	235.6	2.58
1	319.8	3.51

3.6

If there is no thrusting, the propellant will pile up along the walls with a tube of gas running down the centerline, and none of the LH_2 will reach the drain, even with full tanks.

Adding pressurized bladders to force the liquid into position is a possibility, but it could force the internal pressure to higher levels than desired, and gas will build up under the bladders unless some clever mechanism is devised. Another possibility is to design non-symmetrical tanks, but the whole purpose of using symmetrical tanks was to reduce the mass.

3.5 NUMBER OF TANKS

The number of tanks used in the final design will not depend on the total tank mass. The program NUMTANK.FOR (Appendix B) was used to compare the total mass and total surface area for various aspect ratios and number of tanks, and shows that the total mass goes down as the number of tanks is increased. On the other hand, the total surface area goes up, as is seen in Table 3.5. This comparison also conclusively showed that a sphere was the most optimum shape for the tank configuration regardless of the number of tanks used.

Table 3.5: Tank Mass and Area Comparison

AR	H_2 mass (Mkg)	Tanks	Mass (Mkg)	Area (km^2)	Thickness (m)
1	9725.1	1	68.2	1285	.0202
1	9725.1	5	67.6	2197	.0117
1	9725.1	10	67.5	2768	.0092
2	9725.1	1	82.5	1395	.0298
2	9725.1	5	81.6	2385	.0172
2	9725.1	10	81.3	3006	.0136
4	9725.1	1	89.4	1649	.0234
4	9725.1	5	87.6	2820	.0134
4	9725.1	10	87.1	3554	.0106

The heat transfer into the tank will increase with surface area, no matter how good the insulation is, and increasing the number of tanks will increase the LH_2 boil-off rate, as discussed in Chapter 6. From a structural and heat transfer point of view, one large tank is the optimal configuration. One tank exposes the lowest surface area and would require the minimum mass in

3.7

coatings for heat transfer and coverings for meteoroid protection. The primary disadvantage of one tank is the potential loss of the entire propellant load due to tank failure, whether caused by collision or random failure. Redundancy was the primary consideration in dividing the propellant load into two tanks.

3.6 SHAFT

The shafts connecting the torus hub to the two propellant tanks were made as short as practical to reduce the amount of material required and reduce the applied bending moments due to control inputs. The shaft radius was influenced primarily by the requirements of transmitting axial (longitudinal) loads to its associated tank at points where the tank structure could take the stress without an increase in the thickness of the tank. The minimum radius of the shaft was shown to be 100 meters by the analysis of the tank wall stresses in Chapter 4.

3.7 CONCLUSIONS

The configuration of the E-City was driven by the need to minimize the overall mass. A reduction in the mass of the propellant tanks was realized by using a spherical configuration. Dual tanks were used to provide minimum redundancy and to prevent the total loss of propellant should one tank fail. Control of the center of gravity is an added possibility with dual tanks. Mass associated with piping and wiring was assumed to be minimized by locating associated subsystems close together, such as the bubble radiator, engines and power systems. The torus and cosmic radiation shield were fixed by human factors, primarily the desire to have Earth standard gravitational acceleration, Earth normal atmospheric pressure, a biosphere environment, and protection from cosmic and solar radiation to a 5 rad per year level. The tanks were de-spun to eliminate the hydrodynamic effects associated with spinning propellant. The final dimensions of the propellant tanks, spokes, and shafts were dependent on the static and dynamic structural loads.

CHAPTER 4

STRUCTURAL DESIGN

4.1 STATIC STRUCTURE

4.1.0 Introduction

The static structural analysis played a pivotal role in the design of the E-City. It was used as the initial basis for the determination of cross sectional areas and other dimensions, which were then analyzed as a whole for the dynamic behavior of the entire vehicle. Those areas that needed further modifications to meet structural dynamic criteria were then treated and allowances made for the required modifications. The majority of the static structural analysis was included in the parameters program ECITY.FOR since the overall mass was related to the component dimensions and their densities. The reader will note that some of the analyses use different values for variables such as accelerations, masses, and forces. This was the result of using the most current values from analyses as they were completed. If the analysis in question requires the most current data to be valid, it was reaccomplished. In most cases, this was not the case.

4.1.1 Propellant Tanks

The analysis of the WISH hydrogen propellant tank is outlined in the following sections. It was the intent of this design effort to optimize the tank configuration so that the total mass was minimized. Since the tank is the single largest component of the E-City, mass minimization was essential to gain the highest performance possible. Pressure vessel theory is fine for determining the overall stress characteristics of the propellant tank but is inadequate for pinpointing stress concentrations due to the combined loads of thrusting, rotation, and pressure.

Initial Assumptions

As mentioned previously, the working pressure was set at 0.2 atm., which corresponds to the vapor pressure of LH_2 at 16 Kelvin. Using a lower pressure allowed the tank wall to be thinner for the same working stress. Recall also that the aluminum alloy (12% silicon, 0.5% magnesium) has a yield stress of 248 MPa and a working stress of 165 MPa using a safety factor

4.1

of 1.5.

The models were created on I-DEAS[3] utilizing the symmetry of the tank to reduce the size of the model and the number of elements in the finite element model. This method of modeling had two benefits: the first was to reduce the computing time necessary to solve the finite element mesh, the second was to reduce the amount of memory required to execute the mesh solver. The number of elements allowable in the finite element mesh was restricted by the limited amount of memory available in the I-DEAS computer accounts.

Initially the tank structure was optimized, maintaining a constant volume, using only the pressure forces in an attempt to obtain a uniform stress pattern. Considering the magnitude of the thrusting forces, it was felt that the best approach to obtain the optimum configuration would be to initially design only for the pressure forces and then once this was completed, the thrusting force effects would be analyzed. Another advantage to this approach is that the I-DEAS results could be compared and verified with thin pressure vessel theory.

Once the structural configuration was finalized, optimization of the wall thickness was then performed to achieve a maximum principle stress equal to the working stress of the aluminum. Analyzing the effect of the thrusting forces and their affect on the design of the tank concluded the stress analysis.

2-D Modeling

Modeling in 2-D was begun after many attempts at modeling the tank in 3-D failed because of coincident nodes in the finite element mesh along the axis of revolution. It was felt that due to the symmetry of the tank about the longitudinal axis, the finite element model could be reduced to a lengthwise cross section of the tank and cut again in half down the centerline.

Stress analyses were performed on several models covering various height to diameter aspect ratios (ARs) in order to achieve the optimum structural tank shape. The initial model consisted of a cylindrical tank with hemispherical end caps with an AR of 6, corresponding to a height of 1845 m. and radius of 154 m.

The initial wall thickness was determined using:

$$\sigma = pr\,/\,t$$

A thickness of 0.1 m, for the AR=6 tank, was calculated to give the optimum stress level of 165 MPa. This thickness was calculated using a conservative initial estimate of the required internal pressure of 1 atm., which as previously mentioned was later finalized at 0.2 atm.

Modeling the entire propellant tank on I-DEAS as stated proved to be impractical. The proportions of the height and thickness dimensions (a ratio of over 18000 with the AR=6 tank) resulted in a model that resembled a line on the screen. The stress distribution across the tank wall could not visibly be seen. To overcome this problem, the wall thickness and the loads were multiplied by a common scaling factor. This did not affect our results since the stress in a pressure vessel is a function of p / t. Therefore, the stress results are comparable to the values that would occur in a tank constructed using the actual dimensions. The scaling factor was chosen such that the thickness would be large enough in relation to the height to give a good visual representation of the stress patterns across the wall.

Since the main axis of the tank was aligned with the y-axis, restraints were placed on the top and bottom edges along the centerline in the x and z-directions and in all rotations. In applying these restraints it was assumed that an actual tank would expand uniformly. Internal pressure forces were applied to each model using the edge pressure option on I-DEAS. The magnitude of the internal pressure was dependent on the aspect ratio and equal to the minimum pressure of 0.2 atm. multiplied by the appropriate scaling factor.

As shown in Figure 4.1, the maximum principle stress for the AR of 6 case is 5690 MPa, obviously this is above the working stress of aluminum. With this configuration, stresses are concentrated on the insides of the end caps and at the outside midsection of the cylindrical portion. A constant cross sectional area was maintained as the ARs decreased to ensure the same amount of material was used to construct each model. The stress patterns became more uniform and the maximum principle stress decreased as the AR was decreased. Table 4.1 shows a comparison of the maximum principle stresses as the AR was changed from 6 to 1. A comparison of the stress contours can be made by referencing the listed figures.

4.3

Table 4.1: Comparison of Tank Aspect Ratio to Maximum Principle Stress

Figure	AR	σ_{max} (MPa)	% change in σ_{max}
4.1	6	5690	---
4.2	4	2570	-54.8
4.3	2	670	-73.9
4.4	1	89	-86.7

Based on the results of this comparison table and on the stress contours shown in Figures (4.1-4.4), it is easily seen that the optimum configuration for the tank is in fact a sphere and not a cylinder. This agrees with thin pressure vessel theory.

After completing the configuration optimization, the process of optimizing the wall thickness could proceed. For this procedure only the wall thickness was varied, the pressure remained constant. The optimum thickness for the 2-D model was found to be 64.0 m., which scales to an actual thickness of 0.046 m. The maximum principle stress for this model equaled the working stress of our material, see Figure 4.5; however, thin pressure vessel theory predicts an optimum wall thickness of approximately 0.01964 m. or 27.43 m. scaled. After further analysis of the variation in the stress results obtained from I-DEAS and from theory, it was decided that, because the tank was modeled in 2-D, the I-DEAS software was not accounting for the circumferential stress.

3-D Modeling

In an effort to create a finite element model that would correctly represent the actual tank, a 3-D model was generated. Revolving a cross section similar to that used in 2-D modeling 90 degrees about the y-axis created a model of one quarter of the tank. The problem of coincident nodes that was experienced in the previous 3-D modeling attempts was eliminated by offsetting the 2-D profile from the y-axis. When the 2-D profile was revolved about the y-axis, the centerline of the tank and the axis of rotation were not the same because of the offset; therefore, the nodes on the centerline of the tank were not duplicated by the revolution.

In creating the restraints for the 3-D model the assumption of uniform expansion was again made to determine the nature of the restraints. The main axis of the model was again oriented along the y-axis; the edge on the xy-plane was restrained in the z-direction and all rotations, and

Figure 4.1

ORIGINAL PAGE
COLOR PHOTOGRAPH
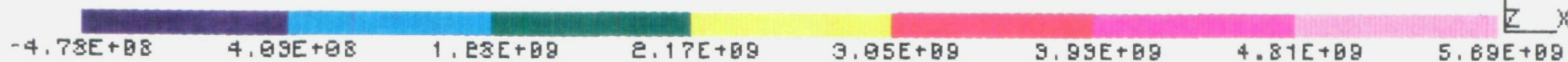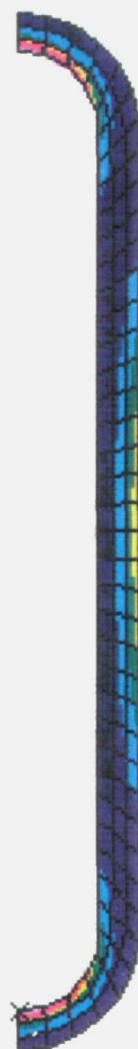
AR = 4.0   WALL THICKNESS = 94.72

LOADCASE:1
FRAME OF REF:GLOBAL
STRESS - MAX PRIN MIN:-2.03E+08 MAX: 2.57E+09

SHELL SURFACE:TOP

-2.03E+08   1.99E+08   5.89E+08   9.88E+08   1.38E+09   1.78E+09   2.17E+09   2.57E+09

Figure 4.2

AR = 2.0   WALL THICKNESS = 121.87
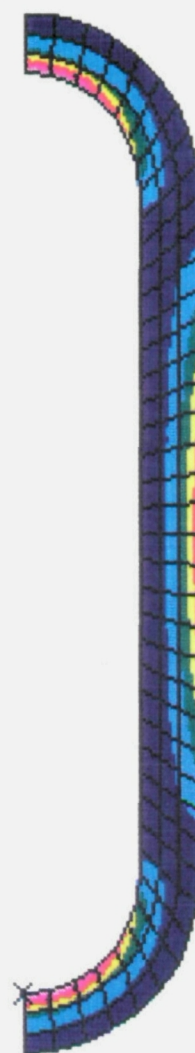
LOADCASE:1
FRAME OF REF:GLOBAL   SHELL SURFACE:TOP
STRESS - MAX PRIN MIN:-8.68E+07 MAX: 6.70E+08

-3.68E+07   2.19E+07   1.29E+08   2.97E+08   3.45E+08   4.59E+08   5.61E+08   6.70E+08
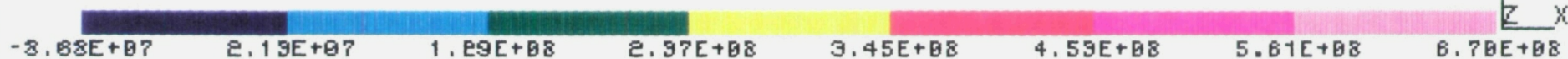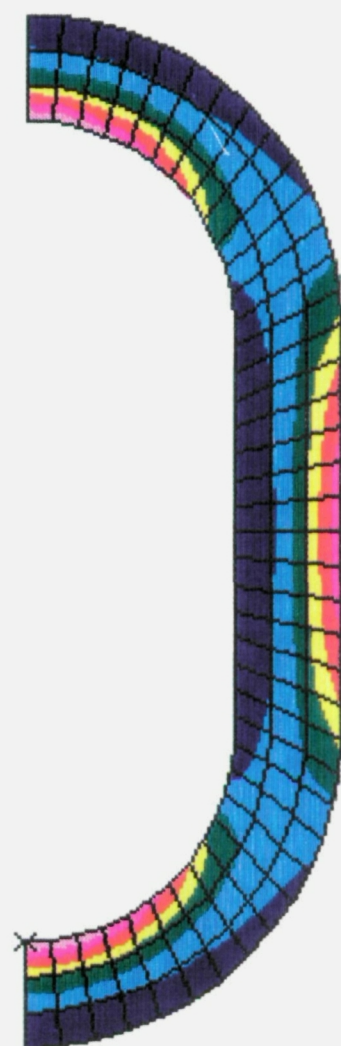
Figure 4.3

LOADCASE:1
FRAME OF REF:GLOBAL
STRESS - MAX PRIN MIN: 4.95E+07 MAX: 8.90E+07

AR = 1.0 WALL THICKNESS = 139.67

SHELL SURFACE:TOP
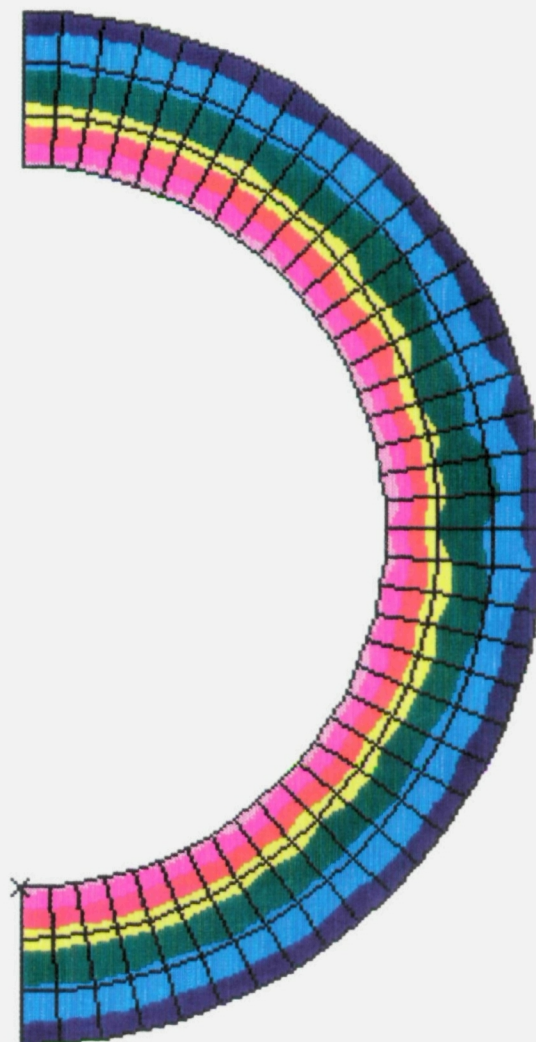
4.95E+07   5.51E+07   6.07E+07   6.64E+07   7.20E+07   7.77E+07   8.33E+07   8.90E+07

Figure 4.4

AR = 1.0   WALL THICKNESS = 64.0

LOADCASE:1
FRAME OF REF:GLOBAL
STRESS - MAX PRIN MIN: 1.23E+08 MAX: 1.65E+08

SHELL SURFACE:TOP
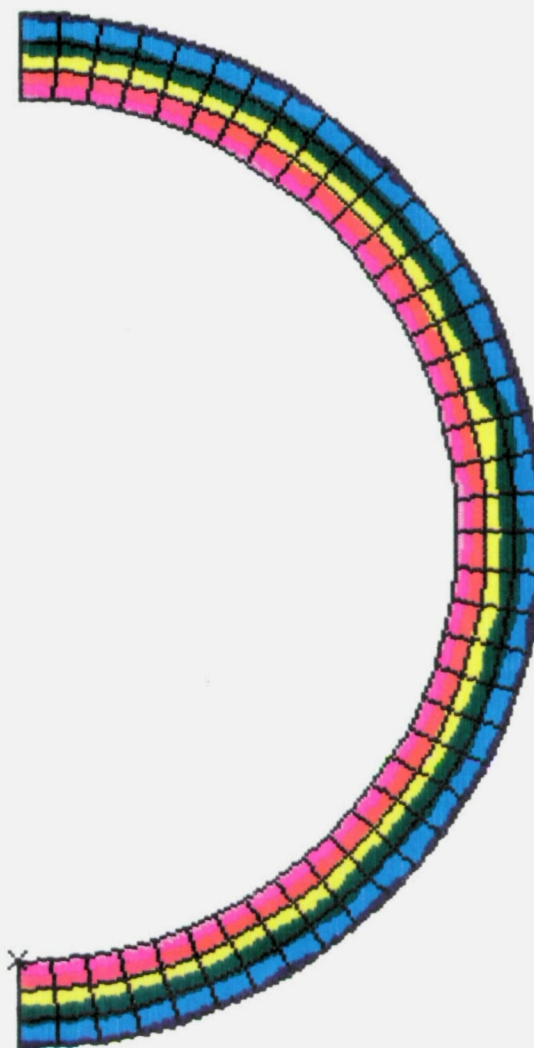
| 1.23E+08 | 1.29E+08 | 1.35E+08 | 1.41E+08 | 1.47E+08 | 1.53E+08 | 1.59E+08 | 1.65E+08 |

Figure 4.5

the edge on the yz-plane was restrained in the x-direction and all rotations. In preparation for a time when the thrusting loads would be applied, the nodes at the top of the tank along the centerline were fully restrained. Those nodes at the bottom of the tank on the centerline were restrained in both the x and z-directions and all rotations. Initial analyses were performed using internal pressure only to obtain the optimum tank wall thickness, as stated previously. Internal pressure was applied using the face pressure option on I-DEAS. As described for the 2-D case, the magnitude of the pressure was 28.304 MPa and held constant throughout.

The first model in 3-D was created using a thickness of 64.0 m to enable a comparison of the results from the optimized 2-D model. As anticipated the results differed greatly. The 3-D model resulted in a maximum principle stress of 97.5 MPa as shown in Figure 4.6. This is a 41 % decrease for the 3-D case.

The next 3-D model was created using a thickness of 27.43 which, as stated previously, is the optimum thickness predicted by thin pressure vessel theory. The maximum principle stress equaled 252 MPa for this model as shown in Figure 4.7. At first glance it appears that this thickness resulted in a stress level far in excess of the working stress. Closer examination reveals that the high stress levels are concentrated at the top and bottom of the tank along the centerline. There is also an obvious horizontal banding of the stress patterns with the center portion of the tank under the lowest stress in the range of 156 to 184 MPa. The working stress of aluminum is within this range indicating that the results from 3-D modeling agree with thin pressure vessel theory. The banding of the stress patterns is symmetric about the xz-plane referenced from the longitudinal midpoint of the tank. Thus, the tank is "flattening out" since the edge shear forces, present in an actual tank, were neglected. However, it is felt that the stress levels in the center portion of the tank are representative of the values that would occur in an actual tank given the agreement with thin pressure vessel theory; therefore, this is the optimum wall thickness.

Comparing the mass of the original aspect AR 6 tank, 99.137 MKg, with the optimized spherical tank mass of 71.696 MKg, shows a mass savings of 27.7%. This also produces a significant savings in propellant mass (see Chapter 3). Optimizing the cylindrical tank to a spherical tank also yields a significant reduction in surface area of 31.4 %, this is especially important from a heat transfer point of view.

4.10

Figure 4.6

AR = 1.0   WALL THICKNESS = 64.0

LOADCASE:1
FRAME OF REF:GLOBAL
STRESS - MAX PRIN MIN: 5.86E+07 MAX: 9.75E+07

5.86E+07   6.41E+07   6.97E+07   7.53E+07   8.08E+07   8.64E+07   9.19E+07   9.75E+07
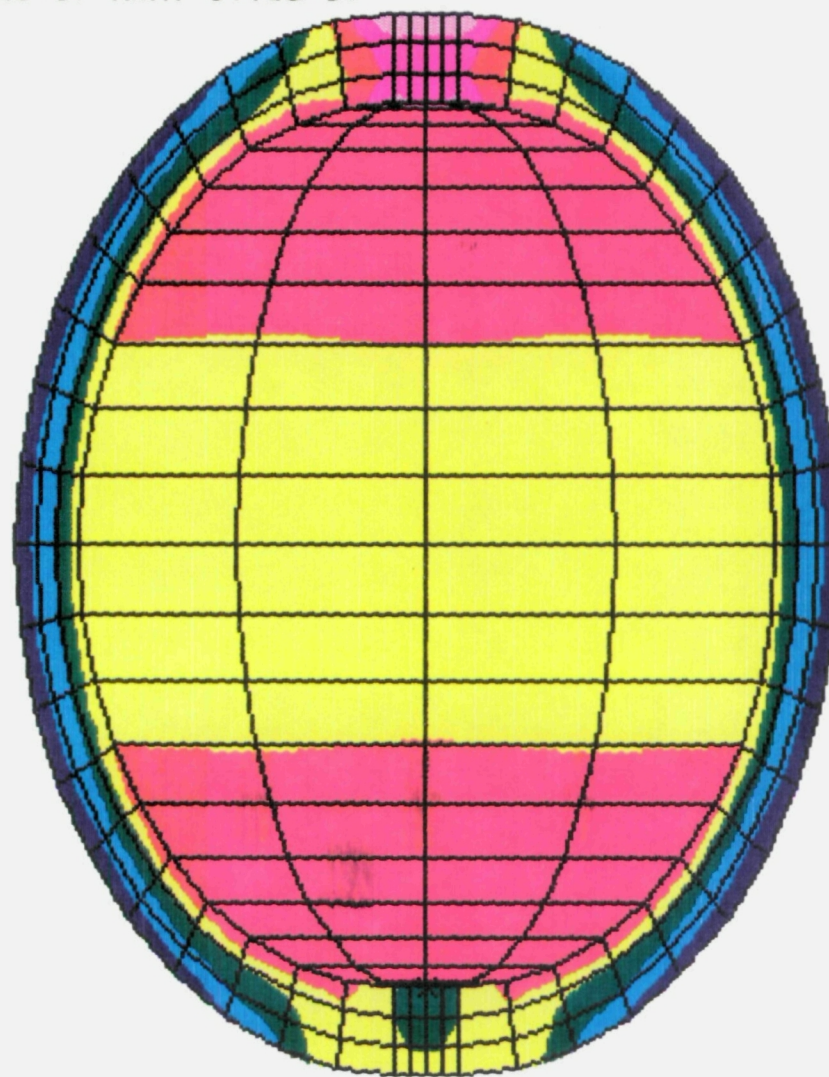
AR = 1.0   WALL THICKNESS = 27.43

LOADCASE:1
FRAME OF REF:GLOBAL
STRESS - MAX PRIN MIN: 1.56E+08 MAX: 2.52E+08

1.56E+08   1.70E+08   1.84E+08   1.98E+08   2.11E+08   2.25E+08   2.39E+08   2.52E+08
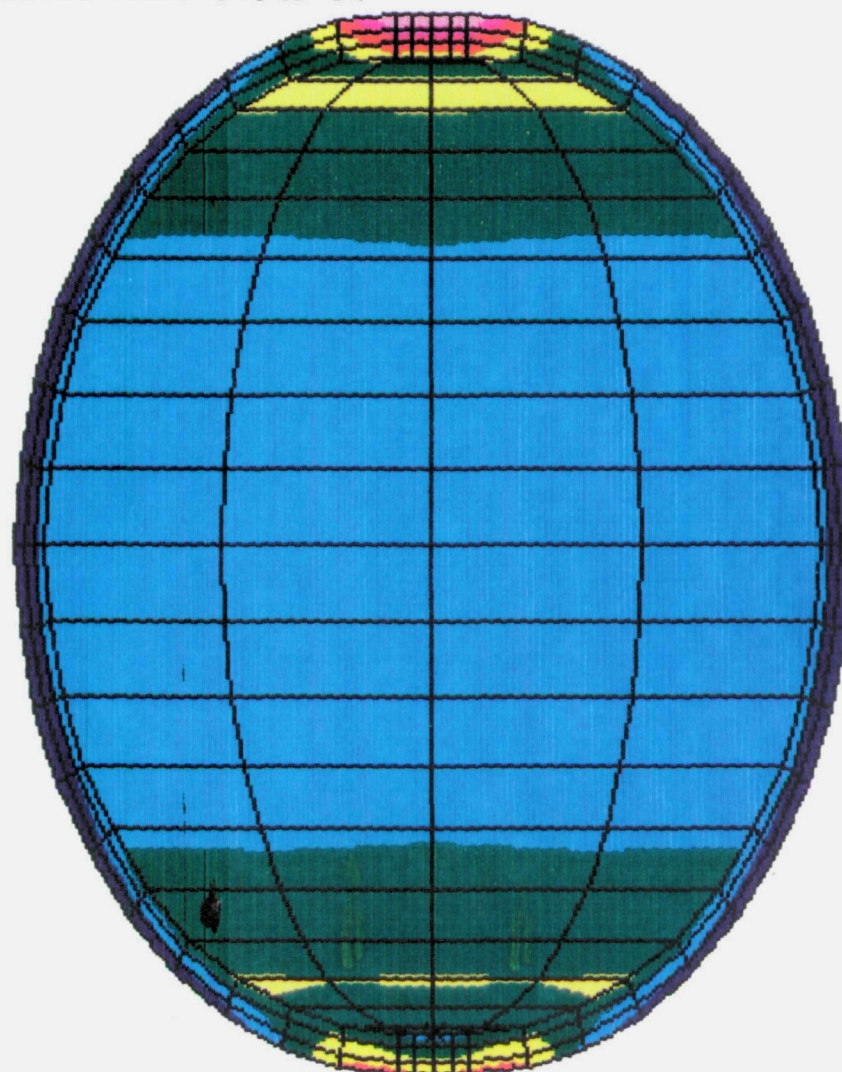
Figure 4.7

In the next case shown in Figure 4.8, the thrusting forces were applied to the previous tank configuration. As this was only a quarter of the tank, only 25 % of the total thrusting forces were applied to the bottom centerline nodes. This value too was multiplied by the same scaling factor used on the pressure. It can be seen from the figure that the stress patterns are nearly identical to those observed for the same model under pressure forces only. The maximum principle stress increases to 516 MPa, which, again, is concentrated at the top and bottom nodes along the centerline. If as before, we look at only the stresses in the center portion of the tank, we see stresses in the range of 200 to 275 MPa, which is significantly larger than the 165 MPa working stress. The displacement profile for this case, shown on the left side of Figure 4.9, gives a maximum displacement of 3.11 m. An actual tank could fail due to exceeding the 248 MPa maximum stress of aluminum.

3-D Modeling with Spar

In an attempt to reduce the stress levels in the tank and to minimize the displacement, a 20 m. thick spar (scaled) was added to close the open end of the 2-D profile. This model was restrained as before with the addition of restraining the spar in the x and z-directions and in all rotations. The loads in this model are also similar to the previous 3-D model, which included the internal pressure and the thrusting forces. The results for this case are given in Figure 4.10. The maximum principle stress for this model was 332 MPa which again is localized in stress concentrations at the top and bottom nodes in this case where the tank wall and spar meet. The addition of the spar did act to eliminate the stress banding that was occurring in the previous models as well as in reducing the maximum displacement to 0.741m, shown on the right side of Figure 4.9. The majority of the tank surface is in the stress range of 129 to 180 MPa further verifying thin pressure vessel theory.

Analysis of Spherical Tank

The analysis of the spherical tank consisted of finite element modeling of the pressure vessel to determine the locations of any stress concentrations using I-DEAS. The objective of the analysis was to determine the optimal placement of the supports to minimized the amount of

4.13
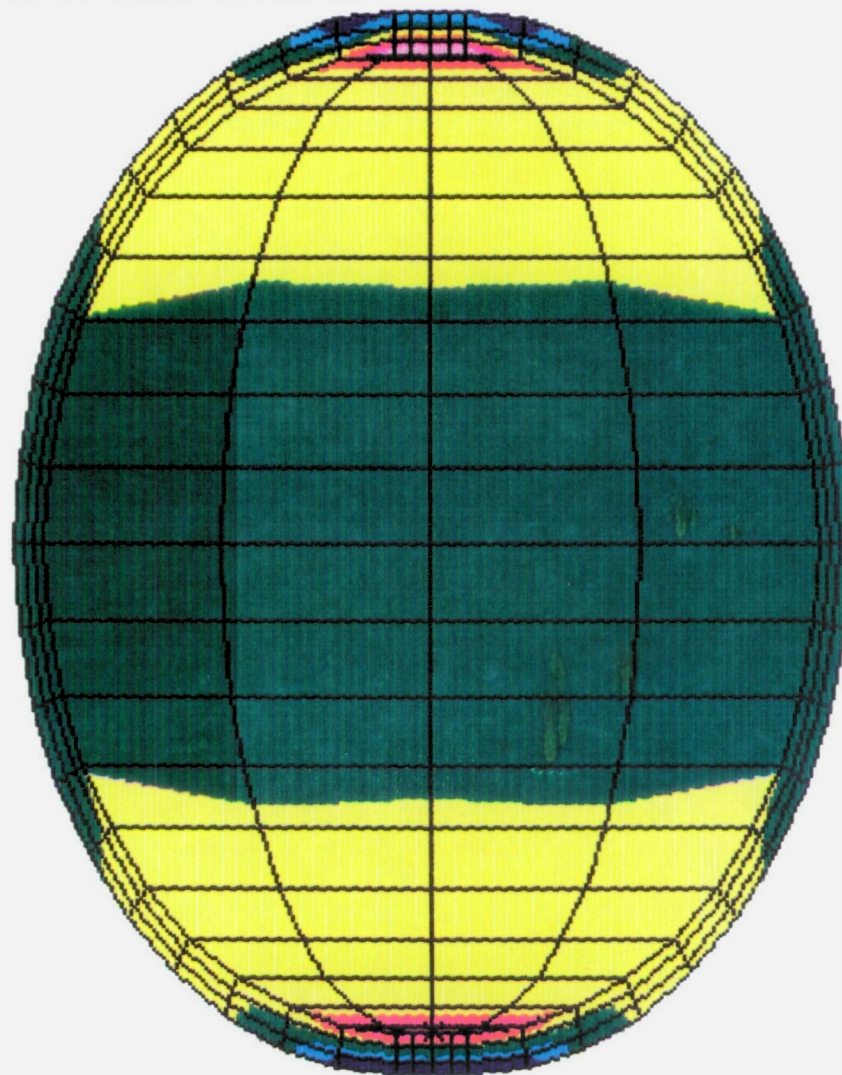
Figure 4.8

Model: 1-FE MODEL1

Associated Workset: 1-WORKING SET1

AR = 1.0   WALL THICKNESS = 27.43
LOADCASE:2
DISPLACEMENT - MAG MIN: 0.00E+00 MAX: 3.11E+00

WALL THICKNESS = 27.43   SPAR THICKNESS = 20.0
LOADCASE:2
DISPLACEMENT - MAG MIN: 0.00E+00 MAX: 7.41E-01

Figure 4.9

WALL THICKNESS = 27.43   SPAR THICKNESS = 20.0
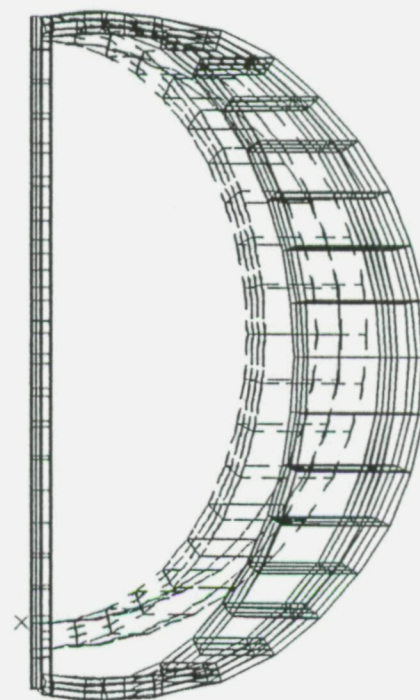
LOADCASE:2
FRAME OF REF:GLOBAL   SHELL SURFACE:TOP
STRESS - MAX PRIN MIN:-2.27E+07 MAX: 3.32E+08

Figure 4.10
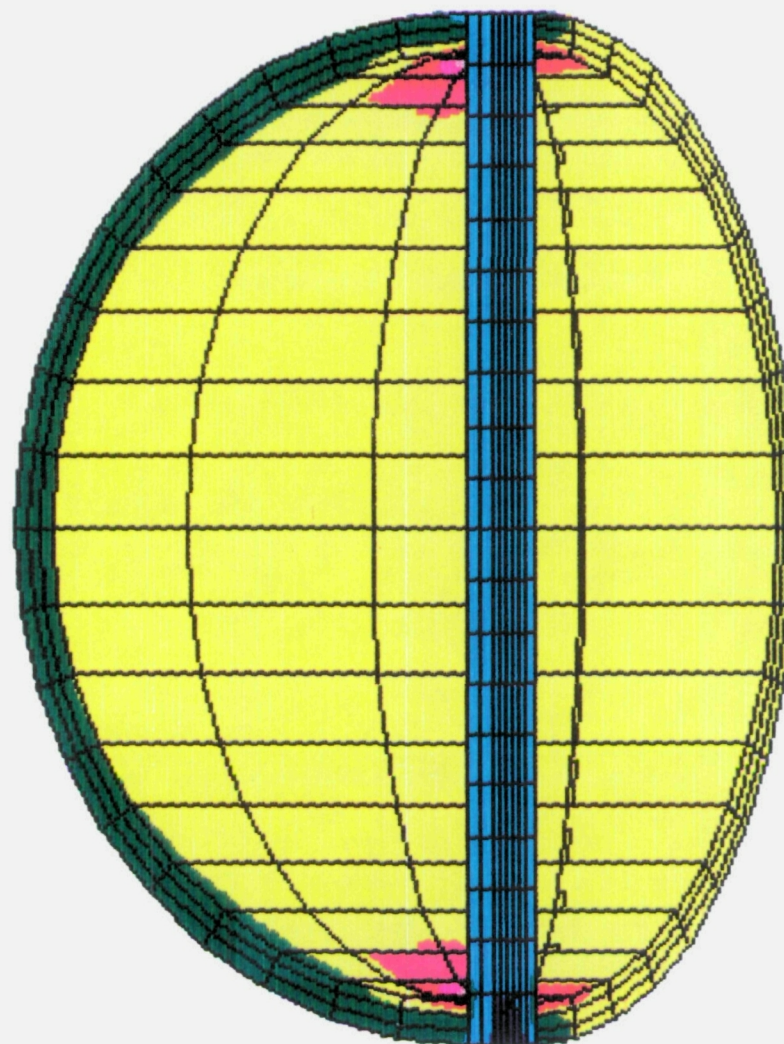
ORIGINAL PAGE
COLOR PHOTOGRAPH

-2.27E+07   2.79E+07   7.86E+07   1.29E+08   1.80E+08   2.30E+08   2.81E+08   3.32E+08

reinforcement that would be needed. The initial assumption used for the thickness of the tank was based on thin pressure vessel theory, using Equation 3.2. The thickness of the tank was 0.0141 meters using a working stress of 165 MPa for the aluminum alloy selected, with a pressure of 0.2 atm (20265 Pa) and a radius of 230 meters. The radius used was about 5% larger than necessary for the initial configuration to give conservative results.

The 3-D model was created by rotating a semi-circle 90 degrees about the y-axis. 240 block elements were created with the cross section composed of a 20 by 3 array of elements. The shell is 3 elements thick and 4 element rows were created by the 90 degree rotation. Symmetry was used to reduce the number of degrees of freedom since it was assumed that the tank would expand uniformly when pressurized. Scaling of the wall thickness was necessary to make the output results visible. The scaling factor was determined by dividing the model thickness by the actual thickness. Point forces were also scaled, but body forces did not require scaling since the scale factor was already introduced by the change in thickness.

The loads on the tank consisted of internal pressure and external forces caused by inertia. A face pressure of 0.2 atm was multiplied by the scale factor and applied to all of the internal elements. The pressure was varied from top to bottom by $p = \rho g h$ to include the effect of acceleration on the propellant pressure. The forces on the tank were modeled using a gravity term instead of applying individual forces because it was easier to include the body forces on the tank elements using this technique. The acceleration used as the gravity term was the rigid body acceleration computed by Newton's second law; dividing the total mass of the E-City by the total thrust. The initial acceleration was 0.014 m/s^2 and the final acceleration at propellant burn out was 0.076 m/s^2. The external forces on the tank consisted of the "weight" of the tank, propellant, and all of the structure above the tank for the bottom tank nodes; and the "weight" of all the structure above the tank for the nodes on the top of the tank. Each tank was analyzed 6 times for one particular load, once with the forces applied at each of the nodal rings around the tank centerline, starting with the centerline (tank longitudinal axis). The nodal rings were spaced at 9 degree intervals. The two loading cases used for the analysis were full tanks at the initial acceleration, and empty tanks at the burnout acceleration. The bottom tank was initially assumed to have the greatest

4.17

stress variations, which was later confirmed after the top tank was analyzed. Figure 4.11 shows the stresses when the load is placed on the centerline of the tank. The maximum stress of 270 MPa is well above the working stress of 165 MPa, and the negative numbers correspond to compression. This is the case of initial acceleration (0.014 m/s^2). The corresponding tank deflection is shown in Figure 4.12, and shows excessive bending. The stresses and deflections reach acceptable values when the tank supports are moved out to at least the fourth node ring, as shown in Figures 4.13 and 4.14 respectively. The radial distance corresponding to the fourth node ring is 104.4 meters with a tank radius of 230 meters. The banding at the tips of the cross section is caused by the extreme distortion of the elements around the central axis of the tank and can be ignored. Also note the banding that occurs across the cross section, which is caused by the transition from tri-axial stress on the interior tank surface to bi-axial stress on the outside surface. Recall that thin pressure vessel theory assumes an average value for the stress across the thickness, and that is nearly the case in Figure 4.13. It was therefore assumed that small areas of higher than working stress were acceptable as long as they were located on the inner surface.

Even though the applied forces at burnout were twice as high as during the initial acceleration, the stress contours were nearly identical, which implies that the tank pressurization was the dominant force. The stress contour at node ring four corresponding to the burnout acceleration of 0.076 m/s^2 is shown in Figure 4.15, and was practically identical to the initial acceleration case. The same was true for node rings five and six.

4.1.2 Torus

A torus was determined to be the most efficient shape for the crew living quarters by the Phase II design team[24]. It is designed as a totally enclosed ecological system, with energy as its only input. Volume requirements were set at 19,000 cubic meters per person to allow extra space for manufacturing, food processing, and other as yet unconsidered needs. It was assumed to be constructed of aluminum alloy and sized so that rotation will provide one-g of artificial gravity. Current dimensions include a major radius of 894.6 meters and a minor radius (tube) of 32.8 m. The pressure of the enclosed atmosphere was set to 1 atm and of the same composition to minimize

4.18

PRESS = .2 ATM   A = .014 M/S/S

LOADCASE:2
FRAME OF REF:GLOBAL
STRESS - MAX PRIN MIN:-2.29E+08 MAX: 2.70E+08
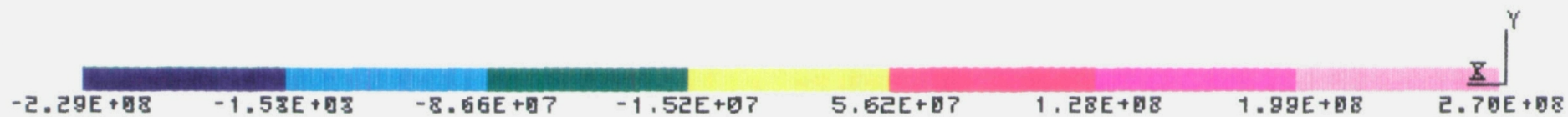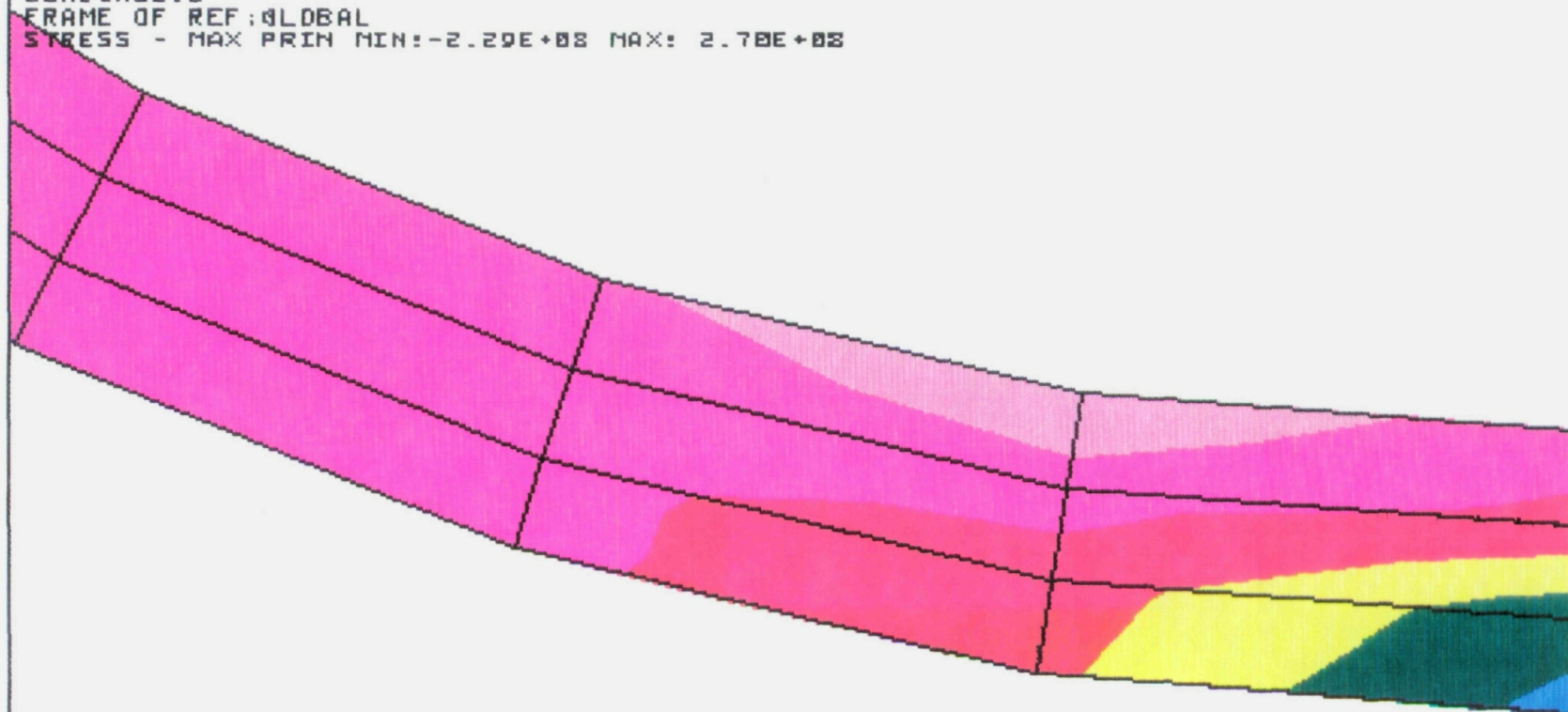
Figure 4.11

| -2.29E+08 | -1.53E+03 | -8.66E+07 | -1.52E+07 | 5.62E+07 | 1.28E+08 | 1.99E+08 | 2.70E+08 |

Figure 4.12

Figure 4.13

PRESS = .2 ATM   A = .014 M??G

LOADCASE:5
DISPLACEMENT - MAG MIN: 9.54E-02 MAX: 5.72E-01
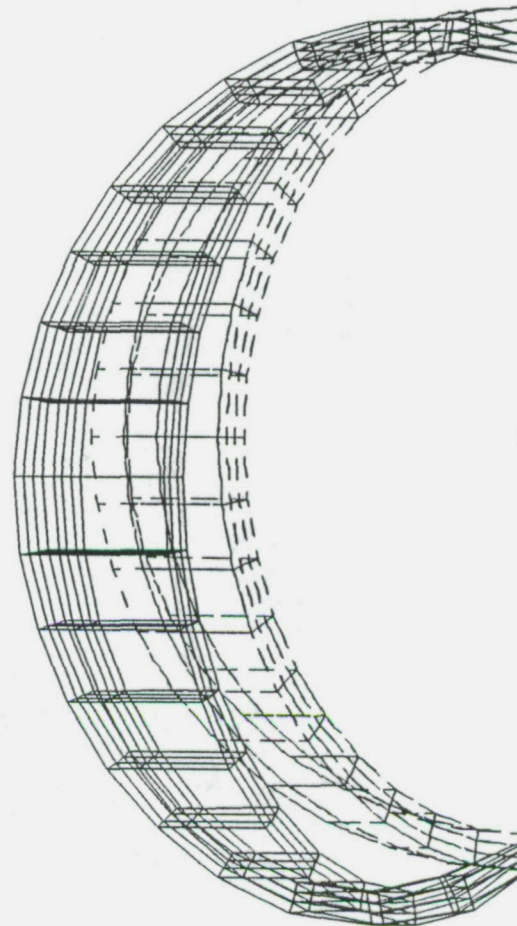
Figure 4.14

PRESS = .2 ATM   A = .076 M/S/S

LOADCASE:5
FRAME OF REF:GLOBAL
STRESS - MAX PRIN MIN: 9.10E+07 MAX: 1.78E+08

Figure 4.15

9.10E+07   1.04E+08   1.16E+08   1.29E+08   1.41E+08   1.54E+08   1.66E+08   1.78E+08
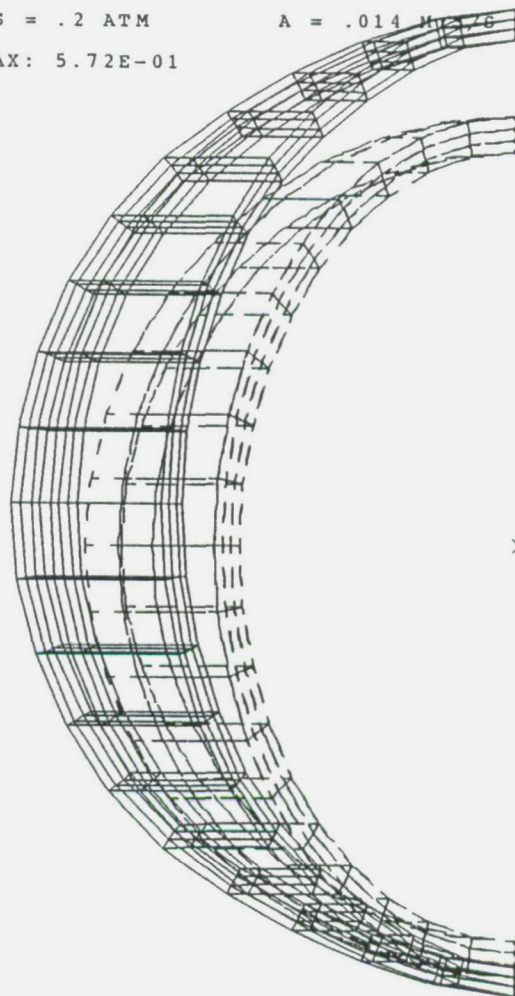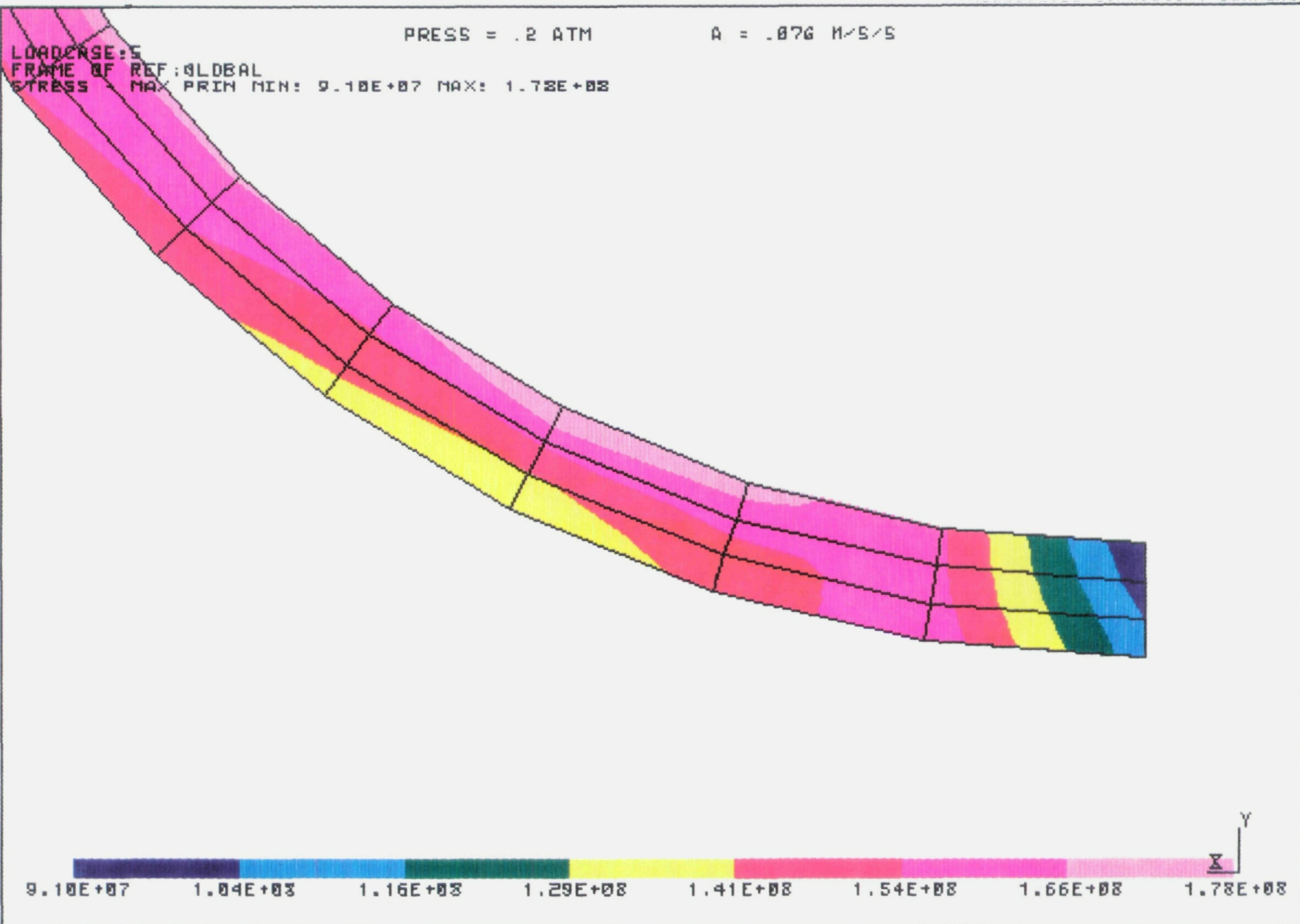
the long term impact on the inhabitants since little is known of such long term effects.

