ELECTRODYNAMIC TETHER OPERATIONS AND CONTROL
FINAL REPORT

Prepared By

bd Systems, Inc.
600 Boulevard South, Suite 304
Huntsville, Alabama 35802

Sponsored By:

GEORGE C. MARSHALL SPACE FLIGHT CENTER
MARSHALL SPACE FLIGHT CENTER, ALABAMA 35812

Under:

Contract No. H-32041D

March 19, 2001

TABLE OF CONTENTS

1.0 INTRODUCTION

This Final Report is organized by tasks from the statement of work (SOW) performed under contract # H-32041D. The paragraphs below contain a brief statement of each task with its task description followed by a discussion of the work performed. The period of performance for this contract phase was from July 21, 2000 to March 19, 2001.

2.0 DESCRIPTION OF WORK PERFORMED

2.1 SOW TASK 1

The contractor shall modify the existing tether dynamics computer simulations, GTOSS or TSSIMR to provide the capability to model the key dynamic/electrodynamic interactions driving the STEP class electrodynamic tether systems.

Much of this task was completed under the initial phase of contract # H-32041D and reported in a previous Final Report. A differential equations model of bare wire tether current collection dynamics was developed using current collection equations supplied by Dr. Robert Estes of the Harvard Smithsonian Astrophysical Observatory. A fortran numerical implementation of this was coded, checked and integrated with GTOSS as part of the initial phase. In the present phase, we re-verified the current generation subroutines and reorganized them into a 6th electrical current/power generation scenario for GTOSS. Five GTOSS electrical/power generation scenarios had been defined for previous applications. This gives the user greater control over electrodynamic tether operations and makes it generic. In this arrangement, the scenario can be applied to PROSEDS as well as STEP-AIRSEDS. These modifications have also made it much easier to model changes in system configurations such as going from a three body configuration (3BC) to a two body configuration (2BC) as has occurred in STEP-AIRESEDS. It is anticipated that additional modifications and adaptations to the tether simulation tools will be an ongoing activity as requirements become better defined and modifications become necessary to investigate effects of such changes or refinements in understanding.

The version of GTOSS for which the bare wire tether current collection algorithm was developed and checked out was an adaptation of GTOSS Version H7.0. This modified version we have called Version H7.0Q. We chose to work primarily with GTOSS because of its built-in ability to handle multiple tether configurations such as those initially being proposed for Step-Airseds (SA). As stated in the previous paragraph, a 3BC was originally proposed, but more recently, the Michigan Technic Corporation (TMTC) has settled on a 2 body configuration (2BC). Consequently, it now becomes reasonable to also bring along TSSIM our own tether simulator. We have also added the electron/ion density subroutine and the bare wire current subroutine to TSSIM.

As the TMTC optimization studies progressed, it was suggested that endbody current collection be considered as an addition or as an alternative to bare wire collection. Thus, it became desirable to add this to our current collection algorithm. A simple, end body collection logic was added to GTOSS. This logic was based on the same structure as used in the bare wire collection algorithm.

The version of GTOSS modified by bd Systems, version H7.0Q has been provided to NASA and also provided to TMTC. This version contains the modifications made to model the dynamics of bare wire current collection. The modifications are contained in a single source, TOSSH6.FOR. The listing for this source file is included as Appendix 1. Discussions continue over modeling tools to be used by TMTC and to be made available to NASA as well. These are based on a Spacecraft Dynamics Tool (SDT) developed by Bob Strunce, a consultant to TMTC. The discussions concern addition of all or parts of GTOSS to SDT.

David D. Lang Associates, developer and author of GTOSS was brought on by TMTC in the October 2000 timeframe as a subcontractor to support them in their development and implementation of a tether dynamics model for their use. The changes and additions to GTOSS Version H7.0Q made by bd Systems for modeling bare wire tether electron current collection were discussed and a copy of the modified source code was made available to MR. Lang. He decided to integrate these as part of his latest version and create a new release. That latest version was called version 9.0. Along with the new release, he provided updates of the Equations of Motion Document in paper form and an electronic copy of the Quick Reference Manuals for version H8.0 followed by version 9.0 documentation after delivery of the version 9.0 GTOSS. Meanwhile, bd Systems added a model of end body current collection based on end body collection area. The form of the model was patterned after the bare wire current collection model. Inputs of test cases and data were solicited from Professor Brian Gilchrist of University of Michigan. He has provided us with a list of references but no test cases have been received to this date.

The updated version of GTOSS, version H9.0 was delivered and modified as required to run in the Windows PC environment. The same process was employed as was used to create version H7.0q. Additional modifications were made to this version to adapt the tether thermal model to the bare wire electrodynamic tether. Lang Associates has incorporated these changes as a standard part of his new beta release version. This version also includes some minor modifications and the addition of new electrical and current collection properties assigned to tethers. A significant change in the beta version is the change to REAL*8 arithmetic. Version H7.0Q was largely operating in REAL*4. Correction of typos as well as refinements and addition of more output formats brought the beta versions of the new GTOSS to version H9 Beta 5. In January 2001, bd Systems implemented minor updates to the GTOSS PC version and this brought our version to H9/Beta 6.

bd Systems recommended to Lang Associates, the GTOSS Developer, that additional attitude control capability be added to GTOSS allowing each TOSS object to have the same attitude control capability already provided for the reference body. This recommendation was accepted and the capability was added in the January 2001 timeframe. As an adjunct to this modification/addition, bd Systems defined additional euler angle types available for GTOSS sufficient in number to fill in all the ones that had not been available previously. GTOSS now has available the complete set of euler angle types. This was necessary to give the

GTOSS users single axis control capability about any desired body axis (such as yaw axis control). These control modes can now be implemented without additional mods to GTOSS or SDT/TOSS. This capability was demonstrated. These control modes are generic and do not implement any sensor or actuator dynamics. They form a basic capability to include end body attitude control relative to an offset local vertical about either a single axis or a full, three axis control. This, more detailed system model capability including sensor and actuator characteristics is to be provided by TMTC in their Satellite Dynamics Tool/TOSS software.

2.2 SOW TASK 2

The contractor shall perform simulations that demonstrate control of various orbital maneuvers, including but not limited to altitude changes (boost or deboost), and inclination changes. The control strategies for affecting all the orbital elements will be examined.

Sample simulations of altitude changes for both orbital boost and deboost have been carried out to demonstrate scenario 6. These were simply to demonstrate the data setup and to show that the scenario can adequately handle both conditions for the 2BC. In this way, the setup required inputs for each situation is shown. For orbital boost, constant power at the high voltage power supply is maintained at the upper spacecraft. For orbital deboost, the current is through the tether passes through a battery and a resistor through the lower spacecraft plasma contactor. GTOSS as presently defined provides a basic capability for electrodynamic tether operations. No controllers have yet been defined to modulate these current flows to achieve orbital navigation/guidance control or to achieve control of tether lateral dynamics. This is expected to come through the development of the Satellite Dynamics Tool and its integration with the TOSS portion of GTOSS. Simulations of steady state orbital deboost operations were performed to demonstrate the operations of the system with current flow and yaw axis only control. These results demonstrated correct operation of the new GTOSS capabilities.

To aid in the visualization of the dynamics of the Step-Airseds system selected configuration. TMTC has proposed use of Satellite Tool Kit's Visualization Option (VO) as an aid to provide animated depictions of the SA dynamics. This would operate as part of the Satellite Dynamics Tool being Developed for TMTC by Robert R. Strunce, Francis H. Maher, and David D. Lang. As an alternative to VO, bd Systems investigated an animation package then under a development sponsorship by the NASA Johnson Space Center (JSC) called Enigma. It was available as Beta test software to bd Systems and operates in the PC computer environment. Since we had already developed animations and graphics for other spacecraft using Enigma, we developed an animation of SA as a demonstration. This Enigma driven animation performed satisfactorily and was shown to Ken Welzyn at MSFC. These were based on dynamic response data generated by TSSIM and GTOSS. The basics required for any animation are virtually the same although the details will vary with the animation software. For this example, a post processor in Matlab© generated an animation file formatted as required to interface with Enigma. However, since Enigma was still in the Beta test stage, we were reluctant to recommend it. Finally, JSC has recently dropped its sponsorship for development of Enigma and it now appears this software is no longer likely to be readily available without cost for the PC. Alternatives have been pursued and as of the time of writing of this report, a package called BLENDER, available as Freeware and download-able from the internet at the site www.blender.nl looks promising. This search continues on the assumption that an available and convenient animation package separate from the simulation tools but compatible with their outputs through post processing is desirable.

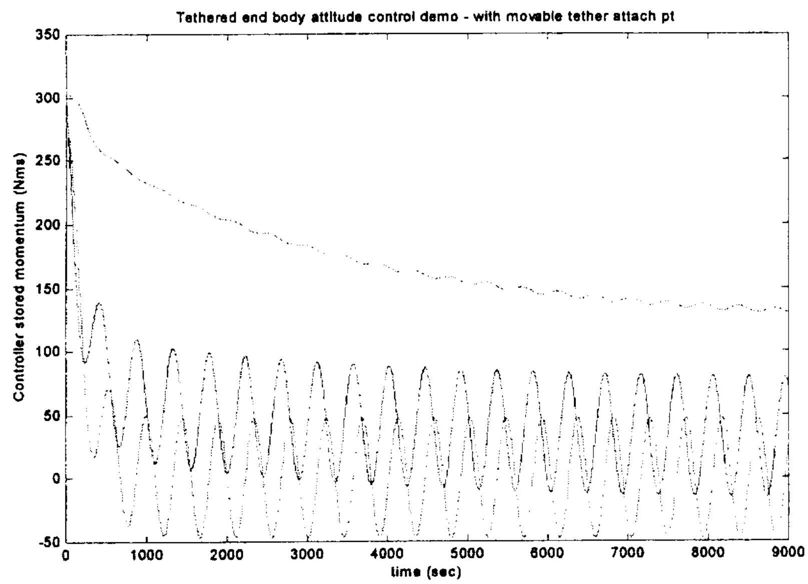

Figure 1. Angular momentum storage required for attitude control of tethered end body. In this example, a momontum management controller based on a controllable tether attach point is active.

Separate from the tether simulations and animation software considerations, a simple rigid body model of a spacecraft in circular earth orbit was built. This simulation employs a simplified model of the tether influence modeled as a constant magnitude force acting approximately along the local vertical, also having a smaller normal component. This force represents tether tension and acts at the tether attach point. The small variations of the force direction away from the local vertical are included to represent the effects of skip rope and other tether dynamics. As stated, this simplified model of the dynamics of end body attitude control simulates a rigid body in a nearly circular orbit with gravity gradient and tether tension disturbances. Momentum storage devices (control moment gyros or reaction wheels) are used for control. The purpose of this simulation is to study methods of end body control in the face of tether and gravity gradient disturbance. A momentum exchange controller is the initial attitude control system to be studied. Actuator size, control and momentum management techniques are items to come out of the study. This model should be adequate to develop endbody control concepts including skip rope observer algorithms. This model was meant to be illustrative of the level of model sophistication required to study the dynamics of tethered end bodies.

Momentum management is being done using a movable tether attachment point. Appendix 2 contains a listing of the C++ source code for the simulation. Sample results are shown in figures 1 and 2. These figures show the components of the accumulated CMG or RW control angular momentum for two sample cases. Each case is initialized with a 300 Nms stored control momentum in each body axis. In figure 1, the momentum management logic is acting to remove the initial momentum offset. In figure 2, the momentum management logic is inhibited so that the initial momentum is not removed but is swapped between body axes as the spacecraft rotates with the local vertical attitude of the orbit.

The momentum management system modeled here uses momentum feedback to position the tether attach point so that tension force induced torque acts to remove accumulated angular momentum. This is an alternative to magnetic momentum control using torque bars. Both systems are similar in that momentum management torque can only be produced about two axes at any instant in time. Since this uncontrolled direction changes with time momentum can be controlled in all three axes over time.

The mass and moment of inertia properties used in this model are as described in TMTC's Preliminary Analysis of the Attitude Determination and Control System, TMTC-SA-PB-ACSv1.0. Our findings from this brief analysis indicate that the momentum wheels selected for the end bodies are significantly undersized. We have made several assumptions in arriving at this result. The first assumption is that the control mode in steady state operation, especially for the upper end body will be local vertical, local horizontal (LVLH) hold with the yaw angle adjusted to allow the solar arrays to track the sun. The principle disturbance is expected to be tether tension of 9-10 N for a 10 km tether. The second assumption is that 10° of skip rope amplitude will be typical. With these assumptions and a fixed attach point of the tether, angular momentum swings approaching 100 Nms peak to peak are possible. The recommended reaction wheel size from the referenced TMTC report is 0.12 Nms, see above report, page 9. Clearly, there is a mismatch here. Methods exist to significantly reduce the required momentum storage but these involve altered attitude hold modes, movable tether attach points, or a combination of these. Also, better control of tether skip rope would reduce necessary momentum storage. This requires some sort of tether skip rope observer. Presently, no sensor is available for direct tether shape observation and no accelerometer is planned which could be

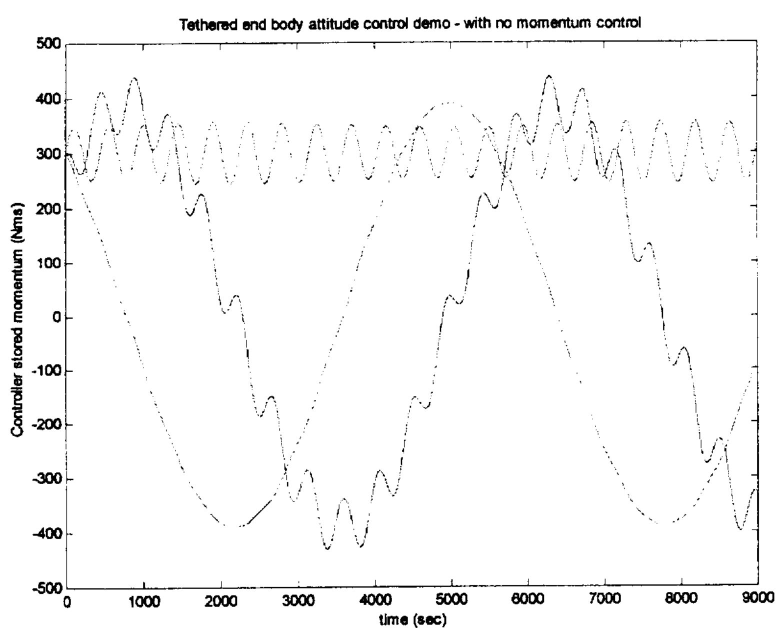

Figure 2. Angular momentum storage required for attitude control of tethered end body. In this example, no momontum management controller is active.

used to estimate tether skip rope from observe tension force vector. A 3-axis-accelerometer currently appears to be the simplest device available to provide substantial insight into tether dynamic behavior.

Since the primary, non-gravitational acceleration of an end body is caused by tether tension and the accelerometer measures only non-gravitational acceleration, it provides an instantaneous readout of the local tether angle. Using this information in conjunction with end body attitude relative to LVLH, both skip rope and libration can be determined. Additional control studies will await better definition of end body attitude control modes during electrodynamics tether operations.

Check-out of GTOSS H9.0 and subsequent beta versions has been completed. We have run cases to show the operation of the bare wire current collection model and to demonstrate the new tether characteristic constants: 1. Tether conductor cross sectional area; 2. Plasma contacting circumference for the tether; and 3. Tether material resistivity. This activity is in preparation for providing the updated GTOSS to TMTC. At a meeting at the TMTC facilities on the 18th and 19th of the December 2000, the updated version of GTOSS was transferred to TMTC and to Fran Maher working with Bob Strunce.

Step-Airseds review meetings in conjunction with attendance at the 39th AIAA Aerospace Sciences Meeting and Exhibit in Reno, Nevada 8-11 January 2001, were held with representatives from TMTC. Bd Systems were representing NASA as no NASA personnel from MSFC were able to attend due to illness and schedule conflicts. A trip report was written and provided to MSFC summarizing the meetings. A copy of this trip report is included in 3.

2.3 SOW TASK 3

The contractor shall develop criteria for defining acceptable envelopes of tether motion during the maneuvers described in Task2 and shall develop and simulate tether control strategies that will keep tether motion within those bounds. The implications of the tether control strategies on the system (e.g. propulsion performance penalties, system requirements for the control architecture, etc.) shall be noted.

Preliminary concepts for system operations have been developed but have not yet been implemented into the simulation tools. These included schemes for tether observers based on satellite IMU accelerometers or departure angle sensors. Currently, parallel efforts are under way by other investigators to determine the feasibility of more direct, laser or optical based tether observers. If such systems prove feasible, we plan to investigate performance comparisons between such systems and to assess implications on system requirements.

Preliminary concepts for system operations have been developed but have not yet been implemented into the simulation tools. When this task was initiated, GTOSS did not have a satisfactory endbody attitude control built-in except for the reference body. The results from the simulations of the previous task provides the database for accomplishing this task and also serve as a test bed for proving control concepts. Preliminary estimates of acceptable envelopes of tether motion have been made based on past history with TSS and SEDS flights as well as previous experience with other spacecraft controls. These need to be verified as part of an integrated system simulation where all aspects of a Step-Airseds mission can be studied. The tools for such an integrated study are now being put into place with the updated GTOSS and TSSIM.