The total mass consists of the mass of the pressure shell and the mass associated with human habitation. NASA SP-428[12] provides a value of 53,000 kg per person for internal furnishings, which includes an agricultural allotment. The determination of the shell structural mass was obtained from NASA SP-413[11] during the Phase II human factors study and are listed below.

$$t_{hoop} = \frac{\frac{P_{atm}}{2} \quad \frac{r}{R} \quad \frac{P_g}{\pi}}{\sigma_w - \rho R} \, R \qquad (4.1)$$

$$M_{ss} = 4 \pi^2 r R \, t_{hoop} \, \rho \qquad (4.2)$$

4.1.3 Cosmic Radiation Shield

The most efficient cosmic radiation shield was determined to be 14 meters of LH_2 by the Phase II design team last year. It was determined that the shield must rotate with the torus. The spinning shield was required because there was no apparent failsafe method to maintain separation between the torus and shield during maneuvering. The torus rotates with a linear velocity of 97.1 m/s and any mechanism to maintain separation induces potentially unacceptable vibrations in the torus and dissipates rotational energy. The difficulty in maintaining separation is exacerbated by the vibrational mode shapes induced by thrusting. A failure of the mechanism separating the rotating torus and stationary shield could have catastrophic consequences, and no viable alternatives were discovered to alleviate this problem, therefore the shield must rotate with the torus.

4.1.4 Spokes

The spokes are the only interface between the crew quarters and the mechanical subsystems of the E-City. They act as cantilever beams and transmit the thrusting forces to the torus and cosmic radiation shield. A simplified analysis was used to calculate the spoke and shaft parameters

with the assumption that the static structure would later be optimized by ASTROS[4]. The spokes were assumed to have a tubular cross section and only the axial stress due to the bending moment was used to determine the thickness and number of spokes. It was assumed that the stress from the axial force on each spoke due to the torus expanding under centrifugal loading was negligible compared to the axial force created by the bending moment due to thrusting. These assumptions allowed the arbitrary selection of the number of spokes. Equation 4.3 is the reduced equation for the total structural mass of the spokes and shows that the number of spokes is no longer a relevant variable.

$$M_{sp} = \frac{L^2 r m a \rho}{\sigma_w} \frac{A}{I} \qquad (4.3)$$

Where L is the length of the spoke, r is the radius, m is the total mass of the torus, a is the rigid body acceleration of the E-City, ρ, σ_w, A, and I are the spoke density, working stress, cross sectional area, and area moment of inertia, respectively.

4.1.4 Shaft

The shaft is the central connecting structure for the tanks, propulsion module, and torus coupling. The primary force on the shaft is due to the axial thrusting load. In this case the bending moment was assumed to be negligible compared to the axial pressure during thrusting and was used to develop the preliminary estimate of the cross sectional area. The radius was determined by the optimal placement of thrusting loads on the tanks. See Section 4.1.1.

4.1.6 Conclusions

The majority of the static structural analysis was performed by the program ECITY.FOR to provide interactive parameter analysis. Using software also reduced the second design iteration to one day as the program reached maturity. A spherical propellant tank provided the optimum configuration with the lowest mass and lowest surface area. Some optimization of the tank wall thickness was provided by calculating the thickness in sections. The minimum mass tank

4.25

configuration was obtained by transmitting the thrusting loads through the tank walls without the addition of special supports or reinforcements. The mass of the spokes and shaft were computed using simplified formulas as a starting place for optimization using the software package ASTROS.

4.2 STRUCTURAL DYNAMICS

4.2.0 Introduction

A structural dynamic analysis is necessary for a complete evaluation of the E-City. The displacements, accelerations, modes, and frequencies of the E-City are needed for the design of the entire ship and its subsystems. Humans inhabiting the torus should not be subjected to intense acceleration and certain structural frequencies that are resonant to the various subsystems must be avoided. In addition, structural mass should be minimized, yet not fail when the E-City is under the influence of external forces. The following material in Section 4.2 includes certain assumptions in the analysis and design aspects addressed. These assumptions were made so that reasonable design activity could still be carried out at the preliminary design stage commensurate with the technical background of the members of the design team.

Although the E-City is a dual-spin spacecraft, hence it is a gyroscopic system, the full-scale gyroscopic structural dynamic analysis and design aspects are bypassed. In the case of attitude disturbances, the anti-symmetric out of plane flexible motions of the torus, spokes, and shaft bending would couple gyroscopically with the attitude dynamics to lead to an area beyond the scope of the design team. Therefore, no attitude disturbances is assumed for the structural dynamics. Pure axial disturbances due to thrusting would characterize non-gyroscopic behavior regardless of the torus spin. Therefore, the axial dynamic behavior which is coupled with out of plane axi-symmetric flexible behavior of the torus and spokes is addressed in this chapter. This motion is still of main significance for crew comfort.

In reality, the spin behavior enters into axial dynamics indirectly. This is because the steady spin of the torus leads to a new static torus radius due to radial expansion due to centripetal forces. Any non-spinning analysis and design should really use the new extended radius for the torus.The static radial expansion of a spinning flexible ring (torus) is given by the following expression[13]

4.26

$$\Delta R = \frac{\Omega^2 \rho R_O^2}{\dfrac{EA}{R_O} - \Omega^2 \rho R_O} = 0.2953 \text{ m} \tag{4.4}$$

with EA = 7.036458E+10 * 12.45246 N, Ω = 0.10472 rad/s, ρ = 2650 kg/m^3, and R_O = 895 m

where Ω, R_O, EA, and ρ are the spin rate, initial undeformed radius, axial circumferential rigidity, and mass per unit length of torus, respectively. Any subsequent analysis can be carried out by considering a new initial radius of R_{ONCW} = R_O + ΔR = 895.2953 m. The static radial expansion ΔR in turn induces constant circumferential tension around the torus which ultimately results in additional stiffening of the out-of-plane axial (bending) deformations of the torus and spokes. Similarly, steady-spin creates additional bending stiffness in the spokes due to centrifugal tension. Inclusion of all such effects in the finite element method software utilized in the following sections would again be beyond the project scope at this point and, therefore, all such centrifugal effects have been neglected; they would add nothing qualitatively different to the structural dynamics considered in this chapter and would not take away from the design experience. Certainly, it is realized that final designs would have to take all these effects into account.

Finally, the axial structural dynamics considered was formulated in a way to lead to a time-invariant system description over the characteristic vibration periods expected. In reality, the system is slowly time-variant due to propellant consumption. However, this variation is deemed insignificant over short term vibrational periods. Thus, wishing to retain a time-invariant system, the axial vibrational dynamics can ultimately be extracted, the propellant consumption effect showing up as an excitation function on the time-invariant structural dynamics. To put it simply and shortly, the rigid body accelerations using the instantaneous mass of the whole ship induce inertial loads on the finite element model grid points to excite the flexible motion. The time-invariant portion of the structural dynamic mass and stiffness matrices of the E-City are obtained as if there is a single-point-constraint (SPC) at the boundary where the engines are thrusting. This modeling approximation is still physically sound engineering and reasonably accurate for the preliminary design stage.

4.27

4.2.1 Finite Element Model

The dynamic analysis was carried out using the Finite Element Method (FEM). To perform the finite element analysis on the E-City, a software package named ASTROS was employed. ASTROS is similar to NASTRAN and also has the capability of optimizing a model for a minimum mass configuration subject to various static or dynamic constraints, a feature used in the final structural design.

The input of ASTROS is divided into two main sections. The first section is the Solution Control Packet and the second is the Bulk Data Packet. Within the Solution Control Packet there are two processes which can be performed, the optimization subpacket and the analysis subpacket. They are done independently so only one is needed, but both may also be done. If both are done, Optimization must be performed before the Analysis. Both subpackets have disciplines and each discipline requires further options to define the execution process. Examples of these disciplines include Statics, Modes, and Frequency. Boundary conditions are also defined in the solution control.

The Bulk Data Packet uses bulk data cards that are similar to the NASTRAN cards. It begins with Begin Bulk and ends with Enddata. Within the Bulk Data Section, the model is defined by setting up a coordinate system, defining grid points, and connecting the grids with various element types. Then the element and material properties must be input, along with any constraints, forces, or moments that are to be applied. Appendix C presents a sample list of input files used for static optimization and dynamic analysis discussed in Sections 4.2.2 - 5.

For a rudimentary analysis and an initial step with becoming familiar with ASTROS, a nondimensional four element rod was used with a concentrated mass representing a rigid torus, and the initial endpoint constrained. The model was changed from using rod elements, which are free only in the axial and torsional directions, to bar elements, which have all six nodal degrees of freedom available.

4.2.2 Design Modeling

An indication of a reliable finite element model is when the first few eigenvalues do not change when the number of elements increase. The number of elements were increased for both

the rod and the bar models. The first few eigenvalues remained about the same at around ten elements for both models. From the lower eigenvalues converging at ten elements and the bar elements having all six degrees of freedom available, it was decided that the ten bar element model was a satisfactory finite element model of the E-City shaft.

An initial evaluation of the torus and spoke sections was done by modeling with four bar elements in a diamond shape for the torus and four bar elements connecting the corners to the center to represent the spokes. The shaft was represented by a concentrated mass at the center. The torus elements were then increased until the lower eigenvalues converged which occurred at about twenty-four elements. The spoke elements for this model were considered to be too long, so each spoke was increased to four elements. This was considered to be an accurate finite element model of the torus and spokes, at least for the preliminary design stage.

The ten bar element model was added to the twenty-four element torus model with the concentrated masses removed. Through interaction with other design members, the original dimensions were defined as follows: torus radius of 895m and a total shaft length of 800m. The model was also changed from solid bars to hollow cylinders by changing the cross sectional areas and area moments of inertia in the material bulk data cards.

When the fuel tank configuration changed from a cylinder to a sphere, the tank, fuel, and engines were represented by a concentrated mass at one end. Another concentrated mass was added to the other end when an additional spherical fuel tank was included in the design. This second concentrated mass only represents a tank and its fuel. With the tanks on each end, it was decided to place the torus in the middle of the shaft to preserve the symmetry of the station which is of consequence for attitude dynamics and control. In addition, considering that the plume radiation problem was eliminated by switching to NLB Engines; it was no longer necessary to place the engines farther out from the torus section, thus the 800m shaft was reduced to 200m with the concentrated masses and engines at the ends.

The spokes connecting the shaft and torus were increased from four to six in number. Therefore the current model has six spokes with four elements, each having a 20m radius; a twenty-four element torus with a 35m tube radius; and a 200m long shaft with ten elements, each having a 100m radius. Thus the total number of degrees of freedom considered in the FEM was

4.29

318. The FEM layout of the configuration is shown in Fig. 4.16 obtained by post processing of ASTROS data by PATRAN graphics package. The list of the ASTROS bulk data is given in Appendix C.

4.2.3 Structural Optimization

For mass minimization, only the spokes and shaft were considered as structural masses. The spherical fuel tanks were treated as nonstructural masses for this purpose since they were already separately optimized (see Section 4.1) based on the propulsion considerations. The torus was also considered as a nonstructural mass and hence was not included among the design variables. This was due to certain numerical conditioning difficulties within the "Optimization Option" of ASTROS. Furthermore, because there would be other nonstructural design requirements on the torus since it is the most vital component of the E-City, a structural optimization of the torus was considered premature at this stage. It should be noted that the optimization model still does include the torus and the fuel tanks along with the shaft and spokes, it is just that there are no design variables associated with them. The design variables are associated with the shaft and spokes and are the respective structural cross sectional areas.

The objective function to be minimized was the structural mass (shaft and spokes) subject to Von Mises yield criterion as stress constraint on the elements under static thrust loading along the shaft. Tables 4.2-3 below show the results of optimization for a typical bar element.

Table 4.2: Optimization Results of Bar Elements

Section	Length	R_{min}	R_{max} Optimized	Cross Sectional Area Initial	Cross Sectional Area Optimized	I_x	I_y
	(m)	(m)	(m)	(m^2)	(m^2)	(m^4)	(m^4)
Torus	233.60	33.00000	33.060000	12.45246	------	13585.414	6792.707
Shaft	20.00	100.00000	100.001808	6.28625	1.13964	11347.424	5673.712
Spokes	223.75	20.00000	20.001000	12.59779	0.12560	50.241	25.120

The original wall thickness of the shaft was 10 mm and of the spokes was 100 mm. As indicated

4.30

in the table, these were reduced to 1.808 mm and 1.00 mm, respectively.

Table 4.3: Optimization Results of Masses

	Mass per Spoke Element (kg)	Mass per Shaft Element (kg)	Structural Mass (10^6 kg) Designed	Total
Initial	7469702.11	333171.25	182.604	367.603
Final	74472.95	60400.92	2.391	187.397

The mass reduction was 49 % as seen in the table above.

4.2.4 Structural Dynamic Analysis

With the structural optimization of the E-CITY spokes and shaft complete, the new cross sectional areas and area moment of inertias were then entered into a new model. A modal analysis was then performed on this model utilizing the ASTROS subpacket ANALYZE, under the MODES discipline. The ASTROS software employed the Modified Givens Method to extract the frequencies and modes of vibration.

To reiterate, the model analyzed had a ten element shaft, 6 spokes with 4 elements each, and a torus of 24 elements (see Figure 4.16). Concentrated masses were placed at each end of the shaft representing the fuel tanks full of fuel and 14 NLB engines. A single point constraint (SPC) was placed on one end of the model (grid point 100) to extract only the flexible modes within the assumptions used to approximate a realistic model of the E-City. Fifty-three grid points were needed to define this model, thus the total number of D.O.F. reached 318 (3 rotations and 3 translations per grid point). The SPC eliminated 6 D.O.F. from grid point 100, so the total number of D.O.F. analyzed was 312.

The output from ASTROS contained all extracted eigenvalues and their respective eigenvectors. These eigenvectors represent the natural modes of vibration of the system. and give the general shape the motions would take. The square root of the eigenvalues gives the frequency at which that particular mode would vibrate, in radians per second.

It is possible, through the help of the graphics software package PATRAN, to illustrate these modes visually. This was done for a few of the modes, and the results can be seen in

4.31

Figures 4.17-4.19. Table 4.4 gives some data on the illustrated modes of the non-optimized initial configuration. Figures (4.17-4.19) show symmetric and antisymmetric mode shapes of the initial design.

Table 4.4: Some Modes of the Initial Configuration

Mode	Description	Eig. Value (rad/sec)2	Frequency (rad/sec)	(Hz)	Period (sec)
16	Rigid Torus	0.0003007	0.01734	0.00276	362.32
17	Ruffle Torus	0.0003728	0.01931	0.00307	325.41
22	Shaft Axial	0.0203428	0.14263	0.02270	44.05

4.2.5 Axial Dynamics

With a rotating torus, a slight increase or decrease of acceleration in the same direction as the gravitation will be tolerable. However motion perpendicular to the plane of rotation, i.e. the out of plane elastic motion of the torus which is parallel to the axial direction of the shaft will be perceived as a lateral "swaying" motion to the crew which can be intolerable for crew comfort and on board activities. It is therefore required to analyze and if necessary redesign the displacements and accelerations due to this motion.

The axial shaft vibration and symmetric out of plane torus elastic displacements are coupled and needed for this study. The axial flexible dynamics is typically excited by the axial thrust forces and therefore is also governed by the propulsion system parameters. We should also note that there is no attitude disturbance so that the coupling between the out of plane torus motion and the gyroscopic wobbling dynamics is avoided. Axial shaft dynamics is coupled with the symmetric out-of-plane torus modes. Furthermore, the steady spin rate of the torus has no direct effect on the axial dynamics considered.

For multidegree of freedom systems, the general linear differential equation of motion is:

$$[M]\,\ddot{q}(t) + [K]\,q(t) = Q(t) \qquad\qquad (4.5)$$

In this case, [M] is the mass matrix, [K] is the stiffness matrix, q(t) is the displacements vector,

4.32

and $Q(t) = [D] F$ where $[D]$ is the distribution matrix and F is the externally applied Forces such as thrusting, gravity, docking, etc. We transform equation 4.5 to modal space by using the mass matrix $[M]$ and the modal matrix $[E]$, such that $[E]^T[M][E] = [I]$ where $[I]$ is the identity matrix, and $[E]^T[K][E] = [\lambda]$ where $[\lambda]$ are the eigenvalues on the diagonal. Recall that $\lambda = \omega^2 = (2\pi f)^2$ where ω is the frequency in rad/s and f is the frequency in Hz.

Introducing the modal coordinate transformation $q=[E] \eta$, and multiplying the general linear differential equation of motion by $[E]^T$ on the left hand side gives:

$$[I] \ddot{\eta}(t) + [\lambda] \eta(t) = [E]^T[D] F \tag{4.6}$$

$F = m_s a_0$, where m_s is the elemental mass vector, i.e. the mass at each grid point connecting the elements, and a_0 is the rigid body acceleration at t . . With substitution of a_0 (See Appendix C) into the equation above gives:

$$\ddot{\eta}(t) + [\lambda] \eta(t) = [E]^T[D] m_s \sqrt{\frac{2 m_{pto} m_w \alpha}{t_{pr} [m_{oto} - m_{pto} t / t_{pr}]^2}} \tag{4.7}$$

where m_{oto} and m_{pto} are the total initial and propellant mass at the beginning of burn time (take-off) and m_w, α, and t_{pr} are the propulsion system mass, propulsion specific power and the propulsion time with $0 << t << t_{pr}$.

Due to the rotation of the torus section creating an outward acceleration of approximately one g, an additional acceleration in the same direction would hardly be noticed. However, a disturbance in a different direction, such as the direction of the spin axis, i.e. the axial direction would be more readily noticed. Therefore, only the symmetric torus modes with significant axial displacements were taken into account in the modal axial dynamics. Three such modes were identified (see Table 4.5) from the results of the FEM analysis of the statically optimized configuration of the E-City in Section 4.2.3.

4.33

Table 4.5: Mode Data of the Optimization Configuration

Mode	Description	Eig. Value (rad/sec)^2	Frequency (rad/sec)	(Hz)	Period (sec)
16	Rigid Torus	0.000294563	0.030760	0.00490	204.29
19	Rigid Torus	0.000945977	0.101903	0.05081	19.68
22	Ruffle Torus	0.101903000	0.155079	0.06267	15.96

Mode 16 in Table 4.5 represents axisymmetric bending of all spokes while the torus ring attached at their ends is simply displaced as a rigid body. The next two modes are also symmetric and they have out of plane axisymmetric bending of the torus sections between the spokes all around the perimeter. However, there are orders of magnitudes of difference between the modal displacements of mode 16 and modes 19, and 22. Therefore it is expected that mode 16 will dominate the axial dynamic response. It is also important to note that the periods of the dominant structural modes are much smaller than the rate of propellant consumption such that over several vibration cycles the system mass can be regarded as time-invariant.

Thus, the next step was to simulate the 3-mode axial dynamics due to thrust loading which required a transformation to the modal state-space with 6 states. The chosen eigenvectors were put into the modal matrix form [E]. The state space equation:

$$\dot{x} = A x + B F \tag{4.8}$$

was then used to find the modal displacements and velocities $\eta(t)$ and $\dot{\eta}(t)$, where

$$X = \begin{bmatrix} \eta \\ \dot{\eta} \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \qquad \text{and } B = [0 \; b_1 \; 0 \; b_2 \; 0 \; b_3]^T$$

where $b_i = i^{th}$ row of $[E]^T[D]$ and $F = m_s \, a_0$, m_s being the elemental mass vector.

Upon simulating equation 4.8 with a given thrust vector the equations used to find the displacements, velocities, and accelerations at desired torus grid locations were:

4.34

$$q(t) = [E] \eta(t) \tag{4.9}$$
$$\dot{q}(t) = [E] \dot{\eta}(t) \tag{4.10}$$
$$\ddot{q}(t) = [E] \ddot{\eta}(t) \tag{4.11}$$

which constitute the output equations. The parameter values used for simulation were,

$m_{oto} = 9.551e+9$ kg

$m_{pto} = 7.711e+9$ kg

$m_w = 20$ engines * 138886.81 kg/engine

$\alpha = 1279038.73$

$t_{pr} = 20$ days.

The displacement and acceleration profiles of the flexible axial dynamics are shown in Figs. 4.20-4.22 for the torus center (grid 25), which is also the E-City center of mass, and grid points 1 and 7 (see Figure 4.16) on the torus, grid point 1 being an attachment point of the torus to a spoke and grid 7 is a non-attached point on the torus. The positive displacements shown in Fig. 4.20 denote compressive action on the shaft with thrust loading applied in the negative axial direction in the simulations. Accordingly, the positive displacements of the torus grid points are in the direction of the induced inertial loading due to thrusting as expected. Simulations show that the dominant axial flexible mode is mode 16 with a period of 204 secs. Grids 1 and 7 have the same response profiles as they should since the dominant flexible axial mode has rigid body torus displacement while the spokes all bend in unison. Figures 4.20-4.22 show the responses at the beginning of the propulsion period for 400 secs. when the propellant tanks are full and therefore when the inertial loads on the E-City are lowest. On the other hand towards the end of powered flight when the propellant tanks are almost empty, higher levels of axial responds are to be expected since the rigid body inertial acceleration term a_0 is the excitation function on the flexible vibrations. At the end of the propulsion time a_0 goes as high as up to six times that of a_0 at the beginning of the powered flight and the typical responses of grids 25, 1, and 7 will be as high as six times as those shown in Figures 4.20-4.22. It is noted from the simulation results that although the vibrational acceleration levels would be tolerable by the crew they may still be intolerable for the precision required for various scientific tasks, etc. More importantly, it is noted

that the maximum vibrational displacement level around the torus is eight meters at the beginning of thrusting and would increase to about 48 meters towards the end of thrusting.

4.2.6 Conclusions

Through the program ASTROS, the E-City model was optimized to reduce structural mass, and then dynamically analyzed in order to gain some knowledge of its natural motion. The axial dynamic simulations show the need for some type of control of the flexible torus motions. A control design, either active or passive, must be implemented in order to reduce excessive torus displacements. By passive control, optimization of the model with frequency and displacement constraints is meant. Active control involves control design theory. A combination of both active and passive control could also be utilized.

4.36
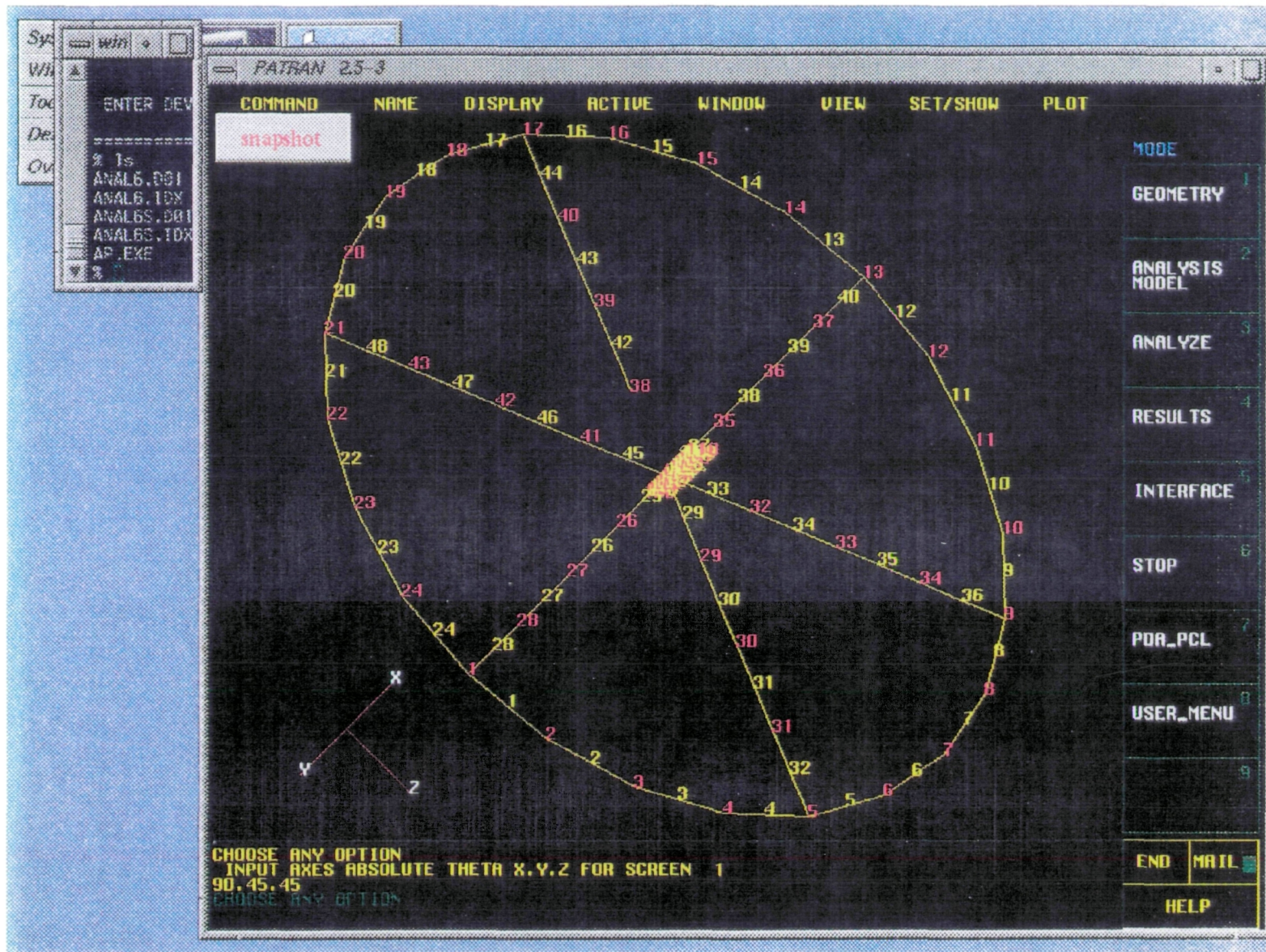
Figure 4.16 - Finite Element Model from PATRAN
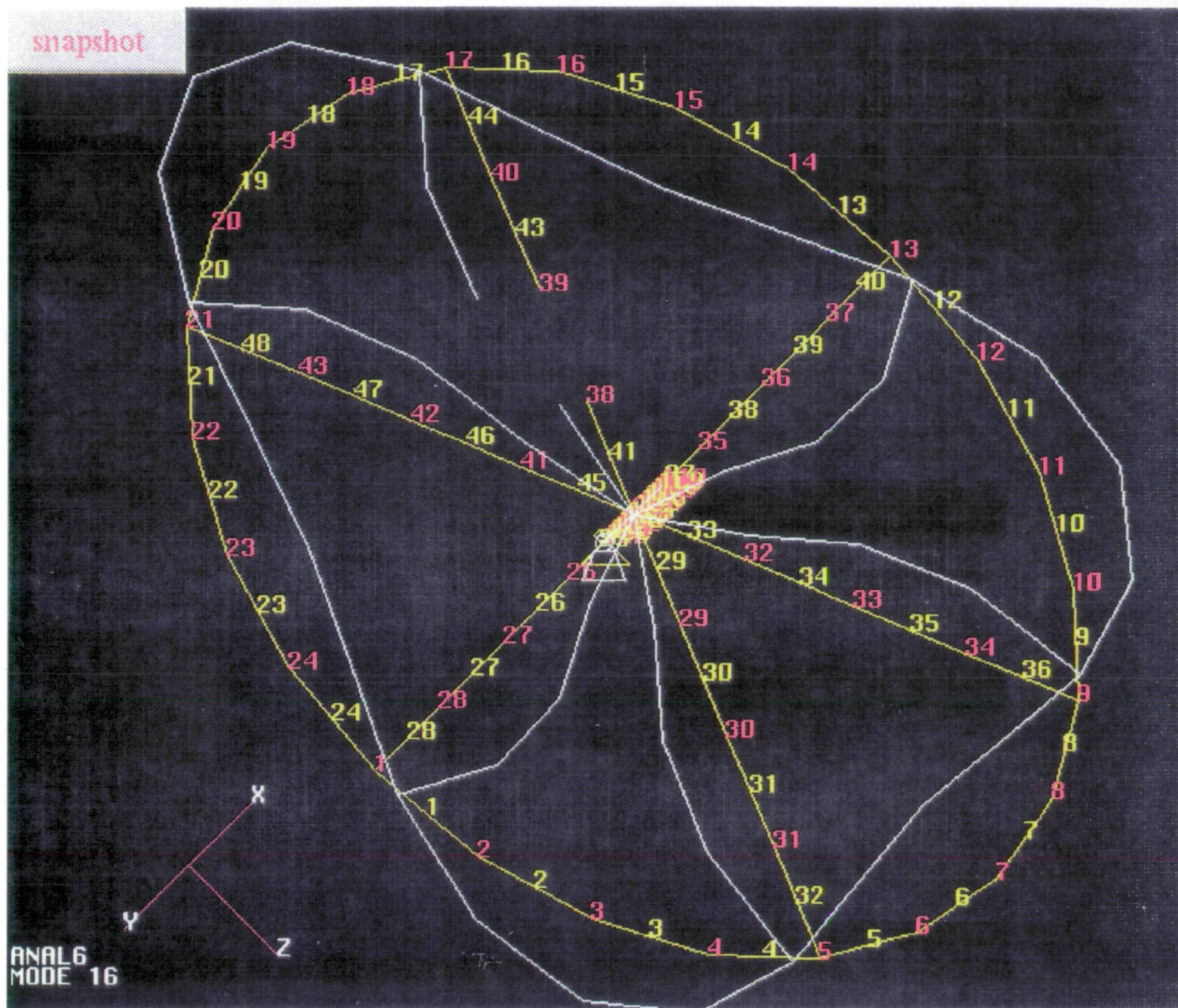
Figure 4.17 - Mode 16, Symmetric: Rigid Torus
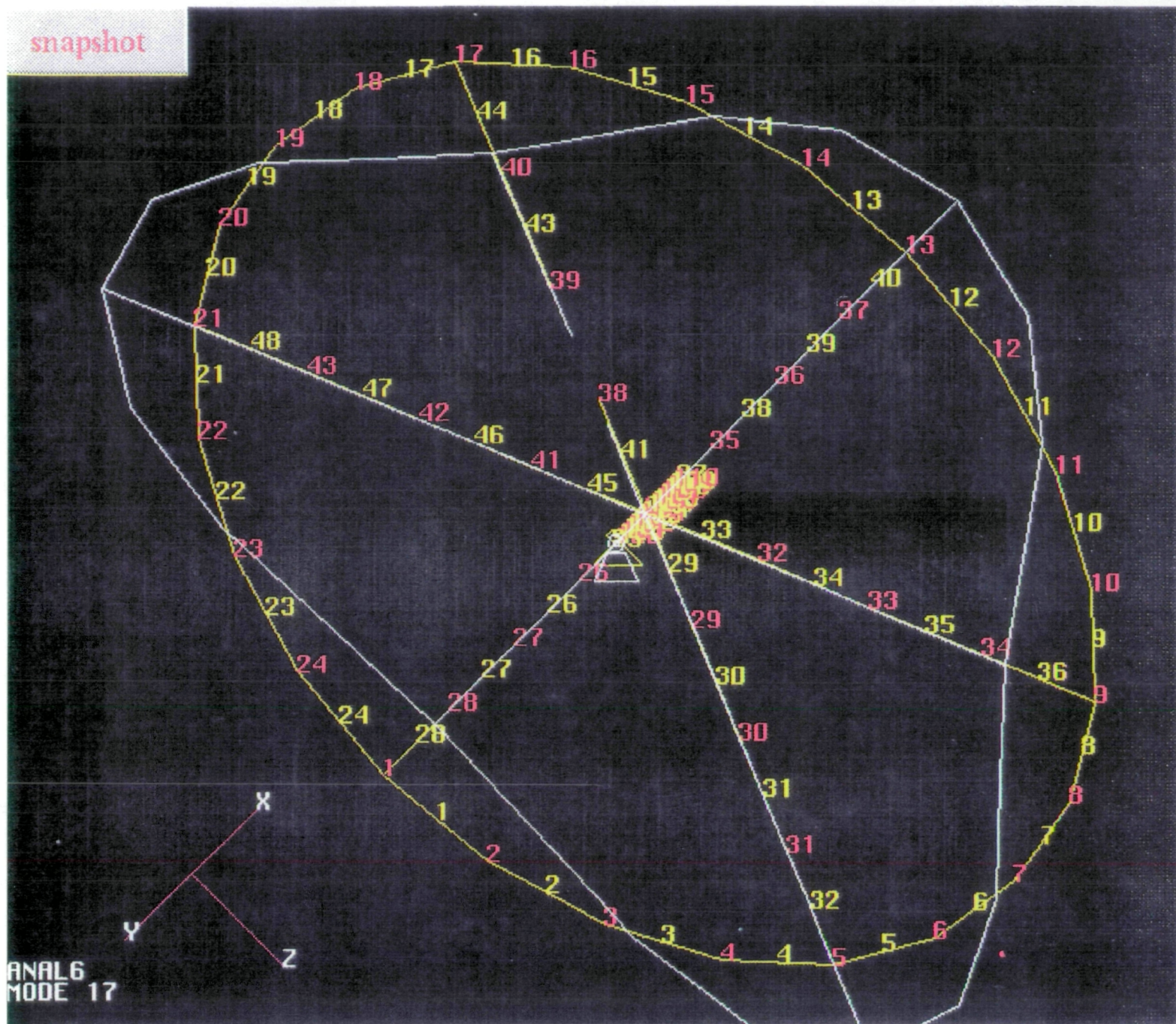
Figure 4.18 - Mode 17, Anti-Symmetric: Ruffle Torus

4.39
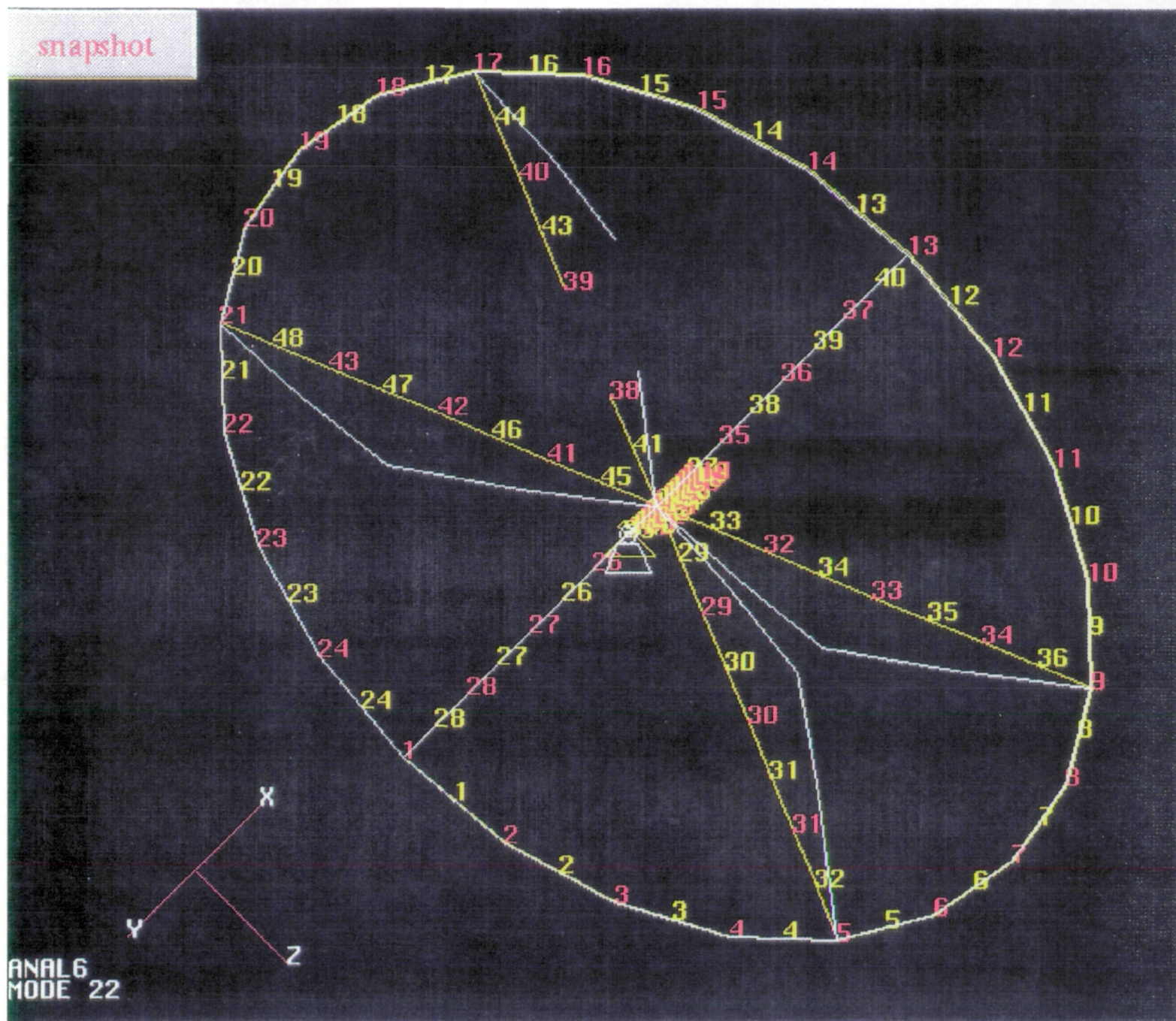
Figure 4.19 - Mode 22, Anti-Symmetric: Shaft Axial
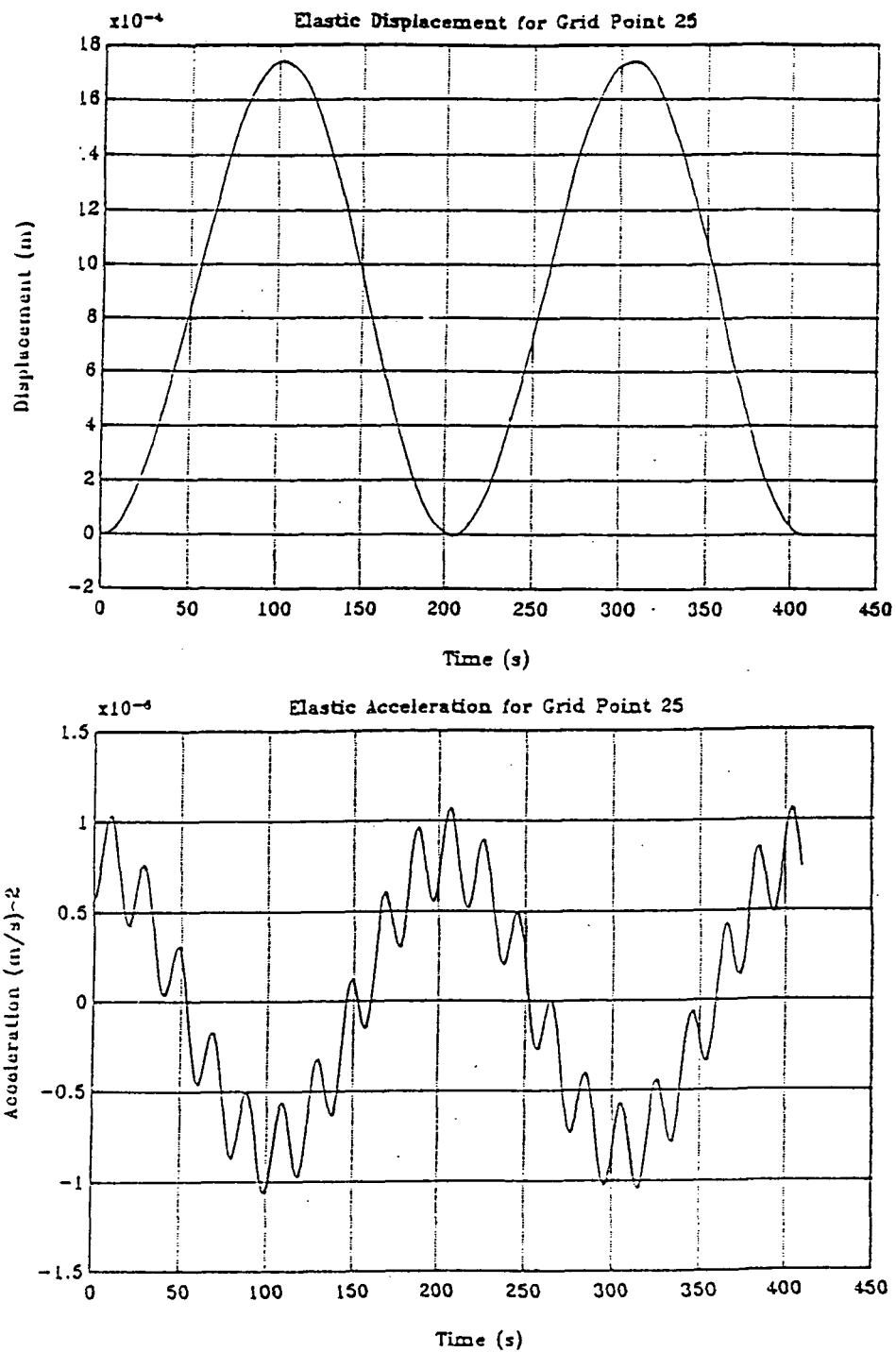
Figure 4.20 - Elastic Displacement and Acceleration for Grid Point 25
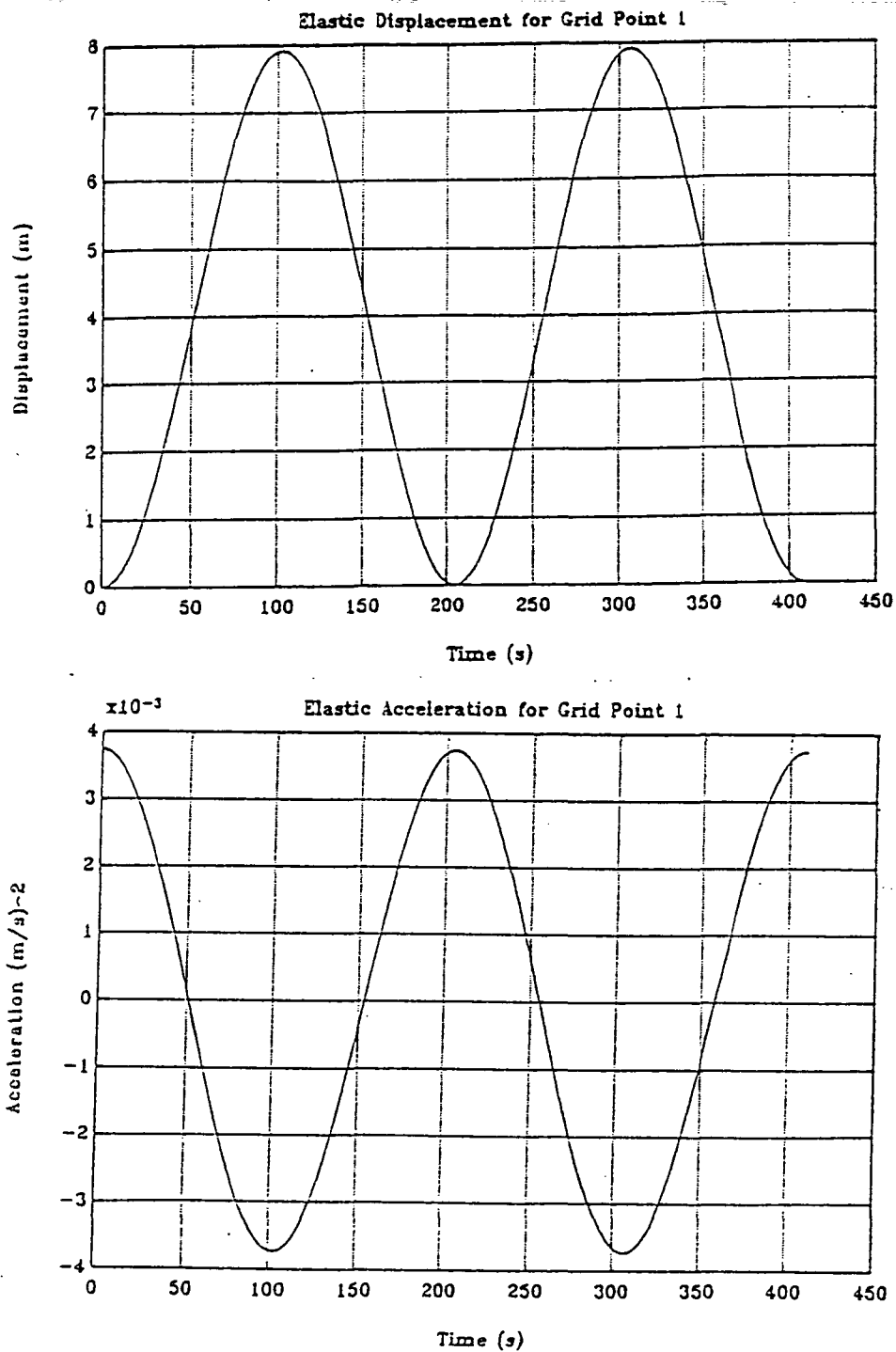
Figure 4.21 - Elastic Displacement and Acceleration for Grid Point 1