At a Technical Interchange Meeting (TIM) held at TMTC held in December 2000, the new version of GTOSS (H9.0) was installed and made operational on TMTC's Mac computers. Also, the bd Systems implementation of GTOSS version H9B3, was given to Fran Maher and Bob Strunce for use in their Dynamic Modeling Tool. Mr. Lang and Dr. Glaese worked with Murshed Khadija who is an Engineering Intern at TMTC. He ran Step-Airseds flyaway simulations based on a GTOSS input file (INGOSS) constructed by Mr. Lang and Dr. Glaese to study the flyaway problem. (see Appendix 4). A basic philosophical approach to flyaway was defined to begin analysis until a database of results could be developed and studied. The initial approach is to separate the end bodies as quickly as the hardware allows. This requires that the deployer deploy tether to match the acceleration obtained with the flyaway thrusters.

The present separation thrust of ½ Newton will accelerate the upper end body at approximately 1 mm per second. After 1,000 seconds, this will produce a 1 m/s separation rate and a separation of approximately 500 m. This neglects the small but not insignificant effects of orbital dynamics in this time period. Also, it requires that the deployer residual forces on the tether be minimal, not more than 0.1 N at most and preferably less. This was discussed with Mr. Neil Rothwell of Double R Controls. His company is to build prototypical hardware for the deployer mechanism. Additional discussions were held with Mr. Kevin Probst. Meeting schedules did not permit more detailed discussions with him but we did discuss deployment philosophy and requirements. Mr. Probst has been identified by TMTC as their Systems Engineer and Project Manager. The importance of the deployer and the ways in which operational requirements will impact deployer design were emphasized including tether deployer testing and control strategies for libration control and end body attitude control. Emphasis was also to be placed on defining operational philosophies for these Step-Airseds modes.

2.4 SOW Task 4

The contractor shall provide regular, informal, technical coordination and communication via telephone and email with the MSFC Study Managers on biweekly or as-needed basis. The contractor shall participate in periodic technical meetings with other organizations working on related technical issues.

Biweekly telecons and other meetings with our MSFC Technical Monitors were held to review GTOSS status and also to jointly review progress. The Tether Observer study took place over the Summer and early Fall of 2000. This study was conducted by TMTC and UAH. The purpose of the study was to determine feasibility of developing a tether observer or shape sensor capable of providing regular measurements of tether shape. A system based on a laser radar was proposed and investigated and but determined to be impractical in power required and expense.

In September, a Technical Interchange Meeting was held in Holland, Michigan, at TMTC headquarters. This meeting reviewed the current status of the SA studies and highlighted growing concerns regarding the aerodynamic drag of the tape tethers, which are regarded as the tether candidates with the greatest lifetime/orbital survivability potential. In the 350 km altitude range, such tethers appear to have nearly as much drag force as they are able to produce in thrust with a 1 kW power supply. This has reopened consideration of endbody current collection as a way to increase thrust by increasing average current or reducing drag by reducing required tether length.

Additionally, in late September and early October, a dynamic model, technical interchange meeting (TIM) was held at MSFC. Bd Systems participated in that meeting. The goal of that meeting was to provide direction for the incorporation of GTOSS as the tether dynamics model generator for TMTC . This was also addressed in biweekly telecons of the October-November timeframe held with MSFC, TMTC and Lang Associates. In December, a second TIM was also held at the TMTC facilities in Holland, MI. Due to the holidays and to face-to-face meetings held with TMTC, fewer than usual telecons were held in the January period. Coordination/review meetings were held at Reno as described previously in conjunction with the conference.

A trip was made to the Goddard Space Flight Center on February 13-14 to attend a proposal planning meeting for an experiment called Gradsat. This experiment is proposed to fly in approximately the year 2004 and will employ a tether to provide separation between end bodies. The end bodies will be equipped to separately measure the earth's magnetic field to sufficient accuracy for gradient techniques to allow determination of earth crustal field sources. This measurement along with a proposed direct measurement of field components above the poles would be data never before obtained. A Trip Report was written for this trip and is included as Appendix 5.

3.0 SUMMARY

The preceding paragraphs describe the work done under this contract. Substantial Progress was made on all the tasks covered. The tether simulator called GTOSS has been enhanced and adopted as the NASA ED tether dynamics modeling simulator with TSSIM as an adjunct. Tools have been developed for modeling the aspects of end body attitude control. Models of ED tether and end body current collection dynamics have been developed and implemented. The work involved is an ongoing activity and items remain to be completed. This work is to be continued under a new contract.