4.42
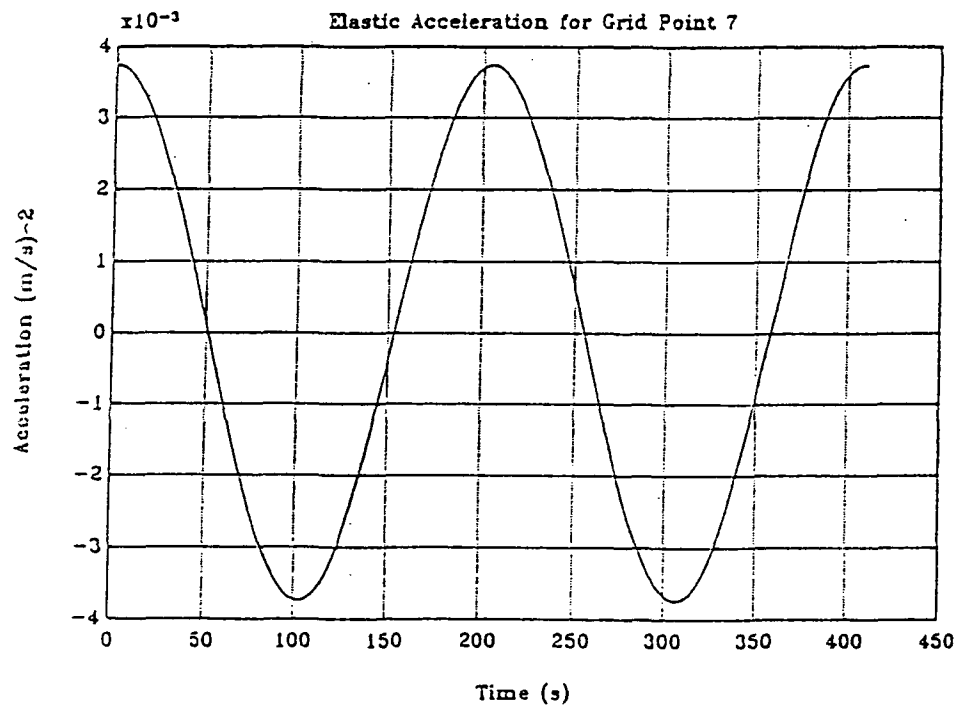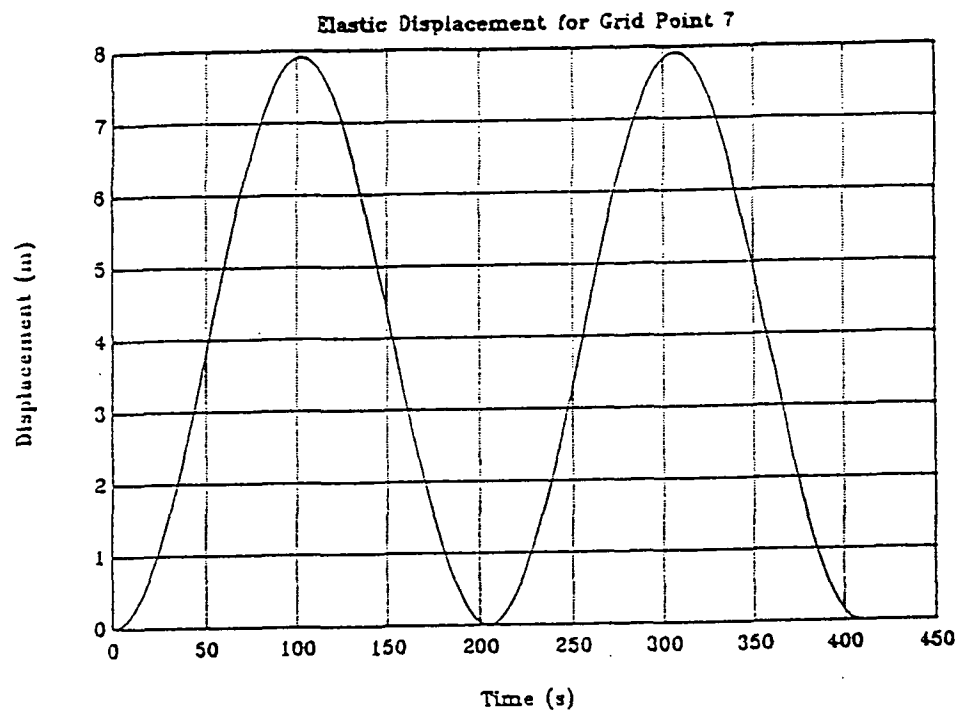
Figure 4.22 - Elastic Displacement and Acceleration for Grid Point 7

CHAPTER 5

ATTITUDE CONTROL SYSTEM DESIGN

5.0 INTRODUCTION

An analysis of the attitude control requirements for the E-City is a necessary step in the optimization and determination of the final station configuration. The system will be gyroscopic in nature due to the spinning of the torus which will produce 1-g artificial gravity for the station's crew. However, as stated in Chapter 3, this year's analysis has deemed that the rest of the ship be de-spun; thus a complete mass moment of inertia analysis (MMI) precedes the attitude control design. Restabilizing the station following a disturbance in a timely and efficient manner is critical with respect to the crew. The propellant required to provide control for the E-City during a given mission is quite significant; therefore, the quality of the control system design is crucial to the viability of the station. The basis for the attitude control design was developed last year during the Phase II study and continued this year in Phase III. The goals of this year's efforts were to optimize the number of control thruster clusters, minimize the number of thrusters required per cluster, and to minimize the control propellant mass while at the same time designing a system that will control the gyroscopic wobble of the station following a disturbance in a manner that will be acceptable to the crew. Chapter 5 will focus on the methods and results of the design and optimization of the attitude control system for a rigid body model of the station.

5.1 MASS MOMENTS OF INERTIA

5.1.1 Introduction

Subject to altering disturbances which must be controlled, a Mass Moment of Inertia (MMI) analysis of the E-City is required to further analyze the vehicle's stability and control characteristics. The MMI study is essential to Project WISH's attitude control system optimization. This section further develops the work done during Phase I and II of Project WISH. It will focus on the effects of the E-City's geometric evolution on its MMIs.

5.1.2 MMI Background

Modeling E-City as a rigid body consisting of lumped masses, Phase I established the basis

5.1

for a MMI study. The work in the previous two years focused on obtaining an order of magnitude estimation of the MMIs and the whole ship was modeled as spinning.

Table 1.1 includes the following variables pertinent to this section. The global E-City structure will be referred to by "ship".

* MMI about ship's x or y axis: $I_x = I_y$
* MMI about ship's z or spin axis: I_z
* Ship's Slenderness Ratio: $r = I_x / I_z$

The x and y-axes are symmetric. Additional mass and geometric variables can be found in Table 1.1.

It is important to note that the lumped modeling of E-City generally disregards the geometrical configuration of each of the ship's components. In fact, the effect of the torus spokes was not initially considered.

On the other hand, Phase III emphasizes a more rigorous approach to the ship's configuration which includes a more accurate and complex MMI analysis. This year's work was devoted to a dual-spin configuration which requires that only the torus and spokes rotate about the spin axis. Note that the definition of r must now change to incorporate the effects of dual-spin. The two new r values, r_1 and r_2 are discussed later.

5.1.3 Goals of MMI Analysis

Developing a reliable and user-friendly FORTRAN program to calculate the MMIs of an evolving complex E-City structure is the motivation behind Phase III of the dual-spin MMI analysis. Accurately calculating the MMI ratios required by an ensuing attitude control study and defining the ship's structure are the goals. A complete program listing (Program Inertsp3) which includes a variable definition list is provided in Appendix D.

5.1.4 MMI Procedure

The following describes the development and progression of the MMI scheme:

5.2

1. Obtain and derive analytical MMI formulation

2. Develop versatile FORTRAN scheme

3. Refine scheme and define vehicle configuration

4. Make FORTRAN program User-Friendly

The input and output to the program are shared by the structural dynamics, attitude control, and propulsion Project WISH team members. The MMI study requires masses and dimensions and calculates the MMIs and MMI ratios required for the stability analysis.

5.1.5 MMI Analysis

E-City evolved through three basic configurations during Phase III. Figures 5.1 to 5.3 are schematic representations illustrating the geometry utilized by the FORTRAN program which is provided in Appendix D. The program was slightly altered with the changes in configuration and only the final program is included. The development of the MMI methodology progressed through the following stages.

The goal of the program is to calculate two MMI ratios which are required for the attitude control analysis. They are defined as follows:

$$r_1 = I_{zspn} / I_x \qquad (5.1)$$

$$r_2 = I_{xspn} / I_x \qquad (5.2)$$

where, I_{zspn} is the sum of the torus and spokes MMI component about the spin axis, I_{xspn} is similar but about the x or y axis, and I_x is the total moment of inertia about the x or y axis.

First, one needs to establish the Center of Mass (CM) for the entire structure. Because of the axial symmetry about the z-axis, the CM will lie on the z-axis. Its location, denoted by Z_{cm}, is defined as the displacement of the mass center from the torus center.

$$Z_{cm} = \Sigma \ (Z_i * m_i)/\Sigma \ (m_i) \qquad i = \# \text{ of components} \qquad (5.3)$$

where, Z_i is the z location of each component with respect to the torus center and m_i denotes the component mass.

Next, the MMIs are defined using formulae which model the ship's components as both

5.3

Configuration #1 (cylindrical tank)
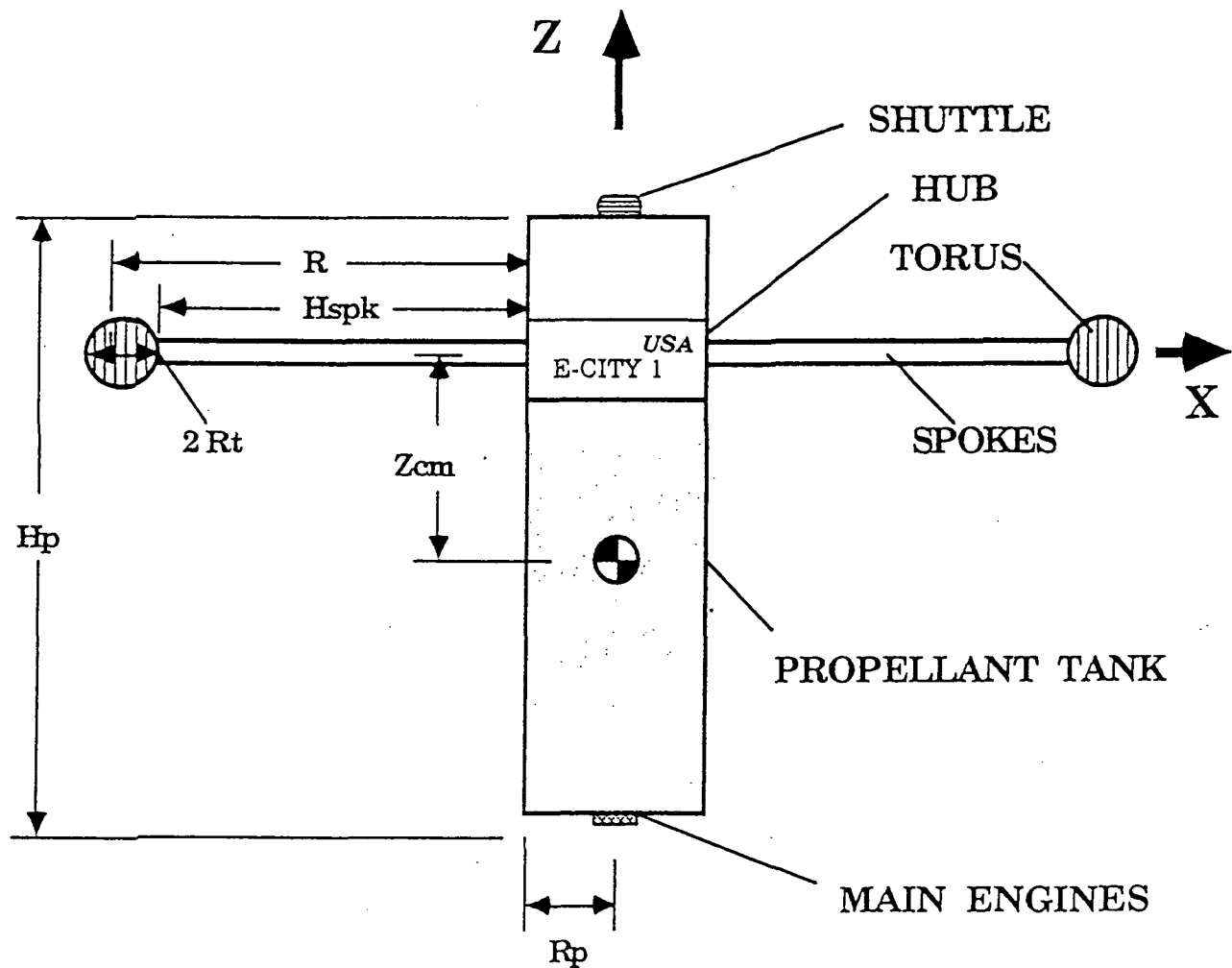
Figure 5.1

Configuration #2 (1 spherical tank)
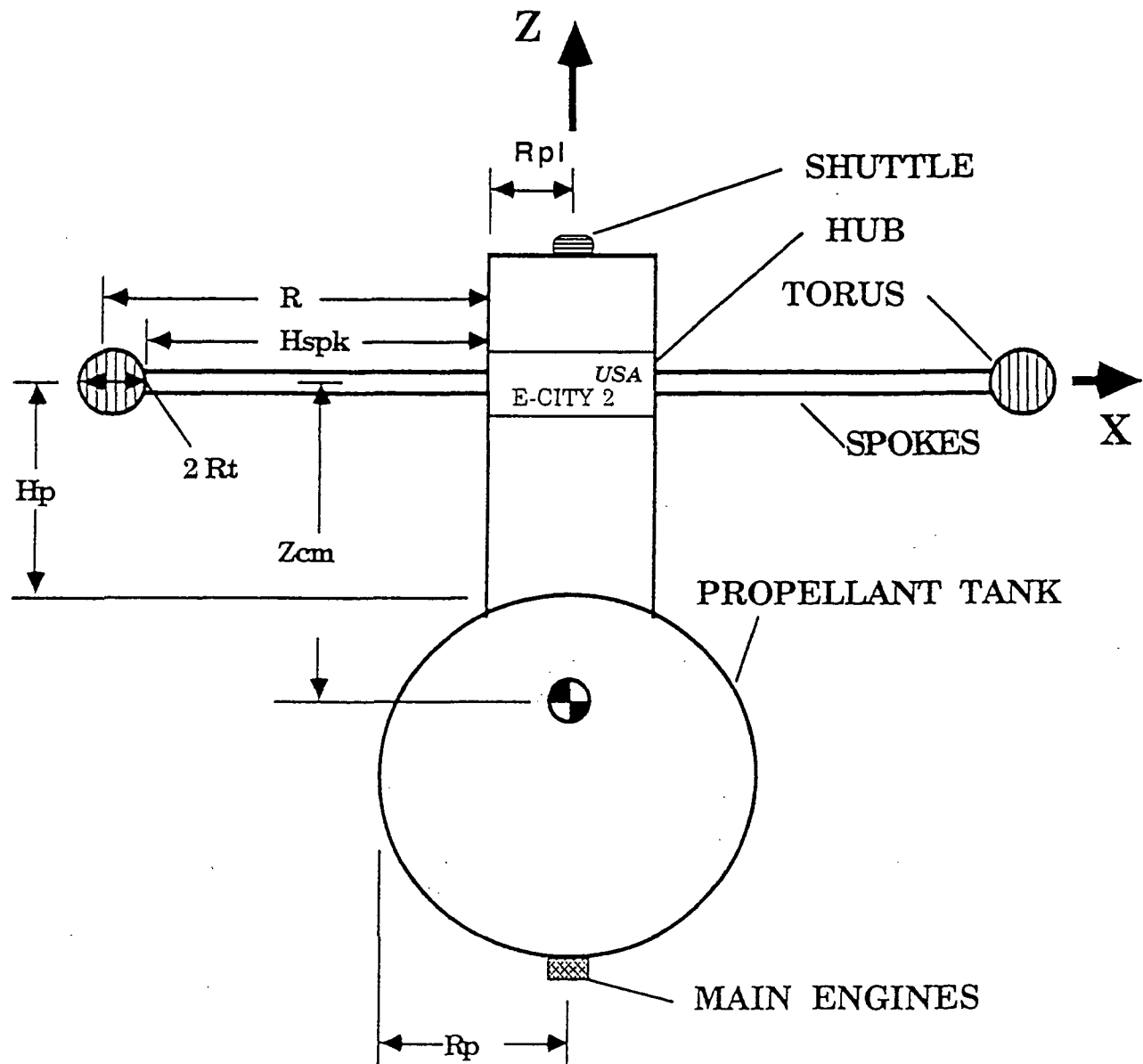
Figure 5.2

Configuration #3 (2 spherical tanks)

Figure 5.3

5.6
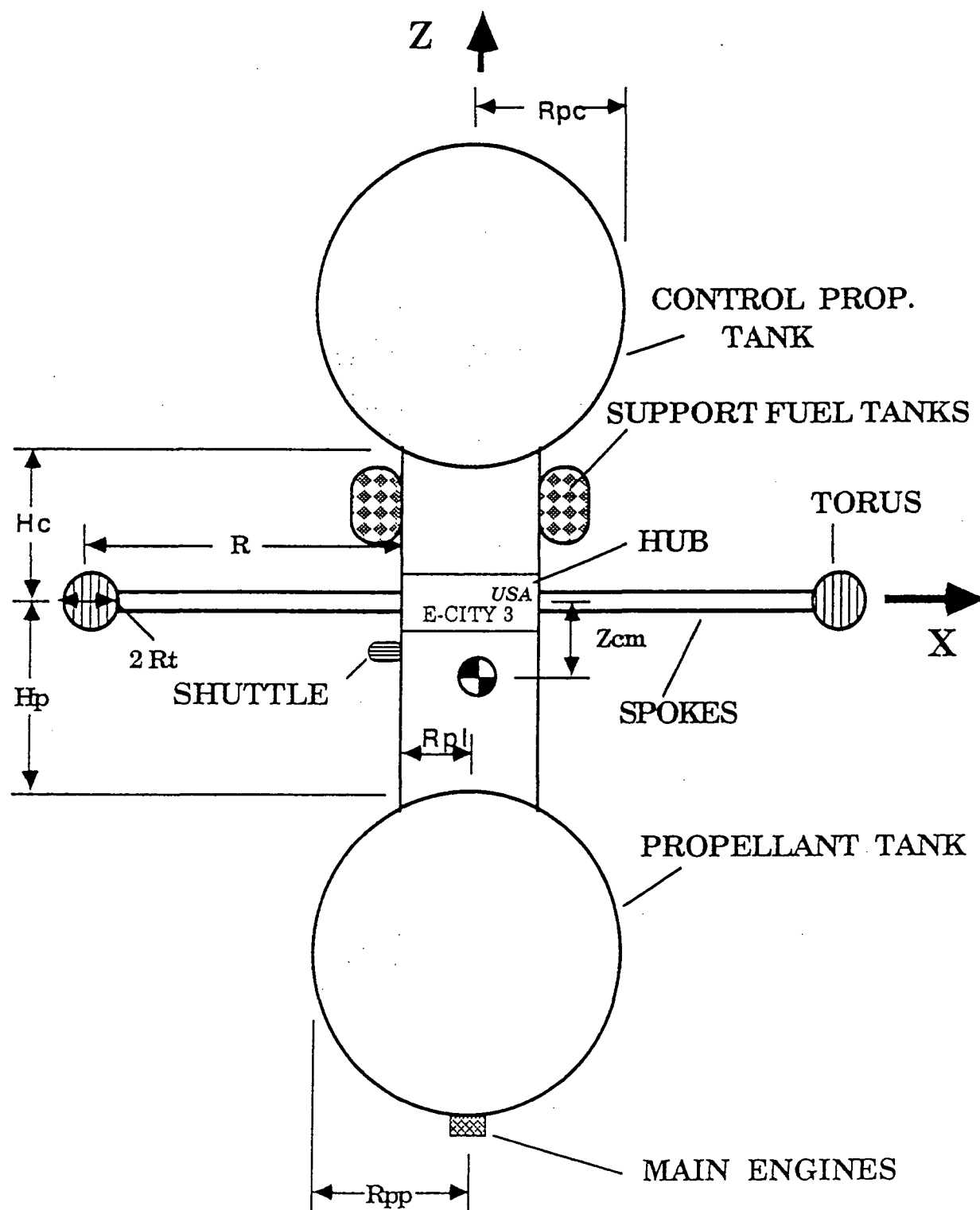
solid and hollow thin shell cylinders, disks, and spheres. For example, the program models the propellant as a solid uniform sphere and the tank structure as a thin shell for the spherical propellant tank. I_{xoi} and I_{zoi} are used to denote the component MMIs, which are defined in Program Inertsp3.

Since this is a configuration consisting of many components, the Parallel Axis Theorem must be incorporated to transfer each component's MMI with respect to the CM location. This component is then added to I_{xoi} or I_{zoi}. The total contribution is defined as:

$$I_x = \sum (I_{xo} + m_i * d_i^2) \qquad (5.4)$$

$$I_z = \sum (I_{zo} + m_i * d_i^2) \qquad (5.5)$$

where, d_i^2 is the sum of the squares of the x and z distances for each component in relation to the CM for the ship, ie. $d_i^2 = x_i^2 + z_i^2$.[14]

Finally, the two ratios r_1 and r_2 can be obtained from eqs. 5.1 and 5.2 and the attitude control analysis is initiated. Note that because of the MMI's dependence on geometry and mass, the ship's configuration is precisely defined within Program Inertsp3.

Program Inertsp3 was successfully developed and refined using the preceding methodology. The program utilizes keyboard-user interaction which allows for easy modification of the twenty five variables required as input.

5.1.6 Discussion of MMI Analysis

Several plots (Figures 5.4 to 5.7) are provided which illustrate the results of Program Inertsp3. All plots use the pole length Hp as a common variable. Note that Hp is the propellant tank length for configuration (cnfg) 1. The characteristics of each plot are summarized as follows.

Ix or Iy

* All of magnitude $10^{15 \to 16}$ kgm^2 (large dimensions and masses)

* All increase (the pole length increases)

5.7

* Cnfg 3 is much higher (two tanks are used)

Iz

* All of magnitude 10^{15} kgm^2 (massive and large)
* Cnfg 1 exhibits hyperbolic behavior (due to specific geometry)
* Cnfg 2 and 3 are constant (no changes in geometry about spin axis)

r1

* All generally decrease (I_x increases faster than I_{zspn})
* Cnfg 1 has maximum (due to specific geometry)
* Cnfg 3 is almost steady below 0.2 (overall symmetry)

r2

* Cnfg 1 has maximum (geometry)
* Cnfg 2 increases (I_{xspn} increases faster than I_x)
* Cnfg 3 is below 0.1 and almost steady (symmetry)

The results show that the final twin sphere configuration exhibits a more stable trend in r_1 and r_2 as the main connecting pole is lengthened. Configuration 3 and the Hp=100 case will be chosen for the attitude control design to complete a thorough analysis.

5.8

Ix=Iy -vs- Pole Length
(Tankage Despun)

Figure 5.4

Iz -vs- Pole Length
(Tankage Despun)

Figure 5.5

5.9

R1=Izspn/Ix -vs- Pole Length
(Tankage Despun)

Figure 5.6

R2=Ixspn/Ix -vs- Pole Length
(Tankage Despun)
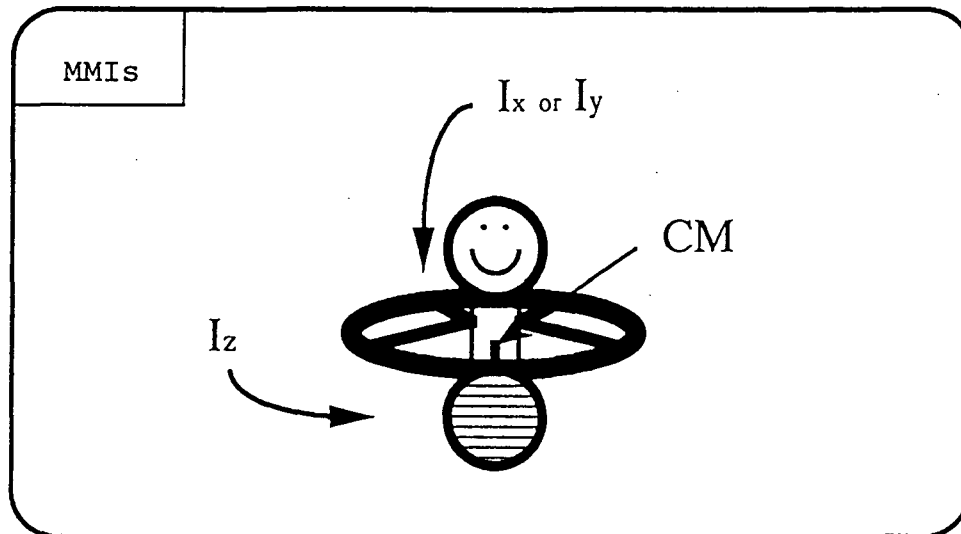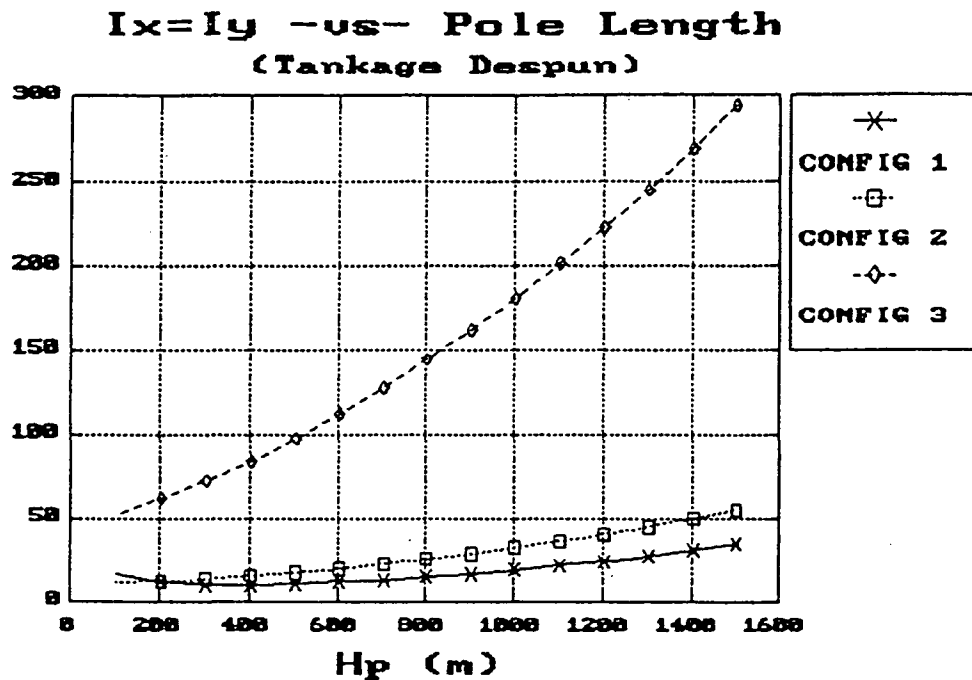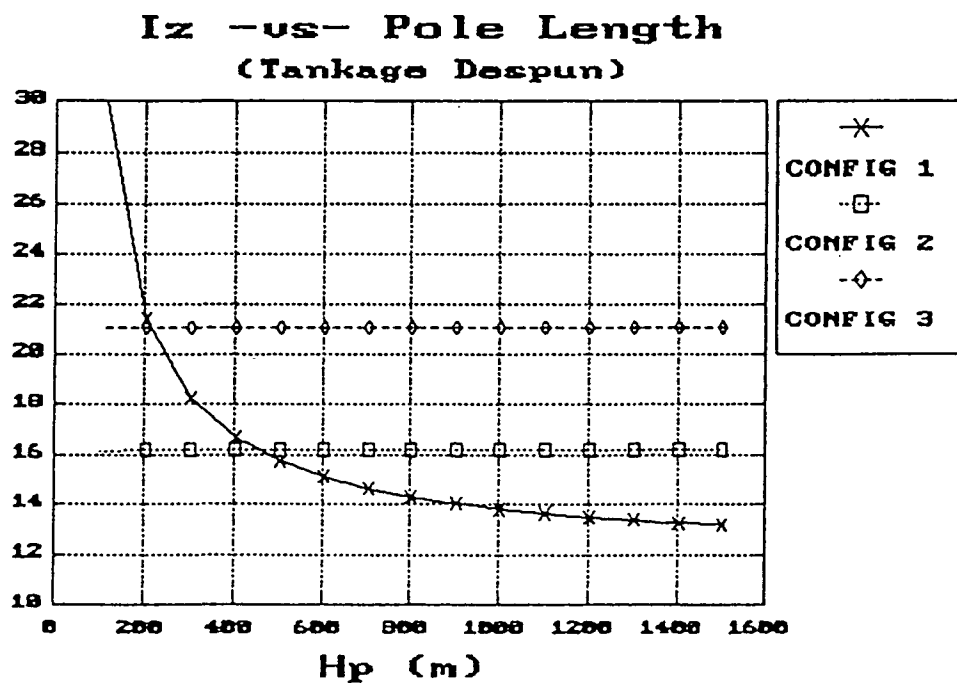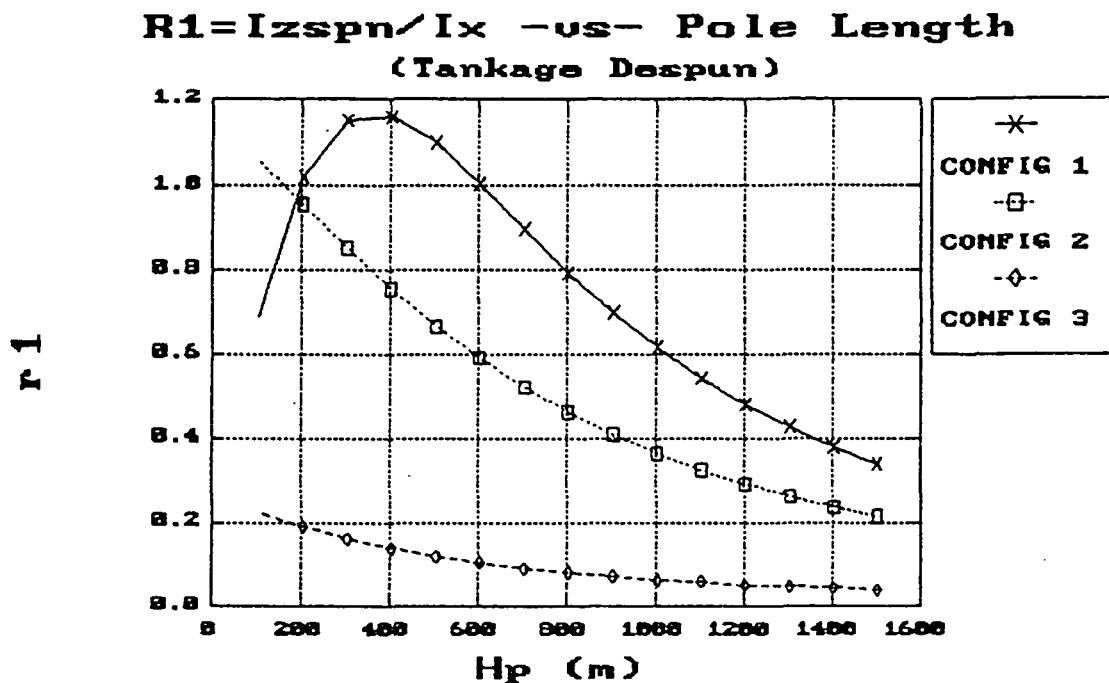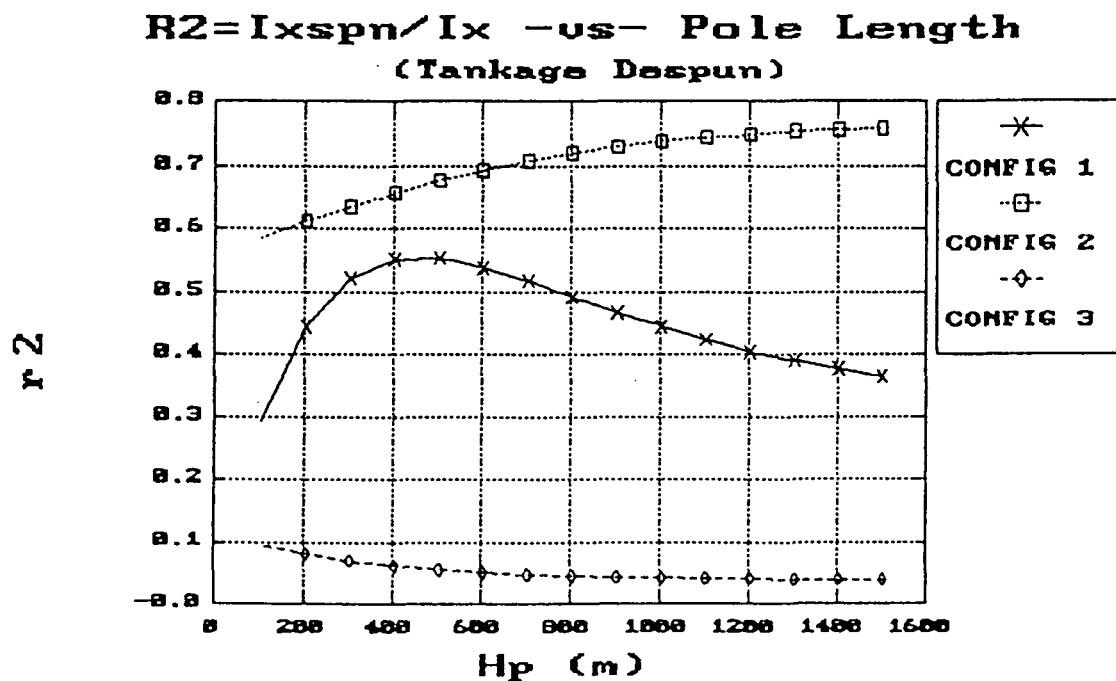
Figure 5.7

5.2 ATTITUDE CONTROL DESIGN

5.2.1 Background

It is understood that with a body the size of the E-City there will be structural deformations due to the thrusting loads and any disturbances encountered. An analysis including the effect of such deformations on the attitude control is beyond the scope of this project; therefore, to reduce the complexity of this analysis the E-City was considered to be a rigid body which could further be utilized to provide starting values for analysis and design of attitude control for the flexible system.

5.2.2 State Feedback Control Design

The first step in performing this task was the determination of the state-space equations that model the motion of the E-City[13]. The equations representing the dual-spin nature of the ship were expressed in the non-dimensional state-space formulation

$$[\dot{\hat{X}}] = [A][\hat{X}] + [B][\hat{T}] \tag{5.6}$$

where

$$[A] = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ r_2\text{-}r_1 & 0 & 0 & 2r_2\text{-}r_1 \\ 0 & r_2\text{-}r_1 & r_1\text{-}2r_2 & 0 \end{bmatrix}$$

$$\tag{5.7}$$

$$[B]^T = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$[\hat{x}]^T = [\hat{\theta}_1 \ \hat{\theta}_2 \ \dot{\hat{\theta}}_1 \ \dot{\hat{\theta}}_2]$$

and $[\hat{T}]$ is the non-dimensional torques provided by the control thrusters (^ denotes non-dimensionalized quantities). The variables r_1 and r_2 in the state matrix A are the moment of inertia

5.11

ratios defined previously in the MMI analysis which are the governing parameters of the attitude motion. θ_1 and θ_2 represent the angular motions of the ship about the x and y axes which lie in the torus plane. Using PRO-MATLAB[5] the linear quadratic regulator control theory was used to obtain the non-dimensional state response that minimizes the control design performance index, defined by

$$CDPI = 1/2 \int_0^\infty (\hat{X}^T Q \hat{X} + \hat{T}^T R \hat{T}) d\tau. \tag{5.8}$$
$$Q = wx\,[1] \text{ and } R = wc\,[1]$$

where Q is the positive semi-definite state weighting matrix and R is the positive-definite control weighting matrix. In the design studies, wx and wc are scalar state and control weighting parameters and [1] denotes an identity matrix of appropriate size. By minimizing this performance index the control effort and deviations in state are minimized and a linear state feedback control law

$$[\hat{T}] = -[G][\hat{X}] \tag{5.9}$$

is obtained where G is the control feedback gain matrix which minimizes the integral in the Equation 5.3 for CDPI.

5.2.3 Attitude Control Power Required

The non-dimensional control power consumed can be found by the non-dimensional power S^* using the control torques of Equation 5.9,

$$S^* = \int_0^\infty \hat{T}^T \hat{T}_u^{-T} \hat{T}_u^{-1} \hat{T} d\tau \tag{5.10}$$

which yields

$$S^* = \hat{X}_o^T P^* \hat{X}_o \tag{5.11}$$

5.12

for any initial disturbance state X_0. The power matrix P^* is the solution to the following Lyapunov equation

$$A_{CL}{}^T P^* + P^* A_{CL} + G \hat{u}^{-T} \hat{u}^{-1} G = 0 \qquad (5.12)$$

where

$$A_{CL} = A - BG \qquad (5.13)$$

and the non-dimensional thruster distribution matrix is defined by

$$\hat{u} = D_t{}^{-1} [D] \qquad (5.14)$$

in which D_t is the torus diameter and the elements of $[D]$ are the moment arms the control thrusters act on to generate torques about the x and y body axes of the station which are parallel to the torus plane. Utilizing PRO-MATLAB the previous quantities and relations were evaluated.

The various dimensional and non-dimensional quantities are related by

$$[\theta_1 \ \theta_2 \ \dot{\theta}_1 \ \dot{\theta}_2] = [\hat{\theta}_1 \ \hat{\theta}_2 \ \hat{\dot{\theta}}_1 \ \hat{\dot{\theta}}_2] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & n & 0 \\ 0 & 0 & 0 & n \end{bmatrix} \qquad (5.15)$$

$$t_{con} = \tau_c / n \qquad (5.16)$$

$$F_i = \hat{F}_i (I_x n^2) / D_t \qquad (5.17)$$

$$T = \hat{T} I_x n^2 \qquad (5.18)$$

where n is the torus spin rate in radians per second, t_{con} is the time required to damp out a disturbance, the elements of F_i are the control thrusts, and the elements of T are the control torques.

5.13

The non-dimensional control thrusts can be obtained with the following equation

$$\hat{F}_{con} = \hat{D}^{-1}\hat{T} \tag{5.19}$$

The root-mean-square power required to damp out a disturbance can then be found from the value of S^*, computed using Equations 5.11 and 5.12. The root-mean-square control power required is then given by the formula

$$P_{rms} = (v_{ex}\, I_x\, n^2 / 2\, D_t)^* \mathrm{sqrt}(S^*/\tau_c) \tag{5.20}$$

where v_{ex} is the exhaust velocity of the control thrusters.

5.2.4 Attitude Control Propellant Requirement

The propellant mass required to damp out a disturbance can be found using the formula

$$m_{p,con} = (I_x\, n / D_t\, v_{ex}) \sum_{j=1}^{f} \int_0^{\tau_c} |\, F_j\, |\, d\tau \tag{5.21}$$

where f is the total number of inputs of control clusters used. The integral in Equation 5.16 is found by numerically integrating the non-dimensional thrust input versus non-dimensional time relationship.

5.2.5 Determination of Torus Acceleration

The acceleration's of the torus relative to the inertial axes are of interest as this is the section of the ship that will be inhabited. Viewing these accelerations from the standpoint of the crew served as the primary design constraint for the control system. Specifically, it was desired that the acceleration felt by the crew in the z-direction, which would be comparable to the rocking of a boat, would immediately settle to zero after the initial disturbance without overshoot. Calculation

of the torus accelerations was performed using the following equations:

$$a_p = \dot{\omega} \times r + \omega \times (\omega \times r) + 2\omega \times \dot{r} + \ddot{r} \qquad (5.22)$$

where ω is the angular velocity vector of the torus relative to the shaft defined by

$$\omega = [\ -\theta_2 \quad \theta_1 \quad 1\ ]\ \Omega \qquad (5.23)$$

and the position vector r is

$$r = (\ D_t / 2\ )\ [\ \cos\theta_p \quad \sin\theta_p \quad 0\ ]. \qquad (5.24)$$

The torus spin rate is Ω and θ_p represents the crew members' position on the torus relative to the torus axes.

5.2.6 Attitude Control Thruster Configuration

For this analysis the control thrusters were considered to be placed in clusters evenly distributed around the main propellant tank, applying forces only in the axial direction. This assumption allows each cluster to act as a single input in the control study which reduces the size of the thruster distribution matrix, D. After finding the total thrust-time history at each cluster location the absolute maximum thrust can be determined and the number of thrusters, which act in parallel, per cluster can be found given the thrust capabilities of a selected thruster.

5.2.7 Attitude Control System Optimization Process

The control theory and the MMI analysis described previously in this chapter were used to study the attitude control of the E-City for a wide range of ship configurations, thruster configurations, and initial disturbances. Programs were generated in PRO-MATLAB (Appendix E) to simulate the state response of the E-City due to an initial disturbance and to calculate control

5.15

system performance parameters. Optimization of the attitude control power required, the number of clusters, the number of thrusters per cluster, the propellant mass requirements, and control weighting parameters, wc and wx, of the control law design were the major objectives of the attitude control study conducted during Phase III of Project WISH. Acceleration levels on a crew moving with uniform velocity calculated at a point on the torus established limits on the state response based on human factors and served as the primary design constraints, as stated previously.