Appendix 1: TOSSH6.FOR
ELECTRODYNAMIC TETHER SUBROUTINES, GTOSS POWER SCENARIO 6

```
C*************************************
C  New routine, developed by John Glaese, patterned after TOSSH5.
C        2-15-00.  For application to ED tethers and STEP-AIRSEDS 3 body model.
C        Revised July 2000 for better compatibility with GTOSS architecture.
C        Makes it easier to change to 2 body model or other future variations
C        as required.
C*************************************

   SUBROUTINE TOSSH6(JS,JTETH)

C*************************************
C*************************************
C
C PURPOSE
C-------
C THIS ROUTINE CURRENTLY INTERPRETS A STANDARD POWER GENERATION SCENARIO
C DATA STRUCTURE IN SUCH A WAY AS TO CREATE TIME SEQUENCES OF VARIOUS
C TYPES OF CURRENT-LENGTH DISTRIBUTIONS (IE. CORRESPONDING TO
C DIFFERENT ELECTRICAL OPERATING MODES) FOR A BARE WIRE TETHER.
C
C ANOTHER PURPOSE FOR THIS ROUTINE IS TO PROVIDE A CONVENIENT TEMPLATE
C TO THE USER FOR BUILDING (AS USER-SUPPLIED SCENARIOS TO TOSS)
C FUNCTIONAL SIMULATIONS OF ON-BOARD COMPUTER OPERATION AND SEQUENCING
C FOR CERTAIN TYPES OF BARE-WIRE ELECTRODYNAMIC TETHER MISSIONS.
C
C PURPOSE DETAILS
C---------------
C THIS SCENARIO PROVIDES A MEANS TO CONVENIENTLY STUDY AND DESIGN THE
C OPERATING CYCLES PROPOSED FOR STEP-AIRSEDS TETHER MISSION.
C
C AN OPERATING CYCLE IS DEFINED AS A TIMED-SERIES OF CURRENT-MODES
C
C A CURRENT MODE IS DEFINED AS A ROUTINE THAT YIELDS CURRENT -VS- LENGTH
C
C
C THIS SCENARIO ALLOWS CONVENIENT STUDY OF ONLY ONE OPERATION CYCLE AT A
C TIME, RATHER THAN A TOTAL MISSION PROFILE IN WHICH DIFFERENT OPERATION
C CYCLES ARE EMPLOYED IN RESPONSE TO ON-GOING SYSTEM STATE CONDITIONS.
C
C
C   THE "STANDARD" SCENARIO      THE BARE-WIRE SCENARIO
C     DATA SET DEFINITION          INTERPRETATION
C   -----------------------      ------------------
C   POWER GENERATION START TIME    MEANS THE SAME
```

```
C   POWER GENERATION STOP TIME     MEANS THE SAME
C   PWR MULTIPLIER                 RATIOS CURRENT VALUES
C
C
C   PERIOD DURATION  - PERIOD 1   XINSFRAC




C   BEGIN PWR LEVEL  - PERIOD 1   CURRENTMAX
C
C   ENDING PWR LEVEL - PERIOD 1   VBIASMAX
C   LIMIT PWR LEVEL  - PERIOD 1   RESERVED FOR FUTURE USE
C
C   PERIOD DURATION  - PERIOD 2   TIME DURATION OF 1ST SPECIFIED MODE
C   BEGIN PWR LEVEL  - PERIOD 2   PVSET (+/- MOTOR/GENERATOR MODE)
C
C   ENDING PWR LEVEL - PERIOD 2   BATTERY SERIES RESISTANCE
C   LIMIT PWR LEVEL  - PERIOD 2   RESERVED FOR FUTURE USE
C
C   PERIOD DURATION  - PERIOD 3   TIME DURATION OF 2ND SPECIFIED MODE
C   BEGIN PWR LEVEL  - PERIOD 3   PVSET (+/- MOTOR/GENERATOR MODE)
C
C   ENDING PWR LEVEL - PERIOD 3   BATTERY SERIES RESISTANCE
C   LIMIT PWR LEVEL  - PERIOD 3   RESERVED FOR FUTURE USE
C
C   PERIOD DURATION  - PERIOD 4   TIME DURATION OF 3RD SPECIFIED MODE
C   BEGIN PWR LEVEL  - PERIOD 4   PVSET (+/- MOTOR/GENERATOR MODE)
C
C   ENDING PWR LEVEL - PERIOD 4   BATTERY SERIES RESISTANCE
C   LIMIT PWR LEVEL  - PERIOD 4   RESERVED FOR FUTURE USE
C
C   PERIOD DURATION  - PERIOD 5   TIME DURATION OF 4TH SPECIFIED MODE
C   BEGIN PWR LEVEL  - PERIOD 5   PVSET (+/- MOTOR/GENERATOR MODE)
C
C   ENDING PWR LEVEL - PERIOD 5   BATTERY INTERNAL RESISTANCE
C   LIMIT PWR LEVEL  - PERIOD 5   RESERVED FOR FUTURE USE
C
C   PERIOD DURATION  - PERIOD 6   TIME DURATION OF 5TH SPECIFIED MODE
C   BEGIN PWR LEVEL  - PERIOD 6   PVSET (+/- MOTOR/GENERATOR MODE)
C
C
C   ENDING PWR LEVEL - PERIOD 6   BATTERY INTERNAL RESISTANCE
C   LIMIT PWR LEVEL  - PERIOD 6   RESERVED FOR FUTURE USE

C@NOS*CALL COM_ALL
C@NOS*CALL COM_RPS
C@NOS*CALL COM_TOSS
C@NOS*CALL EQU_TOSS
C@NOS*CALL COM_FOSS
C@NOS*CALL EQU_FOSS

C@UNX#include "COM_ALL.i"
```

```
C@UNX#include "COM_RPS.i"
C@UNX#include "COM_TOSS.i"
C@UNX#include "EQU_TOSS.i"
C@UNX#include "COM_FOSS.i"
C@UNX#include "EQU_FOSS.i"

C@VAX    INCLUDE 'COM_ALL.FOR'
C@VAX    INCLUDE 'COM_RPS.FOR'
C@VAX    INCLUDE 'COM_TOSS.FOR'
C@VAX    INCLUDE 'EQU_TOSS.FOR'
C@VAX    INCLUDE 'COM_FOSS.FOR'
C@VAX    INCLUDE 'EQU_FOSS.FOR'

      INCLUDE 'COM_ALL.INC'
      INCLUDE 'COM_RPS.INC'
      INCLUDE 'COM_TOSS.INC'
      INCLUDE 'EQU_TOSS.INC'
      INCLUDE 'COM_FOSS.INC'
      INCLUDE 'EQU_FOSS.INC'

C@MAC    INCLUDE '{GTOSS}A_TOSS:COM_ALL.i'
C@MAC    INCLUDE '{GTOSS}A_TOSS:COM_RPS.i'
C@MAC    INCLUDE '{GTOSS}A_TOSS:COM_TOSS.i'
C@MAC    INCLUDE '{GTOSS}A_TOSS:EQU_TOSS.i'
C@MAC    INCLUDE '{GTOSS}A_FOSS:COM_FOSS.i'
C@MAC    INCLUDE '{GTOSS}A_FOSS:EQU_FOSS.i'

      INTEGER IPVSET,IPVSETP,I_EMIT
cjrg  REAL*4 TSCSTR,TSCEND,XINSFRAC,PVSET(5),CURRENTMAX,VBIASMAX,
cjrg $ DUMRMI(3),DUMRMP(3),RBAT(5),AREA_END,DUMMY(5)
   REAL*8 TSCSTR,TSCEND,XINSFRAC,PVSET(5),CURRENTMAX,VBIASMAX,
  $ DUMRMI(3),DUMRMP(3),RBAT(5),AREA_END,DUMMY(5)

C LOCAL DATA ITEMS FOR PLASMA MODEL CALL
   LOGICAL  JFDUM(12),IPVCHG
      DATA IPVSETP /0/
      DATA I_EMIT /0/

C-------------------------------------------------
C FOR CODE-CLARITY, RE-PACKAGE DATA FROM STANDARD SCENARIO DATA
C STRUCTURE INTO THE DATA ITEMS MOST MEANINGFUL TO THIS SCENARIO
C-------------------------------------------------
C 1ST THE MODE/DURATION DATA
      TSCSTR = TGENON(JS)
      TSCEND = TGENOF(JS)
      XINSFRAC   = DPINV1(JS)
      CURRENTMAX = BGPWR1(JS)
      VBIASMAX   = ENPWR1(JS)
      AREA_END   = CULIM1(JS)
```

```
      DURM01   = DPINV2(JS)
      PVSET(1) = BGPWR2(JS)
      RBAT(1)  = ENPWR2(JS)
c     DUMMY(1) = CULIM2(JS)
cjrg  11-29-00 At least temporarily use this parameter to input
cjrg  end body collection area
          I_EMIT   = INT(CULIM2(JS))

      DURM02   = DPINV3(JS)
      PVSET(2) = BGPWR3(JS)
      RBAT(2)  = ENPWR3(JS)
      DUMMY(2) = CULIM3(JS)

      DURM03   = DPINV4(JS)
      PVSET(3) = BGPWR4(JS)
      RBAT(3)  = ENPWR4(JS)
      DUMMY(3) = CULIM4(JS)

      DURM04   = DPINV5(JS)

      PVSET(4) = BGPWR5(JS)
      RBAT(4)  = ENPWR5(JS)
      DUMMY(4) = CULIM5(JS)

      DURM05   = DPINV6(JS)
      PVSET(5) = BGPWR6(JS)
      RBAT(5)  = ENPWR6(JS)
      DUMMY(5) = CULIM6(JS)

C EXTRACT TETHER-SPECIFIC ELECTRO ATTRIBUTE (MINIMIZE INDEXING)
C--------------------------------------------------------
      PWFDUM = FACPWR(JTETH)

C INITIALIZE REDUCED-TIME PARAMETERS (TRIGGERED ON QTIME)
C-----------------------------------------------------
      IF(QTIME .EQ. 0.0) FSGH1(JTETH) = 0.0

C DETERMINE TOTAL DURATION OF ALL DEFINED (IE. NON-ZERO) PERIODS
      IF(QTIME .EQ. 0.0) TSGH1(JTETH) =
     1      DURM01 + DURM02 + DURM03 + DURM04 + DURM05

C--------------------------------------
C DO ABSOLUTE-TIME START/STOP DETERMINATION
C--------------------------------------
C GO TO NEXT TETHER IF ITS NOT TIME TO START THIS SCENARIO
      IF(QTIME .LT. TGENON(JS)) GO TO 2500

C---------------------------------------------------
```

```
C START REDUCED TIME AND SCENARIO PERIOD DETERMINATION
C-----------------------------------------------
C CALCULATE PERIOD-REDUCED TIME
   RTDUM = QTIME - TGENON(JS) - FSGH1(JTETH)*TSGH1(JTETH)

C USE NEW-CYCLE INITIATION TO ZERO CURRENT-INTEGRAL AND ALLOW FLOW
   IF(RTDUM .GE. TSGH1(JTETH)) THEN
      FSGH1(JTETH) = FSGH1(JTETH) + 1.0

C RESET INTEGRAL AND CURRENT-OK FLAG OF CURRENT WHEN A NEW OP CYCLE STARTS
      CURIN(JTETH) = 0.0
      CUROK(JTETH) = 0.0
   END IF


C CAPTURE SCENARIO DATA FOR APPROPRIATE PERIOD
C-----------------------------------------
C ASSUME PERIOD 1
      IPVSET = 1
   JMDUM = NINT(CMOD01)
   DLDUM = DURM01
   RLDUM = RTDUM
   IF(RLDUM .LE. DLDUM) GO TO 2100

C ASSUME PERIOD 2
      IPVSET = 2
   JMDUM = NINT(CMOD02)
   DLDUM = DURM02




   RLDUM = RLDUM - DURM01
   IF(RLDUM .LE. DLDUM) GO TO 2100

C ASSUME PERIOD 3
      IPVSET = 3
   JMDUM = NINT(CMOD03)
   DLDUM = DURM03
   RLDUM = RLDUM - DURM02
   IF(RLDUM .LE. DLDUM) GO TO 2100

C ASSUME PERIOD 4
      IPVSET = 4
   JMDUM = NINT(CMOD04)
   DLDUM = DURM04
   RLDUM = RLDUM - DURM03
   IF(RLDUM .LE. DLDUM) GO TO 2100

C ASSUME PERIOD 5
      IPVSET = 5
   JMDUM = NINT(CMOD05)
   DLDUM = DURM05
   RLDUM = RLDUM - DURM04
   IF(RLDUM .LE. DLDUM) GO TO 2100
```

```
C END OF PERIOD DETERMINATION
2100  CONTINUE
C-------------------------------------
C FETCH THE ASSOCIATED FINITE SOLN DATA IMAGE
C-------------------------------------
   JFTETH = LASIGN(JTETH)
   CALL TISLDW(JFTETH)

        IF (IPVSET .NE. IPVSETP) THEN
         IPVSETP = IPVSET
         IPVCHG = .TRUE.
        ELSE
         IPVCHG = .FALSE.
        ENDIF

C START W/ ZERO CURR AND MAKE OPS CYCLE ACTIVELY PRODUCE CURRENT
C------------------------------------------------------
   DO 111 JSEG = 1, NBEAD+1
      CURRTS(JSEG) = 0.0
111  CONTINUE


C CHECK STOP TIME, GO TO NEXT TETH IF ITS STOP TIME FOR THIS SCENARIO
C NOTE: STOP-TIME = 0.0  WILL DEFAULT TO AN ENDLESS REPETITION
   IF( (TGENOF(JS).GT.0.0) .AND. (QTIME.GT.TGENOF(JS)) ) GO TO 2600


C****************************************************
C****************************************************
C DETERMINE THE PLASMA STATE FOR CURRENT CALCULATIONS
C****************************************************



C****************************************************
C DETERMINE CONNECTIVITY OF THIS TETHER
   JOX = NOBJX(JTETH)
   JAX = LATTX(JTETH)
   JOY = NOBJY(JTETH)
   JAY = LATTY(JTETH)

C FIND A POSITION VECTOR TO MID-POINT OF TETHER IN PLANET FRAME
C------------------------------------------------------
C FORM A POSITION VECTOR FROM REF-END TO FAR-END OF TETHER
   DO 10 J = 1,3
      TOSVX1(J)  = APS(J+6, JAY, JOY) - APS(J+6, JAX, JOX)
10   CONTINUE

C THEN, FIND A POS VECTOR FROM INER PT TO MID POINT OF THIS TETHER
   DO 20 J=1,3
      DUMRMI(J)  = RPI(J) + APS(J+6, JAX, JOX) + .5*TOSVX1(J)
20   CONTINUE

C FIND PLANET FRAME COMPONENTS OF THIS POSITION VECTOR
```

```
      CALL EITEF (LJULOP,RPTIME,DUMRMI,   DUMRMP)

C GO GET ALTITUDE VIA PLANET GLOBE GEOMETRY MODEL
      CALL GEOD (LGLBOP,RPTIME,DUMRMP,   DUMR,DUMLAT,DUMALT)


C-----------------------------------------------------------
C SPECIFY PARAMETERS AND OPTIONS ASSOCIATED WITH PLASMA MODEL
C-----------------------------------------------------------
C DEFINE THE PLASMA EVALUATION INTERVALS
      HBGDUM = DUMALT
      HENDUM = DUMALT + 100.
      JBANDS = 1


C DEFINE VARIOUS LOGICAL OPTIONS FOR PLASMA DETERMINATION
C-------------------------------------------------------
C SPECIFY ELECTRON DENSITY TO BE CALCULATED
      JFDUM( 1) = .TRUE.

C SPECIFY TEMPERATURES TO BE CALCULATED
      JFDUM( 2) = .TRUE.

C SPECIFY ION COMPOSITION TO BE CALCULATED
      JFDUM( 3) = .TRUE.

C SPECIFY USE OF B0 FROM TABLE (FROM GULYEAVA 1987)
      JFDUM( 4) = .TRUE.

C SPECIFY USE OF F2 PEAK FROM CCIR (FROM URSI)
      JFDUM( 5) = .FALSE.

C SPECIFY USE OF ION COMP. STANDARD (DANILOV-YAICHNIKOV-1985)
      JFDUM( 6) = .TRUE.

C SPECIFY USE OF STANDARD IRI TOPSIDE (IRI-79)
      JFDUM( 7) = .TRUE.




C SPECIFY USE OF NMF2 PEAK MODEL (INPUT VALUES)
      JFDUM( 8) = .TRUE.

C SPECIFY USE OF HMF2 PEAK MODEL (INPUT VALUES)
      JFDUM( 9) = .TRUE.

C SPECIFY USE OF TE MODEL (TE-NE MODEL WITH NE INPUT)
      JFDUM(10) = .TRUE.

C SPECIFY USE OF Ne STANDARD (LAY-FUNCTIONS VERSION)
      JFDUM(11) = .TRUE.
```

```
C SPECIFY MESSAGES ARE WRITTEN TO UNIT=6 (=12)
   JFDUM(12) = .TRUE.


C THEN FIND ATMOSPHERIC PLASMA PROPERTIES AT THIS PT (AS HOST COMMANDS)
C-----------------------------------------------------------
C NOTE: THE IF STATEMENT BELOW IS ACTIVATED, BUT CANNOT BE VOUCHED-FOR
C UNTIL EXPERIENCE IS GAINED WITH USE AND MEANING OF THE PLASMA MODEL
C OUTPUT ARGUMENTS, AS INFREQUENT-EVALUATION OF INVIRONMENT IMPLIES THE
C SAVING OF RESULTS (LARGE ARRAY AT PRESENT) FROM THE PREVIOUS CALL FOR
C THE PLASMA STATE OF ALL TETHERS WHICH JUST MIGHT BE USING SPECIFYING
C THIS SCENARIO REGULATOR TYPE
   IF(LEVPLS .EQ. 1) THEN

C NOTE, PLASMA IS DATE SENSITIVE, SO DATE IS DERIVED FROM THE
C     TOSS STANDARD DATE CONTAINED IN COM_ALL.i (SEE PLASMA)
C-----------------------------------------------
     CALL PLASMA (LPLSOP,RPTIME,DUMRMP,HBGDUM,HENDUM,JBANDS,
   1   JFDUM,EDENS)

c  For simplicity at the present time, only a single value for the electron density
c  is used.  This seems adequate but should be periodically reviewed as usage changes.

cjrg  Even if only a single electron density is calculated in PLASMA, that value should
cjrg  be duplicated at each edens(i) corresponding to a bead i.  The following code
cjrg  makes that true.  It would need to be replaced or removed if above periodic
cjrg  review causes a revision

          do i = 2,nbeads(jteth)+1
           edens(i) = edens(1)
          enddo
   END IF

         CALL current_model(jteth,emidum,xinsfrac,pvset(ipvset),
   &  rbat(ipvset),vbiasmax,currentmax,area_end,ipvchg,i_emit)

C PLACE THIS VALUE INTO FINITE SOLN AREA FOR OTHERS TO USE

   EMICUR = EMIDUM

C-----------------------------------------------------------
C DO TOSS-TETHER-SPECIFIC POWER FACTOR RATIOING OF NON-UNIFORM CURRENT
C-----------------------------------------------------------





C (ALLOW EASY RATIOING FOR SENSITIVITY STUDIES)
C IF ZERO IS ENTERED, DEFAULT TO FACTOR OF 1.0


C SAVE THIS IMAGE OF THE FINITE TETHER
C----------------------------
2600  CALL TISSTW(JFTETH)
```

```
C STANDARD RETURN FOR SCENARIO
2500  CONTINUE

      RETURN
      END

c  This subroutine drives bare wire tether current collection model.
c  Developed by John Glaese from equations provided by Bob Estes/SAO.
c  Designed to fit in with GTOSS specifications for generality of application.
c  Originally developed for 3 body Step-Airseds configuration.  Modified to
c  allow user to configure for any model by input adjustment.
c                    Developed 2-15-00 by jrg
c                    Modified 7_27-00 by jrg

         subroutine current_model(jteth,emidum,xinsfrac,pvset,rbat4,
     & vbiasmax,currentmax,area_end,ipvchg,i_emit)

c        implicit none

C@NOS*CALL COM_ALL
C@NOS*CALL COM_RPS
C@NOS*CALL COM_TOSS
C@NOS*CALL EQU_TOSS
C@NOS*CALL COM_FOSS
C@NOS*CALL EQU_FOSS

C@UNX#include "COM_ALL.i"
C@UNX#include "COM_RPS.i"
C@UNX#include "COM_TOSS.i"
C@UNX#include "EQU_TOSS.i"
C@UNX#include "COM_FOSS.i"
C@UNX#include "EQU_FOSS.i"

C@VAX    INCLUDE 'COM_ALL.FOR'
C@VAX    INCLUDE 'COM_RPS.FOR'
C@VAX    INCLUDE 'COM_TOSS.FOR'
C@VAX    INCLUDE 'EQU_TOSS.FOR'
C@VAX    INCLUDE 'COM_FOSS.FOR'
C@VAX    INCLUDE 'EQU_FOSS.FOR'

         INCLUDE 'COM_ALL.INC'
         INCLUDE 'COM_RPS.INC'
         INCLUDE 'COM_TOSS.INC'
         INCLUDE 'EQU_TOSS.INC'
         INCLUDE 'COM_FOSS.INC'
         INCLUDE 'EQU_FOSS.INC'




C@MAC    INCLUDE '{GTOSS}A_TOSS:COM_ALL.i'
C@MAC    INCLUDE '{GTOSS}A_TOSS:COM_RPS.i'
C@MAC    INCLUDE '{GTOSS}A_TOSS:COM_TOSS.i'
```

```
C@MAC    INCLUDE '{GTOSS}A_TOSS:EQU_TOSS.i'
C@MAC    INCLUDE '{GTOSS}A_FOSS:COM_FOSS.i'
C@MAC    INCLUDE '{GTOSS}A_FOSS:EQU_FOSS.i'

          save

          integer*4 nndsmax,jteth,edmode,iins,itcount,itcount_max,i_emit
          parameter (nndsmax = maxseg, itcount_max = 50)

cjrg      real*4 xinsfrac,pvset,rbat4,vbiasmax,currentmax,r_xend(3),
cjrg     $ v_xend(3),area_end,bmag_0end_e(3),bmag_1end_e(3),
cjrg     $ bmag_e(3,nndsmax),bmag_0end_i(3),bmag_1end_i(3),
cjrg     $ bmag_i(3,nndsmax)
          real*8 xinsfrac,pvset,rbat4,vbiasmax,currentmax,r_xend(3),
     $ v_xend(3),area_end,bmag_0end_e(3),bmag_1end_e(3),
     $ bmag_e(3,nndsmax),bmag_0end_i(3),bmag_1end_i(3),
     $ bmag_i(3,nndsmax)
          integer*4 i,j,i3,ii1,ii2,nnodes,nmult,nmulth,nsegsmax,nsegs,
     $ napi0,nobji0,napi1,nobji1,indx(nndsmax),indxb(nndsmax)
          real*8 tethcur(nndsmax),tethpos(3,nndsmax),tethvel(3,nndsmax),
     $ bmag(3,nndsmax),edens_loc(nndsmax),xtemp,xmult,edynti(nndsmax)
          real*8 pos_0end(3),pos_1end(3),vel_0end(3),vel_1end(3),
     $ bmag_0end(3),bmag_1end(3),edens_0end,edens_1end,scrat1,scrat2
          real*8 volts0,current0,power,rbat,vbat,a_end,tethlen(nndsmax),
     $ dpwr,didpwr,dvdpwr,pwr0,pwr,cur0,cur,vlt0,vlt,
     $ v_error_tol,i_error_tol,cursgn

      parameter (nmultmax = 20,
     $       nsegsmax = nndsmax*nmultmax,
     $       v_error_tol = 1.d-5,
     $       i_error_tol = 1.d-5)

          real*8 current(nsegsmax),vbias(nsegsmax),curp(nsegsmax),
     $ vbiasp(nsegsmax),resistivity(nsegsmax),uteth(3,nsegsmax),
     $ pos(3,nsegsmax),vel(3,nsegsmax),bfield(3,nsegsmax),
     $ chgdens(nsegsmax),xins(nsegsmax),resist(nndsmax),
     $ dltlen(nsegsmax),perim(nsegsmax)

cjrg      real*4 emidum,m_f_ft,m_f_in
          real*8 emidum,m_f_ft,m_f_in
          logical ipvchg

          parameter (m_f_ft = 0.3048, m_f_in = 0.0254)
          data dpwr /1.d0/

c Plasma electron density and magnetic field vectors in GTOSS are updated
c at a reduced rate since they vary only slightly over regions of several
c km. This speeds execution of GTOSS with little compromise of accuracy.
c Similarly, we need only execute the current flow calculation when electron
c density or magnetic field or electrodynamic tether properties change.
c Thus, the following IF BLOCK causes the iteration loop of the ED current flow
c to be skipped except on the cycles when those quanties are updated.

          if (levpls .eq. 1 .or. levmag .eq. 1 .or. ipvchg) then
```

```
      nmult = 5
      nmulth = nmult/2

      if (pvset .gt. 0.0) then
        power = 1000.d0*pvset ! pvset in kilowatts
        edmode = 1 ! motor mode
      else
        power = 0.d0
        edmode = -1 ! generator mode
        vbat = - 1000.D0 * pvset
      endif

      nnodes = nbeads(jteth) + 2 ! Nodes = beads + 2 end points
      rbat = rbat4
      a_end = dble(area_end)
      nsegs = nmult * (nnodes - 1) ! Number of tether segments
      xmult = 1.d0 / dble(nmult)

      if (i_emit .eq. 0) then
        cursgn = 1.d0
        do i=1,nnodes-1
          indx(i) = i
          indxb(i) = i
        enddo
c Convert GTOSS stuff to feed bare_wire subroutine, x end = 0 end
        napi0 = lattx(jteth) ! number of attach pt at x end of jteth
        nobji0 = nobjx(jteth)! number of object of ap x end of jteth
        napi1 = latty(jteth) ! number of attach pt at y end of jteth
        nobji1 = nobjy(jteth)! number of object of ap y end of jteth
      else
        cursgn = -1.d0
        do i=1,nnodes-1
          indx(i) = nnodes - i
          indxb(i) = nnodes - 1 - i
        enddo
c Convert GTOSS stuff to feed bare_wire subroutine y end = 0 end
        napi1 = lattx(jteth) ! number of attach pt at x end of jteth
        nobji1 = nobjx(jteth)! number of object of ap x end of jteth
        napi0 = latty(jteth) ! number of attach pt at y end of jteth
        nobji0 = nobjy(jteth)! number of object of ap y end of jteth
      endif

      do i=1,nnodes-1
        resist(indx(i)) = rhoelt(i) / m_f_ft
        tethlen(indx(i)) = m_f_ft * elbsg(i) * xmult
c       tethlen(indx(i)) = m_f_ft * el / dble(nsegs)
      enddo

c This works on the assumption that tethers are active one at a time so that
c there is minimal data that must be carried forward to maintain the integration.

      do j=1,3
        r_xend(j) = rpi(j)  + aps(6+j,lattx(jteth),nobjx(jteth))
        v_xend(j) = rpid(j) + aps(9+j,lattx(jteth),nobjx(jteth))
        pos_0end(j) = m_f_ft*(rpi(j)  + aps(6+j,napi0,nobji0))
```

```
      vel_0end(j) = m_f_ft*(rpid(j) + aps(9+j,napi0,nobji0))
      pos_1end(j) = m_f_ft*(rpi(j)  + aps(6+j,napi1,nobji1))





      vel_1end(j) = m_f_ft*(rpid(j) + aps(9+j,napi1,nobji1))
      bmag_0end_e(j) = 1.e-4*bflseg(indx(1),j) ! Tesla from Gauss
      bmag_1end_e(j) = 1.e-4*bflseg(indx(nnodes-1),j)
     enddo

     call eftei(ljulop,rptime,bmag_0end_e,bmag_0end_i) ! Xform to inertial
     call eftei(ljulop,rptime,bmag_1end_e,bmag_1end_i) ! Xform to inertial
     do j=1,3
      bmag_0end(j) = bmag_0end_i(j)
      bmag_1end(j) = bmag_1end_i(j)
     enddo
     edens_0end = edens(indx(1))
     edens_1end = edens(indx(nnodes-1))

     do i=1,nnodes-1 ! Evaluate at midpoint of each tether segment
      edynti(indx(i)) = edynct(i)
      edens_loc(indx(i)) = edens(i)
      i3 = 3*(i-1)
      do j=1,3
       bmag_e(j,indx(i)) = 1.e-4*bflseg(i,j) ! Tesla from Gauss
      enddo
      call eftei(ljulop,rptime,bmag_e(1,indx(i)),bmag_i(1,indx(i)))
      do j=1,3
       bmag(j,indx(i)) = bmag_i(j,indx(i))
      enddo
     enddo

     do i=1,nnodes-2 ! Evaluate at tether bead
      i3 = 3*(i-1)
      do j=1,3
       tethpos(j,indxb(i)) = m_f_ft*(r_xend(j) + rubi(i3+j))
       tethvel(j,indxb(i)) = m_f_ft*(v_xend(j) + vubi(i3+j))
      enddo
     enddo
c Interpolate to find tether node position, velocity, bfield, charge density,
c needed in the following.
     do i = 1,nsegs
c                                     Quantities defined at a bead
      xtemp = (dble(i) - 0.5d0) * xmult
      ii1 = int(xtemp)
      ii2 = ii1 + 1
      xtemp = dmod(xtemp,1.d0)
      if (ii1 .eq. 0) then
       call linfit3(xtemp,pos(1,i),pos_0end,tethpos(1,ii2))
       call linfit3(xtemp,vel(1,i),vel_0end,tethvel(1,ii2))
      elseif(ii2 .lt. nnodes-1) then
       call linfit3(xtemp,pos(1,i),tethpos(1,ii1),tethpos(1,ii2))
       call linfit3(xtemp,vel(1,i),tethvel(1,ii1),tethvel(1,ii2))
      else
```

```
    call linfit3(xtemp,pos(1,i),tethpos(1,ii1),pos_1end)
    call linfit3(xtemp,vel(1,i),tethvel(1,ii1),vel_1end)
   endif
c                                    Quantities defined at midpoint of a segment
   xtemp = (dble(i) - 0.5d0) * xmult + 0.5d0
   ii1 = int(xtemp)
   ii2 = ii1 + 1



   xtemp = dmod(xtemp,1.d0)
   if (ii1 .eq. 0) then
    scrat1 = 2.d0*resist(ii2) - resist(ii2+1)
    scrat2 = resist(ii2)
    call linfit1(xtemp,resistivity(i),scrat1,scrat2)
    scrat1 = 2.d0*tethlen(ii2) - tethlen(ii2+1)
    scrat2 = tethlen(ii2)
    call linfit1(xtemp,dltlen(i),scrat1,scrat2)
    scrat1 = 2.d0*edynti(ii2) - edynti(ii2+1)
    scrat2 = edynti(ii2)
    call linfit1(xtemp,perim(i),scrat1,scrat2)
    call linfit3(xtemp,bfield(1,i),bmag_0end,bmag(1,ii2))
    call linfit1(xtemp,chgdens(i),edens_0end,edens_loc(ii2))
   elseif(ii2 .le. nnodes-1) then
    call linfit1(xtemp,resistivity(i),resist(ii1),resist(ii2))
    call linfit1(xtemp,dltlen(i),tethlen(ii1),tethlen(ii2))
    scrat1 = edynti(ii1)
    scrat2 = edynti(ii2)
    call linfit1(xtemp,perim(i),scrat1,scrat2)
    call linfit3(xtemp,bfield(1,i),bmag(1,ii1),bmag(1,ii2))
    call linfit1(xtemp,chgdens(i),edens_loc(ii1),edens_loc(ii2))
   else
    scrat1 = resist(ii1)
    scrat2 = 2.d0*resist(ii1) - resist(ii1-1)
    call linfit1(xtemp,resistivity(i),scrat1,scrat2)
    scrat1 = tethlen(ii1)
    scrat2 = 2.d0*tethlen(ii1) - tethlen(ii1-1)
    call linfit1(xtemp,dltlen(i),scrat1,scrat2)
    scrat1 = edynti(ii1)
    scrat2 = 2.d0*edynti(ii1) - edynti(ii1-1)
    call linfit1(xtemp,perim(i),scrat1,scrat2)
    call linfit3(xtemp,bfield(1,i),bmag(1,ii1),bmag_1end)
    call linfit1(xtemp,chgdens(i),edens_loc(ii1),edens_1end)
   endif
  enddo

  if (xinsfrac .ge. 0) then
   iins = nsegs*xinsfrac
   do i = 1,iins
    xins(i) = 0.d0 ! Insulated part
   enddo
   do i = iins+1,nsegs
    xins(i) = 1.d0 ! Bare part
   enddo
```

```
        else
         iins = nsegs*abs(xinsfrac)
         do i = nsegs,nsegs-iins+1,-1
          xins(i) = 0.d0 ! Bare part
         enddo
         do i = nsegs-iins,1,-1
          xins(i) = 1.d0 ! insulated part
         enddo
        endif

c  Integrated DE's for current.  Evaluate at integer points such that above
c  odd half integer points are at midpoints of nmult segments.




         do i=1,nsegs
         if (i .eq. 1) then
          uteth(1,i) = pos(1,i) - pos_0end(1)
          uteth(2,i) = pos(2,i) - pos_0end(2)
          uteth(3,i) = pos(3,i) - pos_0end(3)
         elseif (i .lt. nsegs) then
          uteth(1,i) = pos(1,i+1) - pos(1,i)
          uteth(2,i) = pos(2,i+1) - pos(2,i)
          uteth(3,i) = pos(3,i+1) - pos(3,i)
         else
          uteth(1,i) = pos_1end(1) - pos(1,i)
          uteth(2,i) = pos_1end(2) - pos(2,i)
          uteth(3,i) = pos_1end(3) - pos(3,i)
         endif
         xtemp = dsqrt(uteth(1,i)**2 + uteth(2,i)**2 + uteth(3,i)**2)
     uteth(1,i) = uteth(1,i) / xtemp
     uteth(2,i) = uteth(2,i) / xtemp
     uteth(3,i) = uteth(3,i) / xtemp
        enddo

        call bare_wire(volts0,current0,power,rbat,vbat,a_end,current,
   $   vbias,curp,vbiasp,resistivity,uteth,vel,bfield,chgdens,perim,
   $   xins,dltlen,nsegs,edmode)

c  jrg  New code 10-13-00
c  If motor mode is selected, check for current or voltage exceedance.

c  In motor mode, check for voltage or current exceedance.
c  Reset if either is too high.

        if (edmode .eq. 1) then
         if (volts0 .gt. vbiasmax) then
          pwr = dmin1(power,dble(vbiasmax * currentmax))
          itcount = 0
          vlt = volts0
          do while (dabs(vbiasmax - volts0) .gt. v_error_tol .and.
   $       itcount .lt. itcount_max)
           itcount = itcount + 1
           pwr0 = pwr
```

```
    call bare_wire(vlt0,cur,pwr,rbat,vbat,a_end,current,
$       vbias,curp,vbiasp,resistivity,uteth,vel,bfield,chgdens,
$       perim,xins,dltlen,nsegs,edmode)
            pwr = pwr0 + dpwr
            call bare_wire(vlt,cur,pwr,rbat,vbat,a_end,current,
$       vbias,curp,vbiasp,resistivity,uteth,vel,bfield,chgdens,
$       perim,xins,dltlen,nsegs,edmode)
            dvdpwr = (vlt - vlt0)/dpwr
            if (dvdpwr .ne. 0.d0) then
             pwr = pwr0 + (vbiasmax - vlt0)/dvdpwr
            else
             pwr = pwr0
             itcount = itcount_max
            endif
           enddo
           call bare_wire(volts0,current0,pwr,rbat,vbat,a_end,
$       current,vbias,curp,vbiasp,resistivity,uteth,vel,bfield,
$       chgdens,perim,xins,dltlen,nsegs,edmode)




         endif

          if (current0 .gt. currentmax) then
           pwr = dmin1(power,dble(vbiasmax * currentmax))
           itcount = 0
           cur = current0
           do while (dabs(currentmax - current0) .gt. i_error_tol .and.
$       itcount .lt. itcount_max)
            itcount = itcount + 1
            pwr0 = pwr
            call bare_wire(vlt0,cur0,pwr0,rbat,vbat,a_end,current,
$       vbias,curp,vbiasp,resistivity,uteth,vel,bfield,chgdens,
$       perim,xins,dltlen,nsegs,edmode)
            pwr = pwr0 + dpwr
            call bare_wire(vlt,cur,pwr,rbat,vbat,a_end,current,
$       vbias,curp,vbiasp,resistivity,uteth,vel,bfield,chgdens,
$       perim,xins,dltlen,nsegs,edmode)
            didpwr = (cur - cur0)/dpwr
            if (didpwr .ne. 0.d0) then
             pwr = pwr0 + (currentmax - cur0)/didpwr
            else
             pwr = pwr0
             itcount = itcount_max
            endif
           enddo
           call bare_wire(volts0,current0,pwr,rbat,vbat,a_end,
$       current,vbias,curp,vbiasp,resistivity,uteth,vel,bfield,
$       chgdens,perim,xins,dltlen,nsegs,edmode)
          endif
         endif

c jrg  End new code 10-13-00
```

```
         do i=1,nnodes-1
          tethcur(indx(i)) = cursgn * current(nmult*i-nmulth)
         enddo
        endif

c Sets current flow in each tether segment since these are zeroed each calculation
c cycle by gtoss.

        emidum = current0
        volts_out = volts0
        power_out = volts0 * current0

        do i=1,nbeads(jteth)+1
         currts(i) = tethcur(i)
        enddo

        return
        end


c Bare wire current subroutine based on equations supplied by Bob Estes/SAO
c 2-9-00.  Model developed by John Glaese.




c vbias(i)          ==          Bias of wire relative to plasma
c current(i)        ==          Current flowing through segment
c xins(i)      ==          Insulation factor (0 or 1)

c Routine takes in tether and environmental parameters at nnodes points along
c tether.

c The bare_wire subroutine solves a set of DE's to determine current
c along the tether.  The input parameters are
c          a_end         ==     1 end body current collecting area
c          resistivity   == tether resistivity
c          uteth         == unit tangent to tether at a bead
c          vel           ==     == tether velocity at a bead
c          bfield        == magnetic field at a bead
c          chgdens       == electron density at a bead location
c          perim         == tether perimeter, (assumed constant along tether)
c          xins          == insulation factor (0.0/1.0)
c          dltlen        == length of tether segment for DE integration
c          nsegs         == number of tether segments for DE solution

c The output parameters are
c          current0 == current flowing into end body (neutralizing current)
c          vbias0        ==     0 end body bias voltage
c          current(i)    == current in ith segment
c          vbias(i) == Bias of segment i

        subroutine bare_wire(vbias0,current0,power,rbat,vbat,a_end,
     $ current,vbias,curp,vbiasp,resistivity,uteth,vel,bfield,chgdens,
```

```
$ perim,xins,dltlen,nsegs,edmode)
      implicit none

      integer*4 nsegs,niter_max,iter_count,edmode
      real*8 pi,echarge,emass,curconst,error_tol,a_end,zer_one
      parameter (
$        pi = 3.14159 26535 89793 23846d0,
$    echarge = 1.602176d-19,
$      emass = 9.10938d-31,
$   curconst = 2.d0 * echarge**3 / emass /pi**2,
$  niter_max = 75,
$  error_tol = 2.5d-2)

      real*8 current(nsegs),vbias(nsegs),curp(nsegs),vbiasp(nsegs),
$ uteth(3,nsegs),vel(3,nsegs),bfield(3,nsegs),chgdens(nsegs),
$ xins(nsegs),resistivity(nsegs),perim(nsegs),dltlen(nsegs)
      real*8 current0,power,power0,powerx,vbias0,vbias1,vbias1x,
$ current1,current1x,vb_0,deriv,del_v,del_v0,rbat,vbat

      real*8 dvbias,dvbiasmax,vb1_plus

c This is a boundary value problem since we know current at one end and
c voltage or power at other end.  In generator (orbit lowering) mode, current
c is zero at the 1 end (the nsegs segment here) and voltage is zero or a small
c set bias value at the 0 end of tether.  In motor (orbit raising) mode, current
c is zero and voltage is adjustable.  The bias voltage at the 1 end is set and
c the differential equations are integrated along the tether to calculate
c current, voltage and power at the 0 end.  Numerical procedures are used to




c adjust the value of the bias voltage at the 1 end to reach the desired power
c setting.  There are max values on voltage and current levels.  These limits
c are implemented but not yet tested.

      dvbias = 1.d0
c     del_v0 = 1.d-2
      del_v0 = 5.d0
      del_v = del_v0
      dvbiasmax = 500.d0
      iter_count = 0

c Calculate starting values for 1 end bias voltage and current.

      vbias1 = 0.d0
      current1 = 0.d0

c Calculate starting value for vbias1 by integrating plasma potential along
c tether.  Force current to be 0 along tether by setting perimeter to 0.
      zer_one = 0.d0
      call de_integ(current0,vbias0,current1,vbias1,current,vbias,curp,
$ vbiasp,resistivity,chgdens,perim,xins,dltlen,bfield,uteth,vel,
$ zer_one,nsegs)
```

```
      vbias1 = - 0.5d0*vbias0
c      vbias1 = dmax1(10.d0,vbias1)
      vb_0 = vbias1

c Calculate starting value for current1 by applying endbody collection
c equations using vbias1 as the bias voltage

      vb1_plus = dmax1(0.d0,vbias1)
      current1 = chgdens(nsegs) * a_end * dsqrt(curconst * vb1_plus)

      zer_one = 1.d0
      call de_integ(current0,vbias0,current1,vbias1,current,vbias,
     $  curp,vbiasp,resistivity,chgdens,perim,xins,dltlen,bfield,
     $  uteth,vel,zer_one,nsegs)

      power0 = power + 2.d0*error_tol

c Iteration loop

c     do while (dabs(dvbias) .gt. error_tol .and.
c   $  iter_count .le. niter_max)

      do while (dabs(power - power0) .gt. error_tol .and.
     $  iter_count .le. niter_max)

       vbias1x = vbias1 + del_v
       vb1_plus = dmax1(0.d0,vbias1x)
       current1x = chgdens(nsegs) * a_end * dsqrt(curconst * vb1_plus)

       call de_integ(current0,vbias0,current1,vbias1x,current,vbias,
     $  curp,vbiasp,resistivity,chgdens,perim,xins,dltlen,bfield,
     $  uteth,vel,zer_one,nsegs)

        if (edmode .ge. 0) then
         powerx = vbias0 * current0
        else
         powerx = vbias0 + vbat + rbat * current0
        endif
        deriv = (powerx - power0)/del_v
        if (deriv .ne. 0.d0) then
         if (iter_count .gt. 5) then
          dvbias = 0.25d0*(power - power0)/deriv
         else
          dvbias = (power - power0)/deriv
         endif
         dvbias = dmax1(dmin1(dvbias,dvbiasmax),-dvbiasmax)
        else
         dvbias = 0.d0
        endif
```

```
          vbias1 = vbias1 + dvbias
          vb1_plus = dmax1(0.d0,vbias1)
          current1 = chgdens(nsegs) * a_end * dsqrt(curconst * vb1_plus)

          iter_count = iter_count + 1
c         del_v = dmin1(del_v0,dabs(dvbias))

          call de_integ(current0,vbias0,current1,vbias1,current,vbias,
     $    curp,vbiasp,resistivity,chgdens,perim,xins,dltlen,bfield,
     $    uteth,vel,zer_one,nsegs)
          if (edmode .ge. 0) then
            power0 = vbias0 * current0
          else
            power0 = vbias0 + vbat + rbat * current0
          endif
        enddo

        return
        end

        subroutine linfit3(x,f,f0,f1)
        implicit none

        real*8 x,f(3),f0(3),f1(3)

        f(1) = (1.d0 - x)*f0(1) + x*f1(1)
        f(2) = (1.d0 - x)*f0(2) + x*f1(2)
        f(3) = (1.d0 - x)*f0(3) + x*f1(3)

        return
        end

        subroutine linfit1(x,f,f0,f1)
        implicit none

        real*8 x,f,f0,f1

        f = (1.d0 - x)*f0 + x*f1




        return
        end

c The following subroutine integrates the differential equations for
c barewire current collection with specified initial values at the
c 1 end for current and bias voltage.

        subroutine de_integ(cur0,vol0,cur1,vol1,curi,voli,curip,volip,
     $ resistivity,chgdens,perim,xins,dltlen,bfield,uteth,vel,zer_one,
     $ nsegs)
        implicit none
```

```
      integer*4 i,nsegs
      real*8 pi,echarge,emass,const

 parameter (pi = 3.14159 26535 89793 23846d0,
$    echarge = 1.602176d-19,
$      emass = 9.10938d-31,
$   const = echarge**3 / emass /pi**2)

      real*8 uteth(3,nsegs),vel(3,nsegs),bfield(3,nsegs),chgdens(nsegs),
$ xins(nsegs),resistivity(nsegs),perim(nsegs),dltlen(nsegs),vol0,
$ cur0,vol1,cur1,voli(nsegs),curi(nsegs),volip(nsegs),zer_one,
$ curip(nsegs),dblvp

      voli(nsegs) = vol1
      curi(nsegs) = cur1

      do i = nsegs,1,-1
        if (i .gt. 1) then
          volip(i) = - resistivity(i) * curi(i) +
$   ( uteth(1,i) * (vel(2,i)*bfield(3,i) - vel(3,i)*bfield(2,i))
$   + uteth(2,i) * (vel(3,i)*bfield(1,i) - vel(1,i)*bfield(3,i))
$   + uteth(3,i) * (vel(1,i)*bfield(2,i) - vel(2,i)*bfield(1,i)))
          voli(i-1) = voli(i) - volip(i) * dltlen(i)
          dblvp = dmax1(0.d0,zer_one*(voli(i) + voli(i-1)))
          curip(i) = - xins(i)*chgdens(i)*perim(i)*dsqrt(const*dblvp)
          curi(i-1) = curi(i) - curip(i) * dltlen(i)
        else
          volip(i) = - resistivity(i) * curi(i) +
$   ( uteth(1,i) * (vel(2,i)*bfield(3,i) - vel(3,i)*bfield(2,i))
$   + uteth(2,i) * (vel(3,i)*bfield(1,i) - vel(1,i)*bfield(3,i))
$   + uteth(3,i) * (vel(1,i)*bfield(2,i) - vel(2,i)*bfield(1,i)))
          vol0 = voli(i) - volip(i) * dltlen(i)
          dblvp = dmax1(0.d0,zer_one*(voli(i) + vol0))
          curip(i) = - xins(i)*chgdens(i)*perim(i)*dsqrt(const*dblvp)
          cur0 = curi(i) - curip(i) * dltlen(i)
        endif
      enddo

      return
      end
```