The first step in the optimization process was to find the minimum number of clusters necessary to damp out the gyroscopic wobble dynamics of the E-City for a given disturbance. To reduce the amount of power required for control, the number of clusters was computed against P_{rms} for various weighting parameters, wc and wx, and several initial disturbances. It was determined that 50 clusters could easily handle a basic initial disturbance rate of 0.6 deg/sec about both torus axes, x and y. See Figure 5.8.

After this was accomplished, the state response due to an asteroid impact was modeled to determine its effect on the E-City. Two cases were evaluated assuming perfectly plastic collisions: a head-to-head collision and a tail-to-tail collision. The relative velocities between the E-City and an asteroid consisting of iron were assumed to be 7 km/sec and 27 km/sec. Using our initial disturbance as a reference response, it was found that for a head-to-head collision it would take an asteroid of a 10m radius and for the tail-to-tail asteroid a 6m radius to reciprocate this initial disturbance. See Figures 5.9 and 5.10. This analysis was done to quantify what a 0.6 deg/sec initial disturbance would "feel like" to the ship and what this would represent in physical terms.

Determining the acceleration levels experienced by the crew was the next step in the optimization process. Through this analysis, the number of thrusters in each cluster, propellant mass per control effort, maximum thrust per cluster, settling time, and weighting parameters wc and wx were determined. As stated previously, the primary constraints based on human factors were the acceleration felt by the crew in the z-direction and the control settling time. It was desired that the acceleration in the z-direction would immediately settle to zero after the initial disturbance with minimum overshoot and that the length of time the crew felt the disturbances be a minimum.

The acceleration levels were determined for a person running a 5.5 minute mile in the spin

5.16

direction of the torus. See Figures 5.11 and 5.12. This would be the direction for which the crew would experience the highest g levels. The weighting parameters wc and wx were then varied to determine the control settling time, propellant mass, and maximum thrust. First, the various design parameters were studied against values of wc for a specific wx. See Figures 5.13-5.15. Several iterations were completed to determine the patterns for the various parameters in relation to increasing or decreasing weighting parameter values. It was found that as the state weighting, wx, was increased, the number of engines per cluster significantly increased while the overshoot of the z-direction acceleration experienced by the crew and the settling time decreased. Thus, the control weighting, wc, was used to optimize these parameters while also minimizing the propellant mass, maximum thrust per cluster, and the number of engines per cluster.

In considering the duration of the disturbance, it was decided that an optimal settling time would be approximately 2 minutes. This matches approximately with the period of the lower gyroscopic frequency of the uncontrolled ship and represents an acceptable thrust time for chemical control thrusters. Using the above plots, it was determined that this settling time could be achieved and that the performance parameters could be optimized when wc was equal to 640 and wx was 375. See Table 5.1. The closed loop response and the phase plots for this system are depicted in Figures 5.16-5.21.

Table 5.1: Attitude Control System Design Results

Thrust Available /Engine	2.58 Million (N)
Isp/Engine	437 sec
Control Power Required	722 Billion Watts
Max Thrust/Cluster	289 Million (N)
# of Engines/Cluster	112
Control Propellant	32.1 Million (kg)
wc (control weighting)	640
wx (state weighting)	375
# of Thruster Clusters	50

5.2.8 Conclusion

Chemical control thrusters were chosen for the final design; however, the use of NLB

5.17

engines for control was considered. The advantages to the NLB lie in its higher I_{sp} and thrust levels; thus, the mass of propellant required per control effort is reduced as well as the number of control thrusters required per cluster. The mass of the NLB is much higher in comparison to a chemical thruster; therefore, even though the number of control thrusters required is reduced the mass of the control thrusters actually increases. Due to the uncertainties that remain with respect to the NLB in regards to its startup time and excess heat production it was not considered a valid selection. Hence, the most viable engine selection for the attitude control design of the E-City is chemical based thrusters.

Further design with respect to attitude and control would utilize the results from the structural dynamic design to examine the control requirements for the flexible model of the E-City.
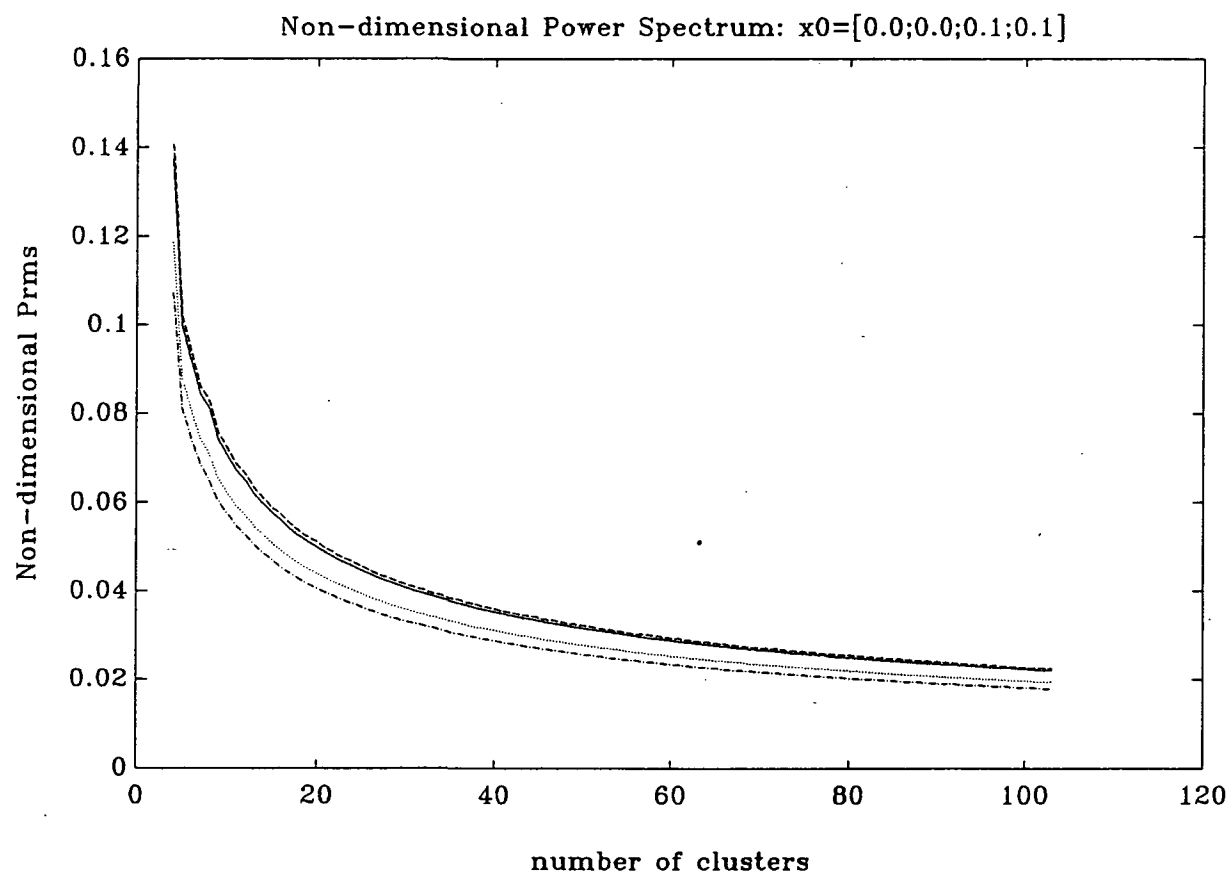
Figure 5.8: Power Spectrum-Prms vs. Number of Clusters

open loop x0=[0.0;0.0;0.1;0.1]
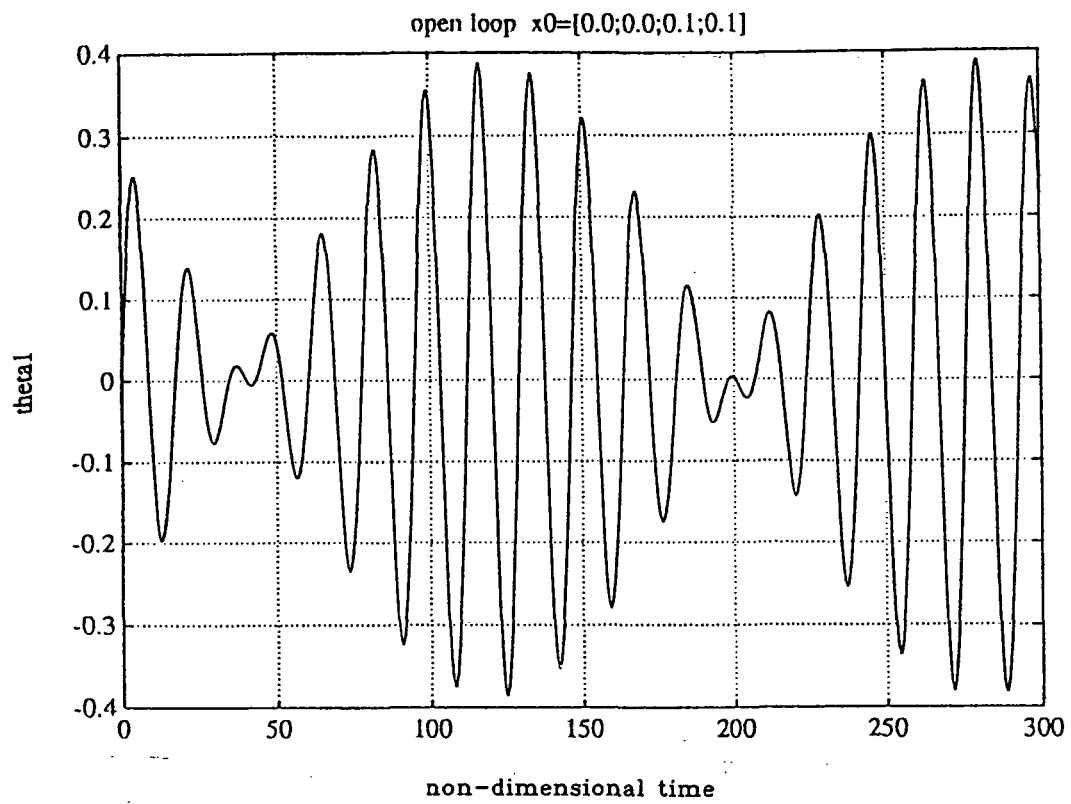
Figure 5.9: Open Loop Response due to an Initial Disturbance

open loop non-dimensional torque=6.603
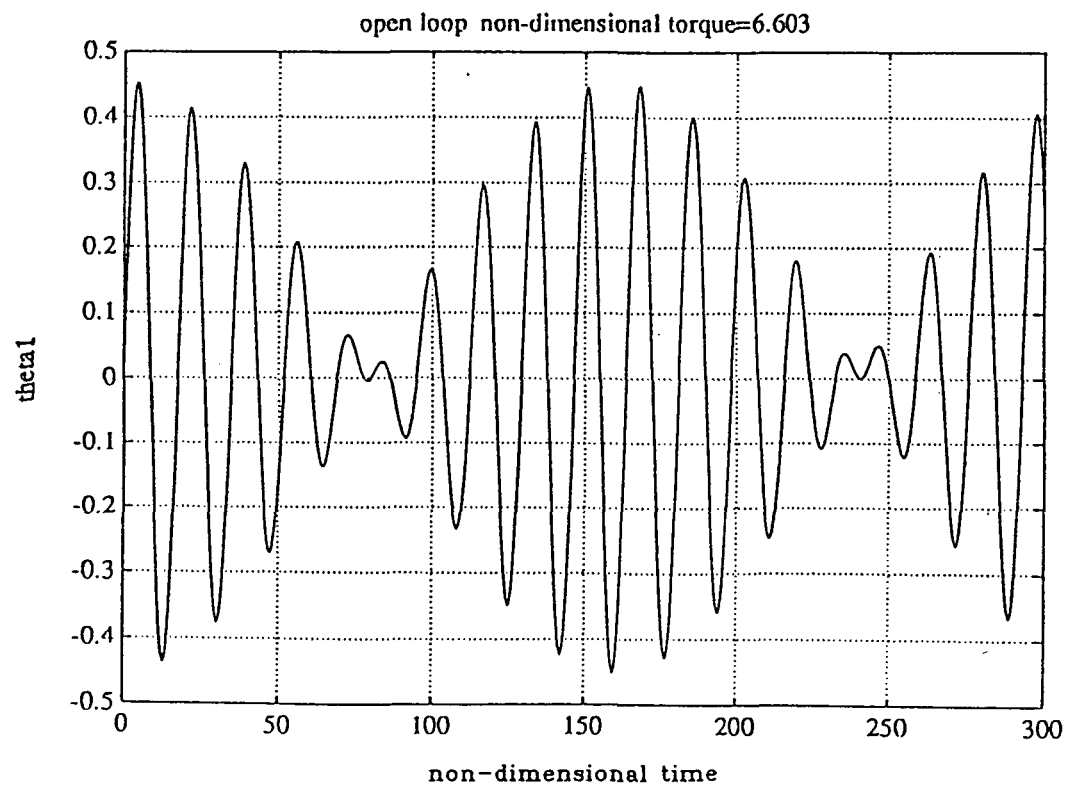
Figure 5.10: Open Loop Response due to an Asteroid Impact
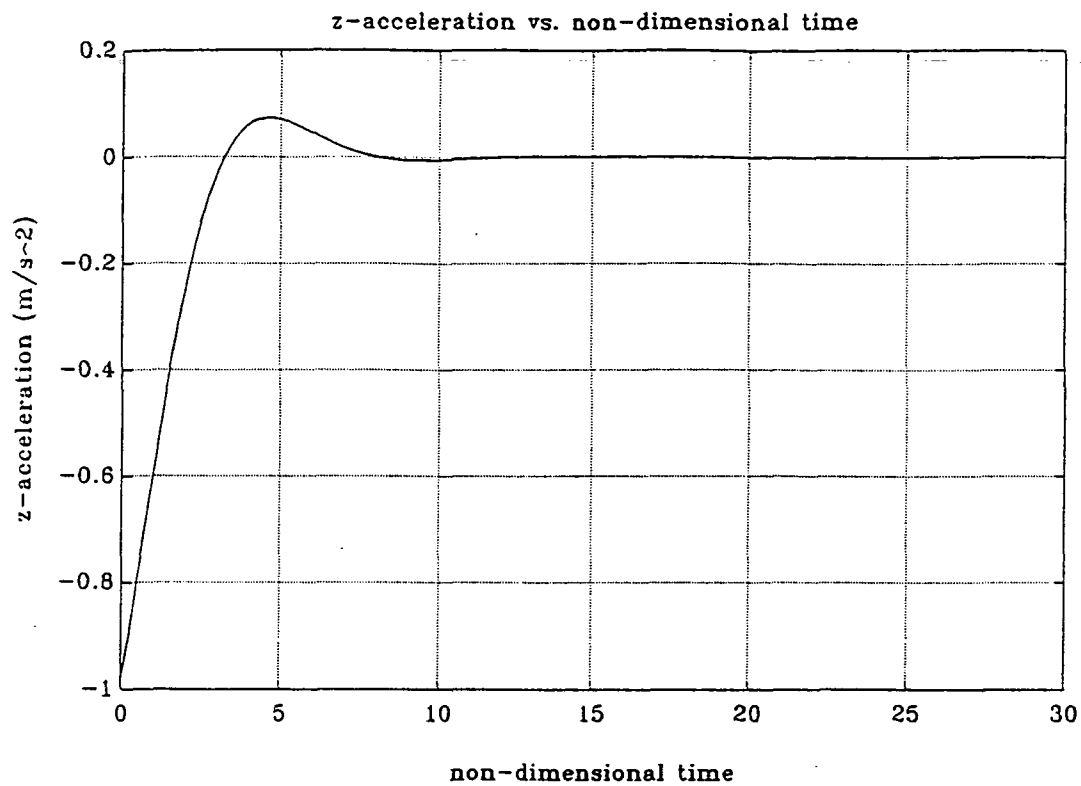
Figure 5.11: Profile of Z-acceleration
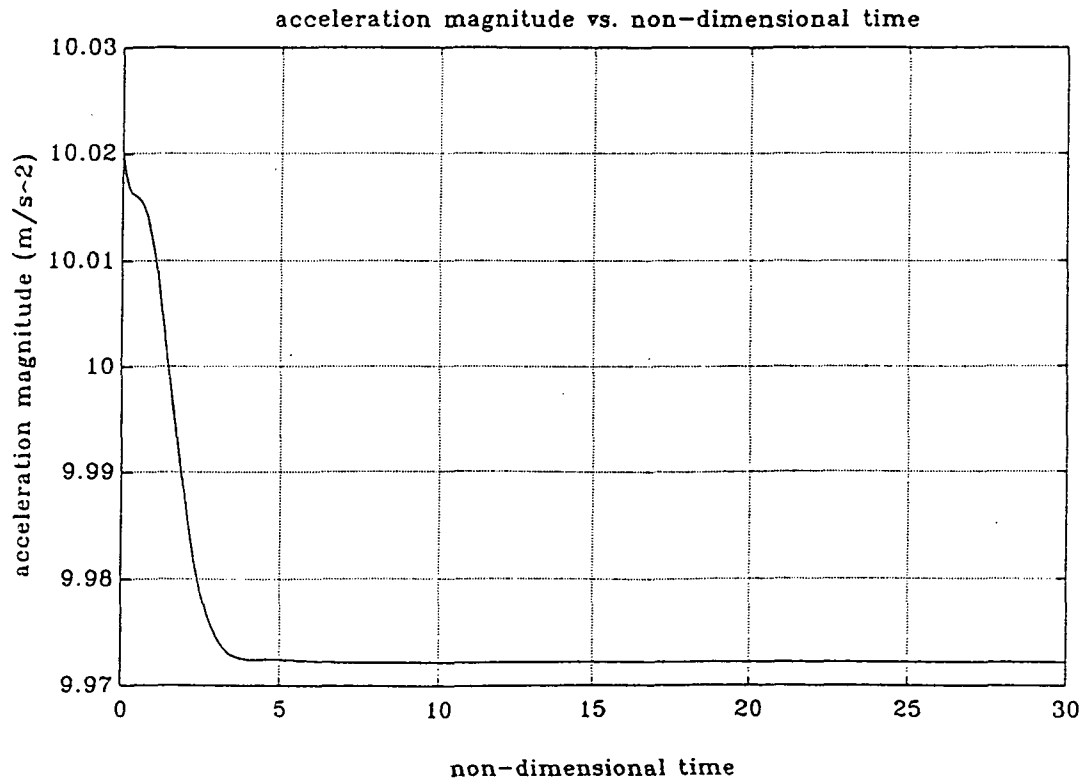
Figure 5.12: Profile of Magnitude of Acceleration

5.21
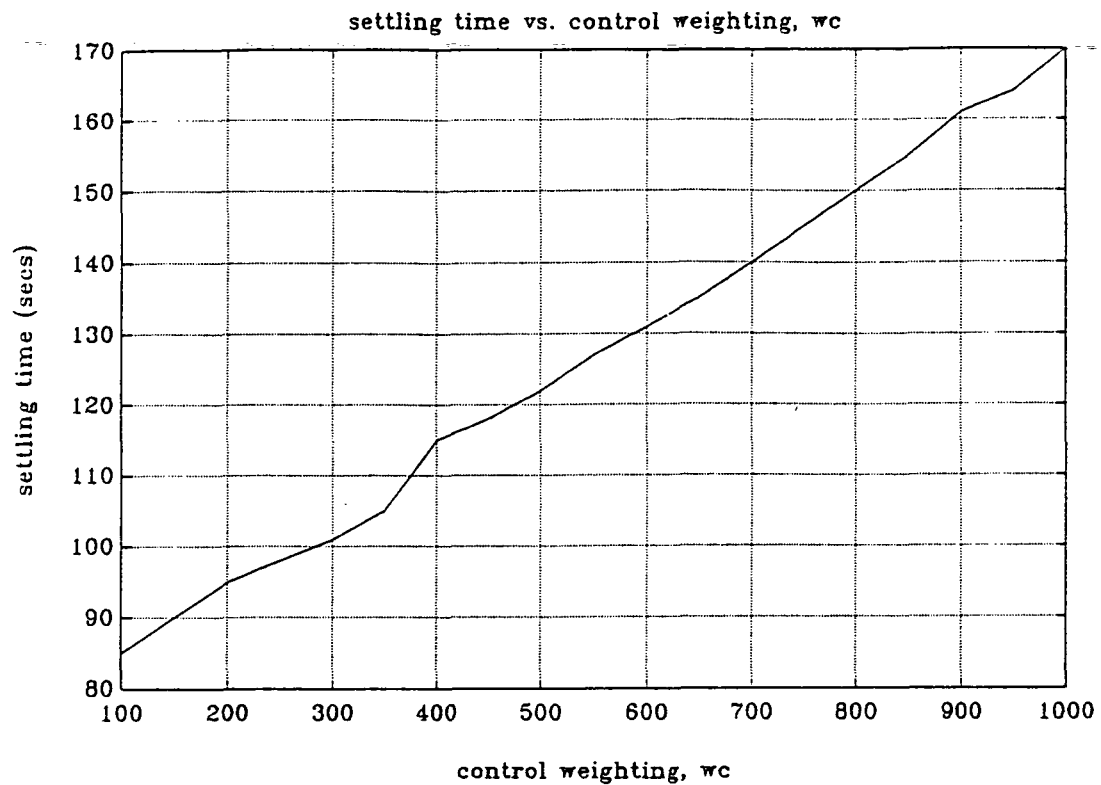
Figure 5.13: Control Settling Time vs. Control Weighting, wc
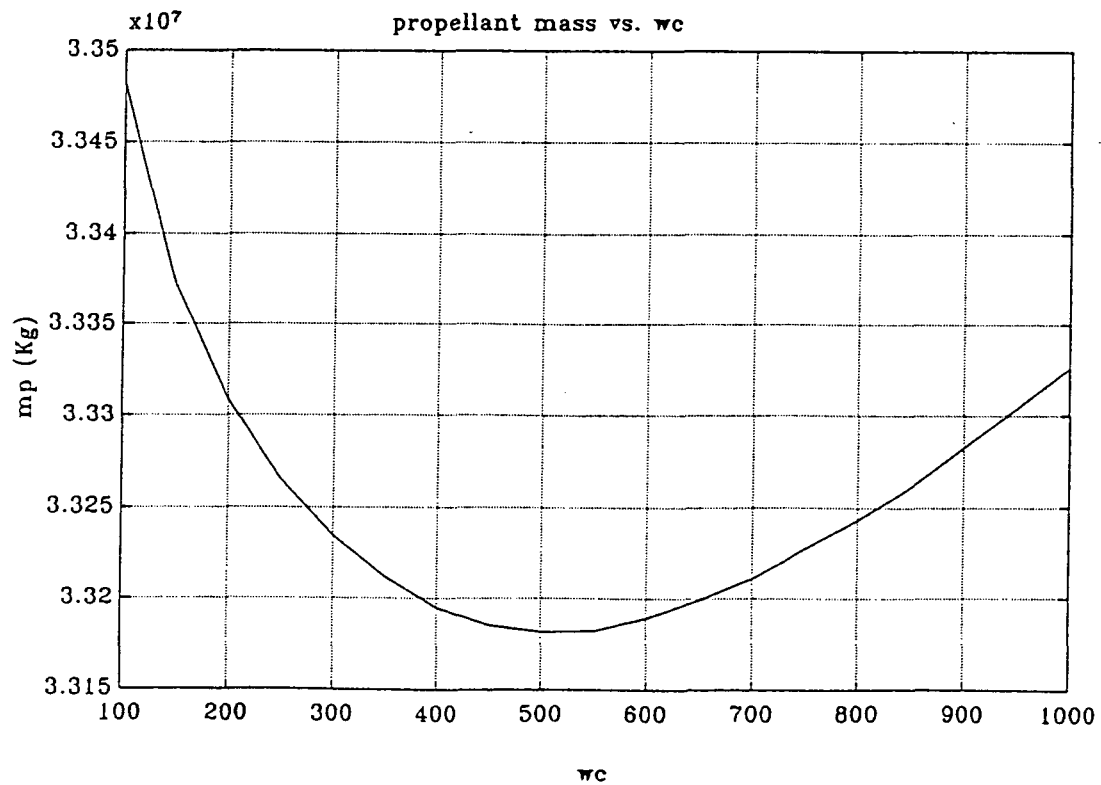
Figure 5.14: Control Propellant Mass vs. Control Weighting, wc
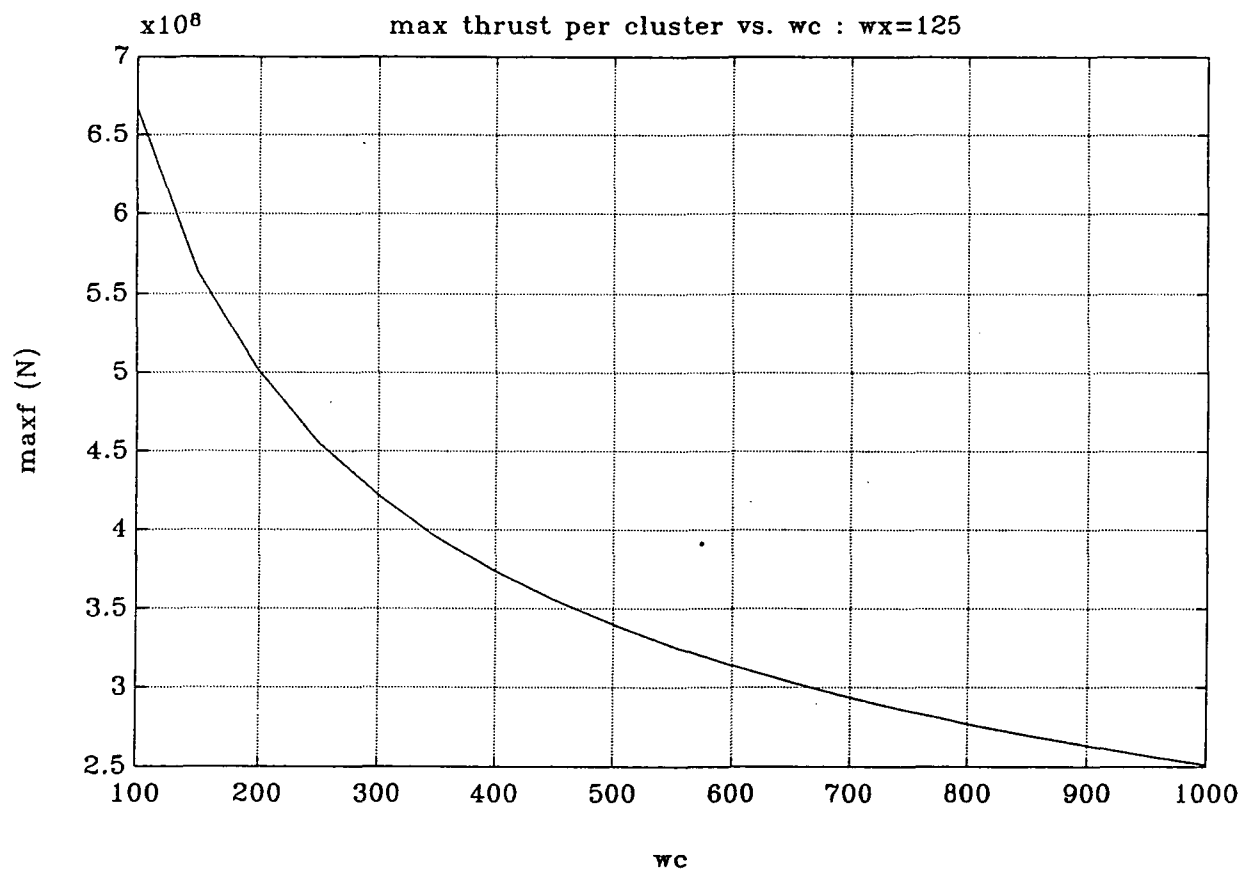
Figure 5.15: Maximum Control Thrust vs. Control Weighting, wc

closed loop response of theta1 and theta1dot
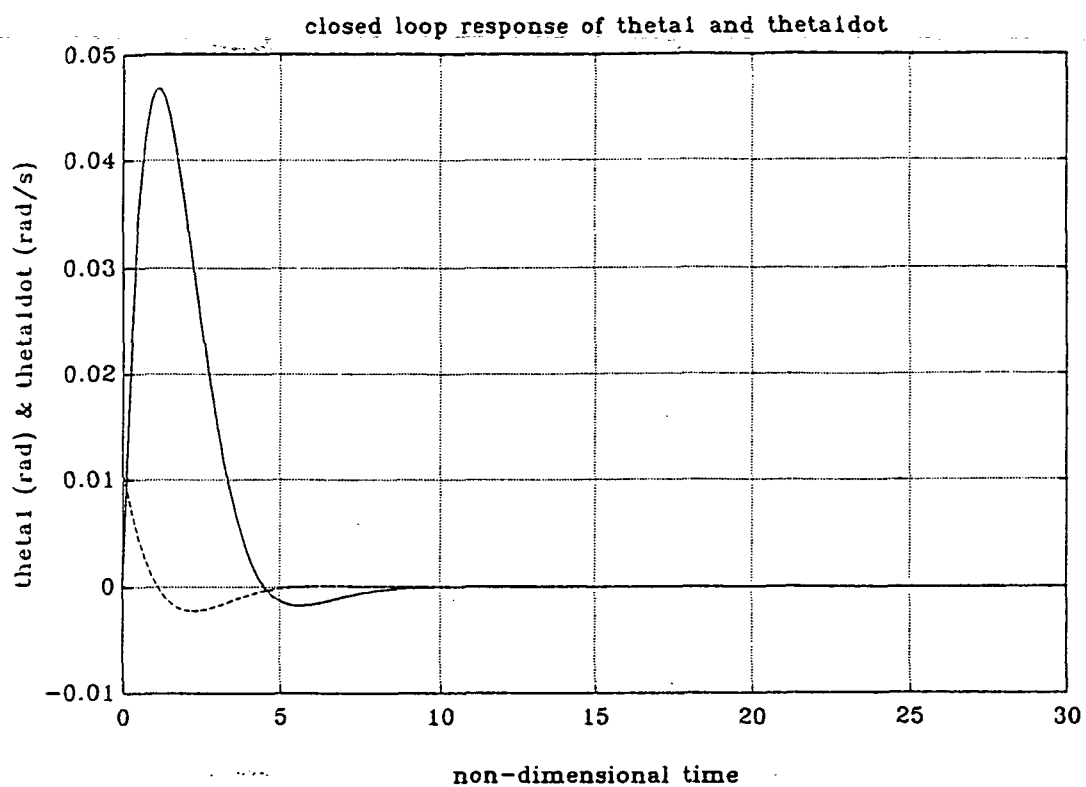
Figure 5.16: Closed Loop Response of State Variables θ_1 and $\dot{\theta}_1$

closed loop response of theta2 and theta2dot
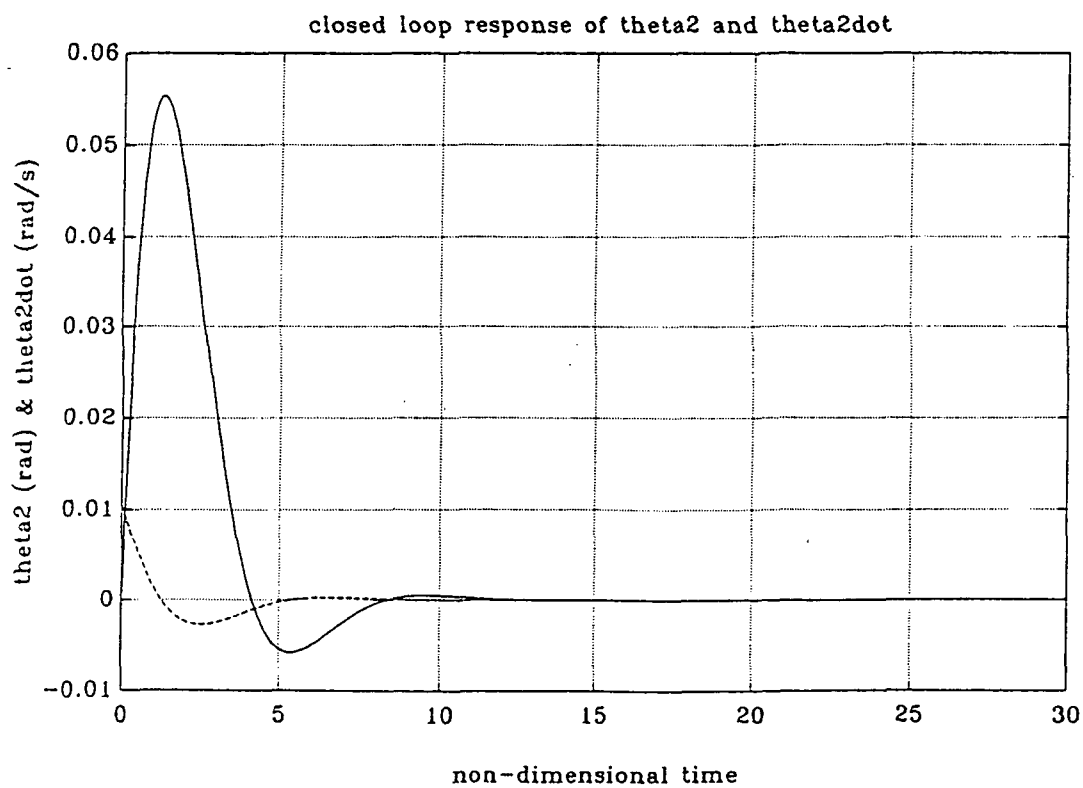
Figure 5.17: Closed Loop Response of State Variables θ_2 and $\dot{\theta}_2$
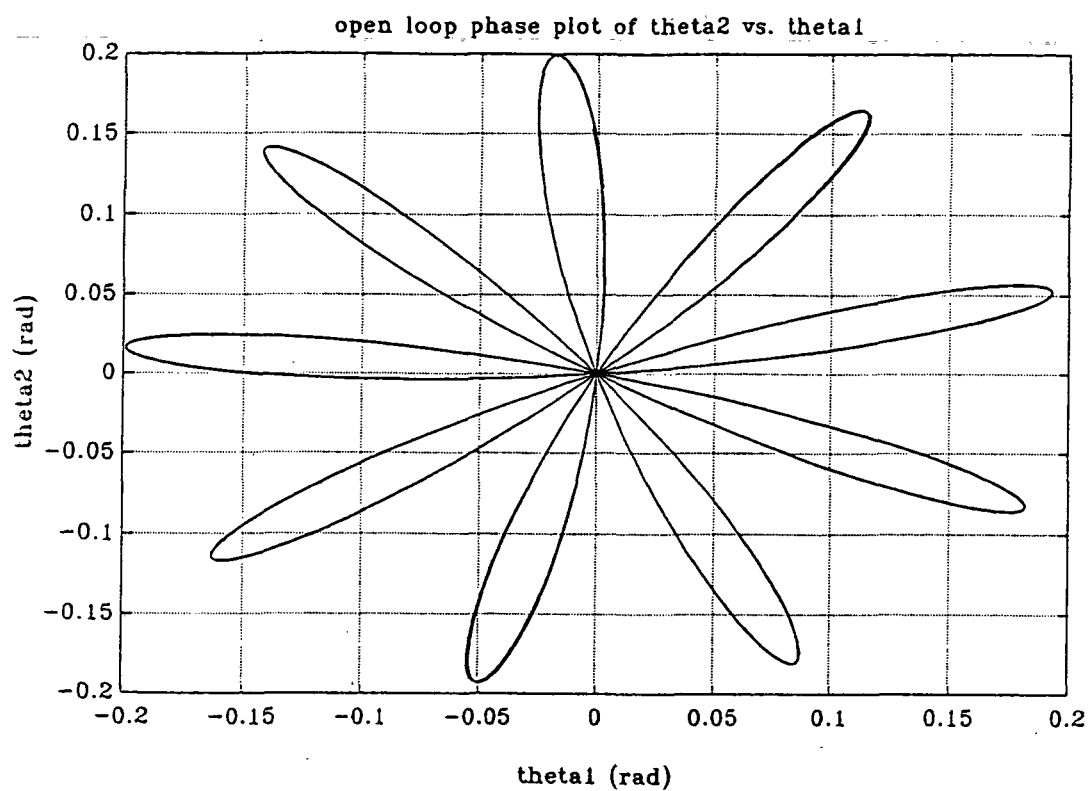
Figure 5.18: Open Loop Phase Plot of θ_1 vs. θ_2
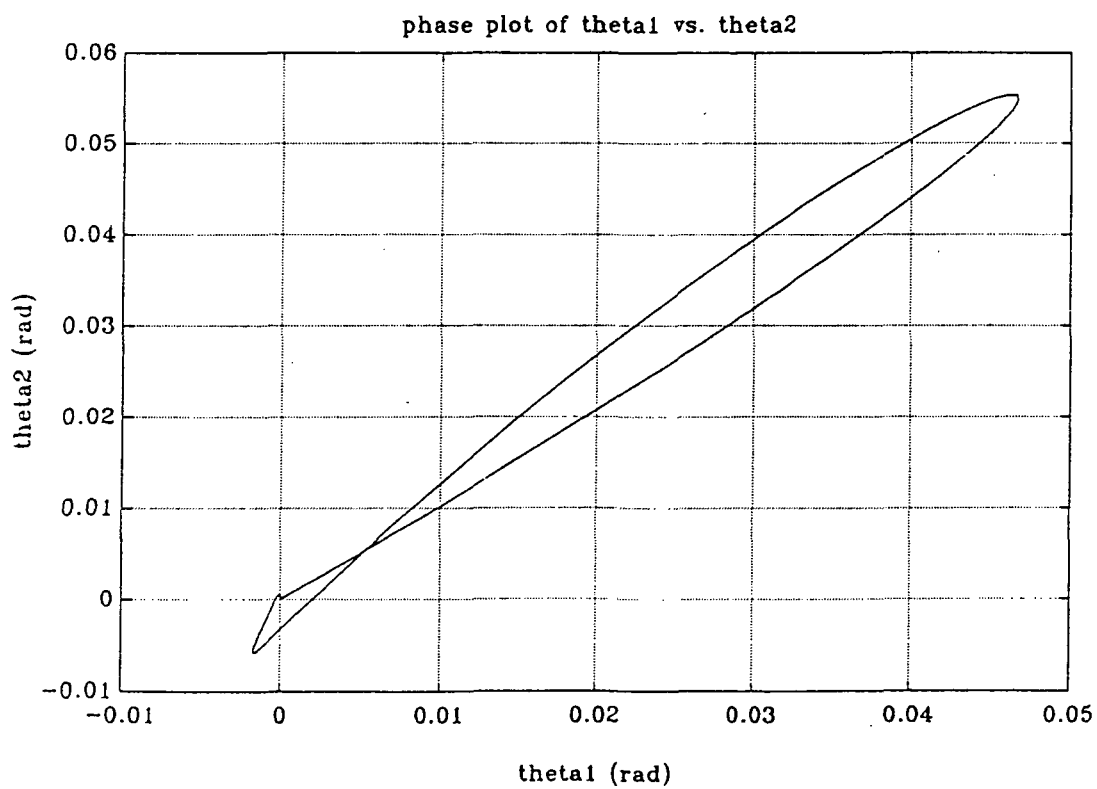
Figure 5.19: Closed Loop Phase Plot of θ_1 vs. θ_2

5.25
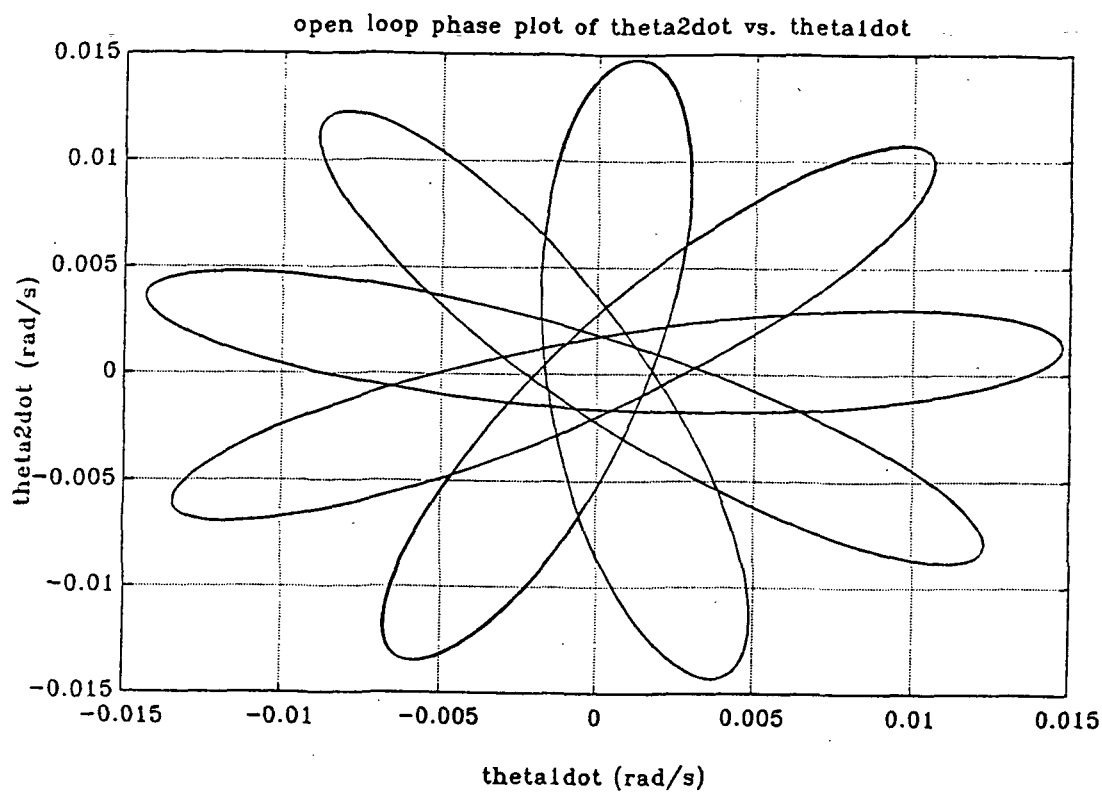
Figure 5.20: Open Loop Phase Plot of $\dot{\theta}_1$ vs. $\dot{\theta}_2$
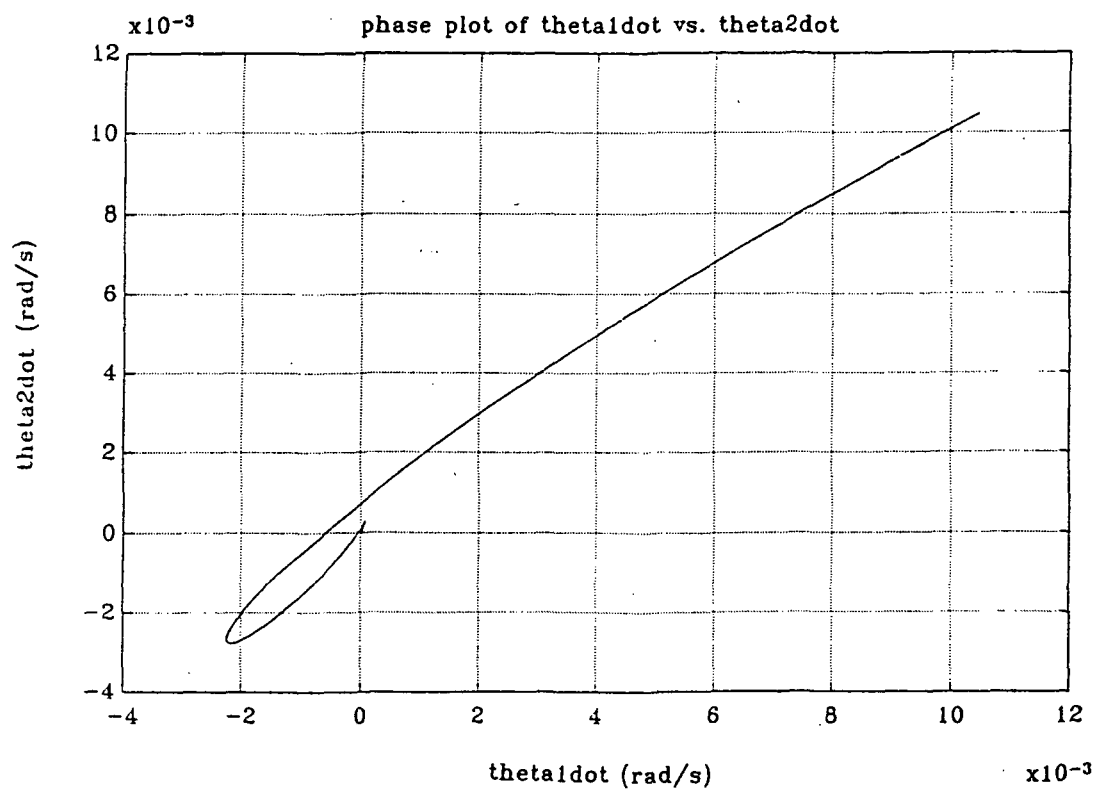
Figure 5.21: Closed Loop Phase Plot of $\dot{\theta}_1$ vs. $\dot{\theta}_2$

CHAPTER 6

POWER AND THERMAL CONTROL SYSTEMS

6.0 INTRODUCTION

The E-City is composed of numerous subsystems that will require electrical and/or thermal energy to operate properly. Many considerations must be taken into account in order to develop feasible power and thermal control systems for the ship[1]. For example, the primary concern of the power system is that it can supply continuous energy to each of the E-City's subsystems for as long as necessary to complete a given mission. The thermal control systems must be able to dissipate the waste heat generated by the various power devices on board the ship. Also, each of these systems must be able to function properly during all modes of operation for E-city. Basic operating modes of the ship include initial startup of the engines, engine burn time, shutdown, and main engines off.

It is the purpose of this chapter to present in detail the governing factors that determined the power supply and thermal control systems for the E-City.

6.1 THE POWER SYSTEM

6.1.1 General Considerations

Before a detailed explanation of the current E-City power system can be presented, it is necessary to consider the existing factors that led to its design. First, the Phase I design team had spent a considerable amount of time and effort towards determining a feasible system. As it turns out, much of the current power system stems largely from the original configuration envisioned by the first year's team. Another consideration is the necessary power needed to startup the NLB engines used for main propulsion. It was through the basic analysis of a fission process that led to the determination of the startup procedure for the main engines. Finally, the power budget allotted for basic ship operations such as life support, navigation, and communications, must be presented. Building on the Phase I team's analysis, the power budget, as determined in the first year of Project WISH, was used.