Appendix 2: END BODY ATTITUDE CONTROL SIMULATION

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
USERES("EulerDynamics.res");
USEFORM("Unit1.cpp", Form1);
USEUNIT("Functions.cpp");
//---------------------------------------------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    return 0;
}
//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

#pragma hdrstop
#include "Unit1.h"
#include "Functions.h"
#include "Math.h"

double ControlTorX, ControlTorY, ControlTorZ;
double dum1,dum2,dum3,dum4,dum5,dum6,dum7,dum8,dum9;
double AcX, AcY;
double MovMomX, MovMomY, MovMomZ,TmmX,TmmY;

void Get_Forces_Moments(double time)
{
double ft, ft0, del_time;
 Form1->FX = Form1->FX;
}
```

```
// ***************************************************************************
// TRANS1 function definitions.  Returns derivitives of integrated variables



// ***************************************************************************

double trans1(int flag, double t, double * w, double * k, int flag2, double*x)
{
double ForceFX, ForceFY, ForceFZ, MomentMX, MomentMY, MomentMZ, answer;
 int i;
double ErrorXprop, ErrorYprop, ErrorZprop, ErrorXrate, ErrorYrate, ErrorZrate;
double GainPropX, GainPropY, GainPropZ, Wc, W0, zeta, GainRateX, GainRateY, GainRateZ;
double vx_global, vy_global, vz_global;



zeta = 0.707; Wc = 0.60;   MomentMX = 0.0, MomentMY = 0.0, MomentMZ = 0.0;
ForceFX = 0.0; ForceFY = 0.0; ForceFZ = 0.0;




/*

            w[1] = u          w[2] = v
            w[3] = w          w[4] = p
            w[5] = q          w[6] = r
            w[7] = h          w[8] = t
        w[9] = s
        w[10] = Hcx
        w[11] = Hcy
        w[12] = Hcz
            etc
            flag 1 = return first dirivitive u  ==> body x acc
            flag 2 = return first dirivitive v ==> body y acc
            flag 3 = return first dirivitive w  ==> body z acc
        flag 4 = return first dirivitive p  ==> body x rot acc
            flag 5 = return first dirivitive q ==> body y rot acc
            flag 6 = return first dirivitive r  ==> body z rot acc
        flag 7 = return first dirivitive h  ==> euler angle phi rate
```

```
                flag 8 = return first dirivitive t ==> euler angle theta rate
                flag 9 = return first dirivitive s  ==> euler angle psi rate
                etc.

*/


switch (flag2)      // flag2 tells which of the four pass kutta operation
{               // is happening
        case (1):
            for (i=1;i<=Form1->no_first_orders;++i)
                        x[i] = w[i];
                        break;
        case (2):
                        for (i=1;i<=Form1->no_first_orders;++i)
                        x[i] = w[i] + k[i]/2.0;
                        break;
        case (3):
                        for (i=1;i<=Form1->no_first_orders;++i)
                        x[i] = w[i] + k[i]/2.0;
                        break;
        case (4):
                        for (i=1;i<=Form1->no_first_orders;++i)
                        x[i] = w[i] + k[i];
                        break;
        default:
                        break;
        }               // End of switch


/// Calculate time dependent forces and moments
// Get_Forces_Moments(t);

W0 = -0.001122; // Orbit rate

double pp = x[4] - W0*cos(x[8])*sin(x[9]);;
double qq = x[5] - W0*(cos(x[7])*cos(x[9]) + sin(x[7])*sin(x[8])*sin(x[9]));
double rr = x[6] - W0*(-sin(x[7])*cos(x[9]) + cos(x[7])*sin(x[8])*sin(x[9]));

// Integrate to get global xyz
// transform body coordinates to global earth coordinates now
// First convert to global ref frame

dum1 = cos(x[9])*cos(x[8]);
dum2 = cos(x[9])*sin(x[8])*sin(x[7]) - sin(x[9])*cos(x[7]);
```

```
dum3 = sin(x[9])*sin(x[7]) + cos(x[9])*sin(x[8])*cos(x[7]);
dum4 = sin(x[9])*cos(x[8]);
dum5 = cos(x[9])*cos(x[7]) + sin(x[9])*sin(x[8])*sin(x[7]);
dum6 = sin(x[9])*sin(x[8])*cos(x[7]) - cos(x[9])*sin(x[7]);
dum7 = -sin(x[8]);
dum8 = cos(x[8])*sin(x[7]);
dum9 = cos(x[8])*cos(x[7]);

vx_global = dum1*x[1] + dum2*x[2] + dum3*x[3];
vy_global = dum4*x[1] + dum5*x[2] + dum6*x[3];
vz_global = dum7*x[1] + dum8*x[2] + dum9*x[3];



//--------------------------------------------------------


// override get_forces for internally calculated force values

Form1->FX = -10.0*sin(x[8]);
Form1->FY =  10.0*cos(x[8])*sin(x[7]);
Form1->FZ =  10*cos(x[8])*cos(x[7]);
Form1->MX =  -5.0*cos(x[8])*sin(x[7]);
Form1->MY =  -5.0*sin(x[8]);

// Now add tether skip-rope force
double A, B, WS, DUMFX, DUMFY, DUMFZ, DUMMX, DUMMY, DUMMZ;

WS = 0.014;  // Skip rope freq
A = 1.5708*cos(WS*t);  B = 1.5708*sin(WS*t);

DUMFX = A*cos(x[8])*cos(x[9]) + B*cos(x[8])*sin(x[9]);
DUMFY = A*(sin(x[7])*sin(x[8])*cos(x[9]) - cos(x[7])*sin(x[9])) +
     B*(sin(x[7])*sin(x[8])*sin(x[9]) + cos(x[9])*cos(x[7]));
DUMFZ = A*(cos(x[7])*sin(x[8])*cos(x[9]) + sin(x[9])*sin(x[7])) +
     B*(cos(x[7])*sin(x[8])*sin(x[9]) - sin(x[7])*cos(x[9]));
DUMMX = -0.5*DUMFY;
DUMMY = 0.5*DUMFX;

ForceFX = Form1->FX + DUMFX;
ForceFY = Form1->FY + DUMFY;
ForceFZ = Form1->FZ + DUMFZ;
MomentMX = Form1->MX + DUMMX;
```

```
MomentMY = Form1->MY + DUMMY;


// Now calculate the controller torque
//Because it is a PD and does not have and integral controller there
//is no need to integrate before applying to plant
// If I had an integral controller it would be easier to take a derivitive of
// everything and then I would have an Moment dot and integrate it before
//passing to plant

 ErrorXprop = sin(x[7])*cos(x[8]) - (cos(x[7])*sin(x[8])*sin(x[9])-sin(x[7])*cos(x[9]));
 ErrorYprop = cos(x[7])*sin(x[8])*cos(x[9]) + sin(x[9])*sin(x[7]) + sin(x[8]);
 ErrorZprop = cos(x[8])*sin(x[9])- (sin(x[7])*sin(x[8])*cos(x[9]) - cos(x[7])*sin(x[9]));

 ErrorXrate = x[4] - W0*(cos(x[8])*sin(x[9]));
 ErrorYrate = x[5] - W0*(sin(x[7])*sin(x[8])*sin(x[9]) + cos(x[9])*cos(x[7]));
 ErrorZrate = x[6] - W0*(cos(x[7])*sin(x[8])*sin(x[9]) - sin(x[7])*cos(x[9]));

 GainPropX = Form1->IXX*Wc*Wc; GainPropY = Form1->IYY*Wc*Wc;  GainPropZ = Form1-
>IZZ*Wc*Wc;
 GainRateX = Form1->IXX*2.0*zeta*Wc; GainRateY = Form1->IYY*2.0*zeta*Wc; GainRateZ = Form1-
>IZZ*2.0*zeta*Wc;

// For Yaw Control
// GainPropX = Form1->IXX*Wc*Wc*0.0; GainPropY = Form1->IYY*Wc*Wc*0.0;  GainPropZ =
Form1->IZZ*Wc*Wc;
// GainRateX = Form1->IXX*2.0*zeta*Wc/10.0; GainRateY = Form1->IYY*2.0*zeta*Wc/10.0;
GainRateZ = Form1->IZZ*2.0*zeta*Wc;



 ControlTorX = -(GainPropX*ErrorXprop + GainRateX*ErrorXrate);
 ControlTorY = -(GainPropY*ErrorYprop + GainRateY*ErrorYrate);
 ControlTorZ = -(GainPropZ*ErrorZprop + GainRateZ*ErrorZrate);

// Put in torque

MomentMX = MomentMX + ControlTorX;
MomentMY = MomentMY + ControlTorY;
MomentMZ = MomentMZ + ControlTorZ;

// Now handle the moving control application point control AcX and AcY
// Tmm = -Kmm(Ht - HtDes)    Assume HtDes = 0.0, Kmm = 0.02

TmmX = -0.005*(w[4]*Form1->IXX + w[10]);
```

```
TmmY = -0.005*(w[5]*Form1->IYY + w[11]);
AcX = -TmmY/ForceFZ;  AcY = TmmX/ForceFZ;

// Now calc moving app point ,moment effect and add it in

 MovMomX = AcY*ForceFZ;
 MovMomY = -AcX*ForceFZ;
 MovMomZ = AcX*ForceFY - AcY*ForceFX;
 MomentMX = MomentMX + MovMomX;
 MomentMY = MomentMY + MovMomY;
 MomentMZ = MomentMZ + MovMomZ;



 switch (flag)
  {
   case (1):
                 answer = (-Form1->mass*Form1->gravity*sin(x[8]) + ForceFX +
              x[2]*x[6]*Form1->mass - x[3]*x[5]*Form1->mass)/Form1->mass;
                 break;
   case (2):
                 answer = (Form1->mass*Form1->gravity*sin(x[7])*cos(x[8]) + ForceFY +
              x[3]*x[4]*Form1->mass - x[1]*x[6]*Form1->mass)/Form1->mass;
                 break;
   case (3):
                 answer = (Form1->mass*Form1->gravity*cos(x[7])*cos(x[8]) + ForceFZ +
              x[1]*x[5]*Form1->mass - x[2]*x[4]*Form1->mass)/Form1->mass;
                 break;
   case (4):
                 answer = (MomentMX + (Form1->IXZ/Form1->IZZ)*(MomentMZ-
              (Form1->IYY-Form1->IXX)*x[4]*x[5]- Form1->IXZ*x[5]*x[6]) +
               Form1->IXZ*x[4]*x[5] - (Form1->IZZ-Form1->IYY)*x[5]*x[6])/
              (Form1->IXX - (Form1->IXZ*Form1->IXZ/Form1->IZZ));
                 break;
   case (5):
                 answer = (MomentMY - (Form1->IXX-Form1->IZZ)*x[4]*x[6] -
               Form1->IXZ*(x[4]*x[4] - x[6]*x[6]))/Form1->IYY;
                 break;
   case (6):
                 answer = (MomentMZ + (Form1->IXZ/Form1->IXX)*(MomentMX +
               Form1->IXZ*x[4]*x[5]-(Form1->IZZ - Form1->IYY)*x[5]*x[6])
              -Form1->IXZ*x[5]*x[6]- (Form1->IYY - Form1->IXX)*x[4]*x[5])/
              (Form1->IZZ - (Form1->IXZ*Form1->IXZ/Form1->IXX));
                 break;
   case (7):
```

```
        answer =  pp + (qq*sin(x[7]) + rr*cos(x[7]))*sin(x[8])/cos(x[8]);
                //answer =  x[4] + (x[5]*sin(x[7]) + x[6]*cos(x[7]))*
        //      sin(x[8])/cos(x[8]);   where pp = x[4] qq = x[5] rr = x[6]

                break;
case (8):
                //answer =  x[5]*cos(x[7]) - x[6]*sin(x[7]);
         answer =  qq*cos(x[7]) - rr*sin(x[7]);
                break;
case (9):
            //  answer =  (x[5]*sin(x[7]) + x[6]*cos(x[7]))/cos(x[8]);
        answer =  (qq*sin(x[7]) + rr*cos(x[7]))/cos(x[8]);
                break;
case (10):

        answer =  -ControlTorX - x[5]*x[12] + x[6]*x[11];
         //answer = 0.0;
                break;
case (11):

        answer =  -ControlTorY - x[6]*x[10] + x[4]*x[12];
        //answer= 0.0;
                break;
case (12):

        answer =  -ControlTorZ - x[4]*x[11] + x[5]*x[10];
         //answer = 0.0;
                break;
case (13):

        answer =  vx_global;
         //answer = 0.0;
                break;
case (14):

        answer =  vy_global;
         //answer = 0.0;
                break;
case (15):

        answer =  vz_global;
         //answer = 0.0;
                break;
case (16):

        answer =  x[4];
```

```
            //answer = 0.0;
                    break;
  case (17):

            answer =  x[5];
            //answer = 0.0;
                    break;
  case (18):

            answer =  x[6];
            //answer = 0.0;
                    break;

  default:
                    answer = 0.0;
                    break;
 }
 return answer;
}

// ****************************************************************************
// Runge Kutta
// ****************************************************************************
void RKutta()
{
 long loop_size, i,j,k;
 double t, delt;
 double *k1, *k2, *k3, *k4, *w, *x;
 double xrel, yrel, zrel, PtX_Global, PtY_Global, PtZ_Global;

 ofstream fout ("string.txt");
 k1 = new double[Form1->no_first_orders + 1];
 k2 = new double[Form1->no_first_orders + 1];
 k3 = new double[Form1->no_first_orders + 1];
 k4 = new double[Form1->no_first_orders + 1];
 w = new double[Form1->no_first_orders + 1];
 x = new double[Form1->no_first_orders + 1];
 FILE *inputdata;
 FILE *outputdata;
  if ((outputdata = fopen("Euler-Glaese.out","w")) == NULL)
  {
        printf("Can not open file 'euler.out'.\n");
        exit(1);
  }
```

```
        if ((inputdata = fopen("initial.dat","r")) == NULL)
        {
         printf("Can not open file.\n");
         exit(1);
        }
// input initial conditions xvel yvel zvel rotx vel, roty vel, rotz vel,
// Euler angles thX, thY, thZ
// Control Moment Gyro Momentums Hcx Hcy Hcz
// These are in body NOT global coordinates
         for ( k = 1; k <= Form1->no_first_orders; ++k )
          fscanf (inputdata, "%lf", &w[k]);
   // Now input x y z global position of cg
  // fscanf(inputdata,"%lf %lf %lf", &x_global, &y_global, &z_global);
   // Now input relative point on body  xrel, yrel, zrel
   fscanf(inputdata,"%lf %lf %lf", &xrel, &yrel, &zrel);
    fclose (inputdata);



/*
              w[1] = u                    w[2] = v
              w[3] = w                    w[4] = p    omega_x
              w[5] = q omega_y            w[6] = r    omega_z
              w[7] = h  Euler roll Phi    w[8] = t Euler pitch theta
         w[9] = s  Euler Yaw Psi
         w[10] = Hcx   Control Moment Gyro Momentums
         w[11] = Hcy
         w[12] = Hcz
         w[13] = xGlobal
         w[14] = YGlobal
         w[15] = ZGlobal
         w[16] = thetax body radians
         w[17] = thetay body
         w[18] = thetaz body
              etc

*/

// Initialize variables

delt = Form1->delt;
//loop_size = 5/delt;
loop_size = 80000;
t = 0.0;

/*
```

```
        Below the following holds:
                trans1(1,,...) returns w[1]dot
                trans1(2,,...) returns w[2]dot
                etc
*/

for (j=1; j<=loop_size; ++j)
{
        for(i=1; i<=Form1->no_first_orders; ++i)
        {
         k1[i] = delt*trans1(i, t, w,k1,1,x);
         //if (i==2) acc_x = k1[i]/delt;
        }
        for(i=1; i<=Form1->no_first_orders; ++i)
        {
         k2[i] = delt*trans1(i, (t+delt/2.),w,k1,2,x);
        }
        for(i=1; i<=Form1->no_first_orders; ++i)
        {
         k3[i] = delt*trans1(i, (t+delt/2.),w,k2,3,x);
        }
        for(i=1; i<=Form1->no_first_orders; ++i)
        {
         k4[i] = delt*trans1(i, (t+delt),w,k3,4,x);
        }
        for(i=1;i<=Form1->no_first_orders;++i)
  {
         w[i] = w[i] + (k1[i] + 2.0*k2[i] + 2.0*k3[i] + k4[i])/6.0;
  }
        t = t + delt;


 // Print Global X Y Z, Euler Roll Pitch Yaw, Body Roll Pitch Yaw

if(j%50 == 0)
{
 fprintf (outputdata,"%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf
%lf %lf %lf %lf %lf %lf %lf %lf %lf\n",
       t, 360.0*w[7]/6.28,360.0*w[8]/6.28, 360.0*w[9]/6.28,
       ControlTorX, ControlTorY, ControlTorZ, w[10], w[11], w[12],
       w[4]*Form1->IXX + w[10], w[5]*Form1->IYY + w[11], w[6]*Form1->IZZ + w[12],
       w[4], w[5], w[6], w[13], w[14], w[15], 360.0*w[16]/6.28, 360.0*w[17]/6.28,
       360.0*w[18]/6.28, AcX, AcY, MovMomX, MovMomY, MovMomZ, TmmX, TmmY);
 }
```

```
PtX_Global = dum1*xrel + dum2*yrel + dum3*zrel;
PtY_Global = dum4*xrel + dum5*yrel + dum6*zrel;
PtZ_Global = dum7*xrel + dum8*yrel + dum9*zrel;


} // End of j loop
 fclose (outputdata);