The power system for E-City was originally investigated by the first year's (Phase I) design team. From their analysis, they concluded that a rotating particle bed reactor was a sufficient power supply for E-City. The reactor, shown in Figure 6.1, operates by storing fissioning uranium fuel in a cylinder with porous walls. As stated in the Phase I report, the cylinder is then rotated to force the reacting fuel against the wall for even distribution, and a cold gas is passed through the porous walls, heated, and carried out through pipes along the axis of the cylinder. Using only 1 m^3 of fuel, the particle bed reactor is capable of producing a projected maximum of 5000 MWth within three seconds of initial startup.

Most of the reactor can be made of modern light-weight alloys because the only extreme temperatures developed will be after the coolant passes through the fuel. This reactor has many advantages. It possesses a quick response time, very high power density, and virtual immunity to thermal stress.
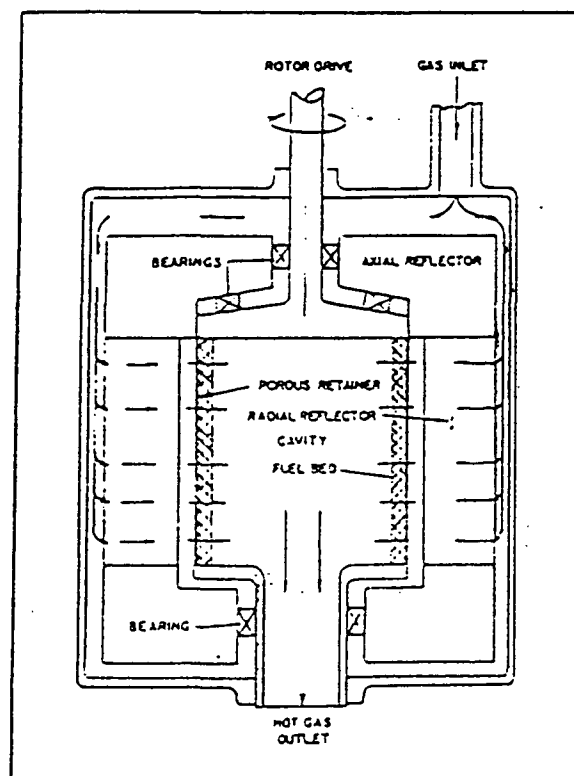
Figure 6.1: Rotating Particle Bed Reactor.

6.2

The thermal energy created by the reactor must be converted into usable electrical energy for E-City's subsystems. The Phase I team had performed extensive research on various types of conversion systems, and decided to use two specific types.

The first was a dynamic system known as the magnetohydrodynamic (MHD) converter. From the first year's report, this device is based on the concept that if a metal conductor is rotated in a magnetic field, an electric potential can be produced. For the MHD, however, the metal conductor is replaced with a conducting gas. If the gas is accelerated to very high speeds through the magnetic field, and if the gas has a very high temperature (approximately 2500 K), then this system can reach efficiencies of up to 40%. It was originally envisioned by the Phase I team as converting the power to be used for the anti-matter propulsion system.

The second power conversion system is a static device. The thermoelectric conversion system relies on the Seebeck Effect in which two dissimilar materials are maintained at different temperatures, forming the basis of a thermocouple. The juncture of the two materials creates an electric potential, which can then be used to drive a current. The thermoelectric converter typically has a lower efficiency than the MHD, and can only be used in low power (1 kW to 100 MW) applications. The Phase I team used this system primarily to run E-City's subsystems, and to serve as an emergency backup converter in case the MHD failed.

The Fission Reaction
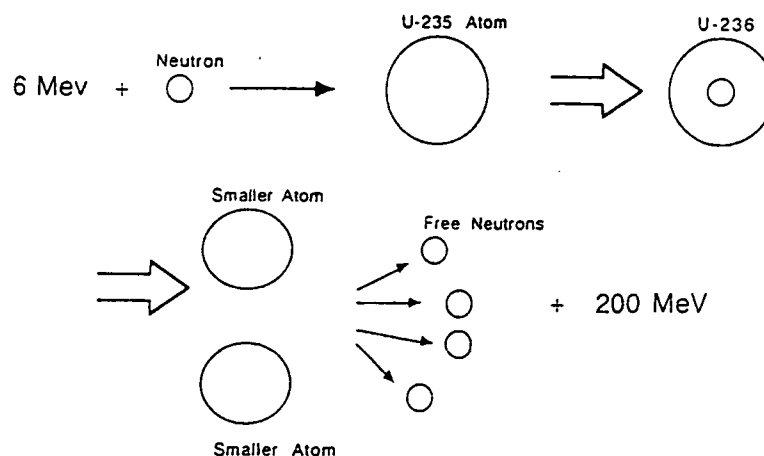
Figure 6.2: Typical Fission Reaction.

6.3

Figure 6.2 shows the chain of events for a typical fission reaction. This is representative of a typical mass of uranium, which also consists of free neutrons. It takes 6 million electron volts of energy to insert one neutron into the nucleus of a uranium atom. Once this has occurred, the atom becomes unstable, and splits into two smaller, dissimilar atoms. With the split of the atoms, additional neutrons and approximately 200 MeV of thermal energy are released. It is the release of the neutrons and energy that create a chain reaction throughout the mass of uranium.

Equation 6.1 simply says that the ratio of power applied to create fission within the uranium is 6 / 200 of the power output created from the reaction. From this, it is possible to calculate the power needed to induce a fission reaction within one NLB engine. From Chapter 2, the output reactor power of one NLB engine used for main propulsion is 160,000 MWth. The power required for startup is then 6 / 200 (or 3%) of this output, or 4800 MWth.

$$P_{rf} = 6 / 200 * (P_{out}) \hspace{3cm} (6.1)$$

Power Budget

The Phase I team determined the approximate quantity of electric power needed to satisfy the operation of E-City's subsystems. Table 6.1 lists the power required for each subsystem. These figures are estimates, and they will probably change as each system is further analyzed.

Table 6.1: Power Budget.

System	Electric Power
Life Support	3 MWe
Communications	5 MWe
Navigation	4.85 MWe
Shuttle/Maintenance	5 MWe
Miscellaneous	5 MWe
TOTAL	22.85 MWe

6.1.2 Phase III Power System

Figure 6.3 displays the current power network of E-City. As mentioned earlier, much of it is based on the analysis performed by the first year's design team. However, there are some exceptions, due to the change from an anti-matter to a nuclear propulsive system.

Starting at the top of Figure 6.3, a rotating particle bed reactor capable of generating a maximum of 5000 MWth is sufficient to supply power to E-City. The converter coupling network is essentially a switch that channels the thermal energy from the reactor to either the MHD or thermoelectric converter. During normal operation, the thermal energy is channeled completely through the highly efficient MHD converter. In an emergency situation, the thermal power can be redirected to the thermoelectric converter as a type of backup system. The usable electrical power generated by the converters is sent to the power coupling network. The remaining unconverted thermal energy, referred to as waste heat, is dissipated by sending it to an external radiator. This thermal control system will be discussed in Section 6.2.

The power coupling network is responsible for supplying power to the propulsion, attitude and control, heat transfer, and ship operations systems. Depending on the operating mode of the ship, which will be explained in the following section, this network is responsible for supplying the appropriate amount of power to each system. It also serves the purpose of redirecting power to systems in emergency situations.

The propulsion system requires power to operate equipment capable of such tasks as monitoring engine status, controlling the thrust output of the engine, and initiating startup or shutdown. The attitude and control system would require power to perform the same type of functions as the main propulsion system. They are separated in the figure on account of the undetermined nature of the thrusters. The heat transfer subsystems require power to either actively and/or passively dissipate the waste heat created by all the power generating devices of E-City. These devices include not only the main reactor, but the NLB engines as well. Above specific impulse ranges of 2100 seconds, the NLB engines cannot thermally control all of the waste heat they generate. Therefore, it is necessary to dissipate a small percentage of the waste heat generated by the engines[9]. Ship operations involves the systems required for navigation, life support, communications, shuttle and maintenance, and other miscellaneous tasks. Although it is

6.5
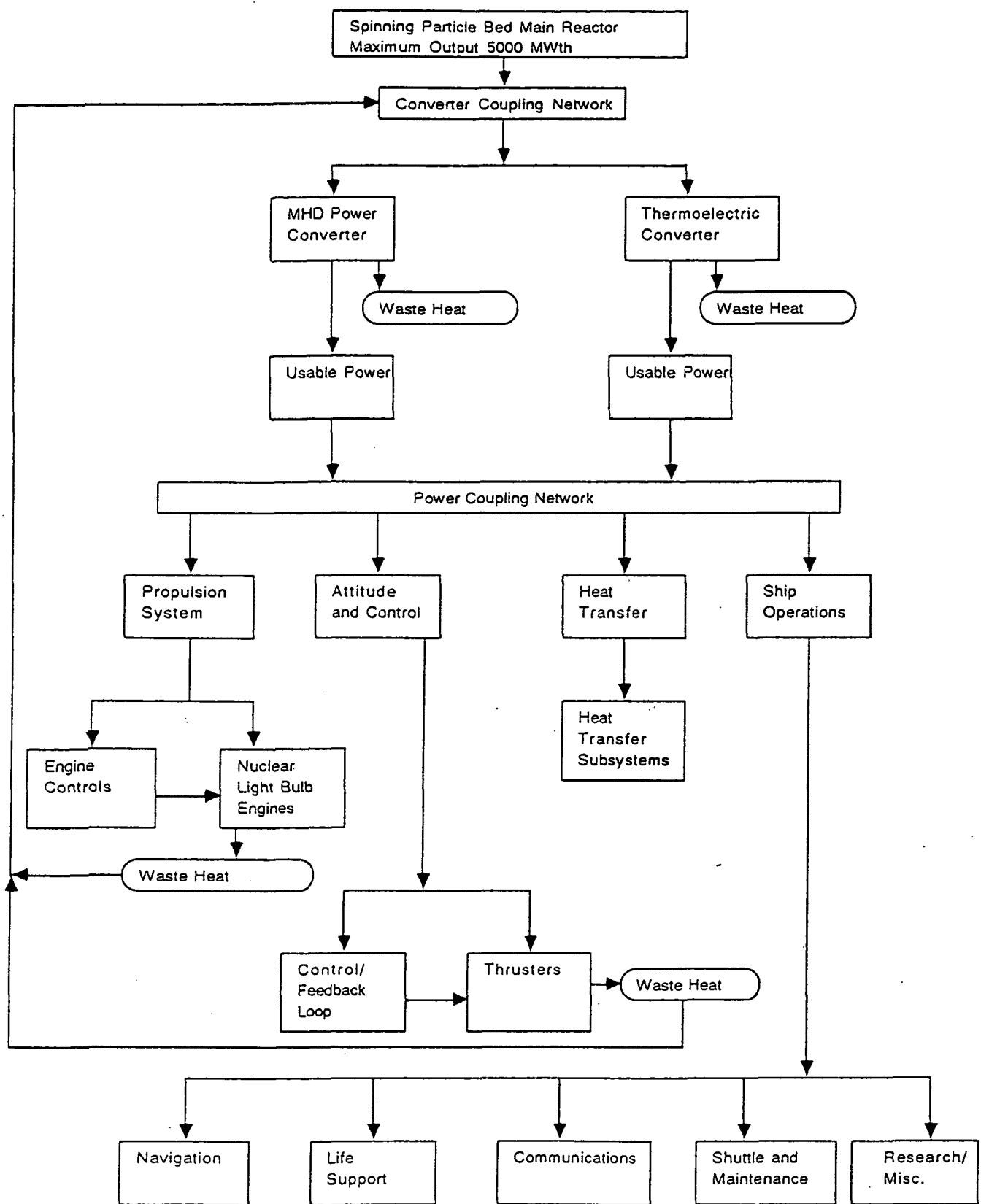
Figure 6.3: Power Network Diagram

essential, the needed 22.85 MWe for ship operations is a small percentage of the total power needs of E-City.

6.1.3 Phase III Ship Configuration and Operating Modes

Because the type of attitude control system has not yet been determined (see Chapter 5), the E-City can be composed of two different ship configurations. The first of which is that the main propulsive system is comprised of nuclear light bulb engines, and that the attitude and control thrusters are made up of some type of high-thrust chemical rockets. The second configuration of the E-City is that both the main propulsive system and the attitude and control thrusters utilize NLB engines. The two different configurations represent a dramatic change in the dynamics of the power and thermal control systems, and will be discussed in more detail in Section 6.3.

Keeping in mind the two possible ship configurations, the next step was to envision typical operating modes of the ship. These operating modes would be based on the startup, burn time, and shutdown of the engines and/or attitude and control thrusters.

The current operating modes of E-city are:

1) Phase I Startup

2) Phase II Startup

3) Main Engines and/or Attitude and Control Thrusters Operating.

4) Shutdown of Main Engines

5) Main Engines Off. Main Reactor and/or Attitude and Control
   Thrusters Operating.

Although most of these operating modes are self explanatory, there are some that require at least a brief explanation.

The startup procedure is essentially a type of chain reaction sequence. The rotating particle bed reactor can supply only 5000 MWth for starting up the engines. As mentioned in Section 6.1.2, the power needed to start one NLB engine for E-City is 4800 MWth. It would be highly impractical to add 13 more particle bed reactors, simply to start the remaining engines. There is another way.

Using the fact that an NLB engine is a power generating device in itself, it does not seem

6.7

unlikely that one modified NLB engine could generate power to start the remaining engines. This modified engine, referred to as the startup engine, would possess some type of moderator/thermal energy network. This network would be capable of using the energy output from the startup engine and supplying it to the remaining engines. In terms of materials and technology, it was felt by this year's design team that such a network would be available by 2050.

Phase I startup refers to the rotating particle bed reactor generating 4800 MWth to start a fission reaction in one NLB engine. Phase II startup refers to the startup engine supplying power to start the rest of the engines.

The startup procedure for the attitude and control thrusters is identical, if the thrusters are NLB engines. In fact, the same startup engine used for main propulsion could also be used to start the thrusters. This is assuming that the power required to start both the main engines and thrusters is not more than the startup engine can provide.

Once the engines and thrusters are operating, they no longer need any external power source. Minimal power for systems monitoring engine and thruster status are needed for propulsion and attitude control. The only systems requiring power are heat transfer and ship operations.

The shutdown of the main engines is a crucial operating mode for E-City. It entails powering down the main engines by terminating the fission reaction occurring within them. Theoretically, this can happen by absorbing the neutrons emitted from the fissioning uranium, and then removing the thermal energy. This process involves the dissipation of enormous amounts of waste heat and will fully tax the thermal control system.

The fifth and final operating mode of E-City takes place when the burn time for the mission has been reached, and the main engines are shutdown. At this time, unless E-City is utilizing NLB engines as attitude control thrusters, the particle bed reactor needs to only supply power to ship operations.

6.2 THERMAL CONTROL SYSTEMS

Thermal control systems can be characterized broadly into two categories: active and passive. This analysis will define the active systems as those required to dissipate internally

generated heat and passive as those required to isolate the station from external heat sources. There is some crossover using these definitions, for instance surface coatings can be tailored to act in either role, but these definitions are convenient for our purposes. Basic heat exchange equations will be used along with several simplifying assumptions to create the thermal model of the E-City.

6.2.1 Passive Thermal Controls

The most critical portions of the E-City needing passive control of heat transfer are the hydrogen propellant storage tanks. The hydrogen propellant is to be stored as a liquid, thus it is very important that heat transfer into the tank be carefully controlled to minimize boiloff. The space environment simplifies the problem somewhat in that the only external source of heat is radiative in nature. There are several possible sources of radiation with the sun, of course, being dominant. The intensity of solar radiation decreases exponentially with distance so only the case of a 1 AU orbit will be considered. This is the closest orbit with which the E-City will be tasked .

In addition to direct radiation from the sun, a body in orbit about a planet or moon experiences reflected radiation known as the Albedo. This source can be considerable and must be taken into account when an orbital mission is contemplated[15]. The extreme additional delta-V required for orbital insertion makes this type of mission impractical for the E-City, therefore Albedo effects will not be considered. The last major source of radiative heat is from external structural components of the station itself. The analysis is very involved, but several further assumptions can be made to reduce this problem.

Most of the heat generated by the nuclear engines will be directed away from the station in the form of exhaust energy. In addition, the necessity for radiation shielding and use of active thermal control systems described later will help limit heat input from the engines into the tank. Waste heat generated in the torus section of the station will be controlled by radiators and by reradiation from the sections facing away from the tank. One final consideration is to look at the shape factor between tank and torus, and it's apparent that the order of magnitude of other sources is small compared to direct radiation from the sun[16]. For these reasons, the following analysis will only consider direct solar radiation.

Requirements and Guidelines

Calculations show that with a bare, uninsulated tank all of the liquid hydrogen would boil off in approximately 1.4 hrs. This is clearly unacceptable and illustrates the necessity for controlling the rate of heat transfer into the tank. Fortunately, improvements are easily attained. Merely by painting the tank white, the boiloff time can be nearly doubled. In order to arrive at a design point, maximum boiloff rate was selected as the sole design criterion and a figure of 0.1% of initial propellant mass per day was chosen. While chosen somewhat arbitrarily this number was thought to be both usefully low and achievable for a first iteration.

The rate of boiloff is determined with the following equation:

$$\dot{m}_{vap} = q / h_{vap} \qquad (6.1)$$

where $\dot{m}_{vap}$ is the mass flow rate of the hydrogen vapor, h_{vap} is the heat of vaporization of liquid hydrogen at 16 K, and q is the total heat flux. However since $\dot{m}_{vap}$ is defined, the above equation can be used to solve for the maximum allowable heat flux. The design goal thus identified was 48.0 MW. The next step is to determine the amount of insulation needed to stay under this limit. The conduction equation can be recast to solve for the insulation thickness as follows:

$$L = K_{ins} * A_s * (\, (T_s - T_c) / q - (L_w / K_w * A_s)) \qquad (6.2)$$

where K_{ins} is the thermal conductance of the insulation, A_s is the tank surface area, T_s is the tank surface temperature, T_c is the inner temp, L_w is the tank wall thickness, and K_w is the tank thermal conductance. All that is needed now is to determine the surface temperature. Using the assumptions made earlier the only heat input will be solar radiation. The equation for finding surface temp is:

$$T_s = (\, \alpha / \varepsilon \,)^{.25} * (\, (\, I_{sun} * A_{perp} \,) / (\, A_s * \sigma_s \,))^{.25} \qquad (6.3)$$

6.10

where (α/ε) is the ratio of absorptance to emissivity of the outermost material or coating, I_{sun} is the solar intensity, A_{perp} is the perpendicular area, and σ_s is the Stefan-Boltzmann constant. Figure 6.4 plots the relationship between solar intensity and skin temperature with a variety of coatings. The constants used throughout have the following values:

$h_{vap} = 427,000$ J/kg

$A_s = 1.3273*10^6$ m^2

$A_{perp} = 331,831$ m^2

$K_{ins} = 0.000106$ W/m*K

$K_w = 163.0$ W/m*K

$L_w = 0.0188$ m

$T_c = 16$ K

$\sigma_s = 5.67*10^{-8}$ W/m^2*K^4

For the first run through it was additionally specified that $\dot{m}_{vap} = 112.56$ kg/s (equal to 0.1% per day), $I_{sun} = 1400$ W/m^2, and the ratio (α/ε) = 0.25. The calculations then gave $q = 48$ MW, $T_s = 198.2$ K and an insulation thickness $L_{ins} = 0.000481$ m . This suggests that the initial guess was too conservative and boiloff can easily be limited to much lower values. It was subsequently decided to specify a maximum heat flux $q = 2$ MW (giving a maximum boiloff rate of 0.02% per day) and use this as the design parameter.

Insulation Materials

Many insulation options are available to the spacecraft designer. As noted earlier, simple external coatings can have a dramatic affect on skin temperatures and will be utilized on the current design. More critical in a cryogenic installation is the insulating material between the outer skin and the inner pressure vessel[17]. Fiberglass could be used, but it would require an insulation thickness of approximately 0.71m. Due to the scale of the E-City this may not be prohibitive, but a

6.11
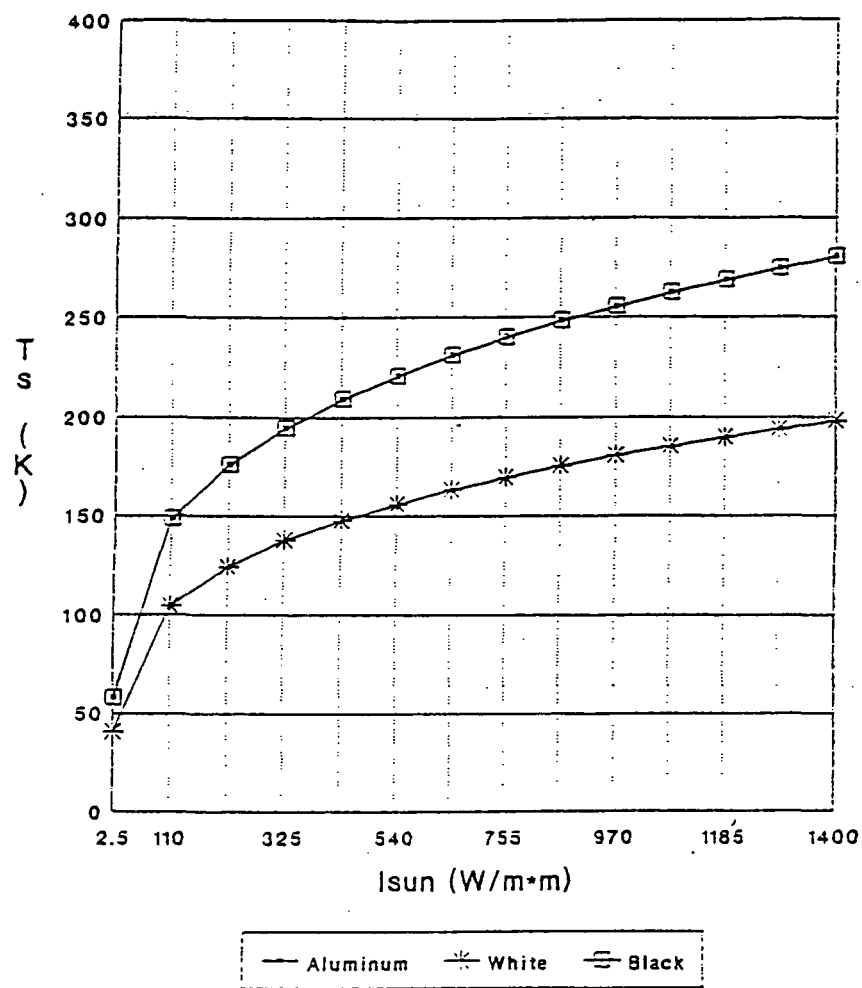
Figure 6.4: Skin Temperature vs. Solar Intensity

Type	multi-layer
Thickness	10.4 mm
Shield Material	aluminum foil
Spacer Material	fiberglass mat
Density	80 kg/m2
Conductance	.000106 W/m*K

Table: 6.2 Insulation Properties

more efficient insulator is available. Multi-Layer Insulations (MLI) can offer a performance increase of up to 600 times that of plain fiberglass. The principle behind MLI's is that of multiple layers of radiation shields separated by low conductance spacers. The shield material chosen should have a low (α/ε) ratio and normally consists of gold, silver, or aluminum foil, or mylar sheets coated with these same metals. The spacer material likewise can take many different forms. Most common are fiberglass cloth, fiberglass mat, and fiberglass paper. Table 6.2 lists the properties of the insulation that was chosen for the E-City. This particular system was chosen for its excellent insulation properties as well as compactness and low density; the additional mass of the insulation adds approximately 6.4% to the dry tank mass.

6.2.2 Active Thermal Control

As stated earlier, because of the tremendous amount of heat generated internally by the E-City, especially during engine start-up and shutdown, some system must be used to actively dissipate this heat. Research done in the Phase I report was utilized in the selection of the particular system, and work done in this report emphasized sizing the system.

There are two basic ways to dissipate thermal energy generated in the E-City: to reject the heat in the form of mass and jettison it overboard; and to emit the energy as a form of thermal radiation [1]. A mass expulsion system was deemed too heavy because of all of the excess mass that must be carried and so a radiator type system was chosen. Of the radiator systems, there are several different designs that make themselves suitable for space use. Several criteria that must be considered when choosing which radiator to use are: external environment; amount of waste heat to be rejected; radiator surface area; circulating fluid system; and micrometeoroid damage sensitivity. Ideally the radiator must not depend on surface area while minimizing the mass.

Two types of radiator that hold the most promise for our application are the liquid droplet radiator (LDR), and the rotating bubble membrane radiator (RBMR). Briefly, the LDR uses nozzles to spray molten metal onto a collector. As the metal droplets travel through space, between the spray nozzle and collector, they radiate their heat to space. The mass of the LDR is low because the metal droplets are the actual radiators and the majority of the mass is concentrated in the supporting structure. No protective shielding is needed because any meteoroids simply pass

6.13

right through the spray carrying some of the molten metal with it. The major disadvantage of the LDR is this loss of metal which also can occur if the spray nozzles are unable to maintain an accurate aim on the collector.

The system chosen, then, is the RBMR. It uses a two phase working fluid with an operating principle similar to the LDR. In this system the molten metal is sprayed onto an outer envelope or bubble. The droplets condense and radiate energy as they hit the bubble. By rotating the radiator, the metal droplets are collected in a trough by centrifugal force and recirculated again for reuse, see Figure 6.5. The advantages of using this type of radiator are its high heat capacity, relatively low mass, and any meteoroid passing through it would simply tear the outer membrane which could be easily repaired. A final consideration is that spray nozzle accuracy would not be a problem since the system is fully enclosed. Since it will be located on the despun portion of the space station, it will need some form of drive to spin it, but this should not entail a great mass penalty.

6.2.3 Analysis

Once the radiator configuration was decided upon, it was then necessary to arrive at a size and placement for it (or them should more than one prove to be needed). A computer program was written to first determine the amount of heat generated by the various E-City modes of operation and then to calculate the needed radiating surface area and the dimensions of such a system. The heat sources that were taken into account were the propulsion units and the power generation system. Details of these two systems were previously related, so there will be no detailed discussion of them here. Table 6.3 summarizes the program inputs and output data for the Phase II start-up procedure which represents the worst case.

The surface area, A, of the radiator envelope is a function of the waste heat, Q_r, and is found by,

$$A = (EQ_r) / \{ \varepsilon\sigma (T_r^4 - T_{sp}^4)\} \tag{6.4}$$

where E is the introduced error, T_r is the radiator surface temperature, and T_{sp} is the temperature of
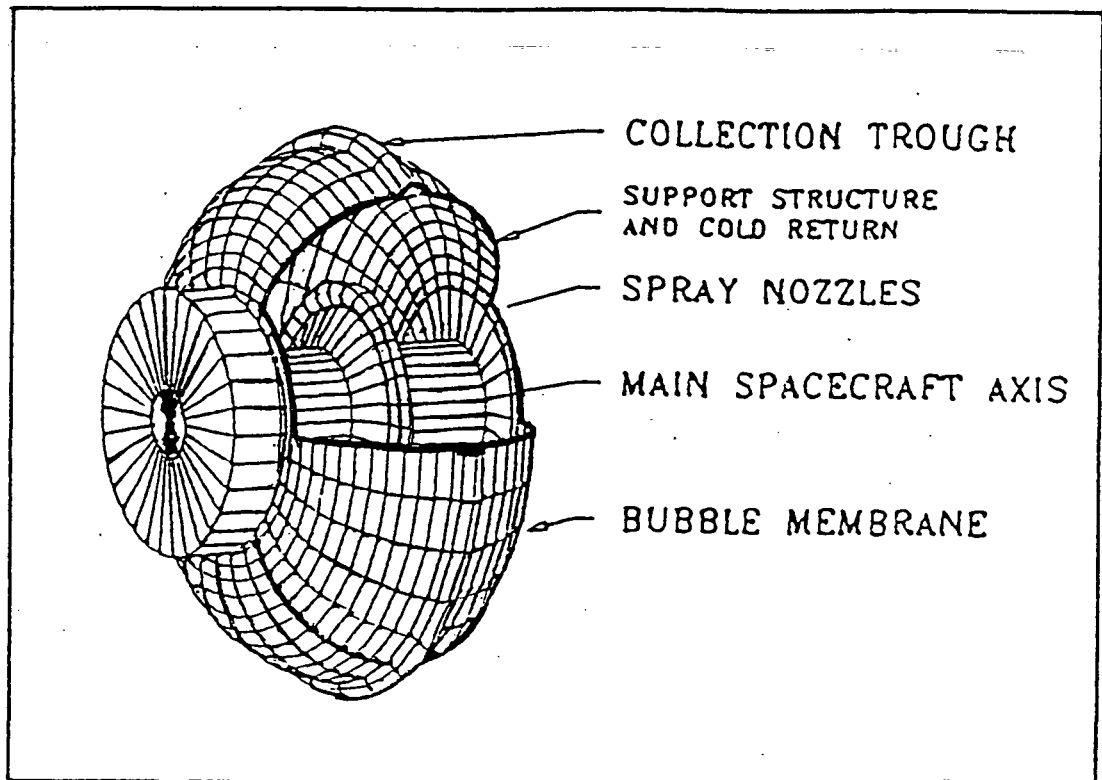
Figure 6.5: Rotating Bubble Membrane Radiator Cutaway

Power of main reactor (MWth)	5,000
Power conversion efficiency (%)	40
Power for ship ops (MWe)	22.85
Number of engines	14
Reactor power of start-up engine (MWth)	160,000
Reactor power of each engine (MWth)	160,000
Engine waste heat percentage	0.0
Waste heat from main reactor (MWth)	4,977
Waste heat from start-up engine (MWth)	97,600
Waste heat from main engines (MWth)	0.0
Total waste heat to dissipate (MWth)	102,577
Surface area needed for radiator (sq. m)	1,149,301
Length of a cylindrical radiator of radius 100 m (m)	1,830
Radius of a spherical radiator (m)	311
Major axis of an ellipsoid radiator w/ minor axis of 50 m (m)	431

Table 6.3: Active Thermal Control System Design Parameters for Phase II Start-up

space. For all calculations E was taken to be 5%, ε was given a value of 1.0, and T_{sp} was 0 K.

Using equation 6.4, the known reactor power and efficiencies of power conversion, the computer code RADIATOR was constructed to determine useful parameters for active thermal control and to arrive at a size of the radiator, see Appendix F. In this program, the inputs were made to account for all of the different combinations of power modes that the ship can operate under. The program asks for inputs such as: configuration and operating mode; reactor powers of power plant, propulsion engines, and control thrusters (if used); power conversion efficiencies; power required by the ship; and number of engines or thrusters. The program then calculates the total amount of waste heat that must be dissipated and the necessary radiating surface area. In addition, the output gives the dimensions for three possible geometric configurations corresponding to that surface area. The results summarized in Table 6.3, assume that nuclear attitude control thrusters will not be utilized due to their slow startup time, however should it become necessary to use them in the future, the radiator size will be greatly increased. It should also be pointed out that while the engines and reactor were assumed to be operating at full power at all times, it is theoretically possible to "throttle" the reactors and thus only a relatively small radiator would be needed.

6.3 Conclusion and Future Work

The largest source of heat transfer to the propellant tank that the E-City will experience is solar radiation while in a 1 AU orbit around the sun. It is critical that the cryogenic storage tank be maintained at an acceptably low temperature to minimize liquid hydrogen boiloff. This analysis has shown that effective means of insulation already exist which can be utilized on this design. The design chosen is rather conservative and can be readily improved as future requirements become apparent. A major area of study that will need to be looked at as the overall station design matures, is that of heat conduction through structural members into the tank and developing a more detailed thermal model. In Saturn's orbit, for example, the solar intensity is only 1.1% that of Earth's orbit and in this case the dominant factor would probably prove to be internal heat conduction. The MLI is not a structural component and even slight compression loads adversely affect its performance, therefore it will be necessary to design attachment points with low thermal

6.16

conductivity characteristics. The ultimate solution may even be to tailor the heat input such that the boiloff rate exactly matches any need the station may have for gaseous hydrogen. A more detailed thermal analysis would take into account other heat sources such as that from shipboard electrical equipment and life support systems. A smaller, low heat capacity, secondary control system will need to be designed to handle this aspect.

For the active control systems, specific materials must be chosen for both the membrane and the molten metal. A membrane skin temperature of 1200 K was chosen because it represents the upper temperature limit of current gas turbine engine components. These normally utilize a titanium alloy and it was felt that these were a realistic option. It may prove possible to develop a high capacity heat storage system to absorb thermal shocks, such as during the relatively short duration startup procedure, thus allowing a smaller radiator.
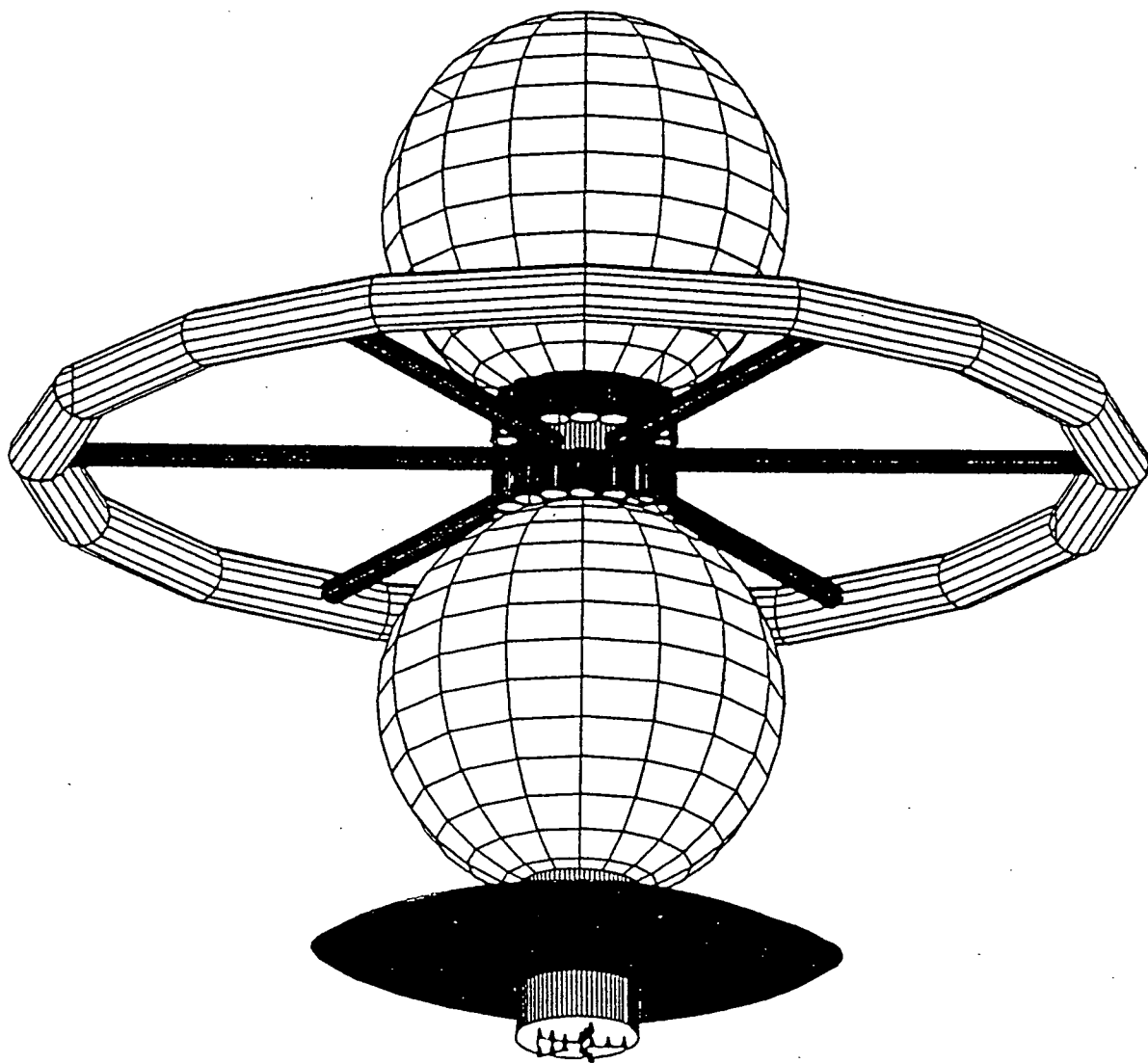
```
      !
      !
      !
      program NLB
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      ! This program calculates the necessary performance parameters
      ! for the Closed-Cycle, Nuclear Gas Core Engine. With the user
      ! input value of engine thrust, this program will generate
      ! values of specific impulse,total engine mass,engine power,
      ! radiating fuel temperature, seeded hydrogen flow rates,
      ! chamber pressure, estimated exhaust gas temperature, and
      ! reactor power required for the rocket engine.
      ! The calculation of these parameters are based on polynomial
      ! curve approximations and interpolations of performance
      ! characteristics given by T. Latham, UTRC, c/o Stan Borowski, NASA
      ! Lewis Research Center.
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      implicit none
      real g,thr,isp,mass,temp,tw,power,pdot,spwr
      real press,rp,exittemp,exvel,est,error,thrust
      integer i,igo,iread,iostat,irep
      open(unit=3,file='NLB.DAT',status='new')
      !Set constants
      g=9.81
      !Common message/ Header format stack.
1     format(8X,'Please specify the desired thrust for the engine
     * within'/8X,'the ranges of 98000 and 9408000 Newtons.')
2     format(8X,'This value is unreadable.')
3     format(8X,'This value is out of range and cannot be used
     * by the program.'/8X,'Please Try Again.')
4     format(/8X,'Would you like to run the program again for
     * another'/8X,'thrust value? (Press "1" for "YES", "2" for
     * "NO".)')
      !Repeat Option Do-Loop.
      irep=1
      dowhile(irep.eq.1)
      !User input sequence.
      igo=1
      dowhile(igo.eq.1)
       write(6,1)
       read(6,*,iostat=iread) thr
       !Data unreadable/mistype check.
       if(iread.gt.0) then
        write(6,2)
       endif
       !If successful read,then check range.
       if(thr.lt.98000.0.or.thr.gt.9408000.0) then
        write(6,3)
       else
        igo=0
       endif
      enddo
      !Calculation Sequence.
```

```fortran
      !Because Temperature Curve Approximations are more
      !accurate than thrust, interpolation will be used
      !to find temperature. Then, the value of temperature
      !will be used to find the rest of the engine parameters.
      !This sequence refines the estimate.
      error=1.0
      if(thr.eq.98000.0) then
      temp=5000.0
      error=1E-6
      elseif(thr.gt.98000.0.and.thr.lt.245000.0) then
      est=5000.0
      elseif(thr.eq.245000.0) then
      temp=7000.0
      error=1E-6
      elseif(thr.gt.245000.0.and.thr.lt.472000.0) then
      est=7000.0
      elseif(thr.eq.784000.0) then
      temp=8333.0
      error=1E-6
      elseif(thr.gt.784000.0.and.thr.lt.784000.0) then
      est=8333.0
      elseif(thr.eq.784000.0) then
      temp=10000.0
      error=1E-6
      elseif(thr.gt.784000.0.and.thr.lt.1470000.0) then
      est=10000.0
      elseif(thr.eq.1470000.0) then
      temp=12000.0
      error=1E-6
      elseif(thr.gt.1470000.0.and.thr.lt.3136000.0) then
      est=12000.0
      elseif(thr.eq.3136000.0) then
      temp=15000.0
      error=1E-6
      elseif(thr.gt.3136000.0.and.thr.lt.9408000.0) then
      est=15000.0
      elseif(thr.eq.9408000.0) then
      temp=20000.0
      error=1E-6
      endif

      dowhile(abs(error.ge.1E-3))
      est=est*1.0001
      thrust=(-1108.6+.42794*est-5.2624D-5*est**2.0+
     *2.8752D-9*est**3.0)*1000.0
      error=thr-thrust
      temp=est
      enddo

      !Calculate Reactor Power
      if(temp.ge.11500.0.and.temp.le.13000.0) then
      rp=-1.1914E+4+4.9912*temp-7.3384E-4*temp**2.0+4.5685D-8*temp**3.0
      else
      rp=-1.1523E+4+4.9533*temp-7.4010E-4*temp**2.0+4.6050D-8*temp**3.0
      endif

      !Calculate Chamber Pressure.
      press=179.13-1.5744E-3*temp+6.4590D-6*temp**2.0-1.7898D-10*temp**3.0

      !Calculate Isp as a function of Chamber Pressure.
```

```fortran
      !(Highest Accuracy)
      isp=-924.66+8.6019*press-7.3704E-3*press**2.0+2.4046D-6*press**3.0


      !Calculate Propellant Flow Rate.
      pdot=-13.989+7.4957E-3*temp-9.3089D-7*temp**2.0+6.8234D-11*temp**3.0

      !Thrust-to-Weight Ratio.
      if(temp.ge.5000.0.and.temp.le.12000.0) then
      tw=-1.3609E-2+9.0717D-5*temp-1.8375D-8*temp**2.0+3.2260D-12*temp**3.0
      else
      tw=-14.950+2.3092E-3*temp-6.0833D-8*temp**2.0
      endif

      !Calculate Engine Mass.
      !First Find Weight, then divide by g.
      mass=thr/tw/g

      !Calculate Specific Power.
      spwr=1000.0*rp/mass

      !Find Exit Temperature (Estimated)
      exittemp=-55.336+.80286*temp

      !Find Exit Velocity
      exvel=isp*g

      write(6,100) thr
      write(3,100) thr
100   format(/8X,'Nuclear Light Bulb Engine Performance Characteristics'
     */8X,'for the Specified Thrust of',F12.2,' Newtons.'/)
      write(6,101) rp,spwr,temp,isp,mass,pdot,tw
      write(3,101) rp,spwr,temp,isp,mass,pdot,tw
      write(6,102) press,exittemp,exvel
      write(3,102) press,exittemp,exvel
101   format(8X,'Reactor Power:',F12.2,' MegaWatts'/8X,
     *'Specific Power:',F9.2,' Kilowatts per Kilogram'/8X,
     *'Uranium Radiating Temperature:',F10.2,' Kelvin'/8X,
     *'Specific Impulse:',F6.1,' Seconds'/8X,'Engine Mass:',
     *F12.2,' Kilograms'/8X,'Propellant Flow Rate:',F8.2,
     *' Kilograms per Second'/8X,'Thrust to Weight Ratio:'
     *,F5.2)
102   format(8X,'Chamber Pressure:',F7.2,' Atmospheres'/
     *8X,'Estimated Exhaust Temperature:',F9.2,' Kelvin'/
     *8X,'Exhaust Velocity:',F12.2,' Meters per Second')
      !Repeat Option Sequence.
      igo=1
      dowhile(igo.eq.1)
      write(6,4)
      read(6,*,iostat=iread) irep
       !Data unreadable/mistype check.
       if(iread.gt.0) then
        write(6,2)
       elseif(irep.eq.1) then
        igo=0
       elseif(irep.ne.1) then
        irep=0
        igo=0
       endif
      !End Repeat Option Loop.
```

```
enddo
!End Main Program Loop.
enddo
close(unit=3)
stop
end
```

```
      !           NLB PARAMETERS FOR RADIATING TEMPERATURE RANGE
      !              OF 5,000 TO 20,0000 KELVIN


      program NLBDATA
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      !
      !This program is a variation of NLB.FOR. Using the same
      !calculation techniques as in NLB, this program will
      !calculate thrust, specific impulse, engine mass,
      !propellant flow rate, thrust to weight ratio, engine power,
      !specific power, and other performance parameters within the
      !uranium radiating temperature range of 5000 to 20000 Kelvin.
      !The output values will then be sent to the screen for review.
      !Because of the number of parameters calculated, the data
      !will be sent to two datafiles: NLBDATA1.DAT, and NLBDATA2.DAT,
      !so that hard copies of the ouput generated by this program
      !may be obtained.
      !
      !Source information provided by Thomas Latham, of United
      !Technologies Research Corporation, c/o Stan Borowski,
      !NASA-Lewis Research Center.
      !
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      implicit none
      real g,thr,isp,mass,temp,tw,power,pdot,spwr
      real press,extemp,weight,exvel
      integer i
      open(unit=7,file='NLBDATA1.DAT',status='new')
      open(unit=12,file='NLBDATA2.DAT',status='new')
      !Set constants
      g=9.81
      write(6,1)
      write(7,1)
      write(12,1)
1     format(8X,'Nuclear Light Bulb Engine Performance Characteristics'
     */8X,'for the Radiating Fuel Temperature Range of 5000 to'/
     *8X,'20000 Kelvin.'/)
      write(6,2)
      write(7,2)
2     format(2X,'Temp.',3X,'Thrust',3X,'Isp',4X,'Mass',8X
     *,'Pdot',3X,'TWRatio')
      write(6,3)
      write(7,3)
3     format(2X,'(K)',6X,'(KN)',4X,'(s)',4X,'(Kg)',7X,'(kg/s)'/)
      write(12,4)
      write(12,5)
4     format(1X,'Temp.',1X,'Reactor Power',1X,'Spec. Pwr.',
     *1X,'Chmbr. Press.',1X,'Est. Ex. Temp.',1X,'Ex. Vel.')
5     format(3X,'(K)',7X,'(MW)',8X,'(KW/Kg)',7X,'(atm)',9X,'(K)'
     *,6X,'(m/s)'/)
      g=9.81
      temp=4500.0
      !Begin calculations for the temperature range of 5000 to 20000 K.
      do i=1,31
      temp=temp+500.0
```

```fortran
      !Calculate Specific Impulse. via chamber pressure
      !Chamber pressure via temperature
      !Polynomial Approximation, Order:3, R**2=1.00
      press=179.13-1.5744E-3*temp+6.4590D-6*temp**2-1.7898D-10*temp**3
      isp=-924.66+8.6109*press-7.3704E-3*press**2+2.4046D-6*press**3


      !Calculate Reactor Power.
      !PA,O3,R**2=1.0
      if(temp.ge.11500.and.temp.le.13000) then
      power=-1.1914E4+4.9912*temp-7.3384E-4*temp**2+4.5685D-8*temp**3
      else
      power=-1.1523E4+4.9533*temp-7.401E-4*temp**2+4.605D-8*temp**3
      endif

      !Calculate Thrust-to-Weight Ratio.
      !PA,O3,R**2=1.0
      if(temp.le.12000) then
      tw=-1.3609E-2+9.0717D-5*temp-1.8375D-8*temp**2+3.226D-12*temp**3
      else
      tw=-14.95+2.3092E-3*temp-6.0833D-8*temp**2
      endif


      !Calculate propellant flow rate.
      pdot=-13.989+7.4957E-3*temp-9.3089D-7*temp**2+6.8234D-11*temp**3

      !Calculate Engine Thrust.
      if(temp.ge.5000.0.and.temp.le.13000.0) then
       thr=pdot*isp*g/1000.0
      else
      !Polynomial Approximation, Order: 3, R**2=1.0
      thr=-1108.6+0.42794*temp-5.2624D-5*temp**2+2.8752D-9*temp**3
      endif

      !Calculate Engine Weight, (kN)
      weight=thr*(1/tw)

      !Calculate Engine Mass, (kg)
      mass=weight/g*1000

      !Calculate Specific Power
      spwr=power/mass*1000.0

      !Calculate Exit Velocity.
      exvel=isp*g

      !Estimated Exit Temperature
      !PA: Order 1, R**2=1.00
      extemp=-55.336+0.80286*temp

      write(6,100) temp,thr,isp,mass,pdot,tw
      write(7,100) temp,thr,isp,mass,pdot,tw
      write(12,101) temp,power,spwr,press,extemp,exvel
100   format(1X,F6.0,F9.0,2X,F5.0,1X,F10.2,3X,
     *F6.2,2X,F5.2)
101   format(1X,F6.0,2X,F8.0,6X,F7.2,7X,F7.1,4X,
     *F7.1,F12.2)
      enddo
```

```
close(unit=7)
close(unit=12)
stop
end
```