}


//------------------------------------
```

Appendix 3: RENO TRIP REPORT

January 19, 2001

To: Randy Baggett and Ken Welzyn, MSFC/NASA

From: John Glaese, Control Dynamics Division of bd Systems

Subject: Reno Trip Report

On the 7th of January, I traveled to Reno, Nevada to attend the 39th Aerospace Sciences Meeting and Exhibit, an AIAA sponsored conference scheduled 8-11 January 2001 at the Reno Hilton in Reno, Nevada. The following paragraphs summarize the meetings and discussions held at this conference.

Because of my lack of coordination, no meetings had been set up for Monday morning and no rendezvous plan established with TMTC/Andrew Santangelo. First part of day was consumed trying to establish contact. Met with Andrew in early afternoon and participated in several side meetings to discuss Step-Airseds status and plans.

Monday, 8 January: Establish initial contact with TMTC.

1:30pm:
Jon Van Noord, Rich Sturmfels and Murshed Khadija gave a presentation on the status of tether studies. They promised to provide electronic-copy of the presentation charts but none has been received yet.

Items discussed:
1. Tether width vs lifetime studies were presented. A tether width of 2.5 cm seems to be favored. Widths greater than this seem to result in little lifetime gain and considerable increase in mass and drag. The knee of the lifetime curve (probability of surviving 1 year vs. tether width) occurs between 2 and 2.5 cm Widths less than this result in shorter lifetimes but it appears that all these trades were done at 10 km tether length. These trades also seem to favor porous over solid tethers. The pores tend to stop crack propagation caused by meteoroid or debris hits. Trades of bare tether length vs. drag and vs. time to raise orbit were also done. These suggest that a bare tether section 1 km to 2 km in length is sufficient for orbit raising. The time to lower the orbit with the shorter bare wire is lengthened significantly. End body current collection on the upper body may compensate for this. An alternate approach would be to use a powered deboost. End body collection on both end bodies as an alternate to a bare wire tether is also being looked at.
2. A consideration, which arose in discussion of the lifetime study, is the likelihood that the long life tether will receive hits that do not sever but cause damage, such as to coatings or broken wires. Such damage may make the tether more likely to jam the deployer during yo-yo or prevent re-deploy after retrieval because of frayed and tangled wires. Alternatively, such damage may increase the likelihood of arcing at or near the HVPS.
3. As part of the bare wire length study, TMTC stated that they felt a requirement for deboost time vs altitude was needed, as well as, a minimum inert system survival time. The first helps set minimum bare wire length or the need for powered deboost or end body collection. The second sets a minimum required

altitude for orbit insertion. TMTC agreed that S-A insertion should be set to the highest achievable altitude. This is expected to approach 500 km.

4. Discussed flyaway/separation studies done with GTOSS by Murshed Khadija. He varied separation thrust level, length of burn and angle of initial separation. Studies were done using massless tether for expediency. Thrust levels were varied between 0.5 N and 2.0 N. The deployer was commanded to deploy tether matching the uniform acceleration theoretically produced by the thrust applied to the upper end body which contains the deployer and the thrusters. Deployer velocity after the end of thrusting was held constant at the velocity achieved to that time. Results seemd to favor the smaller thrust levels in that less slack was produced. An additional study will be done in which shutoff time will be determined by time to reach 500 m separation and the deployer velocity after cutoff will be ramped down to zero, arriving at zero with full tether deployment. As an aside, apparently, the idea of two deployers has been dropped.

Tuesday, 9 January: Deployer Test Plan Review
Items discussed:

1. Slack tether an issue during test phase;
2. Instrumentation/what measurements to be taken;
3. Status of preparations for test: many items will not be done until prototype is built (TMTC wants to call this proto-flight but is not to fly): weights, pack/spool size, resistance, damage tolerance, etc.

Review of Tether Technical Committee Proposal Presentation (copy of presentation available)
Items discussed:

1. Proposed formation of Tethers Technical Committee, presented by Andrew Santangelo;
2. Definition of committee goals;
3. Statement of TC goals and basis for charter;
4. History of Tether activities in conferences and projects – need for a TC
5. Proposed membership, proposed 30-35 members, many from local (SA) tether community;
6. Proposed organization, Andrew as TC chairman;
7. Future plans – meet minimum twice per year, next meeting at JPC, candidates ASM and JPC select each year.

Wednesday, 10 January, 8am:Neil Rothwell, Double R Controls, arrived, continued Deployer Test Plan Discussion, Review with Andrew, et. al.
Neil had questions about TSS deployer hardware and apparently had reviewed some material on its design and development. His questions suggested his material did not describe the upper tether control mechanism since he was unaware of the vernier motor and the pinch pulley and their function in the TSS deployer. Neil wondered if he would be to be able to view TSS hdw. I agreed to ask Randy to see if this is possible/`feasible.
Items discussed:

1. Continued discussion of prototype vs. proto-flight. The intent here is that unit will be made with flight type design, hardware and materials where feasible, but without flight qualification paperwork.
2. Overviewed Deployer Test Plan Outline. Never really got into details. Need to get copy of this presentation. Test plan items related to deployer requirements. First item to be discussed is Deployer Success Rate. This would be a full deploy, retrieve. The take-up reel provides tension. Not clear how will be done, especially with slack tether if, as now, is a part of deploy. Murshed's flyaway studies being

done with minimal tension or some slack. This may be difficult to duplicate in deployer testing because of 1 g conditions and need to avoid tether entanglement due to slack. Probably desirable to maintain some level of tension. Told this to TMTC. Neil Rothwell, Double R Controls seems comfortable to work with minimal outboard tension, though he didn't address slack. He is confident his deployer will deploy with no outboard tension. I am currently favoring length or length rate control of deployer. Neil supports this. Determination of reel size and tether mass/volume will be done analytically early on. Neil seems confident in this. Later on will verify with hdw. Size depends on flight tether selection, i.e. web, mesh, solid, solid with holes. Post test inspection of tether for damage will be visual.
Second item Non-deploy capability – demonstrate hold against tension maximum 100 lbs., use pinch rollers to hold against tension. Apply tension forces with pinch rollers. Keeps loads from being transmitted to reel.
6:30pm: New Tethers TC Organizational Dinner and Meeting.
Dinner meeting to discuss TC organization. Tentative list of TC members. Proposed organizational chart. Randy Baggett, Ken Welzyn, John Glaese suggested for TC. Ken suggested to chair Technical Affairs subcommittee.

Thursday, 11 January, 8am - 12 noon: Attended conference's only tether session -- final day. This session is dominated by papers related to Step-Airseds. Not surprising since S-A is active right now and Andrew Santangelo is vocal sponsor of AIAA, Andrew is also session chairman.

1:30pm: Panel discussion scheduled for 1:30 pm but cancelled because only S-A players showed up. Abbreviated discussion was held but of little consequence. Scheduled meeting for 4:30pm Andrew's suite.
4:30pm: Discussion centered around reports due and what content was expected. Bob Strunce is aiming to finish his part of a report by end of January. I was pushing Bob and others too, to write out their Ops philosophy for Step-Airseds to make it explicit so that the all the issues related to the accomplishment of S-A can be brought out from the shadows into full view. A big item is current control. This will likely play into libration control and skip rope. It also plays into accurate orbital navigation, i.e. getting from a payload pickup point to a payload drop-off point. It's not obvious to me that anyone has really thought about how this might be done. However it is done seems to me will require accurate electrodynamic thrust control, which translates into current control. Another item is end body attitude control. What modes need to be supported and when? Bob Strunce talks about using flywheels for energy storage and says that there is a sponsor (American Flywheel?) who would be willing to defray some of the cost development cost in order to fly a demonstration of a combined energy storage/attitude control system. Current flywheels are significantly undersized for holding attitude in the face of tether tension disturbances. Another item is use of the deployer. We have talked about using yo-yoing to control libration but no analysis has been done to quantify this. How often is it required? What is the amplitude and how does it interact with current operations? Is it necessary if current modulation is being employed to control libration and skip rope? Is partial tether retrieval required to rendezvous and dock with a customer payload? These and probably other issues need to be addressed and resolved to define S-A operations and help get our hands around the mission. Likely, not all of these things are appropriate for the S-A mission but if it is to demonstrate electrodynamic boost technology, it seems that all these are potentially a part of it. Sooner or later to be useful system and compete with alternate

technologies in cost and capabilities, Step-Airseds must address these things. Bob Strunce said he would like to submit sections of the report to me for comments as he completes them. I agreed to this.
5:30pm: Andrew introduced Kevin Probst and told everyone that Kevin was coming in as TMTC's new S-A Project Manager. This action is being taken to allow Andrew to focus more time in the strategic (business) development of the company and free him of day-to-day details of S-A. He still plans to maintain an overview of activities and serve as the customer interface including contract related items. Kevin will assume responsibility for Step-Airseds activities and team leaders will report to him. Kevin addressed the group and reviewed his planned management style which will involve part-time at his home in Boulder, CO and part-time at TMTC. He plans to have a team leads meeting every two weeks to be coordinated with and held in advance of the bi-weekly telecons with NASA. He wants a more focused approach to these telecons and hopes to head off what he apparently feels is a tendency to wander (my words).

The conference ended mid afternoon so our evening meeting concluded activities except for a joint dinner, which closed out activities. My return flight was at 12:32 am on January 12th and since it was snowing outside, I decided to go on to the airport. It was nearing 10 pm by this time. The return trip was uneventful for which I am thankful.

Appendix 4: INGOSS, GTOSS INPUT FILE SETUP FOR S-A FLYAWAY

```
$THR067  RDB FAMILY NAME FOR SERIES (W/ INITIAL SEQ NUMBER XX)
  NOMINAL TMTC RUNSTREAM FOR INITIAL AIRSEDS DEPLOY STUDIES
  PERTINENT DATA PLUS RELATED OPTIONS ONLY INCLUDED

C===========================================================================
C NOTE: THIS PARAMETER STUDY INPUT SKELETON HAS MANY GTOSS/TOSS
C EXECUTION OPTIONS/FEATURES NOT PRESENT SO AS TO SIMPLIFY IT. THE
C REMOVED OPTIONS WOULD NOT PERTAIN TO THIS BASIC DEPLOYMENT SIZING
C STUDY FOR AIR-SEDS. WHILE INITIALLY SET UP FOR A MASSLESS TETHER,
C THE DATA PLACE-HOLDERS FOR SPECIFYING AND ACTIVATING A FINITE
C TETHER HAVE BEEN RETAINED SO THAT THIS SAME INPUT STREAM CAN BE
C EASILY MODIFIED TO ACTIVATE A FINITE TETHER.
C===========================================================================


C YOU MAY INSERT COMMENT CARDS (AS THESE INDEED ARE), OR TOTALLY
C BLANK CARDS AT YOUR DISCRETION.  AS SHOWN, THIS DECK (WITH ALL
C THE COMMENTS AND BLANK CARDS), CAN BE READ, AND IS LOGICALLY
C ACCEPTABLE TO GTOSS, EVEN THOUGH THE DATA IS NOT MEANINGFUL.
C THIS DECK IS TO SERVE AS A TEMPLATE AND CHECK LIST FOR BUILDING
C GTOSS/TOSS INPUT DECKS, AND YOU WOULD USUALLY DELETE A GREAT
C DEAL OF THE CARDS (INCLUDING THESE COMMENTS).

C------------------------------------------
C GTOSS  AND  REF PT DATA IS ENTERED IN I3,F15 FORMAT
C (COMMENTS IN COL 19 OR GREATER ARE IGNORED AND LOST)
C------------------------------------------
C THIS MUST PRECEDE ANY DATA ITEMS YOU WANT TO CHANGE FOR A LATE START
397  0.0        =1.0 TO REQUEST A SNAP SHOT LATE START INITIALIZATION

C GENERAL REF PT PARAMETERS
  1  0.10       INTEGRATION STEPSIZE INITIALLY USED BY GTOSS
  2  10800.     MAX TIME FOR RUN (SEC)
  3  500.       N, THE RDB SOLN OUTPUT INTERVAL = N * STEPSIZE
 16  0.0        RUN NUMBER
 94 -1.0        INHIBIT DOT OUTPUT TO CRT WITH -1.0

C CRT DISPLAY CONTROL (DEFAULT 0. = NO OUTPUT, IE. BATCH MODE OPERATION)
C (NOTE: CRT OUTPUT CAN BE A SLOW PROCESS RELATIVE TO THE SOLUTION OF
C  NON COMPUTE-BOUND TETHER SIMULATIONS. THAT IS, WHILE THE SOLN SEEMS
C  TO BE POURING FORTH AT BREAK-NECK SPEED, IT MAY PROCEED MUCH FASTER
C  IF CRT OUTPUT IS NOT ACTIVE)
 17  1.         = M CAUSES OUTPUT EVERY M SOLN RDB INTERVALS (ITEM 3)

C INTEGRATION STEPSIZE STAGING DATA (IN THE CURRENT BERSION OF GTOSS,
C THIS CAN BE PARTICULARLY USEFUL FOR DEPLOYMENTS OR RETRIEVALS WITH A
C FACTOR OF 2 (OR GREATER) TETHER LENGTH CHANGE

  5  0.0        ABSOLUTE TIME FOR  1ST INTEGRATION STEPSIZE CHANGE
```

```
 9 0.0000    STEPSIZE INVOKED AT 1ST CHANGE
71 0.        RDB OUTPUT INTERVAL 1ST CHANGE

 6 0.0       ABSOLUTE TIME FOR  2ND INTEGRATION STEPSIZE CHANGE
10 0.0000    STEPSIZE INVOKED AT 2ND CHANGE
72 0.        RDB OUTPUT INTERVAL 2ND CHANGE

 7 0.0       ABSOLUTE TIME FOR  3RD INTEGRATION STEPSIZE CHANGE
11 0.0000    STEPSIZE INVOKED AT 3RD CHANGE
73 0.        RDB OUTPUT INTERVAL 3RD CHANGE

 8 0.0       ABSOLUTE TIME FOR  LAST INTEGRATION STEPSIZE CHANGE
12 0.0000    STEPSIZE INVOKED AT LAST CHANGE
74 0.        RDB OUTPUT INTERVAL LAST CHANGE

C REFERENCE DATE (WHICH CORRESPONDS TO ZERO SIMULATION TIME FOR THE RUN)
287 0.0      YEAR, FOR EXAMPLE 1998.0
288 0.0      MONTH, FOR EXAMPLE  5.0 (BETWEEN 1.0 AND 12.0)
289 0.0      DAY,  FOR EXAMPLE  17.0 (DEPENDS ON THE MONTH)
290 0.0      HOUR, FOR EXAMPLE   8.0 (BETWEEN 0.0 AND 24.0)
291 0.0      MIN,  FOR EXAMPLE  42.0 (BETWEEN 0.0 AND 60.0)
292 0.0      SEC,  FOR EXAMPLE  23.0 (BETWEEN 0.0 AND 60.0)

C LATE START SNAP SHOT INITIALIZATION CONTROL
398 0.0      =1.0 IF SNAP SHOTS DATA DUMPS ARE TO BE TAKEN
387 0.0      TIME FOR  1ST SNAP SHOT TO BE TAKEN
388 0.0      TIME FOR  2ND SNAP SHOT TO BE TAKEN
396 0.0      TIME FOR 10TH SNAP SHOT TO BE TAKEN

C QUICK LOOK PAGE FORMAT CONTROL
117 0.       SELECT QUICK LOOK PAGE FORMAT
381 2.0      CHOOSE TOSS OBJ FOR QUICK LOOK (IF APPROP)
382 0.       "    "
421 1.0      CHOOSE TOSS TETHER FOR QUICK LOOK (IF APPROP)
442 0.0      BEAD NUMBER TO DISPLAY FOR QUICK LOOK (IF APPROP)
443 0.       "    "
444 0.       "    "
435 0.0      CHOOSE FINITE SOLN FOR QUICK LOOK (IF APPROP)

C REF PT/GTOSS LOCAL EXECUTION CONTROL DATA
 92 0.       = 1. TO ACTIVATE GRAV GRAD TORQUE ON REF PT OBJECT
110 0.       = 1. TO INHIBIT READING TOSS DATA AND EXECUTING TOSS
111 0.       = 1. TO INHIBIT ROTATIONAL DYNAMICS (PARTICLE OPT)
112 0.       REF PT SPECIFIC EULER ANG TYPE FOR INPUT (DEFAULTS 1)
113 2.       NUMBER OF LAST TOSS OBJECT BEING SIMULATED
114 1.       NUMBER OF ATTACH PTS ON THE REF POINT


C-----------------------------------------------
C REF PT/TOSS GLOBAL PLANETARY ENVIRONMENT MODEL OPTIONS
C-----------------------------------------------
C REF PT/TOSS GLOBAL PLANETARY ENVIR MODEL SELECTION/FIDELITY OPTIONS
481 0.       GRAVITY MODEL (0.0=SPHERICAL; 1.0=SIMPLE OBLATE,ETC)
482 0.       ATMOSPHERIC DENSITY MODEL (0.0=EARTH ARDC 1976)

483 0.       MAGNETIC FIELD (0.0=EARTH TILTED/SHIFTED DIPOLE)
```

```
484 0.        PLANET GLOBE SHAPE (0.0 =SPHERICAL EARTH)
485 0.        ATMOSPHERIC WINDS MODEL (0.0=ROTATING EARTH,ETC)
488 0.        ATMOSPHERIC PLASMA MODEL (0.0=IRI95)
486 0.        INER. FRAME MODEL (0.0=FOR INITIALLY-EARTH-ALIGNED)
487 0.        GLOBAL EULER ANGLE INPUT TYPE (NON-PREEMPTIVE IF =0)
490 0.        CONST PLANET RADIUS (DEFAULT = GEOD VALUE AT T=0)

C REF PT/TOSS GLOBAL PLANETARY ENVIRONMENT MODEL EVALUATION FREQUENCY
561 0.      =1. PREEMPTS ALL ENVIRON'S TO EVAL EVERY RP TIME STEP

C SPECIFY ACTUAL LAPSED TIME BETWEEN ENVIRONMENT EVALUATIONS
C (TIMES LESS THAN REF PT DELTA-T WILL FORCE EVAL EVERY REF PT STEP)
563 0.000     TIME BETWEEN ATMOS EVAL  (=0. DEFAULTS TO EVERY STEP)
564 0.000     TIME BETWEEN MAGNET EVAL (=0. DEFAULTS TO EVERY STEP)
565 0.000     TIME BETWEEN WIND EVAL   (=0. DEFAULTS TO EVERY STEP)
566 0.000     TIME BETWEEN PLASMA EVAL (=0. DEFAULTS TO EVERY STEP)


C BASIC REF PT GEOMETRY
C------------------
20 49.68      REFERENCE PT MASS (SLUGS)
21 121.       IXX = -INTEGRAL (X*X) (SLUG-FT**2)
22 121.       IYY
23 121.       IZZ