MISSION, SYSTEM, and CONFIGURATION ANALYSIS

CONFIGURATION PROGRAM and DATA

```
C###############################################################################
C
C      ECITY.FOR              MicroVax Fortran
C      May 14, 1992   v.4
C
C      Computes the configuration of The Emerald City based on required
C      mission parameters.
C
C###############################################################################
C23456789012345678901234567890123456789012345678901234567890123456789012345678901
C        1         2         3         4         5         6         7         8
C----#--------------------------------------------------------------------#--------
       program EMERALD
       implicit NONE
       logical  LOOP, MORE
       character BEL *1 /7/
       character CASE*50 /'  '/
       integer I, P, Neng, N, NT, NS
C
       common /CONST/PI,ag /ENG/DV,Isp,Tpr,Mi,Fi
      +       /GEN/RPM,G,Vliv,Patm,R1,r2
      +       /PROP/ Dh2, Dsh, Dtor, WStor, Dtank, WStank
C
       real     DV, Isp, Tpr, Mi, Fi, Meng, THRUST
       real     RPM, G, Vliv, Patm, R1, r2
       real     Nvol, Mbio, Ator, Mtor, Mtorl, Ttor, Pext, VELtor
       real     Dh2, Dsh, Dtor, WStor, Dtank, WStank
       real     Mdry, MdryNEW, Mtank, Mpay, Msh, Mstr, Mh2
       real     M0, Ffuel, Fstr, Fpay
       real     PI, ag, Gmax, Gmin, ACCf, ACCe
       real     OD, Vsh, MPP, MpoleT, MpoleB
       real     AR, Rp, Hp, Ttank, Vh2, Ptank, Atank
       real     FTTf,FTTe,FBTPf,FBTPe,FBTTf,FBTTe,FBTBf,FBTBe,MSf,MSe
       real     r3, Lpole
C
       open  (1, file= 'ECITY.IN' ,status='OLD', err=800)
       open  (2, file= 'ECITY.OUT',status='NEW')
       open  (3, file= 'EFORCES.DAT', status= 'NEW')
C
       PI  = 3.141592654
       ag  = 9.81                         ! accel of gravity (on earth)
       N=0
       NS=4
C----#------------------------------------------ INPUT ----------------#--------
       DO 20 WHILE (.true.)
       N=N+1
C
       read (1,*,err=810,end=850)
       read (1,*,err=810,end=830)                        ! read past header.
       read (1,*,err=810,end=830)         CASE           ! case title, 50 char
       read (1,*,err=810,end=830) RPM, G, Nvol, Patm, Mtorl
       read (1,*,err=810,end=830) P, MPP, Mpay, Mstr
       read (1,*,err=810,end=830) DV, Isp, Fi, Mi, Tpr
       read (1,*,err=810,end=830) Dh2, Dsh, Dtor, WStor
       read (1,*,err=810,end=830) Ptank, Dtank, WStank, NT
       read (1,*,err=810,end=830) r3, Lpole
C
       write (2,200) N, CASE               ! header & description
C
       Vliv = P*Nvol
```

```fortran
      Tpr = tpr*60*60*24          ! convert days to seconds...
      Ptank = Ptank * 101325      ! convert atm to N/m^2
      Mbio = P * MPP              ! bio mass based on population
C----#------------------------------------------------------ TORUS MASS --#--------
      call TORUS (Mbio,Ator,Mtor,Ttor,Pext)
      call SHIELD (Vsh, Msh, OD)
      if (Mtor1.gt.0.) Mtor = Mtor1    ! ability to set torus mass manually
      Mtor = Mtor + Mbio          !
      Mdry = Mtor+Msh+Mpay+Mstr
C----#------------------------------------------------------ MAIN LOOP --#--------
      LOOP = .TRUE.
      do 10 while (LOOP.eq..true.)
         call ENGINE (Neng, Meng, Mdry, THRUST)
         Mh2 = Mdry*(exp(DV/Isp/ag)-1.)         ! eq 2.2
         Vh2 = Mh2/Dh2/NT                       ! density definition
         Vh2 = Vh2 + Vh2*.08                    ! 8% ullage
         ACCf = THRUST/(Mdry + Mh2)
         call TANK (Vh2, AR, Rp, Hp, Ttank, Mtank, Atank, ACCf, Ptank)
         MdryNEW = Mtor+Msh+Meng+Mtank*NT+Mpay+Mstr
         if (ABS(MdryNEW - Mdry).le.1000.) LOOP=.FALSE.
         Mdry = MdryNEW
 10   continue
C----#------------------------------------------------------ FINAL CALCULATIONS --#--------
      M0  = Mh2 + Mdry
      Ffuel = Mh2/M0
      Fstr  = (Mdry-Mpay-(P*250.))/M0    ! allow 250 kg/person as payload
      Fpay  = (Mpay+(P*250.))/M0
      Gmax= (RPM*PI/30.)**2.*(R1+r2)/ag   ! eq 3.4a
      Gmin= (RPM*PI/30.)**2.*(R1-r2)/ag   ! eq 3.4a
      VELtor = RPM*PI/30.*(R1+r2)         ! eq 3.5
      ACCe = THRUST/Mdry
C----#------------------------------------------------------ FORCE CALCULATIONS --#--------
C Symbols: first letter is Force or Moment
C          TT=Top Tank; BTT=Bottom Tank Top; BTB= Bottom Tank Bottom
C          e & f correspond to empty and full acceleration cases
C          MS= Spoke Moments @ pole; NS=Number of Spokes; r3=pole radius
C
      FTTe = ACCe*Mtank
      FTTf = ACCf*(Mtank+Mh2/NT)
      FBTTe = ACCe*(Mtank+Mtor+Msh+Mpay+Mstr)
      FBTTf = ACCf*(Mtank+Mh2/NT+Mtor+Msh++Mpay+Mstr)
      FBTBe = ACCe*(Mtank*NT+Mtor+Msh+Mpay+Mstr+MpoleT+MpoleB)
      FBTBf = ACCf*(Mtank*NT+Mh2+Mtor+Msh+Mpay+Mstr+MpoleT+MpoleB)
      MSe = ACCe*((Mtor+Msh)/NS*(R1-r2-r3))
      MSf = ACCf*((Mtor+Msh)/NS*(R1-r2-r3))
C----#------------------------------------------------------ OUTPUT --#--------
      write (2,212)
      write (2,250) P, G, RPM, Gmax, Gmin, VELtor, Ptank
      write (2,214)
      write (2,260) DV/1E3, Isp, Fi/1E3, Tpr/60/60/24, Neng, ACCf, ACCe
      write (2,216)
      write (2,270) (Mtor-Mbio)/1E6, Mtor/1E6,
     +        Mpay/1E6, Msh/1E6, Meng/1E6,
     +        Mtank/1E6*NT, Mstr/1E6, Mdry/1E6, Mh2/1E6, (Mh2+Mdry)/1E6
      write (2,220)
      write (2,290) Ffuel, Fstr, Fpay
      write (2,218)
      write (2,280) R1, R2, OD/2., NT, Rp, Vh2/1E6*NT, Atank/1E3,
     +              Ttank
      write (2,210)
```

B.2

```
C---->
      write (3,400) N, CASE
      write (3,410) ACCf,ACCe,FTTf/1E6,FTTe/1E6,FBTPf/1E6,FBTPe/1E6,
     +           FBTTf/1E6,FBTTe/1E6,FBTBf/1E6,FBTBe/1E6,MSf/1E6,MSe/1E6
C---->
C     write (2,205)                     ! NEW PAGE
C---->
 20   continue                          ! main loop
C----#-------------------------------------------------- ERROR TRAPS --#--------
      if (.false.) then
 800    write (6,*) '           Cannot find ECITY.DAT '
        write (*,*) BEL
      elseif (.false.) then
 810    write (6,*) '           Bad data in ECITY.IN '
        write (6,*) '           Data set ', N
        write (*,*) BEL
      elseif (.false.) then
 830    write (6,*) '           Incomplete data set in ECITY.IN '
        write (6,*) '           Data set ', N
        write (*,*) BEL
      elseif (.false.) then
 850    write (6,*) '       END OF DATA. '
      endif
 900  write (*,*) BEL
      stop '      I M DUN.           Output in file ECITY.OUT'
C----#---------------------------------------------------------- FORMAT --#--------
 200  format ('1',/4X, 13('#####'),//27X, 'EMERALD CITY CONFIGURATION',
     +        '      (Data Set ',I3,')', //5X, 13('====='),
     +        /10X, A50, /2X)
 205  format ('1')                      ! NEW PAGE
 210  format (5X, 13('-----'))
 212  format (/5X, 'GENERAL PARAMETERS -', 9('-----'))
 214  format (/5X, 'PROPULSION ----',        10('-----'))
 216  format (/5X, 'SYSTEM MASSES -',        10('-----'))
 218  format (/5X, 'DIMENSIONS ----',        10('-----'))
 220  format (/5X, 'MASS FRACTIONS ',        10('-----'))
C----#-----------------------------------------------------------------#--------
 250  format (30X, 'Population ',                   I5,
     + /14X, 'Artificial Gravity Desired      ',F4.2,
     +        /31X, 'Spin Rate       ',            F5.2, '  RPM',
     +        /24X, 'Max Artificial G      ',      F4.2,
     +        /24X, 'Min Artificial G      ',      F4.2,
     +        /15X, 'Max Torus linear velocity ',F7.1, '   m/sec',
     +        /27X, 'Tank Pressure ',              F7.0, '    Pa')
C
C----#-----------------------------------------------------------------#--------
 260  format (18X, 'Total Delta-V Required    ', F6.2, '  km/sec',
     +        /37X, 'Isp  ',                      F5.0, '   seconds',
     +        /23X, 'Thrust per Engine ',         F7.1, '   kN',
     +        /23X, 'Total Thrust Time    ',      F5.1, '    days',
     +        /23X, 'Number of Engines ',            I6,
     +        /14X, 'Acceleration: Full H2 load ',F10.3, ' m/s^2',
     +        /14X, 'Acceleration:      Dry ',F10.3, ' m/s^2')
C
C----#-----------------------------------------------------------------#--------
 270  format (5X,   '(1 million kg = 1000 metric tons)',
     +        /35X, 'Torus ',                     F10.3, ' Million kg',
     +        /26X, 'Torus + Insides',            F10.3, ' Million kg',
     +        /33X, 'Payload ',                   F10.3, ' Million kg',
     +        /21X, 'Torus Cosmic Shield ',       F10.3, ' Million kg',
```

```
          +          /28X, 'Total Engine ',              F10.3, ' Million kg',
          +          /25X, 'Propellant Tank ',           F10.3, ' Million kg',
          +          /24X, 'Other Structural ',          F10.3, ' Million kg',
          +          /24X, 11('----'),
          +          /32X, 'DRY MASS ',                   F10.3, ' Million kg',
          +          //25X, 'PROPELLANT MASS ',           F10.3, ' Million kg',
          +          /30X, 'TOTAL MASS ',                 F10.3, ' Million kg')
      C
      C----#-----------------------------------------------------------------#--------
       280  format (5X,   '(in meters unless noted)',
          +          /22X, 'Torus Major Radius ',         F8.1,
          +          /23X, 'Torus Tube Radius ',          F8.1,
          +          /14X, 'Torus & Shield Tube Radius ',F8.1,
          +          //25X, 'Number of Tanks    ',        I4,
          +          /18X, 'Propellant Tank Radius ',     F8.1,
          +    /12X, 'Total Propellant Tank Volume ',   F10.3, ' Million m^3',
          +      /6X, 'Propellant Tank Surface Area (each)',
          +                                              F10.3, ' Thousand m^2',
          +    /15X, 'Propellant Tank Thickness ',      F10.3 )
      C
      C----#-----------------------------------------------------------------#--------
       290  format (30X, 'Propellant ',                  F11.9,
          +          /31X, 'Structure ',                  F11.9,
          +          /33X, 'Payload ',                    F11.9  )
      C
      C----#-----------------------------------------------------------------#--------
       400  format (2X, /2X, 15('*****'),
          +          /10X, 'EFORCES.DAT      (Data Set ',I3,')',
          +          /10X, A50, /2X, /27X, 'Full Load',7X,'Burn Out',
          +          /5X, 13('-----'))
       410  format (6X,   'Acceleration (m/s^2)', F10.3, 5X, F10.3,
          +          //2X,'FORCES: ', 42X, '(millions of N)',
          +          /8X,      'Bottom of top tank', F10.3, 5X, F10.3,
          +          /8X,      'Bottom of top pole', F10.3, 5X, F10.3,
          +          /8X,      'Top of bottom tank', F10.3, 5X, F10.3,
          +          /5X, 'Bottom of bottom tank', F10.3, 5X, F10.3,
          +          //2X, 'MOMENTS:', 42X, '(millions of N-m)',
          +          /6X,   'Spoke-pole interface', F10.3, 5X, F10.3)
      C
      C---->
            end


      C################################################################################
      C
      C     ETORUS.SUB                          VAX Fortran
      C     Feb. 6, 1992
      C
      C     Calculates the properties of the torus and returns values.
      C
      C################################################################################
      C23456789012345678901234567890123456789012345678901234567890123456789012345678901
      C         1         2         3         4         5         6         7         8
      C----#-----------------------------------------------------------------#--------
            subroutine TORUS (B, AREA, MASS, TH,  Pout)
            implicit   NONE
            common /CONST/ PI, ag  /GEN/ RPM, G, Vliv, Patm, R1, r2
          +        /PROP/ Dh2, Dsh, DEN, WS, Dtank, WStank
            real    PI, ag
            real    RPM, G, Vliv, Patm, R1, r2
            real    Dh2, Dsh, DEN, WS, Dtank, WStank
            real    B, AREA, MASS, TH, Pout
      C----#-----------------------------------------------------------------#--------
            R1  = ag*G/((RPM*PI/30.)**(2.))
            r2  = SQRT(Vliv/2./PI**2./R1)
            AREA= 4.*PI**2.*R1*r2
            TH  = (Patm*r2/2./R1 + B*2./PI/AREA)*R1/(WS-DEN*R1)      ! NASA SP-413
            MASS= AREA * TH * DEN
      C----#-----------------------------------------------------------------#--------
            return
            end
```

```fortran
C#################################################################################
C
C     EENGINE.SUB                          VAX Fortran
C     Feb. 10, 1992
C
C     Calculates the number of engines and total propulsion system mass
C     based on delta-V and propulsion time.
C
C#################################################################################
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C        1         2         3         4         5         6         7         8
C----#------------------------------------------------------------------#--------
      subroutine ENGINE (Neng, Meng, Mdry, THRUST)
      implicit NONE
      common    /CONST/ PI, ag     /ENG/ DV, Isp, Tpr, Mi, Fi
      integer Neng
      real    PI, ag
      real    DV, Isp, Tpr, Mi, Fi
      real    N, Mdry, Meng, mfuel, THRUST
C----#------------------------------------------------------------------#--------
      N = Mdry*(exp(DV/Isp/ag)-1.)*Isp*ag/Fi/Tpr
      if ((N-ABS(N)).gt..2) then        ! one more engine of remainder > 20%
         Neng = N+1
      else
         Neng = N
      endif
      Meng = Mi*Neng
      THRUST = Fi * Neng
C----#------------------------------------------------------------------#--------
      return
      end


C#################################################################################
C
C     ESHIELD.SUB                          VAX Fortran
C     Feb. 10, 1992
C
C     Computes the mass and dimensions of the cosmic ray shield for the
C     Emerald City
C#################################################################################
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C        1         2         3         4         5         6         7         8
C----#------------------------------------------------------------------#--------
      subroutine SHIELD (VOL, MASS, OD)
      implicit NONE
      common /CONST/ PI, ag    /GEN/ RPM, G, Vliv, Patm, R1, r2
     +       /PROP/ Dh2, DEN, Dtor, WStor, Dtnk, WStnk
      real   PI, ag
      real   RPM, G, Vliv, Patm, R1, r2
      real   Dh2, DEN, Dtor, WStor, Dtnk, WStnk
      real   VOL, MASS, OD, TH
      real   THi, THo, AREAi, AREAo
      TH   = 14
C----#------------------------------------------------------------------#--------
C     Mass based on 14m of liquid hydrogen
C----#------------------------------------------------------------------#--------
      VOL   = 2.*PI**2.*R1*((R2+TH)**2.-R2**2.)
      OD    = 2.*(R2+TH)
      THi   = 20265./2.*r2/(WStnk-Dtnk*R1)
      THo   = 20265./2.*(r2+TH)/(WStnk-Dtnk*R1)
      AREAi = 4.*PI**2.*R1*r2
      AREAo = 4.*PI**2.*R1*(r2+TH)
      MASS = DEN*VOL + AREAi*THi*Dtnk + AREAo*THo*Dtnk
C----#------------------------------------------------------------------#--------
      return
      end
```

```
C##############################################################################
C
C     ETANK.SUB                            VAX Fortran
C     Feb. 16, 1992
C     Feb. 24, 1992
C
C     Computes the mass and dimensions of the propellant tank(s) for the
C     Emerald City
C##############################################################################
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
C     1         2         3         4         5         6         7         8
C----#-------------------------------------------------------------#--------
      subroutine TANK (VOL, AR, Rp, Hp, TH, MASS, AREA, ACEL, HEAD)
      implicit NONE
      common    /CONST/ PI, ag
     +          /PROP/ Dh2, Dsh, Dtor, WStor, DEN, WS
      real      PI, ag
      real      Dh2, Dsh, Dtor, WStor, DEN, WS
      real      VOL, AR, Rp, Hp, TH, MASS, AREA, ACEL, HEAD
      real      H, AREAs, AREAc, THtop, THcyl, THbot, HHp
C----#-------------------------------------------------------------#--------
C     Spherical end caps assumed.
C     Thickness computed for each section; bottom, cyl, top
C----#-------------------------------------------------------------#--------
      if (AR.lt.1.) AR = 1.                 ! can't have less than a sphere.
      Rp  = (VOL/PI/(2.*AR-2./3.))**(1./3.)     ! eq 5.3    propellant radius
      Hp  = 2.*AR*Rp
      H   = 2.*Rp*(AR-1.)                   ! height of cylindrical portion
      AREAs = 2.*PI*Rp**2.                  ! half of sphere
      AREAc = 2.*PI*Rp*h                    ! Cylinder
      THcyl = (HEAD + Dh2*ACEL*(Hp-Rp))*Rp/WS    ! t=pr/sigma
      THbot = (HEAD + Dh2*ACEL*Hp)*Rp/WS/2.      ! t=pr/2sigma
      HHp = Hp - Rp - H
      if (HHp.lt.0.) HHp = Hp-Rp
      THtop = (HEAD + Dh2*ACEL*HHp)*Rp/WS/2.     ! t=pr/2sigma
      MASS = DEN*(AREAs*THtop + AREAs*THbot + AREAc*THcyl)
      AREA = 2.*AREAs + AREAc
      TH = THbot
      if (AR.ne.1.) then
         if ((THcyl.ge.THbot).and.(THcyl.ge.THtop)) TH = THcyl
         if ((THtop.ge.THbot).and.(THtop.ge.THcyl)) TH = THtop
      else
         if (THtop.ge.THbot) TH = THtop
      endif
C----#-------------------------------------------------------------#--------
      return
      end
```

B.6

```
########## ECITY.IN     data set 1    ############################
----------*----------*----------*----------*----------*------------
'Saturn Mission; Pre-Optimization'
1,  1,   19000,   101325, 0            ! RPM, Grav, Nvol, Patm, Mtorl
1000, 53000, 1000, 000000              ! POP, Mass/person, Mpay, Mstr
50000, 3095, 9402000, 138886.81,  20   ! DV, Isp, Fi, Mi, Tpr
71.,   71.,    2650,   165000000       ! Dh2, Dsh, Dtor, WStor
.2,    2650,   165000000, 2            ! Ptank, Dtank, WStank, # Tanks
100,   200,    6                       ! Shaft r, Shaft L, # Spokes
########## ECITY.IN     data set 2    ############################
----------*----------*----------*----------*----------*------------
'Saturn Mission; Pre-Optimization,   FAT SPOKES & SHAFT'
1, 1,   19000,   101325, 0             ! RPM, Grav, Nvol, Patm, Mtorl
1000, 53000, 1000, 2279400000          ! POP, Mass/person, Mpay, Mstr
50000, 3095, 9402000, 138886.81,  20   ! DV, Isp, Fi, Mi, Tpr
71.,   71.,    2650,   165000000       ! Dh2, Dsh, Dtor, WStor
.2,    2650,   165000000, 2            ! Ptank, Dtank, WStank, # Tanks
100,   200,    6                       ! Shaft r, Shaft L, # Spokes
########## ECITY.IN     data set 3    ############################
----------*----------*----------*----------*----------*------------
'Saturn Mission;  OPTIMIZED VERSION'
1,  1,   19000,   101325, 185006295    ! RPM, Grav, Nvol, Patm, Mtorl
1000, 53000, 1000, 125608671           ! POP, Mass/person, Mpay, Mstr
50000, 3095, 9402000, 138886.81,  20   ! DV, Isp, Fi, Mi, Tpr
71.,   71.,    2650,   165000000       ! Dh2, Dsh, Dtor, WStor
.2,    2650,   165000000, 2            ! Ptank, Dtank, WStank, # Tanks
100,   200,    6                       ! Shaft r, Shaft L, # Spokes
```

```
###################################################################

                    EMERALD CITY CONFIGURATION      (Data Set   1)

=================================================================
          Saturn Mission; Pre-Optimization


    GENERAL PARAMETERS -----------------------------------------
                         Population   1000
             Artificial Gravity Desired   1.00
                         Spin Rate   1.00   RPM
                 Max Artificial G   1.04
                 Min Artificial G   0.96
        Max Torus linear velocity   97.1   m/sec
                    Tank Pressure   20265.   Pa

    PROPULSION -------------------------------------------------
              Total Delta-V Required   50.00   km/sec
                            Isp   3095.   seconds
              Thrust per Engine   9402.0   kN
              Total Thrust Time   20.0   days
              Number of Engines   12
         Acceleration: Full H2 load   0.014 m/s^2
         Acceleration:       Dry   0.073 m/s^2

    SYSTEM MASSES ----------------------------------------------
    (1 million kg = 1000 metric tons)
                            Torus   31.862 Million kg
                 Torus + Insides   84.862 Million kg
                         Payload   0.001 Million kg
             Torus Cosmic Shield   1416.275 Million kg
                    Total Engine   1.667 Million kg
                 Propellant Tank   49.102 Million kg
                 Other Structural   0.000 Million kg
                 ------------------------------------------
                         DRY MASS   1551.907 Million kg

                 PROPELLANT MASS   6503.006 Million kg
                      TOTAL MASS   8054.913 Million kg

    MASS FRACTIONS ---------------------------------------------
                    Propellant 0.807334125
                    Structure 0.192634687
                      Payload 0.000031161

    DIMENSIONS -------------------------------------------------
    (in meters unless noted)
                 Torus Major Radius   894.6
                  Torus Tube Radius   32.8
        Torus & Shield Tube Radius   46.8

                  Number of Tanks   2
             Propellant Tank Radius   227.7
         Total Propellant Tank Volume   98.919 Million m^3
    Propellant Tank Surface Area (each)   651.604 Thousand m^2
             Propellant Tank Thickness   0.014
    -----------------------------------------------------------
```

```
###################################################################

                    EMERALD CITY CONFIGURATION     (Data Set   2)


==================================================================
          Saturn Mission; Pre-Optimization,  FAT SPOKES & SH


   GENERAL PARAMETERS --------------------------------------------
                          Population   1000
             Artificial Gravity Desired    1.00
                          Spin Rate    1.00    RPM
                   Max Artificial G    1.04
                   Min Artificial G    0.96
            Max Torus linear velocity    97.1    m/sec
                      Tank Pressure  20265.    Pa


   PROPULSION ----------------------------------------------------
                 Total Delta-V Required    50.00   km/sec
                                    Isp   3095.    seconds
                       Thrust per Engine   9402.0   kN
                       Total Thrust Time    20.0    days
                       Number of Engines    30
            Acceleration: Full H2 load     0.014 m/s^2
            Acceleration:        Dry       0.072 m/s^2


   SYSTEM MASSES -------------------------------------------------
   (1 million kg = 1000 metric tons)
                                 Torus     31.862 Million kg
                         Torus + Insides    84.862 Million kg
                               Payload     0.001 Million kg
                    Torus Cosmic Shield  1416.275 Million kg
                          Total Engine      4.167 Million kg
                       Propellant Tank   124.398 Million kg
                    Other Structural    2279.400 Million kg
                    ------------------------------------------
                             DRY MASS   3909.102 Million kg

                    PROPELLANT MASS   16380.440 Million kg
                        TOTAL MASS   20289.543 Million kg


   MASS FRACTIONS ------------------------------------------------
                    Propellant 0.807334185
                     Structure 0.192653492
                       Payload 0.000012371


   DIMENSIONS ----------------------------------------------------
   (in meters unless noted)
                    Torus Major Radius    894.6
                    Torus Tube Radius      32.8
            Torus & Shield Tube Radius     46.8

                       Number of Tanks     2
                  Propellant Tank Radius   309.8
          Total Propellant Tank Volume   249.167 Million m^3
   Propellant Tank Surface Area (each)  1206.308 Thousand m^2
          Propellant Tank Thickness       0.020
   ------------------------------------------------------------
```

```
###################################################################
                    EMERALD CITY CONFIGURATION     (Data Set   3)

==================================================================
         Saturn Mission;   OPTIMIZED VERSION


    GENERAL PARAMETERS --------------------------------------------
                         Population   1000
            Artificial Gravity Desired    1.00
                          Spin Rate    1.00   RPM
                  Max Artificial G    1.04
                  Min Artificial G    0.96
            Max Torus linear velocity    97.1   m/sec
                      Tank Pressure  20265.    Pa


    PROPULSION ----------------------------------------------------
              Total Delta-V Required    50.00  km/sec
                             Isp   3095.    seconds
                  Thrust per Engine  9402.0   kN
                  Total Thrust Time    20.0   days
                  Number of Engines    14
            Acceleration: Full H2 load     0.014 m/s^2
            Acceleration:       Dry     0.072 m/s^2


    SYSTEM MASSES -------------------------------------------------
    (1 million kg = 1000 metric tons)
                             Torus   185.006 Million kg
                   Torus + Insides   238.006 Million kg
                         Payload     0.001 Million kg
               Torus Cosmic Shield  1416.275 Million kg
                      Total Engine     1.944 Million kg
                    Propellant Tank    58.260 Million kg
                  Other Structural   125.609 Million kg
                -----------------------------------------------
                         DRY MASS  1840.095 Million kg

                    PROPELLANT MASS  7710.612 Million kg
                        TOTAL MASS  9550.708 Million kg


    MASS FRACTIONS ------------------------------------------------
                      Propellant 0.807334125
                       Structure 0.192639589
                         Payload 0.000026281


    DIMENSIONS ----------------------------------------------------
    (in meters unless noted)
                  Torus Major Radius    894.6
                  Torus Tube Radius     32.8
          Torus & Shield Tube Radius     46.8

                  Number of Tanks       2
              Propellant Tank Radius    241.0
        Total Propellant Tank Volume   117.288 Million m^3
    Propellant Tank Surface Area (each)  729.962 Thousand m^2
          Propellant Tank Thickness      0.015
    ---------------------------------------------------------------
```

```
*********************************************************************************
            EFORCES.DAT      (Data Set    1)
            Saturn Mission; Pre-Optimization

                              Full Load        Burn Out
            --------------------------------------------------------------
            Acceleration (m/s^2)    0.014            0.073

FORCES:                                               (millions of N)
        Bottom of top tank     45.887            1.785
        Bottom of top pole      0.000            0.000
        Top of bottom tank     66.913          110.918
     Bottom of bottom tank    112.801          112.703

MOMENTS:                                              (millions of N-m)
     Spoke-pole interface    4004.243         20783.357

*********************************************************************************
            EFORCES.DAT      (Data Set    2)
            Saturn Mission; Pre-Optimization,   FAT SPOKES & SH

                              Full Load        Burn Out
            --------------------------------------------------------------
            Acceleration (m/s^2)    0.014            0.072

FORCES:                                               (millions of N)
        Bottom of top tank    114.723            4.488
        Bottom of top pole      0.000            0.000
        Top of bottom tank    167.279          277.271
     Bottom of bottom tank    282.002          281.759

MOMENTS:                                              (millions of N-m)
     Spoke-pole interface    3974.194         20627.391

*********************************************************************************
            EFORCES.DAT      (Data Set    3)
            Saturn Mission;  OPTIMIZED VERSION

                              Full Load        Burn Out
            --------------------------------------------------------------
            Acceleration (m/s^2)    0.014            0.072

FORCES:                                               (millions of N)
        Bottom of top tank     53.535            2.084
        Bottom of top pole      0.000            0.000
        Top of bottom tank     78.066          129.405
     Bottom of bottom tank    131.601          131.489

MOMENTS:                                              (millions of N-m)
     Spoke-pole interface    4341.917         22535.998
```

```fortran
C##########################################################################
C
C      ENGINES.FOR                         Vax Fortran
C      Jan. 29, 1992    v.3
C
C      Calculates the number of engines required to attain a specified
C      delta-V in a specified amount of time (tpr) and the mass of propellant
C      required.
C
C          tpr must be in seconds for units to agree
C          deltaV must be in meters/sec for units to agree.
C          tprdays is the tpr in days
C
C##########################################################################
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
C        1         2         3         4         5         6         7         8
C----#----------------------------------------------------------------#--------
       program ENGINES
       implicit NONE
       logical  DONE
       character BEL *1 /7/
       integer  I
       real    n1, n2, n3, n4, n5, Mdry, Mfuel
       real    Isp, g, Fi, tpr, deltaV, tprdays
       data   g/9.81/
C----#----------------------------------------------------------------#--------
       open (1, file= 'ENGINES.IN', status='old', err=900)
       open (2, file= 'ENGINES.OUT',status='new')
       read (1,*,err=920,end=910)                   ! read past header
       read (1,*,err=920,end=910)                   ! read past header
       read (1,*,err=920,end=910) Mdry, Isp, Fi, tprdays
C----#------------------------------- CALCULATE # OF ENGINES -----------#--------
       DONE = .FALSE.
       do 50 while (DONE.eq..false.)
          write (2,200)                             ! header
          write (2,210) Mdry/1E6, Isp, Fi/1E3
          write (2,220) tprdays/5., 2.*tprdays/5., 3.*tprdays/5.,
     +                             4.*tprdays/5., tprdays
          tpr = tprdays*86400.             ! convert to seconds
C
          do 10 I=1,20
             deltaV = 5000.*I                       ! convert to meters/sec
             Mfuel = Mdry*(exp(deltaV/Isp/g)-1.)
             n1 = Mfuel*Isp*g/Fi/(tpr/5.)
             n2 = Mfuel*Isp*g/Fi/(2.*tpr/5.)
             n3 = Mfuel*Isp*g/Fi/(3.*tpr/5.)
             n4 = Mfuel*Isp*g/Fi/(4.*tpr/5.)
             n5 = Mfuel*Isp*g/Fi/tpr
             write (2,230) deltaV/1E3, Mfuel/1E6, n1, n2, n3, n4, n5
 10       continue
          read (1,*,err=920,end=930) Mdry, Isp, Fi, tprdays
 50    continue
C----#--------------------------------------- ERROR TRAPS -------------#--------
 930   if (.false.) then
 900      write (6,*) '     Cannot find ENGINES.IN '
          write (6,*) BEL
       elseif (.false.) then
 910      write (6,*) '     No data in ENGINES.IN '
          write (6,*) BEL
       elseif (.false.) then
```

```
920     write (6,*) '      Bad data in ENGINES:IN '
        write (6,*) BEL
      endif
      write (6,*) BEL
      stop      '      I M DUN           Data in ENGINES.OUT'
C----#--------------------------------------------------------------------#--------
 200  format ('1',/5X, 14('#####'),
      +       /5X, 'Number of Engines Required for ')
 210  format (5X,'Mdry = ',F10.3, ' Million kg'
      +       /6X,'Isp = ', F8.1, '    Sec'
      +       /7X,'Fi =  ',F7.1, '    kN')
 220  format (32X,7('-'),'Powered Flight Time (days)',7('-'),
      +       /5X, ' delta-V      Fuel Mass ', 5(3X,F5.1),
      +       /5X, '(km/sec)  (kg, millions)', 1X, 5(2X,'# Engs'),
      +       /5X, 14('-----'))
 230  format (7X, F5.1, 6X, F8.2, 3X, 5(3X,F5.1))
 300  format (2X)
C----#--------------------------------------------------------------------#--------
      end
```

Mdry	Isp	Fi	tpr	ENGINES.IN
1480000000,	2500,	4000000,	30	
1480000000,	5000,	444000,	30	
1480000000,	2600,	4000000,	30	
1480000000,	2700,	4000000,	30	
1480000000,	2800,	4000000,	30	
1480000000,	2900,	4000000,	30	
1480000000,	3000,	4000000,	30	
1480000000,	2499,	1647397,	30	
1480000000,	2626,	2529275,	30	
1480000000,	2701,	3236534,	30	
1480000000,	2800,	4476433,	30	
1480000000,	3012,	9484996,	30	

```
##############################################################################
Number of Engines Required for
Mdry =   1480.000 Million kg
 Isp =   2500.0   Sec
 Fi =    4000.0   kN
```

		-------Powered Flight Time (days)-------				
delta-V	Fuel Mass	6.0	12.0	18.0	24.0	30.0
(km/sec)	(kg, millions)	# Engs	# Engs	# Engs	# Engs	# Engs
5.0	334.69	4.0	2.0	1.3	1.0	0.8
10.0	745.07	8.8	4.4	2.9	2.2	1.8
15.0	1248.26	14.8	7.4	4.9	3.7	3.0
20.0	1865.23	22.1	11.0	7.4	5.5	4.4
25.0	2621.73	31.0	15.5	10.3	7.8	6.2
30.0	3549.31	42.0	21.0	14.0	10.5	8.4
35.0	4686.66	55.4	27.7	18.5	13.9	11.1
40.0	6081.21	71.9	36.0	24.0	18.0	14.4
45.0	7791.12	92.1	46.1	30.7	23.0	18.4
50.0	9887.72	116.9	58.5	39.0	29.2	23.4
55.0	12458.46	147.3	73.7	49.1	36.8	29.5
60.0	15610.54	184.6	92.3	61.5	46.2	36.9
65.0	19475.45	230.3	115.2	76.8	57.6	46.1
70.0	24214.38	286.4	143.2	95.5	71.6	57.3
75.0	30024.99	355.1	177.6	118.4	88.8	71.0
80.0	37149.63	439.4	219.7	146.5	109.8	87.9
85.0	45885.45	542.7	271.3	180.9	135.7	108.5
90.0	56596.83	669.4	334.7	223.1	167.3	133.9
95.0	69730.49	824.7	412.4	274.9	206.2	164.9
100.0	85834.26	*****	507.6	338.4	253.8	203.0

```
###############################################################################
Number of Engines Required for
Mdry =    1480.000 Million kg
 Isp =    5000.0    Sec
 Fi  =     444.0    kN

                              -------Powered Flight Time (days)-------
  delta-V     Fuel Mass       6.0     12.0    18.0    24.0    30.0
 (km/sec)   (kg, millions)   # Engs  # Engs  # Engs  # Engs  # Engs
-------------------------------------------------------------------------
     5.0        158.82        33.8    16.9    11.3     8.5     6.8
    10.0        334.69        71.3    35.7    23.8    17.8    14.3
    15.0        529.43       112.8    56.4    37.6    28.2    22.6
    20.0        745.07       158.8    79.4    52.9    39.7    31.8
    25.0        983.85       209.7   104.8    69.9    52.4    41.9
    30.0       1248.26       266.0   133.0    88.7    66.5    53.2
    35.0       1541.04       328.4   164.2   109.5    82.1    65.7
    40.0       1865.23       397.5   198.7   132.5    99.4    79.5
    45.0       2224.22       474.0   237.0   158.0   118.5    94.8
    50.0       2621.73       558.7   279.4   186.2   139.7   111.7
    55.0       3061.91       652.5   326.3   217.5   163.1   130.5
    60.0       3549.31       756.4   378.2   252.1   189.1   151.3
    65.0       4089.03       871.4   435.7   290.5   217.8   174.3
    70.0       4686.66       998.7   499.4   332.9   249.7   199.7
    75.0       5348.42       *****   569.9   379.9   284.9   228.0
    80.0       6081.21       *****   648.0   432.0   324.0   259.2
    85.0       6892.63       *****   734.4   489.6   367.2   293.8
    90.0       7791.12       *****   830.2   553.4   415.1   332.1
    95.0       8786.04       *****   936.2   624.1   468.1   374.5
   100.0       9887.72       *****   *****   702.4   526.8   421.4


###############################################################################
Number of Engines Required for
Mdry =    1480.000 Million kg
 Isp =    3000.0    Sec
 Fi  =    4000.0    kN

                              -------Powered Flight Time (days)-------
  delta-V     Fuel Mass       6.0     12.0    18.0    24.0    30.0
 (km/sec)   (kg, millions)   # Engs  # Engs  # Engs  # Engs  # Engs
-------------------------------------------------------------------------
     5.0        274.07         3.9     1.9     1.3     1.0     0.8
    10.0        598.88         8.5     4.2     2.8     2.1     1.7
    15.0        983.85        14.0     7.0     4.7     3.5     2.8
    20.0       1440.11        20.4    10.2     6.8     5.1     4.1
    25.0       1980.85        28.1    14.1     9.4     7.0     5.6
    30.0       2621.73        37.2    18.6    12.4     9.3     7.4
    35.0       3381.29        48.0    24.0    16.0    12.0     9.6
    40.0       4281.51        60.8    30.4    20.3    15.2    12.2
    45.0       5348.42        75.9    38.0    25.3    19.0    15.2
    50.0       6612.91        93.9    46.9    31.3    23.5    18.8
    55.0       8111.56       115.1    57.6    38.4    28.8    23.0
    60.0       9887.72       140.3    70.2    46.8    35.1    28.1
    65.0      11992.80       170.2    85.1    56.7    42.6    34.0
    70.0      14487.69       205.6   102.8    68.5    51.4    41.1
    75.0      17444.59       247.6   123.8    82.5    61.9    49.5
    80.0      20949.04       297.3   148.7    99.1    74.3    59.5
    85.0      25102.46       356.3   178.1   118.8    89.1    71.3
    90.0      30024.99       426.1   213.1   142.0   106.5    85.2
    95.0      35859.08       508.9   254.5   169.6   127.2   101.8
   100.0      42773.53       607.1   303.5   202.4   151.8   121.4
```

```
C##################################################################################
C
C     NUMTANK.FOR                          Vax Fortran
C
C     Computes the total mass and area of a number of identically sized
C     propellant tanks based on the total mass of propellant required.
C
C     Subroutines:
C         TANK      Computes the tank area, mass, thickness based on AR,
C                   Working stress, density, and the applied accelerations.
C
C##################################################################################
C234567890123456789012345678901234567890123456789012345678901234567890
C         1         2         3         4         5         6         7         8
C----#-----------------------------------------------------------------#--------
      program SHELL
      implicit NONE
      character BEL *1 /7/
      integer Num, N
C
      common /CONST/PI,ag
     +       /PROP/ Dh2, Dsh, Dtor, WStor, Dtank, WStank
C
      real     Dh2, Dsh, Dtor, WStor, Dtank, WStank
      real     Mdry, MdryNEW, Mtank, Mpay, Msh, Mstr, Mh2
      real     PI, ag, ACCf, Mass, Area, Vol
      real     AR, Rp, Hp, Ttank, Vh2, Ptank, Atank
      open  (1, file= 'NUMTANK.IN' ,status='OLD', err=800)
      open  (2, file= 'NUMTANK.OUT',status='NEW')
      open  (3, file= 'NUMTANK2.OUT',status='NEW')
      PI  = 3.141592654
      ag  = 9.81                        ! accel of gravity (on earth)
C----#-----------------------------------------------------------------#--------
      Dtank = 2650
      WStank= 165e6
      ACCf = .012
      Dh2  = 71
      N=0
      read (1,*,err=810,end=830)
      read (1,*,err=810,end=830)                        ! read past header.
      write (2,200)                     ! header
      write (3,210)                     ! header
C----#-------------------------------------------- INPUT ---------------#--------
      DO 20 WHILE (.true.)
      N=N+1
      read (1,*,err=810,end=850) Mh2, Num, AR, Ptank
      Ptank = Ptank * 101325            ! convert atm to N/m^2
C----#---------------------------------------------------- MAIN LOOP --#--------
      Vh2 = Mh2*1E6/Dh2/Num
      call TANK (Vh2, AR, Rp, Hp, Ttank, Mtank, Atank, ACCf, Ptank)
      Mass = Mtank * Num
      Area = Atank * Num
      Vol = Vh2 * Num
      write (2,250)
     +        Mh2, AR, Ptank, Num, Vol/1E6, Mass/1E6, Area/1E3, Ttank
      write (3,250)
     +        Mh2, AR, Rp, Num, Vh2/1E6, Mtank/1E6, Atank/1E3, Ttank
20    continue
C----#---------------------------------------------------- OUTPUT --#--------
      if (.false.) then
```

```
800     write (6;*) '          Cannot find NUMTANK.IN '
        write (*,*) BEL
      elseif (.false.) then
810     write (6,*) '          Bad data in NUMTANK.IN '
        write (6,*) '          Data set ', N
        write (*,*) BEL
      elseif (.false.) then
830     write (6,*) '          Incomplete data set in NUMTANK.IN '
        write (6,*) '          Data set ', N
        write (*,*) BEL
      elseif (.false.) then
850     write (6,*) '      END OF DATA. '
      endif
900   write (*,*) BEL
      stop '       I M DUN.        Output in file NUMTANK,NUMTANK2.OUT'
C----#------------------------------------------------------------ FORMAT --#--------
 200  format (5X, //15X, 'NUMTANK.OUT        ',
C---->              xxxxxx.xxx-xx.x-xxxxxx.xxx--xxxx-xxxxxx.xxx-
     +       /2X,'     Mh2      AR      Ptank       Num       Vh2    ',
C---->              xxxxxx.xxx-xxxxxx.xxx--x.xxxx
     +       '      Mass       Area       Thick',
     +       /2X,'    (Mkg)            (N/m^2)                (M m^3)  ',
     +       '    (Mkg)        (K m^2)      (m)',
     +       /2X, 15('-----'))
 210  format (5X, //15X, 'NUMTANK2.OUT    (Individual Tanks)   ',
C---->              xxxxxx.xxx-xx.x-xxxxxx.xxx--xxxx-xxxxxx.xxx-
     +       /2X,'     Mh2      AR       Rp        Num       Vh2    ',
C---->              xxxxxx.xxx-xxxxxx.xxx--x.xxxx
     +       '      Mass       Area       Thick',
     +       /2X,'    (Mkg)            (N/m^2)                (M m^3)  ',
     +       '    (Mkg)        (K m^2)      (m)',
     +       /2X, 15('-----'))
 250  format (2X, F10.3,1X,F4.1,1X,F10.3,2X,I4,3(1X,F10.3),2X,F6.4)
      end
```

Mass H2 (kg)	Num	AR	Ptank (atm)
9725.131	1	1	.2
9725.131	5	1	.2
9725.131	10	1	.2
9725.131	1	2	.2
9725.131	5	2	.2
9725.131	10	2	.2
9725.131	1	4	.2
9725.131	5	4	.2
9725.131	10	4	.2

NUMTANK.OUT

Mh2 (Mkg)	AR	Ptank (N/m^2)	Num	Vh2 (M m^3)	Mass (Mkg)	Area (K m^2)	Thick (m)
9725.131	1.0	20265.000	1	136.974	68.219	1285.017	0.0202
9725.131	1.0	20265.000	5	136.974	67.660	2197.349	0.0117
9725.131	1.0	20265.000	10	136.974	67.497	2768.486	0.0092
9725.131	2.0	20265.000	1	136.974	82.497	1395.230	0.0298
9725.131	2.0	20265.000	5	136.974	81.562	2385.809	0.0172
9725.131	2.0	20265.000	10	136.974	81.290	3005.931	0.0136
9725.131	4.0	20265.000	1	136.974	89.414	1649.658	0.0234
9725.131	4.0	20265.000	5	136.974	87.627	2820.875	0.0134
9725.131	4.0	20265.000	10	136.974	87.107	3554.080	0.0106

```
C###########################################################################
C
C       ROTPRESS.FOR                        Vax Fortran
C
C
C       Subroutines:
C               PRVAR.SUB       calculates the pressure on the wall of a
C                               spinning tank, where area varies with radius
C
C               PRCAR.SUB       Calculates pressure in a constant area spinning
C                               vessels
C
C###########################################################################
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
C        1         2         3         4         5         6         7         8
C----#--------------------------------------------------------------#--------
        program SHELL
        implicit NONE
        character BEL *1 /7/
        integer N
        real    PI, ag
        real    RHO, RPM, RO, RI, PRVA, PRCA
C
        common /CONST/PI,ag
C
        open (1, file= 'ROTPRESS.IN' ,status='OLD', err=800)
        open (2, file= 'ROTPRESS.OUT',status='NEW')
        PI  = 3.141592654
        ag  = 9.81                      ! accel of gravity (on earth)
C----#--------------------------------------------------------------#--------
        read (1,*,err=810,end=830)
        read (1,*,err=810,end=830)                 ! read past header.
        write (2,200)                   ! header
C----#------------------------------------------------- INPUT ------------#--------
        DO 20 WHILE (.true.)
        N=N+1
        read (1,*,err=810,end=850) RPM, RHO, RO, RI
C----#------------------------------------------------- MAIN LOOP --#--------
        call PRVAR (RHO, RPM, RO, RI, 500, PRVA)       ! Varying area
        call PRCAR (RHO, RPM, RO, RI, 500, PRCA)       ! Constant area
        write (2,250) RPM, RHO, RO, RI, PRCA, PRVA
 20     continue
C----#------------------------------------------------- OUTPUT --#----------
        if (.false.) then
 800       write (6,*) '            Cannot find .IN '
           write (*,*) BEL
        elseif (.false.) then
 810       write (6,*) '            Bad data in .IN '
           write (6,*) '            Data set ', N
           write (*,*) BEL
        elseif (.false.) then
 830       write (6,*) '            Incomplete data set in .IN '
           write (6,*) '            Data set ', N
           write (*,*) BEL
        elseif (.false.) then
 850       write (6,*) '        END OF DATA. '
        endif
 900    write (*,*) BEL
        stop '       I M DUN.        Output in file ROTPRESS.OUT'
C----#------------------------------------------------- FORMAT --#--------

 200    format ('1',/10X, 'ROTPRESS.OUT',
     +     //55X,'Constant', 5X, 'Varying',
     +      /57X, 'Area',     8X, 'Area',
     + /8X,'RPM',9X,'RHO',10X,'RO',10X,'RI',8X,'Pressure',4X,'Pressure',
C xxxxxx.xxx   xxxxx.xxx   xxxxx.xxx   xxxxx.xxx   xxxxxxxxx.   xxxxxxxxx.
     +         /5X,13('====='))
 250    format (5X,4(F10.3,2X),2(F10.0,2X))
        end
```

```
C############################################################################
C
C     PRVAR.SUB
C
C     Calculates the pressure in a cylindrical tank that is spinning where
C     the area varies with radius.
C
C############################################################################
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C         1         2         3         4         5         6         7         8
C----#---------------------------------------------------------------------#--------
      subroutine PRVAR (RHO, RPM, RO, RI, N, PRESS)
      common /CONST/ PI, ag
      integer N
      real PI, ag
      real RHO, PRM, RO, RI, PRESS
      real SLICE, P, R2, R1
C----#---------------------------------------------------------------------#--------
      SLICE = (RO - RI)/N
      PRESS = 0.
      do R2 = RO,RI,-SLICE
         R1 = R2 - SLICE
         if(R1.gt.0.)P=RHO*(R2**2.-R1**2.)*(PI*RPM)**2.*(R2+R1)/R2/3600.
         PRESS = PRESS + P
      enddo
      return
      end




C############################################################################
C
C     PRCAR.SUB
C
C     Calculates the pressure in tank that is rotating, where the area is
C     constant along the radius.
C
C############################################################################
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C         1         2         3         4         5         6         7         8
C----#---------------------------------------------------------------------#--------
      subroutine PRCAR (RHO, RPM, RO, RI, N, PRESS)
      common /CONST/ PI, ag
      integer N
      real PI, ag
      real RHO, PRM, RO, RI, PRESS
      real SLICE, P, R2, R1
C----#---------------------------------------------------------------------#--------
      SLICE = (RO - RI)/N
      PRESS = 0.
      do R2 = RO,RI,-SLICE
         R1 = R2 - SLICE
         if(R1.gt.0.)P=RHO*SLICE*(PI*RPM)**2.*(R2+R1)/1800.
         PRESS = PRESS + P
      enddo
      return
      end
```

RPM	RHO	RO	RI	ROTPRESS.IN
1	71	156.7	0	
1	71	940.6	847.8	
1	71	847.8	0	
1	71	847.8	400	
1	71	847.8	600	

ROTPRESS.OUT

RPM	RHO	RO	RI	Constant Area Pressure	Varying Area Pressure
1.000	71.000	156.700	0.000	9559.	9540.
1.000	71.000	940.600	847.800	64610.	64606.
1.000	71.000	847.800	0.000	279820.	279265.
1.000	71.000	847.800	400.000	217804.	217654.
1.000	71.000	847.800	600.000	139899.	139852.

```
C#################################################################################
C
C      TANKPRESS.FOR
C      Feb 18, 1992
C
C      Calculates the fluid pressure inside a tank due to acceleration and
C      head pressure for spacecraft applications.
C
C#################################################################################
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
C        1         2         3         4         5         6         7         8
C----#----------------------------------------------------------------#--------
       program PRESS
       implicit NONE
       real PI, THRUST, Mdry, Mass, Height, Radius, Head, DEN, VOL, VOLt
       real P, Pt, Accel, h, h2, top
       data PI /3.1415926/
       open (2, file = 'TANKPRESS.OUT' , status = 'new')
C----#----------------------------------------------------------------#--------
       THRUST = 4E6 *34
       Mdry   = 1455.663E6
       DEN    = 71
       Height = 1880.3
       Radius = 156.7
       Head   = 101325 * .2
C----#----------------------------------------------------------------#--------
       VOLt = PI*Radius**2.*(Height-2.*Radius) + 4./3.*PI*Radius**3.
       top = height - radius
       do 10 h2 = height-radius, radius, -radius
         VOL = PI*Radius**2.*(h2-Radius) + 2./3.*PI*Radius**3.
         MASS = DEN * VOL
         Accel = Thrust/(Mass + Mdry)
         write (2,200) Height, Radius, h2, Vol/Volt*100, Accel, Head

         do 20 h=0,h2,radius/4.
         P = DEN * accel * (h2-h)
         Pt = P + Head
         write (2,220) h, P, Pt
20     continue
10     continue
       stop '       I M DUN.              Data in TANKPRESS.OUT'
C----#----------------------------------------------------------------#--------
200    format ('1',4X,'PRESSURE PROFILE DUE TO THRUSTING.',
      +        //9X, 'Tank Height ', F8.1, ' meters',
      +         /9X, 'Tank Radius ', F8.1, ' meters',
      +    /3X, 'Propellant Height ',F8.1, ' meters',
      +        /14X, '% Full ',       F8.1,
      +        /8X, 'Acceleration ',F8.4, ' m/sec',
      +        /16X, 'Head ',         F8.1, ' N/m^2',
      +        //5X, '         Hydrostatic  Combined',
      +         /5X, '        y     Pressure   Pressure',
      +         /5X, 13('-----'))
220    format (5X, 5(3X,F8.1))
       end
```

PRESSURE PROFILE DUE TO THRUSTING.

Tank Height	1880.3	meters
Tank Radius	156.7	meters
Propellant Height	1723.6	meters
% Full	94.1	
Acceleration	0.0128	m/sec
Head	20265.0	N/m^2

y	Hydrostatic Pressure	Combined Pressure
0.0	1568.7	21833.7
39.2	1533.0	21798.0
78.3	1497.3	21762.3
117.5	1461.7	21726.7
156.7	1426.0	21691.0
195.9	1390.4	21655.4
235.1	1354.7	21619.7
274.2	1319.1	21584.1
313.4	1283.4	21548.4
352.6	1247.8	21512.8
391.7	1212.1	21477.1
430.9	1176.5	21441.5
470.1	1140.8	21405.8
509.3	1105.2	21370.2
548.4	1069.5	21334.5
587.6	1033.9	21298.9
626.8	998.2	21263.2
666.0	962.5	21227.5
705.1	926.9	21191.9
744.3	891.2	21156.2
783.5	855.6	21120.6
822.7	819.9	21084.9
861.8	784.3	21049.3
901.0	748.6	21013.6
940.2	713.0	20978.0
979.4	677.3	20942.3
1018.5	641.7	20906.7
1057.7	606.0	20871.0
1096.9	570.4	20835.4
1136.1	534.7	20799.7
1175.3	499.1	20764.1
1214.4	463.4	20728.4
1253.6	427.7	20692.7
1292.8	392.1	20657.1
1332.0	356.4	20621.4
1371.1	320.8	20585.8
1410.3	285.1	20550.1
1449.5	249.5	20514.5
1488.7	213.8	20478.8
1527.8	178.2	20443.2
1567.0	142.5	20407.5
1606.2	106.9	20371.9
1645.4	71.2	20336.2
1684.5	35.6	20300.6

```
C########################################################################
C
C    I-TANK.FOR
C    VAXFortran
C    Feb. 1, 1992
C    Calculates the mass moment of inertia for a given volume, density, and
C    several aspect ratios.
C
C########################################################################
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
C        1         2         3         4         5         6         7         8
C----#-----------------------------------------------------------------#--------
       program INERTIA
       implicit NONE
       logical  LOOP
       real     AR, Iyy, Den, Vol
       real     PI /3.141592654/
       open (1, file= 'I-TANK.IN', status='old',err=900)
       open (2, file= 'I-TANK.OUT',status='new')
       read (1,*,err=900,end=900)
       read (1,*,err=900,end=900)
       LOOP = .TRUE.
       do 10 while (loop.eq..true.)
         read (1,*, err=900, end=910) Den, Vol
         write (2,200) Den, Vol/1E6
         do 20 AR=1.,50.
           Iyy = Den*Vol/2.*(Vol/2./PI/AR)**(2./3.)
           write (2,210) AR, Iyy/1E9
20       continue
         if (.false.) then
910        LOOP = .FALSE.
         endif
10     continue
       if (.false.) then
900      write (6,*) ' Something wrong with I-TANK.IN   '
       endif
       stop     '  I M DUN.      Data in I-TANK.OUT'
200    format ('1',15X, 'I-TANK.OUT', /5X, 14('#####'),
      +         /5X, 'Density =', F11.4,' kg/cubic meters',
      +         /6X, 'Volume =', F11.4, ' Million cubic meters',
      +         /5X,  /5X, 'Aspect Ratio       Iyy'
      +         /20X,  '(kg*m^2)(billions)'
      +         /5X, 14('-----') )
210    format (8X, F5.1, 6X, F13.3)
       end
```

```
                 I-TANK.OUT
###############################################################
Density =     71.0000 kg/cubic meters
 Volume =    100.0000 Million cubic meters

Aspect Ratio        Iyy
                (kg*m^2)(billions)
-------------------------------------------------------------
     1.0          224616.641
     2.0          141499.625                I-TANK.OUT
     3.0          107984.414    ###############################################
     4.0           89139.172    Density =     71.0000 kg/cubic meters
     5.0           76817.805     Volume =    200.0000 Million cubic meters
     6.0           68025.914
     7.0           61382.309    Aspect Ratio        Iyy
     8.0           56154.160                    (kg*m^2)(billions)
     9.0           51913.492    --------------------------------------------
    10.0           48392.188         1.0          713113.375
    11.0           45412.992         2.0          449233.281
    12.0           42853.645         3.0          342829.156
    13.0           40626.832         4.0          282999.250
    14.0           38668.434         5.0          243881.344
    15.0           36930.152         6.0          215968.828
    16.0           35374.902         7.0          194876.688
    17.0           33973.684         8.0          178278.344
    18.0           32703.449         9.0          164815.063
    19.0           31545.652        10.0          153635.609
    20.0           30485.164        11.0          144177.266
    21.0           29509.535        12.0          136051.828
    22.0           28608.391        13.0          128982.172
    23.0           27773.035        14.0          122764.617
    24.0           26996.104        15.0          117245.914
    25.0           26271.320        16.0          112308.320
    26.0           25593.303        17.0          107859.719
    27.0           24957.402        18.0          103826.984
    28.0           24359.584        19.0          100151.203
    29.0           23796.324        20.0           96784.375
    30.0           23264.535        21.0           93686.938
    31.0           22761.496        22.0           90825.984
    32.0           22284.793        23.0           88173.891
    33.0           21832.289        24.0           85707.289
    34.0           21402.078        25.0           83406.242
    35.0           20992.453        26.0           81253.664
    36.0           20601.883        27.0           79234.813
    37.0           20228.984        28.0           77336.867
    38.0           19872.514        29.0           75548.617
    39.0           19531.346        30.0           73860.305
    40.0           19204.449        31.0           72263.242
    41.0           18890.902        32.0           70749.805
    42.0           18589.844        33.0           69313.195
    43.0           18300.500        34.0           67947.367
    44.0           18022.158        35.0           66646.883
    45.0           17754.164        36.0           65406.898
    46.0           17495.916        37.0           64223.023
    47.0           17246.857        38.0           63091.305
    48.0           17006.479        39.0           62008.160
    49.0           16774.303        40.0           60970.328
    50.0           16549.895        41.0           59974.871
                                    42.0           59019.070
                                    43.0           58100.465
                                    44.0           57216.781
                                    45.0           56365.961
                                    46.0           55546.070
                                    47.0           54755.359
                                    48.0           53992.207
                                    49.0           53255.094
                                    50.0           52542.641
```

```
          I-TANK.OUT
##################################################################
Density =    71.0000 kg/cubic meters
 Volume =   300.0000 Million cubic meters

Aspect Ratio        Iyy
                 (kg*m^2)(billions)
-----------------------------------------------------------------
    1.0         1401664.375
    2.0          882993.188                    I-TANK.OUT
    3.0          673849.875     ###############################################
    4.0          556250.875     Density =    71.0000 kg/cubic meters
    5.0          479362.438      Volume =   400.0000 Million cubic meters
    6.0          424498.844
    7.0          383041.031     Aspect Ratio        Iyy
    8.0          350416.063                      (kg*m^2)(billions)
    9.0          323953.250     ----------------------------------------------
   10.0          301979.406          1.0         2263994.000
   11.0          283388.500          2.0         1426226.750
   12.0          267417.500          3.0         1088414.750
   13.0          253521.688          4.0          898466.563
   14.0          241300.719          5.0          774275.000
   15.0          230453.422          6.0          685658.313
   16.0          220748.281          7.0          618694.938
   17.0          212004.313          8.0          565998.500
   18.0          204077.734          9.0          523255.188
   19.0          196852.797         10.0          487762.688
   20.0          190235.109         11.0          457734.281
   21.0          184146.922         12.0          431937.656
   22.0          178523.578         13.0          409492.844
   23.0          173310.719         14.0          389753.375
   24.0          168462.453         15.0          372232.563
   25.0          163939.641         16.0          356556.688
   26.0          159708.641         17.0          342433.250
   27.0          155740.469         18.0          329630.125
   28.0          152009.938         19.0          317960.250
   29.0          148495.063         20.0          307271.219
   30.0          145176.563         21.0          297437.500
   31.0          142037.438         22.0          288354.531
   32.0          139062.703         23.0          279934.656
   33.0          136238.969         24.0          272103.656
   34.0          133554.359         25.0          264798.313
   35.0          130998.188         26.0          257964.344
   36.0          128560.930         27.0          251554.875
   37.0          126233.945         28.0          245529.234
   38.0          124009.492         29.0          239851.922
   39.0          121880.508         30.0          234491.828
   40.0          119840.609         31.0          229421.484
   41.0          117883.969         32.0          224616.641
   42.0          116005.297         33.0          220055.688
   43.0          114199.719         34.0          215719.438
   44.0          112462.789         35.0          211590.688
   45.0          110790.445         36.0          207653.969
   46.0          109178.906         37.0          203895.406
   47.0          107624.734         38.0          200302.406
   48.0          106124.711         39.0          196863.641
   49.0          104675.867         40.0          193568.750
   50.0          103275.508         41.0          190408.344
                                    42.0          187373.875
                                    43.0          184457.484
                                    44.0          181651.969
                                    45.0          178950.766
                                    46.0          176347.781
                                    47.0          173837.438
                                    48.0          171414.578
                                    49.0          169074.406
                                    50.0          166812.484
```

B.24

APPENDIX C

ASTROS BULK DATA CARDS

CBAR   - Defines a simple beam element (BAR) of the structural model.

CONM2   - Defines a concentrated mass at a grid point of the structural model.

CROD   - Defines a tension-compression-torsion element (ROD) of the structural model.

DCONVMM - Defines a Von-Mises stress constraint by specifying material identification numbers.

DESELM   - Designates design variable properties when the design variable is uniquely associated with a single finite element.

EIGR   - Defines data needed to perform real eigenvalues extraction.

FORCE   - Defines a static load at a grid point be specifying a vector.

GRID   - Defines the location of a geometric grid point of the structural model, the directions of its displacement, and its permanent single-point constraints.

MAT1   - Defines the material properties for linear, temperature-independent, isotropic materials.

MODELIST - Defines a list of modes at which outputs are desired.

PBAR   - Defines the properties of a simple beam (bar) which is used to create bar elements via the CBAR entry.

SPC   - Defines sets of single-point constraints and enforced displacements.

SAMPLE ASTROS INPUT FILE-OPTIMIZATION

```
ASSIGN DATABASE PROJECT WISH TEMP
SOLUTION
TITLE=COMPLETE MODEL -- TRUE DIMENSION
  OPTIMIZE STRATEGY = MP
     PRINT (RECT,ITER=LAST) GDES=ALL
     BOUNDARY  SPC = 250,   METHOD = 20
     STATICS(MECH=1),
            CONSTRAINT(STRESS=1,STRESS=2,STRESS=3)
  END
BEGIN BULK NOECHO
$234567812345678123456781234567812345678123456781234567812345678123456781234567
$      2        3        4        5        6        7        8        9       10
MFORM,COUPLED
 GRID,  1,,  100.0,       895.0,          0.0,,6
 GRID,  2,,  100.0,       864.5,        231.6
 GRID,  3,,  100.0,       775.1,        447.5
 GRID,  4,,  100.0,       632.9,        632.9
 GRID,  5,,  100.0,       447.5,        775.1
 GRID,  6,,  100.0,       231.6,        864.5
 GRID,  7,,  100.0,         0.0,        895.0,,5
 GRID,  8,,  100.0,      -231.6,        864.5
 GRID,  9,,  100.0,      -447.5,        775.1
 GRID, 10,,  100.0,      -632.9,        632.9
 GRID, 11,,  100.0,      -775.1,        447.5
 GRID, 12,,  100.0,      -864.5,        231.6
 GRID, 13,,  100.0,      -895.0,          0.0,,6
 GRID, 14,,  100.0,      -864.5,       -231.6
 GRID, 15,,  100.0,      -775.1,       -447.5
 GRID, 16,,  100.0,      -632.9,       -632.9
 GRID, 17,,  100.0,      -447.5,       -775.1
 GRID, 18,,  100.0,      -231.6,       -864.5
 GRID, 19,,  100.0,         0.0,       -895.0,,5
 GRID, 20,,  100.0,       231.6,       -864.5
 GRID, 21,,  100.0,       447.5,       -775.1
 GRID, 22,,  100.0,       632.9,       -632.9
 GRID, 23,,  100.0,       775.1,       -447.5
 GRID, 24,,  100.0,       864.5,       -231.6
$
$ -- Grid 25 is the connecting point between the shaft and the torus
GRID,25,,100.0,0.0,0.0,,4
$
$ -- Grid points for the spokes
 GRID,26,,  100.0,       223.8,          0.0,,5
 GRID,27,,  100.0,       447.5,          0.0,,5
 GRID,28,,  100.0,       671.3,          0.0,,5
 GRID,29,,  100.0,       111.9,        193.8,,6
 GRID,30,,  100.0,       223.8,        387.5,,6
 GRID,31,,  100.0,       335.6,        581.3,,6
 GRID,32,,  100.0,      -111.9,        193.8,,6
 GRID,33,,  100.0,      -223.8,        387.5,,6
 GRID,34,,  100.0,      -335.6,        581.3,,6
 GRID,35,,  100.0,      -223.8,          0.0,,5
 GRID,36,,  100.0,      -447.5,          0.0,,5
 GRID,37,,  100.0,      -671.3,          0.0,,5
 GRID,38,,  100.0,      -111.9,       -193.8,,6
 GRID,39,,  100.0,      -223.8,       -387.5,,6
 GRID,40,,  100.0,      -335.6,       -581.3,,6
 GRID,41,,  100.0,       111.9,       -193.8,,6
 GRID,42,,  100.0,       223.8,       -387.5,,6
 GRID,43,,  100.0,       335.6,       -581.3,,6
```

C.3

```
$
$ -- Grid points for the shaft
  GRID,100,,     0.0,0.0,0.0,,4
  GRID,101,,    20.0,0.0,0.0,,4
  GRID,102,,    40.0,0.0,0.0,,4
  GRID,103,,    60.0,0.0,0.0,,4
  GRID,104,,    80.0,0.0,0.0,,4
  GRID,106,,   120.0,0.0,0.0,,4
  GRID,107,,   140.0,0.0,0.0,,4
  GRID,108,,   160.0,0.0,0.0,,4
  GRID,109,,   180.0,0.0,0.0,,4
  GRID,110,,   200.0,0.0,0.0,,4
$
$
CBAR,1,100,1,2,1.0,0.0,0.0
CBAR,2,100,2,3,1.0,0.0,0.0
CBAR,3,100,3,4,1.0,0.0,0.0
CBAR,4,100,4,5,1.0,0.0,0.0
CBAR,5,100,5,6,1.0,0.0,0.0
CBAR,6,100,6,7,1.0,0.0,0.0
CBAR,7,100,7,8,1.0,0.0,0.0
CBAR,8,100,8,9,1.0,0.0,0.0
CBAR,9,100,9,10,1.0,0.0,0.0
CBAR,10,100,10,11,1.0,0.0,0.0
CBAR,11,100,11,12,1.0,0.0,0.0
CBAR,12,100,12,13,1.0,0.0,0.0
CBAR,13,100,13,14,1.0,0.0,0.0
CBAR,14,100,14,15,1.0,0.0,0.0
CBAR,15,100,15,16,1.0,0.0,0.0
CBAR,16,100,16,17,1.0,0.0,0.0
CBAR,17,100,17,18,1.0,0.0,0.0
CBAR,18,100,18,19,1.0,0.0,0.0
CBAR,19,100,19,20,1.0,0.0,0.0
CBAR,20,100,20,21,1.0,0.0,0.0
CBAR,21,100,21,22,1.0,0.0,0.0
CBAR,22,100,22,23,1.0,0.0,0.0
CBAR,23,100,23,24,1.0,0.0,0.0
CBAR,24,100,24,1,1.0,0.0,0.0
$
$ -- CBARs for the spokes
CBAR,25,101,25,26,1.0,0.0,0.0
CBAR,26,101,26,27,1.0,0.0,0.0
CBAR,27,101,27,28,1.0,0.0,0.0
CBAR,28,101,28,1,1.0,0.0,0.0
CBAR,29,101,25,29,1.0,0.0,0.0
CBAR,30,101,29,30,1.0,0.0,0.0
CBAR,31,101,30,31,1.0,0.0,0.0
CBAR,32,101,31,5,1.0,0.0,0.0
CBAR,33,101,25,32,1.0,0.0,0.0
CBAR,34,101,32,33,1.0,0.0,0.0
CBAR,35,101,33,34,1.0,0.0,0.0
CBAR,36,101,34,9,1.0,0.0,0.0
CBAR,37,101,25,35,1.0,0.0,0.0
CBAR,38,101,35,36,1.0,0.0,0.0
CBAR,39,101,36,37,1.0,0.0,0.0
CBAR,40,101,37,13,1.0,0.0,0.0
CBAR,41,101,25,38,1.0,0.0,0.0
CBAR,42,101,38,39,1.0,0.0,0.0
CBAR,43,101,39,40,1.0,0.0,0.0
CBAR,44,101,40,17,1.0,0.0,0.0
```

```
CBAR,45,101,25,41,1.0,0.0,0.0
CBAR,46,101,41,42,1.0,0.0,0.0
CBAR,47,101,42,43,1.0,0.0,0.0
CBAR,48,101,43,21,1.0,0.0,0.0
$
$ -- CBABs for the shaft
CBAR,100,102,100,101,0.0,0.0,1.0
CBAR,101,102,101,102,0.0,0.0,1.0
CBAR,102,102,102,103,0.0,0.0,1.0
CBAR,103,102,103,104,0.0,0.0,1.0
CBAR,104,102,104,25,0.0,0.0,1.0
CBAR,105,102,25,106,0.0,0.0,1.0
CBAR,106,102,106,107,0.0,0.0,1.0
CBAR,107,102,107,108,0.0,0.0,1.0
CBAR,108,102,108,109,0.0,0.0,1.0
CBAR,109,102,109,110,0.0,0.0,1.0
$
$ -- Design Elements for the spokes
$
DESELM,25,25,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,26,26,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,27,27,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,28,28,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,29,29,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,30,30,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,31,31,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,32,32,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,33,33,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,34,34,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,35,35,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,36,36,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,37,37,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,38,38,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,39,39,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,40,40,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,41,41,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,42,42,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,43,43,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,44,44,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,45,45,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,46,46,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,47,47,CBAR,0.1256,128.8052,12.5978,,SPOKES
DESELM,48,48,CBAR,0.1256,128.8052,12.5978,,SPOKES
$
$ -- Design Elements for the shaft
$
DESELM,100,100,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,101,101,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,102,102,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,103,103,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,104,104,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,105,105,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,106,106,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,107,107,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,108,108,CBAR,0.160,62.8632,6.28625,,SHAFT
DESELM,109,109,CBAR,0.160,62.8632,6.28625,,SHAFT
$
DCONVMM,2,165.0E6,165.0E6,1281004.,2
DCONVMM,3,165.0E6,165.0E6,2624776.,3
$
```

```
FORCE,1,110,,188.04E6,-1.0,0.0,0.0
$
CONM2,41,110,,3.9345E9
CONM2,42,100,,3.9365E9
$
PBAR,100,1,12.4525,6792.71,6792.71,13585.41
PBAR,101,2,1.,2532.20,2532.20,5064.39
PBAR,102,3,1.,31434.75,31434.75,62869.50
$
MAT1,1,7.036E10,,0.33,2650.0
MAT1,2,7.036E10,,0.33,2650.0
MAT1,3,7.036E10,,0.33,2650.0
$
MODELIST,100,1,THRU,100
$
SPC,250,100,123456
EIGR,20,MGIV,0.00001,1000.0,,,,,123
+23,MASS
$
$
ENDDATA
```

SAMPLE ASTROS INPUT FILE-ANALYSIS

```
ASSIGN DATABASE PROJECT WISH TEMP
SOLUTION
TITLE=COMPLETE MODEL -- TRUE DIMENSION
   ANALYZE
      PRINT DISPLACEMENTS(MODE=100)=ALL
      BOUNDARY SPC=250, METHOD=20
      MODES
   END
BEGIN BULK NOECHO
$2345678123456781234567812345678123456781234567812345678123456781234567812345678
$         2         3         4         5         6         7         8         9        10
MFORM,COUPLED
 GRID,  1,,  100.0,     895.0,       0.0
 GRID,  2,,  100.0,     864.5,     231.6
 GRID,  3,,  100.0,     775.1,     447.5
 GRID,  4,,  100.0,     632.9,     632.9
 GRID,  5,,  100.0,     447.5,     775.1
 GRID,  6,,  100.0,     231.6,     864.5
 GRID,  7,,  100.0,       0.0,     895.0
 GRID,  8,,  100.0,    -231.6,     864.5
 GRID,  9,,  100.0,    -447.5,     775.1
 GRID, 10,,  100.0,    -632.9,     632.9
 GRID, 11,,  100.0,    -775.1,     447.5
 GRID, 12,,  100.0,    -864.5,     231.6
 GRID, 13,,  100.0,    -895.0,       0.0
 GRID, 14,,  100.0,    -864.5,    -231.6
 GRID, 15,,  100.0,    -775.1,    -447.5
 GRID, 16,,  100.0,    -632.9,    -632.9
 GRID, 17,,  100.0,    -447.5,    -775.1
 GRID, 18,,  100.0,    -231.6,    -864.5
 GRID, 19,,  100.0,       0.0,    -895.0
 GRID, 20,,  100.0,     231.6,    -864.5
 GRID, 21,,  100.0,     447.5,    -775.1
 GRID, 22,,  100.0,     632.9,    -632.9
 GRID, 23,,  100.0,     775.1,    -447.5
 GRID, 24,,  100.0,     864.5,    -231.6      .
$
$ -- Grid 25 is the connecting point between the shaft and the torus
GRID,25,,100.0,0.0,0.0
$
$ -- Grid points for the spokes
 GRID,26,,  100.0,     223.8,       0.0
 GRID,27,,  100.0,     447.5,       0.0
 GRID,28,,  100.0,     671.3,       0.0
 GRID,29,,  100.0,     111.9,     193.8
 GRID,30,,  100.0,     223.8,     387.5
 GRID,31,,  100.0,     335.6,     581.3
 GRID,32,,  100.0,    -111.9,     193.8
 GRID,33,,  100.0,    -223.8,     387.5
 GRID,34,,  100.0,    -335.6,     581.3
 GRID,35,,  100.0,    -223.8,       0.0
 GRID,36,,  100.0,    -447.5,       0.0
 GRID,37,,  100.0,    -671.3,       0.0
 GRID,38,,  100.0,    -111.9,    -193.8
 GRID,39,,  100.0,    -223.8,    -387.5
 GRID,40,,  100.0,    -335.6,    -581.3
 GRID,41,,  100.0,     111.9,    -193.8
 GRID,42,,  100.0,     223.8,    -387.5
 GRID,43,,  100.0,     335.6,    -581.3
$
```

```
$ -- Grid points for the shaft
  GRID,100,,      0.0,0.0,0.0
  GRID,101,,     20.0,0.0,0.0
  GRID,102,,     40.0,0.0,0.0
  GRID,103,,     60.0,0.0,0.0
  GRID,104,,     80.0,0.0,0.0
  GRID,106,,    120.0,0.0,0.0
  GRID,107,,    140.0,0.0,0.0
  GRID,108,,    160.0,0.0,0.0
  GRID,109,,    180.0,0.0,0.0
  GRID,110,,    200.0,0.0,0.0
$
$
CBAR,1,100,1,2,1.0,0.0,0.0
CBAR,2,100,2,3,1.0,0.0,0.0
CBAR,3,100,3,4,1.0,0.0,0.0
CBAR,4,100,4,5,1.0,0.0,0.0
CBAR,5,100,5,6,1.0,0.0,0.0
CBAR,6,100,6,7,1.0,0.0,0.0
CBAR,7,100,7,8,1.0,0.0,0.0
CBAR,8,100,8,9,1.0,0.0,0.0
CBAR,9,100,9,10,1.0,0.0,0.0
CBAR,10,100,10,11,1.0,0.0,0.0
CBAR,11,100,11,12,1.0,0.0,0.0
CBAR,12,100,12,13,1.0,0.0,0.0
CBAR,13,100,13,14,1.0,0.0,0.0
CBAR,14,100,14,15,1.0,0.0,0.0
CBAR,15,100,15,16,1.0,0.0,0.0
CBAR,16,100,16,17,1.0,0.0,0.0
CBAR,17,100,17,18,1.0,0.0,0.0
CBAR,18,100,18,19,1.0,0.0,0.0
CBAR,19,100,19,20,1.0,0.0,0.0
CBAR,20,100,20,21,1.0,0.0,0.0
CBAR,21,100,21,22,1.0,0.0,0.0
CBAR,22,100,22,23,1.0,0.0,0.0
CBAR,23,100,23,24,1.0,0.0,0.0
CBAR,24,100,24,1,1.0,0.0,0.0
$
$ -- CBARs for the spokes
CBAR,25,101,25,26,1.0,0.0,0.0
CBAR,26,101,26,27,1.0,0.0,0.0
CBAR,27,101,27,28,1.0,0.0,0.0
CBAR,28,101,28,1,1.0,0.0,0.0
CBAR,29,101,25,29,1.0,0.0,0.0
CBAR,30,101,29,30,1.0,0.0,0.0
CBAR,31,101,30,31,1.0,0.0,0.0
CBAR,32,101,31,5,1.0,0.0,0.0
CBAR,33,101,25,32,1.0,0.0,0.0
CBAR,34,101,32,33,1.0,0.0,0.0
CBAR,35,101,33,34,1.0,0.0,0.0
CBAR,36,101,34,9,1.0,0.0,0.0
CBAR,37,101,25,35,1.0,0.0,0.0
CBAR,38,101,35,36,1.0,0.0,0.0
CBAR,39,101,36,37,1.0,0.0,0.0
CBAR,40,101,37,13,1.0,0.0,0.0
CBAR,41,101,25,38,1.0,0.0,0.0
CBAR,42,101,38,39,1.0,0.0,0.0
CBAR,43,101,39,40,1.0,0.0,0.0
CBAR,44,101,40,17,1.0,0.0,0.0
CBAR,45,101,25,41,1.0,0.0,0.0
```

```
CBAR,46,101,41,42,1.0,0.0,0.0
CBAR,47,101,42,43,1.0,0.0,0.0
CBAR,48,101,43,21,1.0,0.0,0.0
$
$ -- CBABs for the shaft
CBAR,100,102,100,101,0.0,0.0,1.0
CBAR,101,102,101,102,0.0,0.0,1.0
CBAR,102,102,102,103,0.0,0.0,1.0
CBAR,103,102,103,104,0.0,0.0,1.0
CBAR,104,102,104,25,0.0,0.0,1.0
CBAR,105,102,25,106,0.0,0.0,1.0
CBAR,106,102,106,107,0.0,0.0,1.0
CBAR,107,102,107,108,0.0,0.0,1.0
CBAR,108,102,108,109,0.0,0.0,1.0
CBAR,109,102,109,110,0.0,0.0,1.0
$
$
$
CONM2,41,110,,3.9345E9
CONM2,42,100,,3.9365E9
$
PBAR,100,1,12.4525,6792.71,6792.71,13585.41
PBAR,101,2,0.1256,25.12045,25.12045,50.24090
PBAR,102,3,1.13964,904.78,904.78,1809.56
$
MAT1,1,7.036E10,,0.33,2650.0
MAT1,2,7.036E10,,0.33,2650.0
MAT1,3,7.036E10,,0.33,2650.0
$
MODELIST,100,1,THRU,50
$
SPC,250,100,123456
EIGR,20,MGIV,0.00001,1000.0,,,,,123
+23,MASS
$
$
ENDDATA
```

DERIVATION OF a_0, ACCELERATION DUE TO THRUSTING

Now from $F = m_S a_0$, where m_S is the elemental mass vector, and a_0 is the rigid body acceleration at t , the derivation of a_0 is as follows:

$$F_{TH} = m_p v_{ex} = \sqrt{2 m_p W} \tag{C.1}$$

where $W = \dfrac{m_p v_{ex}^2}{2}$ (the propulsive power). $\hspace{3cm}$ (C.2)

v_{ex} is the exit velocity of the propellant, and $m_p = \dfrac{m_{PTO}}{t_{PR}}$ (mass of propellant at takeoff) .

(burn time)

$$F_{TH} = \sqrt{\frac{2 m_{PTO} W}{t_{PR}}} = \sqrt{\frac{2 m_{PTO} m_W W}{t_{PR}}} \tag{C.3}$$

where m_W is the power system mass. Let α be the ratio of propellant power to propellant mass;

i.e. $\alpha = \dfrac{W}{m_W}$. Also m_S is changing during thrusting time from the expended propellant; so :

$$m_S = m_{OTO} - \frac{m_{PTO} \, t}{t_{PR}} \tag{C.4}$$

where m_{OTO} is the total initial mass. Therefore:

$$a_0(t) = \frac{F_{TH}}{m_S} = \sqrt{\frac{2 m_{PTO} m_W \alpha}{t_{PR} \left(m_{OTO} - \dfrac{m_{PTO} \, t}{t_{PR}} \right)^2}} \tag{C.5}$$

C.11

```
c          Program  inertia (E-CITY)      ROBERT HELLSTROM

c          *Torus & Spokes Spin Only*     AAE 515/416 s

c             -2 spherical tanks-

c

c       6    sec run time on pc 10 mhz    14 May 1992

c

c   ------------------------------------------------------------

c

      PROGRAM INERTSP3
      IMPLICIT REAL (A-I,L-Z)
      INTEGER Nspk
      DATA PI/3.14159265/
26    OPEN (1,FILE='INERTSP3.DAT',STATUS='NEW')
      OPEN (2,FILE='ECITY3.DAT',STATUS='OLD')
      WRITE(*,700)
700   FORMAT(5X,'**PROGRAM (E-CITY) INERTIA "TANKS DESPUN"**'
     ,//2X,'The program will compute the mass moments of inertia'
     ,/2X,'MMIs(kg m^2) for several spherical propellant tank axis '
     ,/2X,'locations given a certain mass of liquid hydrogen')
      WRITE(*,800)
800   FORMAT(2X,'propellant for both the control and the'
     ,/2X,'primary tanks.  The support fuel mass is also needed'
     ,/2X,'in addition to several other variables. With the exception'
     ,/2X,'of the torus and shuttle all other structural components'
     ,/2X,'are modeled as hollow spheres or cylinders.')
      WRITE(*,900)
900   FORMAT(2X,'The data is read from a data file or'
     ,/2X,'keyboard. The output is required for a stability and'
     ,/2X,'control analysis utilizing a program such as MATLAB.')
      WRITE(*,901)
901   FORMAT(//5X,'The following configurations will/can be analyzed:'
     ,/2X,'I.   Shuttle (Orbiter type) docked'
     ,/2X,'II.  Shuttle NOT docked'
     ,/2X,'III. Spokes filled with propellant '
     ,/2X,'IV.    "  NOT  "    "     "'
     ,/2X,'V.   All propellant and support fuel tanks are empty')
      WRITE(*,902)
902   FORMAT(/15X,'TYPE "1" TO RUN PROGRAM INERTIA')
      READ(*,*) GO
c
c  The following data is required:
c  1) Mass of the torus & shield if desired (Mt)
c  2)   "   "   "    engines (Mw)
c  3)   "   "   "    shuttle (Mshu)
c  4)   "   "   "    torus shield (Msh)
c  5)   "   "   "    LH2 primary propellant (Mpp)
c  6)   "   "   "    LH2 control     "    (Mpc)
c  7)   "   "   "    support fuel (Ms)
c  8) Radius of torus: from hub to center of cross section (R)
c  9)    "    "  torus cross section (Rmin)
c 10)    "    "  engine cluster (Rw)
c 11)    "    "  propellant & control tank connecting pole (Rpl)
c 12)    "    "  shuttle (Rshu)
c 13)    "    "  Spoke (Rspk)
c 14) Height (or length) of engine cluster (Hw)
```

```
c 15)    "              " primary tank pole (Hppi): initial length
c 16)    "              " control  "    " (Hpc):  stays constant
c 17)    "              " shuttle (Hshu)
c 18) Number of spokes (Nspk): Be sure they will fit on hub and torus!
c 19) Thickness of Torus Shield: assumed LH2 (Tsh)
c 20)    "       " both pole walls (Tpl)
c 21)    "       " propellant tank wall (Tp)
c 22)    "       " spoke wall (Tspk)
c 23) Density of liquid hydrogen (ROLH2)
c 24)    "    " support fuel (ROS)
c 25)    "    " hardened aluminum (ROAl)
c
c Read required data line by line from the previously generated file.
c Keyboard interactive input is also an option.
C #################################################################
      READ(2,*) Mt,Mw,Mshu,Msh
      READ(2,*) Mpp,Mpc,Ms
      READ(2,*) R,Rw,Rmin,Rpl,Rshu,Rspk
      READ(2,*) Hw,Hppi,Hpc,Hshu
      READ(2,*) Tsh,Tpl,Tp,Tspk
      READ(2,*) ROLH2,ROS,ROAl
      READ(2,*) Nspk
92    WRITE(*,1) Mt
      WRITE(*,2) Mw
      WRITE(*,3) Mshu
      WRITE(*,4) Msh
      WRITE(*,5) Mpp
      WRITE(*,6) Mpc
      WRITE(*,7) Ms
      WRITE(*,8) R
      WRITE(*,9) Rw
      WRITE(*,10) Rmin
      WRITE(*,11) Rpl
      WRITE(*,12) Rshu
      WRITE(*,45)
      READ(*,*) KNL
      IF(KNL.EQ.0) GOTO 46
      GOTO 55
46    WRITE(*,94)
94    FORMAT(//2X,'HIT ANY KEY FOR MORE DATA')
      READ(*,*)
95    WRITE(*,13) Rspk
      WRITE(*,14) Hw
      WRITE(*,15) Hppi
      WRITE(*,16) Hpc
      WRITE(*,17) Hshu
      WRITE(*,18) Tsh
      WRITE(*,19) Tpl
      WRITE(*,20) Tp
      WRITE(*,21) Tspk
      WRITE(*,22) ROLH2
      WRITE(*,23) ROS
      WRITE(*,24) ROAl
      WRITE(*,25) Nspk
```

D.2

```
 1      FORMAT(2X,'1) Torus mass (kg): Mt= ',E20.6)
 2      FORMAT(2X,'2) Total engine mass (kg): Mv= ',E20.6)
 3      FORMAT(2X,'3) Shuttle mass (kg): Mshu= ',E20.6)
 4      FORMAT(2X,'4) Torus shield mass (kg): Msh= ',E20.6)
 5      FORMAT(2X,'5) Primary propellant mass (kg): Mpp= ',E20.6)
 6      FORMAT(2X,'6) Control propellant mass (kg): Mpc= ',E20.6)
 7      FORMAT(2X,'7) Support fuel mass (kg): Ms= ',E20.6)
 8      FORMAT(2X,'8) Torus radius (m): R= ',F15.3)
 9      FORMAT(2X,'9) Main engine cluster radius (m): Rv= ',F15.3)
10      FORMAT(2X,'10) Torus minor radius (m): Rmin= ',F15.3)
11      FORMAT(2X,'11) Connecting pole radius (m): Rpl= ',F15.3)
12      FORMAT(2X,'12) Wing span of shuttle (m): Rshu= ',F15.3)
13      FORMAT(2X,'13) Spoke radius (m): Rspk= ',F15.3)
14      FORMAT(2X,'14) Engine length (m): Hv= ',F15.3)
15      FORMAT(2X,'15) Initial primary pole length (m): Hppi=',F15.3)
16      FORMAT(2X,'16) Connecting pole length (m): Hpc= ',F15.3)
17      FORMAT(2X,'17) Shuttle length (m): Hshu= ',F15.3)
18      FORMAT(2X,'18) Torus shield thickness (m): Tsh= ',F15.3)
19      FORMAT(2X,'19) Connecting pole wall thickness (m): Tpl= ',F15.3)
20      FORMAT(2X,'20) Propellant tank wall thicknesses (m): Tp= ',F15.3)
21      FORMAT(2X,'21) Spoke wall thickness (m): Tspk= ',F15.3)
22      FORMAT(2X,'22) Density of liquid hydrogen (kg/m3): ROLH2= ',F15.3)
23      FORMAT(2X,'23) Density of support fuel (kg/m3): ROS= ',F15.3)
24      FORMAT(2X,'24) Density of aluminum walls (kg/m3): ROAl= ',F15.3)
25      FORMAT(2X,'25) Number of spokes: Nspk= ',I4)
        WRITE(*,45)
45      FORMAT(15X,'TYPE NUMBER OF VARIABLE TO MODIFY OR "0" IF OK')
        READ(*,*) KNL
        IF(KNL.EQ.0) GOTO 56
55      IF(KNL.EQ.1) GOTO 1001
        IF(KNL.EQ.2) GOTO 1002
        IF(KNL.EQ.3) GOTO 1003
        IF(KNL.EQ.4) GOTO 1004
        IF(KNL.EQ.5) GOTO 1005
        IF(KNL.EQ.6) GOTO 1006
        IF(KNL.EQ.7) GOTO 1007
        IF(KNL.EQ.8) GOTO 1008
        IF(KNL.EQ.9) GOTO 1009
        IF(KNL.EQ.10) GOTO 1010
        IF(KNL.EQ.11) GOTO 1011
        IF(KNL.EQ.12) GOTO 1012
        IF(KNL.EQ.13) GOTO 1013
        IF(KNL.EQ.14) GOTO 1014
        IF(KNL.EQ.15) GOTO 1015
        IF(KNL.EQ.16) GOTO 1016
        IF(KNL.EQ.17) GOTO 1017
        IF(KNL.EQ.18) GOTO 1018
        IF(KNL.EQ.19) GOTO 1019
        IF(KNL.EQ.20) GOTO 1020
        IF(KNL.EQ.21) GOTO 1021
        IF(KNL.EQ.22) GOTO 1022
        IF(KNL.EQ.23) GOTO 1023
        IF(KNL.EQ.24) GOTO 1024
        IF(KNL.EQ.25) GOTO 1025
```

```
1001   WRITE(*,1)
       READ(*,*)Mt
       GOTO 92
1002   WRITE(*,2)
       READ(*,*)Mw
       GOTO 92
1003   WRITE(*,3)
       READ(*,*)Mshu
       GOTO 92
1004   WRITE(*,4)
       READ(*,*)Msh
       GOTO 92
1005   WRITE(*,5)
       READ(*,*)Mpo
       GOTO 92
1006   WRITE(*,6)
       READ(*,*)Mpc
       GOTO 92
1007   WRITE(*,7)
       READ(*,*)Ms
       GOTO 92
1008   WRITE(*,8)
       READ(*,*)R
       GOTO 92
1009   WRITE(*,9)
       READ(*,*)Rv
       GOTO 92
1010   WRITE(*,10)
       READ(*,*)Rmin
       GOTO 92
1011   WRITE(*,11)
       READ(*,*)Rpl
       GOTO 92
1012   WRITE(*,12)
       READ(*,*)Rshu
       GOTO 92
1013   WRITE(*,13)
       READ(*,*)Rspk
       GOTO 95
1014   WRITE(*,14)
       READ(*,*)Hv
       GOTO 95
1015   WRITE(*,15)
       READ(*,*)Hppi
       GOTO 95
1016   WRITE(*,16)
       READ(*,*)Hpc
       GOTO 95
1017   WRITE(*,17)
       READ(*,*)Hshu
       GOTO 95
1018   WRITE(*,18)
       READ(*,*)Tsh
       GOTO 95
```