24 0.0        XCG: REF PT CENTER OF MASS LOC (FT)
25 0.         YCG
26 0.         ZCG

27 0.0        IXY = -INTEGRAL (X*Y) (SLUG-FT**2)
28 0.         IXZ
29 0.         IYZ

30 0.         INER VARIATION W/MASS OPTION (1.NONE,2.LINEAR,3.QUAD)


C REF PT TRANSLATION STATE INITIALIZATION
C--------------------------------
C =0. WR/T TOPO FRAME;  =1. WR/T INER FRAME;   =2 CIRCULAR ORBIT
100 2.        SELECT REF PT TRANSLATION STATE IC OPTION

C FOR TRANSLATION IC OPTION = 1.0
81 0.0        XIO REF PT POSITION (FT) [INER FRAME]  (OPT 1)
82 0.         YIO "      "                 (OPT 1)
83 0.         ZIO "      "                 (OPT 1)
84 0.0        XIDO REF PT RATE (FT/SEC) [INER FRAME] (OPT 1)
85 0.         YIDO "     "                 (OPT 1)
86 0.         ZIDO "     "                 (OPT 1)

C FOR TRANSLATION IC OPTION = 0.0 AND 2.0
101 1312336.   REF PT  ALT (FT)         (OPT 0 AND OPT 2)
102 -80.6      REF PT  TOPO LONGITUDE (DEG)  (OPT 0 AND OPT 2)
103  28.5      REF PT  TOPO LATITUDE  (DEG)  (OPT 0 AND OPT 2)
107 0.0        VXTO INER VEL TOPO COMP(FPS)  (OPT 0)

108 0.         VYTO "     "        (OPT 0)
```

```
109 0.      VZTO  "      "        (OPT 0)
80  0.0     CIRC VEL AZIM WR/T TOPO FRAME (DEG,EAST=0) (OPT=2)


C REF PT BODY ATTITUDE STATE INITIALIZATION
C-------------------------------------
C =0. EULER TO ORBIT FRM;  =1. EULER TO TOPO FRM;  =2.EULER TO INER FRM
90  0.0     SELECT BODY ATTITIDE OPTION

104 -30.     EULER PITCH ANGLE (DEG)
105 0.      EULER ROLL  ANGLE
106 0.      EULER YAW   ANGLE

C REF PT BODY RATE STATE INITIALIZATION
C----------------------------------
C =0.BODY COMP;  =1.ORB RATE;  =2.EUL RATE/INER FRM, =3.EUL RATE/ORB FRM
93  1.0     SELECT BODY RATE OPTION

87  0.0     BODY-X ANG VEL, CAN BE EULER ROLL  RATE (DEG/SEC)
88  0.      BODY-Y ANG VEL, CAN BE EULER PITCH RATE
89  0.      BODY-Z ANG VEL, CAN BE EULER YAW   RATE


C ATTACH PT COORDS (1 SHOWN, 8 POSSIBLE)
134 0.0      PXBT1: REF PT (FT) X COORD OF ATTACH PT
142 0.0      PYBT1: REF PT (FT) Y COORD OF ATTACH PT
150 1.64     PZBT1: REF PT (FT) Z COORD OF ATTACH PT

C SIMPLE REF POINT AERO DRAG OPT DATA
455 0.       =1. TO ACTIVATE SIMPLE DRAG OPTION ON REF PT
456 0.0      AERO REFERENCE AREA FOR REF PT OBJECT (FT**2)
457 0.0      SIMPLE DRAG COEFFICIENT (NON-DIMENS)


C---------------------------------------------
C ACTIVATE ATTITUDE CONTROL ALGORITHMS ON HOST OBJECT
C---------------------------------------------
C (NOTE SPECIFYING GREATER THAN 2. RESULTS IN ADDITIONAL OPTIONS)
807 1.      0.=OFF, 1.=WR/T ORB FRM, 2.=WR/T INER FRM, 3.=ETC

C RATE AND ERROR FEEDBACK GAINS FOR OPTION ACTIVATED BY ITEM 807
808  50.0     RATE FEEDBACK GAIN (X-AXIS)(Modified rate gains)
809  50.0     RATE FEEDBACK GAIN (Y-AXIS)
810  50.0     RATE FEEDBACK GAIN (Z-AXIS)
811  10.0     ATT ERR FEEDBACK GAIN (X-AXIS)
812  10.0     ATT ERR FEEDBACK GAIN (Y-AXIS)
813  10.0     ATT ERR FEEDBACK GAIN (Z-AXIS)


C--------------------------------
C FLAG TO ACTIVATE ARBITRARY BODY FORCES
C--------------------------------
817 1.      0.=OFF, 1.=TABLE LOOKUP, 2.=ALGORITHM-A, 3.=ALGOR-B


C TABLE DATA DEFINING ARBITRARY HOST X-BODY AXIS FORCES VS TIME
```

```
819 0.0      MAX TIME COVERED BY THIS TABLE
820 0.0      VALUE OF X-BODY FORCE AT MAX TIME

821 0.0      FIRST TIME VALUE
822 0.0      FIRST VALUE OF X BODY FORCE
C ...........ETC...........UP TO 20 TIME/FORCE POINT SETS ALLOWED
859 0.0      TWENTIETH TIME VALUE
860 0.0      TWENTIETH VALUE OF X BODY FORCE


C TABLE DATA DEFINING ARBITRARY HOST Y-BODY AXIS FORCES VS TIME
862 0.0      MAX TIME COVERED BY THIS TABLE
863 0.0      VALUE OF Y-BODY FORCE AT MAX TIME

864 0.0      FIRST TIME VALUE
865 0.0      FIRST VALUE OF Y BODY FORCE
C ...........ETC...........UP TO 20 TIME/FORCE POINT SETS ALLOWED
902 0.0      TWENTIETH TIME VALUE
903 0.0      TWENTIETH VALUE OF Y BODY FORCE


C TABLE DATA DEFINING ARBITRARY HOST Z-BODY AXIS FORCES VS TIME
905 20000.     MAX TIME COVERED BY THIS TABLE
906 0.0000     VALUE OF Z-BODY FORCE AT MAX TIME

907 0.0        1ST TIME VALUE
908 -0.0674    1ST VALUE OF Z BODY FORCE
909 2000.      2ND TIME VALUE
910 -0.0674    2ND VALUE OF Z BODY FORCE
911 2000.1     3RD TIME VALUE
912  0.0000    3RD VALUE OF Z BODY FORCE
913 20000.     4TH TIME VALUE
914  0.0000    4TH VALUE OF Z BODY FORCE
915 10000.     5TH TIME VALUE
916  0.0000    5TH VALUE OF Z BODY FORCE
917 10000.     6TH TIME VALUE
918  0.0000    6TH VALUE OF Z BODY FORCE
919 10000.     7TH TIME VALUE
920  0.0000    7TH VALUE OF Z BODY FORCE

C ...........ETC...........UP TO 20 TIME/FORCE POINT SETS ALLOWED
945 0.0      TWENTIETH TIME VALUE
946 0.0      TWENTIETH VALUE OF Z BODY FORCE

C GTOSS DATA TERMINATOR CARD IS MANDATORY, DEFINED AS  0 IN I3 FIELD
 0          END OF GTOSS AND REF POINT DATA


C*********************************************
C*********************************************
C BEGIN READ IN OF TOSS GENERAL INTEGER DATA
C*********************************************
C*********************************************
C A MANDATORY ALPHANUMERIC CARD PRECEDES EACH TOSS DATA CATEGORY

C A MANDATORY DATA TERMINATOR CARD FOLLOWS EACH TOSS DATA CATEGORY
```

```
C----------------------------------------------------
C TOSS DATA ENTERED AS:  INTEGER: I4,I6,A50  REAL:I4,E16,A50
C (THE A50 FIELD ALLOWS RE-DISPLAY OF COMMENT DATA IF DESIRED)
C
C AS IT PERTAINS TO FINITE SOLNS, ALL DATA SHOWN IS FOR SOLN 1.
C TO FIND INPUT DATA ITEM NUMBER FOR SOLN 2, ALL SHOWN INPUT ID
C NUMBERS ARE INCREMENTED BY 1. (ETC FOR SOLN 3). SIMILARLY FOR
C TOSS TETHERS, EXCEPT TYPICALLY ITEM NUMBERS ARE SHOWN FOR 3.
C----------------------------------------------------

======> GENERAL <====== READ-IN LTOSQ ARRAY HERE - INTEGER CONSTANTS

C GENERAL ACTIVATION OPTIONS APPLYING TO ALL OF TOSS
C----------------------------------------------
 22    0 ACTIVATE WRITE-OUT OF TOSS INPUT DATA STREAM
 23    0 ACTIVATE PARTICLE DYNAMICS            PREEMPTS ALL OBJECTS
 126   0 ACTIVATE GRAVITY GRAD BODY TORQUE     PREEMPTS ALL OBJECTS
 128   0 ACTIVATE DYNAMICS VERIFICATION CALCS (=1 ACTIVATES)
 197   0 ACTIVATE DEPLOYING TETHER MASS FLOW EFFECTS
 155   0 INITIALIZE PASSIVE DAMPER MODE (0=ACTIVE, 1=QUIESCENT)

C GENERAL SPECIFICATION OPTIONS APPLYING TO ALL OF TOSS
C----------------------------------------------
 125   0 SPECIFY EULER ANGLE TYPE FOR DATA INPUT  PREEMPTS ALL OBJECTS
 127   0 SPECIFY LIBR. ANGLE TYPE FOR DATA INPUT  PREEMPTS ALL OBJECTS


C DATA APPLYING TO ALL TETHERS
C----------------------
 24    1 SPECIFY TOTAL NUMBER OF TOSS TETHERS (FINITE+MASSLESS)

 194   0 ACTIVATE GRAVITATIONAL  FORCES FOR ALL TOSS TETHERS
 195  -1 ACTIVATE AERODYNAMIC    FORCES FOR ALL TOSS TETHERS
 196  -1 ACTIVATE ELECTRODYNAMIC FORCES FOR ALL TOSS TETHERS

C DEFINITION OF TOSS TETHER CONNECTIVITY
 25    1 OBJECT TO WHICH "X" END ATTACHES-TOSS TETHER #1
 50    1 ATT PT FOR "X" END OF TOSS TETHER #1

 75    2 OBJECT TO WHICH "Y" END ATTACHES-TOSS TETHER #1
 100   1 ATT PT FOR "Y" END OF TOSS TETHER #1


C ASSOCIATE FINITE SOLN NUMBERS WITH TOSS TETHER NUMBERS
 130   0 ASSIGN A FINITE SOLN NUMBER TO TOSS TETHER #1
 284   0 SET UP STABLE GRAVITY GRADIENT TENSION IN TOSS TETHER #1
 259   1 ASSIGN A DEPLOYMENT SCENARIO DATA SET TO TOSS TETHER #1
 234   0 ASSIGN AN ELEC POWER SCENARIO DATA SET TO TOSS TETHER #1


C----------------------------------------
C THERMAL SIMULATION CONTROL FOR ALL TOSS TETHERS
C----------------------------------------

 177   0 DIRECT SOLAR RADIATION HEATING        (=1 ACTIVATES)
```

```
178   0 PLANET ALBEDO HEATING            (=1 ACTIVATES)
179   0 PLANET BLACK BODY RADIATION HEATING   (=1 ACTIVATES)
180   0 AERODYNAMIC HEATING              (=1 ACTIVATES)
181   0 ELECTRICAL RESISTIVE HEATING        (=1 ACTIVATES)

182   0 HEAT RADIATION FROM A TETHER        (=1 ALLOWS)
183   0 HEAT CONDUCTION ALONG A TETHER        (=1 ALLOWS)
184   0 THERMAL EXPANSION CHANGES TO TETHER LENGTH  (=1 ALLOWS)


C DATA APPLYING TO FINITE SOLUTIONS (1ST IN SERIES OF 9 POSSIBLE SHOWN)
C--------------------------------------------------------------------
129   0 NFINIT, LARGEST FINITE SOLN NUMBER ALLOWED TO BE ACTIVE (ALL)
198   0 SPECIFY SOLN TYPE (1=STD, 2=HST, 3=HST_UN-SYM)  FINITE SOLN 1
207   0 NUMBER OF BEADS ASSIGNED TO              FINITE SOLN 1
185   0 WAVE SHAPE-VELOCITY IC OPTION             FINITE SOLN 1
343   0 SKIP ROPE IC ACTIVATOR (1 = CW, 2 = CCW)      FINITE SOLN 1
352   0 SKIP ROPE MODE SHAPE   (1, 2, 3, 4, ...)      FINITE SOLN 1

C DO NOT USE THESE NEXT TWO ITEMS UNLESS YOU KNOW WHAT YOU ARE DOING
164   0 IF=-1, INHIBITS STOPS ON HST CONSTRAINT EXCEEDANCE  ALL SOLNS
165   0 PREEPTS HST SOLN TENS REF END (X-END =1, Y =2)  FINITE SOLN 1

  0    0       END OF DATA

 ------------------ READ-IN JTOSQ ARRAY HERE - INTEGER VARIABLES
  0    0       END OF DATA


C*****************************************
C*****************************************
C BEGIN READ IN OF TOSS GENERAL REAL DATA
C*****************************************
C*****************************************

 ------------------ READ-IN FTOSQ ARRAY HERE - REAL CONSTANTS

C************************************************************
C NOMINAL, BASIC PHYSICAL DATA APPLYING TO ALL UNIFORM TETHERS
C************************************************************
C TOSS TETHER INITIAL DEPLOYED, UNDEFORMED LENGTH
 50       2.033 UN-STRETCHED LENGTH (FT)........TOSS TETHER #1

C LINEAL DENSITY FOR TOSS TETHERS (ASSUMED UNIFORM)
1223        15.5 LINEAL DENSITY (LBM/1000FT).....TOSS TETHER #1

C YOUNGS MODULUS FOR TOSS TETHERS (ASSUMED UNIFORM)
1248       5.0E+6 YOUNGS MODULUS (PSI)............TOSS TETHER #1

C EFFECTIVE ELASTIC DIAMETER (ASSUMED UNIFORM)
1273       0.1405 ELASTIC DIAMETER (IN)...........TOSS TETHER #1

C STRESS-PROPORTIONAL STRAIN RATE DISSIPATION COEFF (ASSUMED UNIFORM)
1298        0.010 BETA, DISSIPATION COEFF (SEC)...TOSS TETHER #1
```

```
C EFFECTIVE AERODYNAMIC DIAMETER (ASSUMED UNIFORM)
1323        0.0 AERODYNAMIC DIAMETER (IN).......TOSS TETHER #1


C NON-LINEAR STIFFNESS MODEL STRAIN BIAS CONSTANT (ASSUMED UNIFORM)
C-------------------------------------------------------
1348        0.0 NON-LINEAR STRAIN BIAS (ND).....TOSS TETHER #1

C NON-LINEAR STIFFNESS MODEL EXPONENT, >0 TO ACTIVATE (ASSUMED UNIFORM)
1373        0.0 NON-LINEAR EXPONENT (ND)........TOSS TETHER #1


C VALUE TO INITIALIZE AMPERAGE IN TOSS TETHERS (EASY CONST CURRENT OPT)
C------------------------------------------------------------
 181    0.0     INITIAL/CONST UNIFORM CURRENT (AMPS) TOSS TETHER #1


C MULTIPLIERS FOR DEPLOY AND POWER GENERATION SCENARIOS
C----------------------------------------------
C MULTIPLIER USED BY TOSS TETHER FOR ANY ASSIGNED POWER GEN DATA SET
 335    0.0     PWR MULTIPLIER (DEFAULTS TO 1.0) - TOSS TETHER #1

C MULTIPLIER USED BY TOSS TETHER FOR ANY ASSIGNED DEPLOYMENT DATA SET
 360    0.0     MULTIPLIER (DEFAULTS TO 1.0) - TOSS TETHER #1


C****************************************
C DATA APPLYING ONLY TO MASSLESS TETHERS
C****************************************
C ADDITIONAL ELASTIC TETHER LENGTH FROM ANCHOR PT TO TOSS ATTACH PT
C AFFECTS ONLY SPRING RATE CALCULATION (AND IS ALWAYS IN EFFECT)
C---------------------------------------------------
1137        0.0  ANCHOR LENGTH BIAS (FT) - MASSLESS TOSS TETHER #1

C USE CONSTANT SPRING-RATE INSIDE THIS: ANCHOR + DEPLOYED LENGTH
C SPRING-RATE CLAMPS AT A VALUE EQUIVALENT TO THIS PROXIMITY LENGTH
1162      1000.0  PROXIMITY LENGTH (FT)   - MASSLESS TOSS TETHER #1

C ALTERNATIVE MASSLESS TETHER STIFFNESS AND DISSIPATION SPECIFICATION
C------------------------------------------------------------
C (THESE VALUES ARE BASED ON INITIAL LENGTHS, IE. ITEM 50,..., ABOVE)
C THIS IS USED ONLY IF TOSS TETHER YOUNGS MODULUS IS ZERO
  25        0.0  SPRING RATE (LB/FT) - MASSLESS TOSS TETHER #1

C THIS IS USED ONLY IF TOSS TETHER DISSIPATION COEFFICIENT IS ZERO
  75        0.0  END-END DAMPNG COEF (LB/FPS) MASSLESS TOSS TETHER #1



C THERMAL PROPERTIES FOR FINITE TETHERS
C------------------------------
1506    0.0     PROPERTIES BASELINE TEMPERATURE       FINITE SOLN 1
1398    0.0     TETH HEAT CONDUCTIVITY /UNIT AREA     FINITE SOLN 1
1407    0.0     THERMAL LINEAR EXPANSION COEFF (/K)   FINITE SOLN 1