D.4

```fortran
1019  WRITE(*,19)
      READ(*,*)Tpl
      GOTO 95
1020  WRITE(*,20)
      READ(*,*)Tp
      GOTO 95
1021  WRITE(*,21)
      READ(*,*)Tspk
      GOTO 95
1022  WRITE(*,22)
      READ(*,*)ROLH2
      GOTO 95
1023  WRITE(*,23)
      READ(*,*)ROS
      GOTO 95
1024  WRITE(*,24)
      READ(*,*)ROA1
      GOTO 95
1025  WRITE(*,25)
      READ(*,*)Nspk
      GOTO 95
C ****************************************************************
c *Propellant (Jspk=2) or none (Jspk=1) in spokes: chose one
56    Jspk=1
      WRITE(*,60)
60    FORMAT(//2X,') Do you wish to put propellant in the spokes?',/2X,
     ,'Type "1" for no or "2" for yes')
      READ(*,*) Jspk
c *
c *Empty configuration ( Mpp=Mpc=Ms= 0 ) and Jempty = 3 else = 1
c *
      Jempty=1
      WRITE(*,62)
62    FORMAT(//2X,') Finally, type "3" if you wish to analyze the tanks
     ,empty case "0" if not.')
      READ(*,*) Jempty
c
      WRITE(1,105)
105   FORMAT(15X,'*MASS MOMENTS OF INERTIA (MMI) & RATIOS*')
      WRITE(1,110)
110   FORMAT(13X,'-Spherical Propellant(2) and Support(1) Tanks-',
     ,/23X,'(Ix,y,z =) kg m^2 * 10^+14)',//30x,'*Shuttle docked*')
      WRITE(*,105)
      WRITE(*,110)
      Hspk = R - Rmin
      Mspkw = 2.*PI*Rspk*Hspk*Tspk*ROA1
c
c Calculate spherical propellant tank volumes (Vpw,Vpctrl,Vpsupp),
c and, spoke volume (Vspk) and mass (Mspk), required for the given
c masses of liquid hydrogen and support fuel.
c
      V = Mpp/ROLH2
      Vspk = PI*Rspk*Rspk*Hspk
      Mspk = Vspk*ROLH2 + Mspkw
```

D.5

```
          IF(Jspk.EQ.1) Mspk = Mspkw
          Vpp = V = Nspk*Vspk
          IF(Jspk.EQ.1) Vpp=V
   c The percentage of fuel in the spokes
          VspkV = 100.*Vspk*Nspk/V
          IF(Jspk.EQ.1) VspkV = 0.
          IF(Jempty.EQ.3) VspkV = 0.
          Mpp = Vpp*ROLH2
          Vpc = Mpc/ROLH2
          Vs = Ms/ROS
   c Calculate the radius of the propellant and support fuel filled spheres
   c (Rpp,Rpc,and Rps).
          Rpp = (3.*Vpp/4./PI)**(1./3.)
          Rpc = (3.*Vpc/4./PI)**(1./3.)
   c Check to see if all tanks are empy ( set Mpp=Mpc=Ms = 0. )
          IF(Jempty.NE.3) GOTO 114
          Mpp = 0.
          Mpc = 0.
          Ms = 0.
   c Calculate the mass of the primary and control tank,
   c and control tank connecting pole.
   114    Mppw = 4.*PI*Rpp*Rpp*Tp*ROAl
          Mpcw = 4.*PI*Rpc*Rpc*Tp*ROAl
          Mhpcw = 2.*PI*Rpl*Hpc*Tpl*ROAl
          WRITE(1,115)VspkV,Nspk,Mt,Mpp,Vpp,Rpp,Mpc,Vpc,Rpc,Ms,Vs,Hpc,Rpl
   115    FORMAT(10x,'The percentage of propellant in the spokes is: ',
         ,F8.4,/10x,'The number of spokes is: ',I4,/10x,
         ,'The Mass of the torus is: ',E12.4,'kg',/10x,
         ,'The Mass of primary propellant is: ',E12.4,'kg',/10x,
         ,'The Volume for the primary tank is: ',E12.4,'m^2',/10x,
         ,'The Radius of the primary tank is: ',F9.3,'m',/10x,
         ,'The Mass of the control propellant is: ',E12.4,'kg',/10x,
         ,'The Volume for the control tank is: ',E12.4,'m^2',/10x,
         ,'The Radius of the control tank is: ',F9.3,'m',/10x,
         ,'The Mass of the support fuel is: ',E12.4,'kg',/10x,
         ,'The Volume for the support fuel tank is: ',E12.4,'m^2',/10x,
         ,'The Length of the control tank connecting pole is: 'F9.3,'m',
         ,/10x,'The Radius for both connecting poles is: ',F9.3,'m',/)
          WRITE(*,115)VspkV,Nspk,Mt,Mpp,Vpp,Rpp,Mpc,Vpc,Rpc,Ms,Vs,Hpc,Rpl
          WRITE(*,117)
   117    FORMAT(//8X,'TYPE "1" TO VIEW MORE DATA',//)
          READ(*,*) MORE
          WRITE(1,120)
   120    FORMAT(5X,'Hpp (m)',3x,'Zcm(m)',3x,'Ix & Iy',6X,'Iz',
         ,7x,'Iz/Ix',3x,'Izspn/Ix',2x,'Ixspn/Ix')
          WRITE(*,120)
   c
   c The centroid (Xspk) of half of the spokes combined is required
   c to calculate the MMI's Ix & Iy. Mspk includes wall mass.
   c
          Jnspk = Nspk/2 + 1
          SumIM = 0.
          Theta = 0.
          DO 150 JJ=1,Jnspk
```

D.6

```fortran
          Xspk = (Rpl + Hspk/2.)*SIN(Theta)
          SumXM = SumXM + Xspk*Mspk
          Theta = Theta + 2.*PI/Nspk
150   CONTINUE
      SumMsp = Nspk*Mspk/2.
      Xspk = SumXM/SumMsp
c
c The section MMI's are calculated for all sections except the
c primary connecting pole, which changes. This is done before
c the loop to reduce run time.
c
      Ixwo = Mw*(3.*Rw*Rw + Hw*Hw)/12.
      Ixshuo = Mshu*(3.*Rshu*Rshu + Hshu*Hshu)/12.
      Ixspko = Nspk*Mspk*Xspk*Xspk
      Ixto = 4.*(Mt + Msh)*(R + Rpl)*(R + Rpl)/PI/PI
c Exact formula for hollow sphere is not used: I=2(r^5-ri^5)/(r^3-ri^3)/5
c This formula may cause arithmetic problems with such large numbers.
c However a good approx. for very thin shells is I=2.*(r^2)/3.
      Ixppo = 2.*Mpp*Rpp*Rpp/5. + 2.*Mppw*Rpp*Rpp/3.
      Ixpco = 2.*Mpc*Rpc*Rpc/5. + 2.*Mpcw*Rpc*Rpc/3.
      Ixso = 4.*Ms*Rpl*Rpl/PI/PI
      Ixhpco = Mhpcw*(3.*((Rpl+Tpl)*(Rpl+Tpl) + Rpl*Rpl) + Hpc*Hpc)/12.
c In addition to the Shuttle docked and undocked cases, fifteen different
c propellant tank locations (Hpp changes) are considered.
c The connecting pole initial minimum length.
      Hpp = Hppi
      DO 500 J=1,2
        IF(J.EQ.1) GOTO 250
        Write(1,200)
        Write(*,200)
200     Format(//30x,'*Shuttle Undocked*')
        Write(1,120)
        WRITE(*,117)
        READ(*,*) MORE
        Write(*,120)
        Mshu=0.
        Ixshuo=0.
250     DO 400 K=1,15
c Calculate the mass of the primary connecting pole wall.
        Mhppw = 2.*PI*Rpl*Hpp*Tpl*ROA1
c
c The moment "arms" along the axis relative to the torus centriod
c are determined for each section. Note the shuttle is located half
c the control pole length forevard of the torus center.
c
          Zw = -(Hpp + 2.*Rpp + Hw/2.)
          Zpp = Zw + Rpp + Hw/2.
          Zhpp = -Hpp/2.
          Zhpc = Hpc/2.
          Zshu = 0.
          IF(J.EQ.2) Zshu=0.
          Zpc = Hpc + Rpc
          Zspk = 0.
          Zt = 0.
```

D.7

```fortran
              Zs = 0.
      c
      c The axis location of the vehicle's mass center (Zcm) is determined:
      c
              SumZM = Zv*Mv + Zpp*(Mpp+Mppv) + Zhpp*Mhppv + Zshu*Mshu
              SumZM = SumZM + Zpc*(Mpc+Mpcv) + Zhpc*Mhpcv
              SumM = Msh+Mt+Mv+Mpp+Mppv+Mhppv+Mshu+Mpc+Mpcv+Mhpcv+Ms
              SumM = SumM + Nspk*(Mspk)
              Zcm = SumZM/SumM
      c
      c The global MMI's (Ix=Iy and Iz) are calculated:
      c
              Izpp = 2.*Mpp*Rpp*Rpp/5. + 2.*Mppv*Rpp*Rpp/3.
              Izpc = 2.*Mpc*Rpc*Rpc/5. + 2.*Mpcv*Rpc*Rpc/3.
              Izhpp = Mhppv*((Rpl+Tpl)*(Rpl+Tpl) + Rpl*Rpl)/2.
              Izhpc = Mhpcv*((Rpl+Tpl)*(Rpl+Tpl) + Rpl*Rpl)/2.
              Izs = Ms*Rpl*Rpl
              Izshu = Mshu*(Rpl+Rshu/2.)*(Rpl+Rshu/2.)
              Izt = (Mt + Msh)*(R + Rpl)*(R + Rpl)
              Izv = Mv*Rv*Rv/2.
              Izspk = Nspk*Mspk*(Rpl + Hspk/2.)*(Rpl + Hspk/2.)
      c The local value of Ix for the primary and connecting pole.
              Ixhppo = 3.*((Rpl+Tpl)*(Rpl+Tpl) + Rpl*Rpl)
              Ixhppo = Mhppv*(Ixhppo + Hpp*Hpp)/12.
              Ixhpco = 3.*((Rpl+Tpl)*(Rpl+Tpl) + Rpl*Rpl)
              Ixhpco = Mhpcv*(Ixhpc0 + Hpc*Hpc)/12.
      c The Parallel-Axis Theorem is incorporated (note that Zcm is < 0).
              Ixv = Ixvo + Mv*(Zv - Zcm)*(Zv - Zcm)
              Ixhpp = Ixhppo + Mhppv*(Zhpp - Zcm)*(Zhpp - Zcm)
              Ixhpc = Ixhpco + Mhpcv*(Zhpc - Zcm)*(Zhpc - Zcm)
              Ixs = Ixso + Ms*Zcm*Zcm
              Ixshu = Ixshuo + Mshu*(Zshu - Zcm)*(Zshu - Zcm)
              Ixpp = Ixppo + (Mpp + Mppv)*(Zpp - Zcm)*(Zpp - Zcm)
              Ixpc = Ixpco + (Mpc + Mpcv)*(Zpc - Zcm)*(Zpc - Zcm)
              Ixt = Ixto + (Mt + Msh)*Zcm*Zcm
              Ixspk = Ixspko + Nspk*Mspk*(Xspk*Xspk + Zcm*Zcm)
      c Non-dimensionalized MMI's.
              Ix = (Ixv+Ixhpp+Ixhpc+Ixs+Ixshu+Ixpp+Ixpc+Ixt+Ixspk)/1.E+14
              Iz = (Izv+Izhpp+Izhpc+Izs+Izshu+Izpp+Izpc+Izt+Izspk)/1.E+14
      c The MMI's of the spinning parts, Ixspn and Izspn are found.
              Ixspn = (Ixt + Ixspk)/1.E+14
              Izspn = (Izt + Izspk)/1.E+14
      c Calculation of the total and spin slenderness ratios, rtot and r1.
              rtot = Iz/Ix
              r1 = Izspn/Ix
      c Calculation of the ratio r2.
              r2 = Ixspn/Ix
      c Increment for the length of the connecting pole and output.
              Write(1,300) Hpp,Zcm,Ix,Iz,rtot,r1,r2
              Write(*,300) Hpp,Zcm,Ix,Iz,rtot,r1,r2
      300     Format(5x,F7.2,,F9.3,1x,F11.4,1x,F10.4,1x,3F9.6)
              Hpp = Hpp + 100.
      400     CONTINUE
              Hpp = Hppi
```

```fortran
        WRITE(*,117)
        READ(*,*) MORE
500     CONTINUE
        WRITE(*,600)
600     FORMAT(//2X,'An output data file "INERTSP3.DAT" may now be'
       ,/2X,'accessed for this run; EDIT to view with Pathainder.')
        WRITE(*,650)
650     FORMAT(//15X,'>>Type "0" to QUIT  or  "1" to RUN again<<')
        READ(*,*) RUN
        IF(RUN.EQ.1) GOTO 26
        STOP
        END
```

```
3.5992E+7,2.244E+7,1.5E+5,1.397E+9
10.21129E+9,10.21129E+9,9.3E+8
794.6,50.,32.3,100.,15.,20.
10.,100.,100.,40.
14.,0.1,0.1,0.1
71.,1410.,2710.
6
c  Data file for program inertia; see program
c  for variable definitions.
c  Mt,Mv,Mshu,Msh,Mpp,Mpc,Ms
c  R,Rv,Rmin,Rpl,Rshu,Rspk
c  Hv,Hppi,Hpc,Hshu
c  Tsh,Tpl,Tp,Tspk
c  ROLh2,ROS,ROAl
c  Nspk
```

```
% PROGRAM control.m

% This program will compute the attitude control requirements for
% the Emerald City with a given geometry for various control
% weighting parameters, wc and wx.

clear

%% r1 is ratio of torus Iz to ECITY Ix %%

r1=.789894;

%% r2 is ratio of torus Ix to ECITY Ix %%

r2=.320189;

%% ncl is the number of control thruster clusters %%

ncl=50;
rtank=237.0;
dtorus=1789.2;
inertx=16.8032e+14;
nspin=1.0;
vex=4370.;
omega=2.*pi*nspin/60.;

a=[    0.,       0.,          1.,          0.;
       0.,       0.,          0.,          1.;
   r2-r1,       0.,          0.,    2.*r2-r1;
       0.,    r2-r1,   r1-2.*r2,         0.];

b=[0., 0.;
   0., 0.;
   1., 0.;
   0., 1.];


%Initial disturbance matrix: dimensional displs., non-dimensional rates
x0=[0.0;0.0;0.1;0.1];

c=eye(4);

n=4;
m=2;
nm=4;

wx=375;

for in=1:19
    wc=100+(in-1)*50;
```

```
%wc=640
q=wx*eye(n);
r=wc*eye(m);
[g,s]=lqr(a,b,q,r);
    check=s*a+a'*s-s*b*inv(r)*b'*s+q;
acl=a-b*g;

% This following section is for simulating the results of EMACS program
% via MATLAB routines
% n is the number of states
% m is the number of inputs
% nm is the number of outputs

ti=0.;tf=30.;
stepsize=0.05;
t=[ti:stepsize:tf]';
u=zeros(length(t),m);

%non-dimensional torque input is assigned to u matrix
%u(1,1)=10;

delt=stepsize;
aa=acl;
bb=zeros(n,m);
cc=c;
dd=zeros(nm,m);

% open loop analysis
%aa=a;
%bb=b;

[y,x]=lsim(aa,bb,cc,dd,u,t,x0);
theta1=y(:,1);
theta2=y(:,2);
theta1dot=y(:,3).*omega;
theta2dot=y(:,4).*omega;

% end of simulation

% Determination of accelerations on torus

xdottrans=acl*y';
xdot=xdottrans';
theta1dbldot=xdot(:,3).*omega^2;
theta2dbldot=xdot(:,4).*omega^2;

% Calls to subroutine accel2

accel2

% Initialization of thruster distribution matrix

ntheta=2*pi/ncl;
for i=1:ncl
    ntheta2=ntheta*(i-1);
    dist(1,i)=abs(rtank/dtorus*cos(ntheta2));
    dist(2,i)=abs(rtank/dtorus*sin(ntheta2));
end

% Calculation of attitude control requirements
```

```
        dinv=pinv(dist);
        pstar=lyap(acl,g'*dinv'*dinv*g);
        xx0=x0*x0';
        sstar=sum(diag(pstar*xx0));
        tau=g*x';
        nf=dinv*tau;
        sumnf=sum(abs(nf'));
        nmp=sum(sumnf)*delt;
        mp=inertx*omega/dtorus/vex*nmp;
        actf=inertx*omega^2/dtorus*nf;
        maxf=max(abs(max(abs(actf))));

        % Calculation of rms Power given non-dimensional settling time ndt

        rmsp=vex*inertx*omega^2/2/dtorus*sqrt(sstar/ndt);

        % Stores theta responses for various values of control weighting, wc

        ptheta1(:,in)=theta1;
        ptheta1dot(:,in)=theta1dot;
        ptheta2(:,in)=theta2;
        ptheta2dot(:,in)=theta2dot;

        pwx(in,1)=wx
        pwc(in,1)=wc;
        pmp(in,1)=mp;
        pmaxf(in,1)=maxf;
        pneng(in,1)=maxf/2579968.5;
        psetime(in,1)=ndt/omega;
        prmspower(in,1)=rmsp;
        eigacl=eig(acl);
        pzovershoot(in,1)=max(vap(:,3));

end

plot(pwc,pmaxf);
grid;
xlabel('wc');
ylabel('maxf (N)');
title('max thrust per cluster vs. wc : wx=125')
meta lewey

plot(pwc,pmp);
grid;
xlabel('wc');
ylabel('mp (Kg)');
title('propellant mass vs. wc : wx=125')
meta

plot(pwc,pmaxf,pwc,pmp);
grid;
xlabel('wc');
ylabel('mp & maxf');
title('propellant mass & max thrust vs. wc : wx=125')
meta

plot(pwc,pzovershoot);
grid;
xlabel('wc');
```

```
ylabel('zovershoot (m/s^2)');
title('zovershoot vs.wc : wx=125')
meta


% PROGRAM accel2.m

% This program will calculate the acceleration vector, vap, which
% represents the accelerations experienced by a crew member running
% around the torus at an angular rate of thetadotp.

%Calculation of radius vectors
rad=dtorus/2;
nr=length(theta1);
for kz=1:nr
    w(kz,:)=omega*[-theta2(kz,1),theta1(kz,1),1];
    wdot(kz,:)=omega*[-theta2dot(kz,1),theta1dot(kz,1),0];
end
% crew members angular rate

thetadotp=0.00086;

tist=(tf-ti)/stepsize+1;
for ts=1:tist
    time=(ts-1)*stepsize;
    thetap=thetadotp*time;
    rv(ts,1)=rad*cos(thetap);
    rv(ts,2)=rad*sin(thetap);
    rv(ts,3)=0.0;
    rvdot(ts,1)=-thetadotp*rad*sin(thetap);
    rvdot(ts,2)=thetadotp*rad*cos(thetap);
    rvdot(ts,3)=0.0;
    rv2dot(ts,1)=-thetadotp^2*rad*cos(thetap);
    rv2dot(ts,2)=-thetadotp^2*rad*sin(thetap);
    rv2dot(ts,3)=0.0;
end

%Calculation of acceleration at a point on the torus.

v1=wdot;
v2=rv;

% Calls subroutine cross to calculate the cross products of v1 and v2

cross

apprime=v3+rv2dot;
v1=w;
cross
v2=v3;
cross
cent=v3;
v1=2*w;
v2=rvdot;
cross
```

```
corio=v3;

% vap is torus acceleration in vector form

vap=apprime+cent+corio;

magapsqrd=vap(:,1).^2+vap(:,2).^2+vap(:,3).^2;

% magap is the magnitude of the acceleration vector, vap

magap=sqrt(magapsqrd);




% PROGRAM cross.m

%This program will calculate the cross product vector V3 given two vectors V1
%and V2.

v3(:,1)=v1(:,2).*v2(:,3)-v2(:,2).*v1(:,3);
v3(:,2)=v2(:,1).*v1(:,3)-v1(:,1).*v2(:,3);
v3(:,3)=v1(:,1).*v2(:,2)-v2(:,1).*v1(:,2);
```

ACTIVE THERMAL CONTROL PROGRAM

```fortran
      program RADIATOR
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! This program calculates the waste heat generated by the
! Emerald City during its various modes of operation. Also
! accounting for the two possible ship configurations, this
! program then finds the geometries of various types of active
! thermal control systems.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      implicit none
      real whe,whc,upe,upc,rw,qr,mhd,tw,tpe,tpr,tp,pe,pc
      real rpe,rpc,rp,ep,spn,nav,ls,com,sm,rm,ht,stpn
      real fpr,rps,rqp,rqpc,pbe,pbc,a,k,tr,ts,pi,e,tpb
      real length,radius,c,b,ea,aa,d
      integer i,iostat,iread,go,in,ne,nc,conf,mode,rep
      open(unit=3,file='RADIATOR.DAT',status='new')
      !initialize variables.
      k=5.67D-8          !Stefan-Boltzmann constant
      pi=3.141592654
      tr=1150.0          !Radiator surface temparature
      ts=0.0             !Temperature of space
      e=.9               !Emissivity of surface
      go=1               !Loop control variable
      fpr=6./200.        !Fission power ratio. (Pin/Pout)
      spn=22.85          !Ship operation power needs
      !Welcome sequence to go here.(Was not completed.)
      !Common message format stack.
1     format(8X,'Press RETURN to Continue.')
2     format(8X,'What is the ship configuration?'/
     */8X,'1. NLB Engines used solely for propulsion.'/
     *8X,'2. NLB Engines used for BOTH propulsion and attitude
     * control.')
3     format(/8X,'Please Enter a "1" or "2" for the desired
     * configuration.')
4     format(8X,'What operating mode is the ship in currently?'/
     */8X,'1. Phase 1 Startup.'/8X,'2. Phase 2 Startup.'
     */8X,'3. Reactor and Engines Operating Simultaneously.'
     */8X,'4. Shutdown of Main Engines.'
     */8X,'5. Main Reactor Only.')
5     format(/8X,'Please Enter a "1", "2", "3", "4", or "5"
     * for the'/8X,'desired option.')
6     format(8X,'Is the waste heat from the engines already known?'/
     */8X,'1. Yes.'/8X,'2. No.'//8X,'Please Enter "1" or "2".')
7     format(8X,'How many NLB Engines are there?')
8     format(8X,'What is the reactor power of each engine in
     * megawatts?')
9     format(8X,'What is the percentage of waste heat from each
     * engine?')
10    format(8X,'How many NLB Attitude Control Thrusters are there?')
11    format(8X,'What is the reactor power of each thruster
     * in megaWatts?')
```

```fortran
12      format(8X,'What is the percentage of waste heat from each
     * thruster?')
13      format(8X,'What is the reactor power of the startup engine
     * in megawatts?')
14      format(8X,'How much thermal energy is generated by the
     * Main'/8X,'Reactor? (in megawatts)')
15      format(8X,'What is the conversion efficiency of the MHD
     * Power'/8X,'Converter? (in percent)')
16      format(8X,'The Startup Engine does NOT generate enough
     * thermal'/8X,'power to startup the engines. Calculations will
     * not be'/8X,'carried out.')
17      format(8X,'Power required for ship operations is currently
     * 22.85 Megawatts.'/8X,'Do you wish to change this value?'//8X,
     *'1. Yes.'/8X,'2. No'//8X,'Please Enter a "1" or "2".')
18      format(8X,'How much power is allotted to the navigation
     * system (MWe)?')
19      format(8X,'How much power is used by the life support
     * system (MWe)?')
20      format(8X,'How much power is allotted to the communication
     * system (MWe)?')
21      format(8X,'How much power is used for shuttle and maintenance
     * systems (MWe)?')
22      format(8X,'How much power is allotted for research and other
     * systems (MWe)?')
23      format(8X,'How much power is used for the heat transfer
     * system (MWe)?')
100      format(8X,'Calculations cannot be performed at this
     * time')
        !Repeat loop.
        go=1
        dowhile(go.eq.1)
                        !Initial Values of:
        whe=0.          !Waste Heat of Engines
        whc=0.          !Waste Heat of Control Thrusters
        rp=0.           !Reactor Power
        qr=0.           !Waste Heat to be Dissipated
        pbe=0.0         !Percentage bleed-off for engines.
        pbc=0.0         !Percentage bleed-off for thrusters.
        rps=0.0
        stpn=0.         !Start-up Engine Power Needs from Reactor
        tpe=0.0
        !Configuration/Operating Mode input option.
        write(6,2)              !Configuration Prompt.
        write(6,3)
        read(5,*) conf
        write(6,4)              !Operating Mode Prompt.
        write(6,5)
        read(5,*) mode
        !Input Sequence.
        !Main Reactor Input.
        write(6,14)             !Reactor Power Prompt.
        read(5,*) rp
        write(6,15)             !Electric Power Conversion Efficiency.
        read(5,*) mhd
        mhd=mhd/100.0           !Adjust percentage to fraction.
        write(6,17)
        read(5,*) i
        if(i.eq.1) then
          write(6,18)           !Navigation Systems Power.
          read(5,*) nav
```

```fortran
      write(6,19)              !Life Support Power.
      read(5,*) ls
      write(6,20)              !Communications.
      read(5,*) com
      write(6,21)              !Shuttle and Maintenance.
      read(5,*) sm
      write(6,22)              !Reasearch/Misc.
      read(5,*) rm
      write(6,23)              !Heat transfer Systems.
      read(5,*) ht
      spn=nav+ls+com+sm+rm+ht
   else
      spn=22.85
   endif
!Mode option input/processing sequence.
   if(mode.eq.1) then     !Phase 1 Startup Sequence.
      write(6,13)              !Startup Engine Power Prompt.
      read(5,*) rps
      stpn=fpr*rps
      rp=rp-stpn
      whe=0.0
      whc=0.0
   elseif(mode.eq.2) then !Phase 2 Startup.
      write(6,13)              !Startup Engine Power Prompt.
      read(5,*) rps
      write(6,7)               !Number of Engines Prompt.
      read(5,*) ne
      write(6,8)               !Reactor Power of Each Engine.
      read(5,*) rpe
      if(conf.eq.2) then       !Attitude and Control Info..
       write(6,10)             !Number of Thrusters.
       read(5,*) nc
       write(6,11)             !Thruster Reactor Power
       read(5,*) rpc
      endif
      rqp=fpr*(ne-1)*rpe       !Required Power for Fission.
      pbe=rqp/rps
      rqpc=fpr*nc*rpc
      pbc=rqpc/rps
      tpb=pbe+pbc
      if(tpb.gt.1.0) then  !Startup Engine Overload
       write(6,16)
       write(6,1)
       read(5,*)
       go=0
      endif
!"whe" represents waste heat generated by startup engine.
      whe=rps*(1.-tpb)
      whc=0.0
   elseif(mode.eq.3) then !Steady State Conditions.
      write(6,7)               !Number of Engines Prompt.
      read(5,*) ne
      write(6,8)               !Reactor Power of Each Engine.
      read(5,*) rpe
      write(6,9)               !Percentage of Waste Heat
      read(5,*) pe
      pe = pe/100.
       if(conf.eq.2) then      !Attitude and Control Info..
       write(6,10)             !Number of Thrusters.
       read(5,*) nc
```

F.3

```fortran
        write(6,11)              !Thruster Reactor Power
        read(5,*) rpc
        write(6,12)              !Thruster Waste Heat Percentage.
        read(5,*) pc
        pc = pc/100.
      endif
      whe=ne*pe*rpe
      whc=nc*pc*rpc
    elseif(mode.eq.4) then   !Shutdown of Main engines
      write(6,100)    !Informs user this routine will be
                      !will be added later.
      write(6,1)
      read(5,*)
      go=0
    elseif(mode.eq.5) then !Main Engines Off.
      whe=0.0
      if(conf.eq.1) then     !No Thrusters.
       whc=0.0
      elseif(conf.eq.2) then   !Attitude and Control Needed.
       write(6,10)             !Number of Thrusters.
       read(5,*) nc
       write(6,11)             !Thruster Reactor Power.
       read(5,*) rpc
       write(6,12)             !Thruster Waste Heat Percentage.
       read(5,*) pc
       whc=nc*pc*rpc
      endif
    endif
    !Main Calculations
    if(go.ne.0) then          !Calculations can be performed.
      tpr=rp*mhd                !
      rw=rp*(1-mhd)           !Reactor Waste.
      tw=whe+whc              !Waste from engines and thrusters.
      if(mode.eq.3) then
       upe=mhd*whe            !Usable Elec. Power from Engine Waste Heat
       upc=mhd*whc            !Usable Elec. Power from Ctrl. Trusters
       tpe=upe+upc            !Elec. Power Available from previous 2
      endif
      tp=tpe+tpr              !Total Usable Power
      ep=tp-spn               !Available Power After Ship-Op. Needs
      qr=ep+tw+rw             !Total Waste Heat to be dissipated
      a=qr*1E6/(e*k*(tr**4.0-ts**4.0))
      !Convert all percentages for output sequence.
      mhd=mhd*100.
      pe=pe*100.
      pc=pc*100.
      pbe=pbe*100.
      pbc=pbc*100.
      rp=rp+stpn              !Adjusts for specified reactor power.
     c=50           !Calc. of Area of Elipsoid Radiator of Minor Ax. = 50
     b=5            !and Iterate the Major Starting at 5
     aa=a                 !Effective Area = Area Needed + Surf. Area of
                          !Connecting Shaft = 62832
333  d=(c**2+b**2)**.5!Function of c and b to Simpilify Calc.
     ea=2*pi*c*b*((b/c)+(c/d)*log((b+d)/c))!Ellipsoid Area Funct.
     if((aa-ea).lt.3500) then !Loop Until Convergence Within 3500
       if((aa-ea).lt.0)then    !If Over-Shoot, Go Backward
          b=b-.05
          goto 333
       endif
```

F.4

```fortran
          if((aa-ea).lt.200) then !Loop to Within 200 for Better Accuracy
          !write (3,98) ea
          !write (6,98) ea
          !98      format (8x,'ea:',F9.0)
            goto 334
           endif
           b=b+.025
           goto 333
          endif
         b=b+.5
         goto 333
         !Ellipsoid Area Solved
           !Most excellent output sequence.
334       write(6,101)
          write(3,101)
101    format('1')
          if(mode.eq.1) then
            write(6,35)             !Phase 1 Header.
            write(3,35)             !Phase 1 Header.
35     format(8X,'Phase 1 Startup Conditions -- Both Configurations:'/)
          elseif(mode.eq.2) then    !Phase 2 Startup Output.
            write(6,38)             !Phase 2 Startup Header
            write(3,38)             !Phase 2 Startup Header
38     format(8X,'Phase 2 Startup Conditions:')
          elseif(mode.eq.3) then    !Steady State Conditions.
            write(6,44)             !Steady State Prompt.
            write(3,44)             !Steady State Prompt.
44     format(8X,'Steady-State Conditions.'/8X,'The main reactor,
     * engines, and thrusters (if NLB) are'/8X,'operating
     * simultaneously.'/)
          elseif(mode.eq.5) then    !Reactor and AC only.
            if(conf.eq.2) then
              write(6,49)              !Mode 5 Conf. 2 Header.
              write(3,49)              !Mode 5 Conf. 2 Header.
49     format(8X,'Main Engines Off.'/8X,'Reactor and NLB
     * Attitude Control Thrusters On.'/)
            else
              write(6,51)              !Mode 5-1 Header.
              write(3,51)              !Mode 5-1 Header.
51     format(8X,'Main Engines Off -- Reactor Only.'/)
            endif
          endif
          write(6,30) rp        !Specified Reactor Power.
          write(3,30) rp        !Specified Reactor Power.
30     format(/8X,'The specified power of the Main Reactor was:',F7.2,
     *' MWth.')
          write(6,31) mhd
          write(3,31) mhd
31     format(8X,'The MHD power converter efficiency was:',F5.2,
     *' percent.')
          if(i.eq.1) then          !Ship Power Output.
            write(6,32)             !Revised Power Prompt.
            write(3,32)             !Revised Power Prompt.
32     format(/8X,'The power required for ship operations was changed.')
            write(6,33) nav,ls,com,sm,rm,ht
            write(3,33) nav,ls,com,sm,rm,ht
33     format(8X,'Navigation Systems:',F5.2,' MWe.'/
     *8X,'Life Support Systems:',F5.2,' MWe.'/
     *8X,'Communications Systems:',F5.2,' MWe.'/
     *8X,'Shuttle and Maintenance Systems:',F5.2,' MWe.'/
```

```fortran
     *8X,'Research and Miscellaneous Systems:',F5.2,' MWe.'/
     *8X,'Heat Transfer Systems:',F5.2,' MWe.')
           write(6,34) spn
           write(3,34) spn
34    format(8X,'The total power used for ship operations is:'
     *F5.2,' MWe.'/)
           else                   !Standard Power.
           write(6,34) spn
           write(3,34) spn
           endif
           if(mode.eq.1) then
           write(6,36) rps         !Startup Engine Prompt.
           write(3,36) rps         !Startup Engine Prompt.
36    format(8X,'Power of the Startup Engine:',F10.2,' MWth.')
           write(6,37) stpn        !Power needed from the Main Reactor.
           write(3,37) stpn        !Power needed from the Main Reactor.
37    format(8X,'The power that the Main Reactor has to supply to
     * the Startup'/8X,'Engine is:',F10.2,' MWth.'/)
           elseif(mode.eq.2) then    !Phase 2 Startup Output.
           write(6,36) rps         !Startup Engine Prompt.
           write(6,39) ne,rpe      !Engine data.
           write(3,36) rps         !Startup Engine Prompt.
           write(3,39) ne,rpe      !Engine data.
39    format(8X,'Number of Engines:',I2/8X,'Reactor Power of Each
     * Engine:',F10.2,' MWth.')
           if(conf.eq.2) then    !Attitude and Control Info..
           write(6,40) nc,rpc      !AC Data.
           write(3,40) nc,rpc      !AC Data.
40    format(8X,'Number of Control Thrusters:',I2/8X,'Reactor Power
     * of Each Engine:',F10.2,' MWth.')
           endif
           write(6,41) rqp,pbe       !Power required and percentage.
           write(3,41) rqp,pbe       !Power required and percentage.
41    format(8X,'The power needed to startup the Engines is:',
     *F10.2,' MWth.'/8X,'This takes up ',F6.2,' percent of the power
     * generated by the'/8X,'Startup Engine.')
           if(conf.eq.2) then         !rqpc and pbc.
           write(6,42) rqpc,pbc       !rqpc,pbc.
           write(3,42) rqpc,pbc       !rqpc,pbc.
42    format(8X,'The power needed to startup the Thrusters is:',
     *F10.2,' MWth.'/8X,'This takes up ',F5.2,' percent of the power
     * generated by the'/8X,'Startup Engine.')
           endif
           write(6,43) whe                !Waste heat of the startup engines.
           write(3,43) whe                !Waste heat of the startup engines.
43    format(8X,'The waste heat from the Startup Engine is:',
     *F10.2,' MWth.'/)
           elseif(mode.eq.3) then      !Steady State Conditions.
           write(6,39) ne,rpe      !Engine data.
           write(6,45) pe
           write(3,39) ne,rpe      !Engine data.
           write(3,45) pe
45    format(8X,'The percentage of waste heat generated by the
     * engines'/8X,'is ',F5.2,' percent.')
           if(conf.eq.2) then
           write(6,40) nc,rpc       !AC Data.
           write(6,46) pc
           write(3,40) nc,rpc       !AC Data.
           write(3,46) pc
46    format(8X,'The percentage of waste heat generated by the
```

```fortran
     * control'/8X,'thrusters is ',F5.2,' percent.')
           endif
           write(6,47) rw,whe
           write(3,47) rw,whe
47         format(8X,'The waste heat from the reactor is ',
     *F10.2,' MWth.'/8X,'The waste heat from the engines is ',
     *F10.2,' MWth.')
           if(conf.eq.2) then
             write(6,48) whc
             write(3,48) whc
48         format(8X,'The waste heat generated by the control
     * thrusters'/8X,'is ',F10.2,' MWth.')
           endif
         elseif(mode.eq.5) then        !Reactor and AC only.
           write(6,50) rw
           write(3,50) rw
50         format(8X,'The waste heat from the reactor is:',
     *F10.2,' MWth.')
           if(conf.eq.2) then
             write(6,40) nc,rpc    !AC Data.
             write(6,46) pc
             write(6,48) whc
             write(3,40) nc,rpc     !AC Data.
             write(3,46) pc
             write(3,48) whc
           endif
         endif
           !Final Output to File.
           write(6,25) qr
           write(3,25) qr
25         format(/8X,'The total waste heat is:',F12.2,' MWatts.')
           write(6,24) a
           write(3,24) a
24         format(8X,'The necessary surface area for the radiator is:
     *',F12.2,' m**2.')
           length=a/(200*pi)
           radius=(aa/(4.*pi))**.5
           write(6,26) length
           write(3,26) length
26         format(8X,'The length of a cylindrical radiator is:',F10.2,'
     * meters.')
           write(6,27) radius
           write(3,27) radius
27         format(8X,'The radius of a spherical radiator is:',F10.2,'
     * meters.')
           write(6,97) c
           write(3,97) c
97         format(8x,'The minor axis of an ellipsoid radiator is fixed at:'
     * ,F6.1,' meters.')
           write(6,99) b
           write(3,99) b
99         format(8x,'The major axis of the ellipsoid is:',F6.1,'
     * meters.')
           write(6,28) conf,mode
           write(3,28) conf,mode
28         format(/8X,'This was for the case of configuration ',I1,
     *' and operating'/8X,'mode ',I1,'.'//)
         endif
         go=1
         write(6,29)
```

```
29      format(8X,'Do you wish to repeat the program?'//8X,
       *'1.Yes.'/8X,'2. No'//8X,'Enter "1" or "2".')
        read(5,*) go
        enddo
        close(unit=3)
        stop
        end
```

```
c          Program  HEAT TRANSFER        Robert Hellstrom
c                                              George Owens
c          E-CITY Propellant tank        AAE 515 / 541
c    25    sec run time on pc 10 mhz     4 March 1992
c_____
c
c The program will compute the outer surface temperature of the
c spherical propellant tank due to solar radiative heating.
c In addition, several insulative methods and materials are analyzed.
c No planetary interferences are included.
c
          PROGRAM INERTIA
          IMPLICIT REAL (A-I,K-Z)
          INTEGER N
 c        The following data is required:
 c        1) Radius of Propellant Tank (Rp)
 c        2) Temperature of surrounding space: in Kelvin (Tspc)
 c        3)    "      " propellant (Tinner)
 c        4) Absorptivity of outer  material (absor1,2,3)
 c        5) Emissivity   "    "      "    (emiss1,2,3)
 c        6) Internal Heat Generation (Qin)
 c        7) Stefan Boltzmann constant (sigma)
 c        8) Wall thickness (Lw)
 c        9) Thermal conductivity of exterior wall: in W/m/K (Kw)
 c        10)  "       "      " insulation (Kins)
c The above data must be entered below sequencially using SI units
c (Kelvin and m) and separated by commas.
          DATA PI,Rp,Tspc,Tinner/3.141593,2.370E+2,0.0,23./
          DATA absor1,emiss1,absor2,emiss2/.3,.3,.2,.8/
          DATA absor3,emiss3/.9,.9/
          DATA Qin,sigma,Lw,Kw/0.0,5.67E-8,.0188,163./
          DATA Kins1,Kins2,Kins3/0.036,.01,.000106/
          DATA Rp2,Lw2/237.07,.0148/
          OPEN (1,FILE='heat.DAT',STATUS='NEW')
c Calculate spherical propellant tank exposed perpendicular area
c (Aperp) and total surface area (As).
          Aperp = PI*Rp*Rp
          As = 4.*Aperp
          WRITE(1,10)
10   FORMAT(/18X,'*ONE SPHERICAL PROPELLANT TANK*')
          DO 400 Jnum=1,2
          IF(Jnum.EQ.1) GOTO 25
          Rp = Rp2
          Lw = Lw2
          Aperp = 2.*PI*Rp*Rp
          As = 4.*Aperp
          WRITE(1,20)
20   FORMAT(/18X,'*TWO SPHERICAL PROPELLANT TANKS*')
25   DO 300 JK=1,3
```

```
c The maximum allowable heat flow in MW to the propellant.
      Qmax = 2.0
c The ratio of absorptivity/emissivity is calculated.
      IF(JK.EQ.1) absor=absor1
      IF(JK.EQ.1) emiss=emiss1
      IF(JK.EQ.2) absor=absor2
      IF(JK.EQ.2) emiss=emiss2
      IF(JK.EQ.3) absor=absor3
      IF(JK.EQ.3) emiss=emiss3
      absem = absor/emiss
      WRITE(1,30) absem,Qmax
30    FORMAT(/15X,'*Heat Transfer in a Spherical Prop. Tank (SI)*',
     ,/15x,'absor/emiss = ',F8.5,5x,'Qmax=',F9.4,' MW',
     ,//7X,'Ts (K)',6x,'Isun(W/m^2)',
     ,2x,'DIST. TO SUN',3x,'Qr (MW)',5x,'INSUL.(m)')
      WRITE(1,40)
4     FORMAT(6x,'_____
     ,_____')
c Initial value of Solar Intensity is for the earth Isun=1400W/m/m.
c The intensity is assumed to decrease: ie. the space vehicle moves
c farther from the sun. However, a formula is needed.
c LOOP for different insulative materials.
      Qmax = Qmax*1.E+6
      DO 200 JN=1,3
c Calculate the insulation thickness (Lins).
      IF(JN.EQ.1) Kins=Kins1
      IF(JN.EQ.2) Kins=Kins2
      IF(JN.EQ.3) Kins=Kins3
      WRITE(1,50)Kins,Lw
50    FORMAT(5x,'Kins= ',F9.6,/5x,'WALL (m)= ',F6.4,/)
c LOOP for changing solar intensity (Isun).
      Isun = 1400.
      DO 100 J=1,14
c Calculate the surface temperature of the tank (Ts).
      Ts = (absor*Aperp*Isun + Qin)/emiss/sigma/As
      Ts = Ts**(1./4.)
      DELTAT = Ts - Tinner
c Calculate the total radiative heat flow (Qr) in watts.
      Qr = emiss*sigma*As*(Ts*Ts*Ts*Ts - Tspc*Tspc*Tspc*Tspc)
      Lins = Kins*As*(DELTAT/Qmax - Lw/Kw/As)
      Qr = Qr/1.E+6
      WRITE(1,60) Ts,Isun,Dsun,Qr,Lins
60    FORMAT(5x,3(F9.3,4x),F10.4,2x,F12.8)
      Isun = Isun - 107.5
100   CONTINUE
200   CONTINUE
300   CONTINUE
400   CONTINUE
      STOP
      END
```

F.10

REFERENCES

1   *Project WISH: The Emerald City*, The Ohio State University Department of Aeronautical and Astronautical Engineering, Columbus, OH, 1990.

2   *Project WISH: The Emerald City, Phase II*, The Ohio State University Department of Aeronautical and Astronautical Engineering, Columbus, OH, 1991.

3   Eisley, J.G., *Tutorial Lessons for SDRC I-DEAS, Level 4.1*, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 1991.

4   Neill, D.,J.,Johnson, E., H., Heredoen, D.,L., *Automated Structural Optimization System*, Vol. II - User's Manual, April 1988.

5   Moler, C., et. al., PRO-MATLAB for VAX/VMS Computers User's Guide, The MathWorks Inc., Sherborn, Massachusetts, October 16, 1987.

6   Ragsdale, R.G., *Relationship Between Engine Parameters and the Fuel Mass Contained in an Open-Cycle Gas-Core Reactor*, NASA Technical Memorandum X-52733, January 1970.

7   McLafferty, G.H., Baver, H.E., *Studies of Specific Nuclear Light Bulb and Open-Cycle Vortex-Stabilized Gaseous Nuclear Rocket Engines*, United Aircraft Research Laboratories Report F-910093-37, Also issued as NASA CR-1030, April 1968.

8   Kendall, J.S., *Investigation of Gaseous Nuclear Rocket Technology-Summary Technical Report*, UARL Report L-910905, September 1972.

9   Latham, T.S., Rodgers, R.J., *Small Nuclear Light Bulb Engines with Cold Beryllium Reflectors*, AIAA Paper No. 72-1093, presented at AIAA 8th Joint Propulsion Specialist Conference, New Orleans, Louisiana, November-December 1972.

10 Latham, T.S., *Summary of the Performance Characteristics of the Nuclear Light Bulb Engine*, AIAA Paper No. 71-642, presented at AIAA 7th Joint Propulsion Specialist Conference, Salt Lake City, Utah, June 1971.

11 Johnson, R.D., Holbrow,C., *Space Settlements: A Design Study*, National Aeronautics and Space Administration SP-413, 1977.

12 Billingham, J., Gilbreath, W., "Space Resources and Space Settlements", *National Aeronautics and Space Exploration*, SP-428, 1979.

13 Oz, H., The Ohio State University, Aeronautical/Astronautical Engineering, AAE 416A, Class Notes, Spring 1990, 1992.

14 Beer, F.P., Johnston, E.R. Jr., *Vector Mechanics for Engineers: Dynamics*, McGraw-Hill Book Company, New York, 1988.

15 Griffin, M.D., French, J.R., *Space Vehicle Design*, American Institute of American Astronautics, Washington, DC, 1991.

16 Krieth, F., Bohn, M.S., *Principles of Heat Transfer, 4th Edition*, Harper & Row Publishers, New York, 1986.

17 Cunnington, G.R., "Thermodynamic Optimization of a Cryogenic Storage System for Minimum Boiloff", No. AIAA-82-0075, AIAA 20th Aerospace Sciences Meeting, Orlando, FL, 1982.

ADDITIONAL REFERENCES:

Anderson, J.D., *Introduction to Flight, 2nd Edition*, McGraw-Hill Book Company, New York, 1985.

Bednar, H. M., Pressure Vessel Design Handbook, 2nd Edition, Van-Nostran Reinhold Company Inc., New York, 1986.

Gere, J.M., Timoshenko, S.P., *Mechanics of Materials, 2nd Edition*, PWS-Kent Publishing Company, Boston, MA, 1984.

The Handling and Storage of Liquid Propellants, Office of the Director of Defense Research and Engineering, Washington, DC, 1963.

Meirovitch, L., Van Landingham, H.F., Oz, H., "Control of Spinning Flexible Spacecraft by Modal Synthesis", *Acta Astronautica*, Vol. 4, pp. 985-1010, Great Britain, 10 March 1976.

Meirovich, L., Oz, H., "Observer Modal Control of Dual-Spin Flexible Spacecraft", *Journal of Guidance and Control*, Vol. 2, No. 2, March-April 1979.

Rittenhouse, J.B., Singletary, J.B., *Space Materials Handbook*, National Aeronautics and Space Administration, Washington, DC, 1969.