1416    0.0     ABSORPTIVITY IN SOLAR SPECTRUM        FINITE SOLN 1
```

```
1425    0.0    EMISSIVITY IN RADIANT SPECTRUM      FINITE SOLN 1
1434    0.0    SPECIFIC HEAT/UNIT MASS J/(KG-K)    FINITE SOLN 1
1443    0.0    SLOPE OF SPECIFIC HEAT WR/T TEMP    FINITE SOLN 1
1470    0.0    CONDUCTOR RESISTIVITY /UNIT LENGTH   FINITE SOLN 1
1479    0.0    SLOPE OF RESISTIVITY WR/T TEMP      FINITE SOLN 1

C THERMAL EFFECTS IMPLIED BELOW ARE NOT CURRENTLY IMPLEMENTED
1452    0.0    SLOPE OF YOUNGS MODULUS WR/T TEMP   FINITE SOLN 1
1461    0.0    SLOPE OF DAMPING WR/T TEMP          FINITE SOLN 1
1488    0.0    HEAT CONDUCTIVITY AT X-END TETH AP  FINITE SOLN 1
1497    0.0    HEAT CONDUCTIVITY AT Y-END TETH AP  FINITE SOLN 1


C FRACTIONAL SEG LENGTH ERROR AT WHICH HST SOLN WILL NOTIFY USER
C-----------------------------------------------------
C (GENERALLY NOT USED UNLESS YOU KNOW WHAT YOU ARE DOING)
 207    0.0    FRACTONAL ERROR LIMIT (DEFAULT = 0.01) FINITE SOLN 1

C HST CONSTRAINT STABILIZATION FEEDBACK GAINS FOR SEG/SEG-RATE ERRORS
C (GENERALLY NOT USED UNLESS YOU KNOW WHAT YOU ARE DOING)
 216    0.0    ERROR FDBK GAIN MULTIPLIER (DEFAULT =1.0)  ALL SOLNS
 217    0.0    RATE  FDBK GAIN MULTIPLIER (DEFAULT =1.0)  ALL SOLNS


C-----------------------------------
C DATA FOR DEPLOYMENT DATA SETS 1 THRU 10
C-----------------------------------
C (SEE REF MANUAL FOR DEFINITIONS OF SCENARIO TYPES)

C DEFINE DEPLOYMENT DATA SET NUMBER 1
C-------------------------------
 520    1.0     DEPLOYMENT SCENARIO TYPE  FOR DATA SET 1
 530    0.0     ABSOLUTE TIME TO START DEPLOY  DATA SET 1

 540  500.0     PERIOD DURATION      - PERIOD 1, DATA SET 1
 550    0.000   BEGINING DEPLOY LEVEL - PERIOD 1, DATA SET 1
 560    1.093   ENDING  DEPLOY LEVEL - PERIOD 1, DATA SET 1
 570    0.00   LIMIT CRITERIA  VALUE - PERIOD 1, DATA SET 1

 580  500.0     PERIOD DURATION      - PERIOD 2, DATA SET 1
 590    1.093   BEGINING DEPLOY LEVEL - PERIOD 2, DATA SET 1
 600    2.186   ENDING  DEPLOY LEVEL - PERIOD 2, DATA SET 1
 610    0.00   LIMIT CRITERIA  VALUE - PERIOD 2, DATA SET 1

 620  500.0     PERIOD DURATION      - PERIOD 3, DATA SET 1
 630    2.186   BEGINING DEPLOY LEVEL - PERIOD 3, DATA SET 1
 640    3.279   ENDING  DEPLOY LEVEL - PERIOD 3, DATA SET 1
 650    0.00   LIMIT CRITERIA  VALUE - PERIOD 3, DATA SET 1

 660  500.0     PERIOD DURATION      - PERIOD 4, DATA SET 1
 670    3.279   BEGINING DEPLOY LEVEL - PERIOD 4, DATA SET 1
 680    4.372   ENDING  DEPLOY LEVEL - PERIOD 4, DATA SET 1
 690    0.00   LIMIT CRITERIA  VALUE - PERIOD 4, DATA SET 1


 700 20000.0     PERIOD DURATION      - PERIOD 5, DATA SET 1
```

```
710   4.372   BEGINING DEPLOY LEVEL - PERIOD 5, DATA SET 1
720   4.372   ENDING  DEPLOY LEVEL - PERIOD 5, DATA SET 1
730   0.00    LIMIT CRITERIA  VALUE - PERIOD 5, DATA SET 1

740  20000.0    ABSOLUTE TIME TO STOP  DEPLOYMENT DATA SET 1


C FIXED SUN ANGLE OPTION (FOR DAY/NITE DETERMINATION)
C--------------------------------------------------
222   0.0    SUN LONGITUDE (DEG) WR/T GREENWICH-EQUATOR (INER)
223   0.0    SUN LATITUDE  (DEG) WR/T GREENWICH-EQUATOR (INER)
224   0.0    SOLAR CONSTANT IN VICINITY OF PLANET (W/M**2)
225   0.0    PLANET ALBEDO (FRAC OF SOLAR RAD BEING REFLECTED)

 0       0.0    END DATA

 ---------------- READ-IN DTOSQ ARRAY HERE - REAL VARIABLES
 0       0.0    END DATA


C****************************************
C BEGIN READ IN DATA FOR EACH TOSS OBJECT
C****************************************

C-----------------------------------
C TOSS OBJECT 2 DATA (INTEGER FOLLOWED BY REAL)
C-----------------------------------
=====> OBJECT 2 <===== READ-IN LTOS2 ARRAY HERE - INTEGER CONSTANTS
 18   1 NUMBER OF ATTACH POINTS ON THIS OBJECT
 19   0 AERO CALCULATION OPTION  (=1 ACTIVATES)
 20   0 CONTROL SYS CALC OPTION  (.GE. 1 ACTIVATES CONTROL)
 21   3 TRANSLATIONAL STATE IC OPTION
 22   3 ATTITUDE STATE     IC OPTION
 29   1 ATTITUDE RATE STATE IC OPTION
 23   0 SPECIAL          IC OPTIONS (NOT CURRENTLY USED)
 24   0 OBJECT-SPECIFIC PARTICLE DYN OPTION (=1 INHIBITS ROT DYN)
 25   0 OBJECT-SPECIFIC EULER ANGLE TYPE FOR INPUT (DEFAULTS TYPE 1)
 26   0 TOSS OBJECT GRAVITY INHIBIT OPTION ON (=-1 INHIBITS GRAV)
 27   0 OBJECT-SPECIFIC GRAVITY GRADIENT BODY TORQUE (=1 ACTIVATES)
 28   0 OBJECT-SPECIFIC LIBRATION ANGLE TYPE FOR INPUT (DEFAULT TO 1)
 30   0 INERTIA VARIATION WR/T MASS (1=NONE, 2=LINEAR, 3=QUADRATIC)

C INTEGERS APPLYING TO CONTROL OPTION = 4
 37   0 OTHER TOSS OBJ (WR/T WHICH INLINE-THRUST/LIB CONTROL IS DONE)
 38   0 ATTACH PT NUMBER ON OTHER TOSS OBJ (WR/T WHICH INLINE-......)
 39   0 ATTACH PT NUMBER ON CONTROLLED OBJ (I.E. THIS OBJECT)
 40   0 TOSS TETHER NUMBER (FOR LENGTH ACTIVATION OF IN-LINES)

 0   0       END OF DATA

 ---------------- READ-IN JTOS2 ARRAY HERE - INTEGER VARIABLES
 0   0       END OF  DATA

 ---------------- READ-IN FTOS2 ARRAY HERE - REAL CONSTANTS

 3        20.9 INITIAL MASS FOR THIS OBJECT (SLUGS)
```

```
4     51. INITIAL IXX FOR OBJ = -INTEGRAL (X*X) (SLUG-FT**2)
5     51.  "  IYY   "    "
6     51.  "  IZZ   "    "

7     0.0 INITIAL IXY FOR OBJ = -INTEGRAL (X*Y) (SLUG-FT**2)
8     0.   "  IXZ   "    "
9     0.   "  IYZ   "    "

10    0.0 INITIAL X CG POS WITHIN THIS OBJECT (FT)
11    0.   "  Y CG    "  "  "
12    0.   "  Z CG    "  "  "

C ATTACH POINT LOCATIONS IN BODY REF FRAME
13    0.0 X COORD FOR ATTACH PT # 1 ON THIS OBJECT (FT)
21    0.0 Y COORD FOR ATTACH PT # 1 ON THIS OBJECT (FT)
29   -1.64 Z COORD FOR ATTACH PT # 1 ON THIS OBJECT (FT)

37    0.0 X COORD OF AERO REF PT FOR TOSS OBJECT (FT)
38    0. Y  "     " " "    "  "
39    0. Z  "     " " "    "  "

C TRANSLATION STATE INITIALIZATION DATA FOR OBJECT (FOR MOST IC OPTIONS)
40    0.0 INITIAL POSITION X COORD (OF OBJECT WR/T RP) (FT)
41    0.   "    "  Y  "    "  "
42    4.92  "   "  Z  "    "  "
43    0.0 INITIAL RATE X COORD (OF OBJECT WR/T RP) (FPS)
44    0.   "  " Y  "    "  "
45    0.   "  " Z  "    "  "

C TRANSLATION STATE INITIALIZATION FOR TRANSL IC OPTION = 5
170   0.0 INITIAL IN-PLANE  LIB     (DEG) CAN ALSO BE ELEV
171   0.   "  OUT-PLANE LIB       "  CAN ALSO BE AZIM

172   0.0 INITIAL IN-PLANE  LIB RATE  (D/S) CAN ALSO BE ELEV
173   0.   "  OUT-PLANE LIB RATE,  "  CAN ALSO BE AZIM

174   0.0 INITIAL RANGE    TO TOSS RP (FT)
175   0.0  "  RNG-RATE TO TOSS RP (F/S)

C ROTATIONAL STATE INITIALIZATION DATA
46    0.0 BODY-X ANG VEL, CAN BE EULER ROLL  RATE (DEG/SEC)
47    0.  BODY-Y  "   " , CAN BE EULER PITCH RATE
48    0.  BODY-Z  "   " , CAN BE EULER YAW   RATE

49    0.0 EULER PITCH ANGLE OF OBJECT (DEG)
50    0.  EULER ROLL  ANGLE
51    0.  EULER YAW   ANGLE

C DATA FOR SIMPLE AERO DRAG ON OBJECT
102   0.0 AERO REF AREA (FT)
103   0.  SIMPLE DRAG COEFF FOR OBJECT


C-----------------------------------------------------------

C DATA APPLYING TO CONTROL SYSTEM "OPTION 2" (SPECIAL FORCES/MOMENTS)
```

```
C-----------------------------------------------------------------
C (ALL THE EFFECTS FOR THIS OPTION ARE ACCUMULATIVE ON THE OBJECT)
 62        0.0 CONSTANT BODY AXIS MOMENT, X-AXIS (FT-LB)
 63        0.   "   "   "   Y-AXIS
 64        0.   "   "   "   Z-AXIS

C SEVEN PERIODS AVAILABLE, ONLY 1ST TWO ARE SHOWN.....
 65        0.0 TIME TO START CONSTANT FORCE, 1ST PERIOD (SEC)
 66        0.  CONSTANT BODY X-AXIS FORCE,   1ST PERIOD (LB)
 67        0.   "   " Y "   "     1ST PERIOD
 68        0.   "   " Z "   "     1ST PERIOD
 69        0.0 TIME TO START CONSTANT FORCE, 2ND PERIOD
 70        0.  CONSTANT BODY X-AXIS FORCE,   2ND PERIOD
 71        0.   "   " Y "   "     2ND PERIOD
 72        0.   "   " Z "   "     2ND PERIOD
C          ....................


C-----------------------------------------------------------------
C DATA APPLYING TO OBJECT CONTROL SYS OPTION 4 (LIB CONT/INLINE THRUST)
C-----------------------------------------------------------------
 176       0.0 DEADBAND FOR LIB CONTROL (DEG)  (=0.0 DE-ACTIVATES)
 177       0.0 IN-PLANE CMD ANGLE (DEG) OF LIB TYPE USED FOR INPUT
 183       0.0 START TIME FOR LIB CONTROL (0.0 MEANS ON IMMEDIATE)
 184       0.0 STOP  TIME FOR LIB CONTROL (0.0 MEANS NO TIME STOP)
C ZERO FOR ACTIVATION TRIGGERS ITEMS BELOW CAN MEAN, IGNORE-OR-ACTIVATE
C SEE TOSS REF MANUAL FOR LOGICAL INTERACTION OF THESE ITEMS
 178       0.0 INLINE THRUSTER LEVEL (LB) (.EQ. 0.0  DE-ACTIVATES)
 179       0.0 TIME TO   ACTIVATE (.EQ. 0. MAY MEAN TURN ON)
 180       0.0 TIME TO DE-ACTIVATE (.EQ. 0. MAY MEAN NO TURN OFF)
 181       0.0 DEP LENGTH ACTIVATE (L .LT. = ON, TIMES PREEMPT)

  0        0.0    END DATA

 ---------------- READ-IN DTOS2 ARRAY HERE - REAL VARIABLES
  0        0.0    END DATA




$THR089  RDB FAMILY NAME FOR SERIES (W/ INITIAL SEQ NUMBER XX)
  NOMINAL TMTC RUNSTREAM FOR INITIAL AIRSEDS DEPLOY STUDIES
  PERTINENT DATA PLUS RELATED OPTIONS ONLY INCLUDED

C=================================================================
C NOTE: THIS PARAMETER STUDY INPUT SKELETON HAS MANY GTOSS/TOSS
C EXECUTION OPTIONS/FEATURES NOT PRESENT SO AS TO SIMPLIFY IT. THE
C REMOVED OPTIONS WOULD NOT PERTAIN TO THIS BASIC DEPLOYMENT SIZING
C STUDY FOR AIR-SEDS. WHILE INITIALLY SET UP FOR A MASSLESS TETHER,
C THE DATA PLACE-HOLDERS FOR SPECIFYING AND ACTIVATING A FINITE

C TETHER HAVE BEEN RETAINED SO THAT THIS SAME INPUT STREAM CAN BE
```

```
C EASILY MODIFIED TO ACTIVATE A FINITE TETHER.
C===========================================================================


C YOU MAY INSERT COMMENT CARDS (AS THESE INDEED ARE), OR TOTALLY
C BLANK CARDS AT YOUR DISCRETION. AS SHOWN, THIS DECK (WITH ALL
C THE COMMENTS AND BLANK CARDS), CAN BE READ, AND IS LOGICALLY
C ACCEPTABLE TO GTOSS, EVEN THOUGH THE DATA IS NOT MEANINGFUL.
C THIS DECK IS TO SERVE AS A TEMPLATE AND CHECK LIST FOR BUILDING
C GTOSS/TOSS INPUT DECKS, AND YOU WOULD USUALLY DELETE A GREAT
C DEAL OF THE CARDS (INCLUDING THESE COMMENTS).

C-------------------------------------------
C GTOSS  AND  REF PT DATA IS ENTERED IN I3,F15 FORMAT
C (COMMENTS IN COL 19 OR GREATER ARE IGNORED AND LOST)
C-------------------------------------------
C THIS MUST PRECEDE ANY DATA ITEMS YOU WANT TO CHANGE FOR A LATE START
397  0.0        =1.0 TO REQUEST A SNAP SHOT LATE START INITIALIZATION

C GENERAL REF PT PARAMETERS
  1  0.10       INTEGRATION STEPSIZE INITIALLY USED BY GTOSS
  2 10800.       MAX TIME FOR RUN (SEC)
  3  500.       N, THE RDB SOLN OUTPUT INTERVAL = N * STEPSIZE
 16  0.0        RUN NUMBER
 94 -1.0        INHIBIT DOT OUTPUT TO CRT WITH -1.0

C CRT DISPLAY CONTROL (DEFAULT 0. = NO OUTPUT, IE. BATCH MODE OPERATION)
C (NOTE: CRT OUTPUT CAN BE A SLOW PROCESS RELATIVE TO THE SOLUTION OF
C  NON COMPUTE-BOUND TETHER SIMULATIONS. THAT IS, WHILE THE SOLN SEEMS
C  TO BE POURING FORTH AT BREAK-NECK SPEED, IT MAY PROCEED MUCH FASTER
C  IF CRT OUTPUT IS NOT ACTIVE)
 17  1.         = M CAUSES OUTPUT EVERY M SOLN RDB INTERVALS (ITEM 3)

C INTEGRATION STEPSIZE STAGING DATA (IN THE CURRENT BERSION OF GTOSS,
C THIS CAN BE PARTICULARLY USEFUL FOR DEPLOYMENTS OR RETRIEVALS WITH A
C FACTOR OF 2 (OR GREATER) TETHER LENGTH CHANGE
  5  0.0        ABSOLUTE TIME FOR  1ST INTEGRATION STEPSIZE CHANGE
  9  0.0000       STEPSIZE INVOKED AT 1ST CHANGE
 71  0.         RDB OUTPUT INTERVAL 1ST CHANGE

  6  0.0        ABSOLUTE TIME FOR  2ND INTEGRATION STEPSIZE CHANGE
 10  0.0000       STEPSIZE INVOKED AT 2ND CHANGE
 72  0.         RDB OUTPUT INTERVAL 2ND CHANGE

  7  0.0        ABSOLUTE TIME FOR  3RD INTEGRATION STEPSIZE CHANGE
 11  0.0000       STEPSIZE INVOKED AT 3RD CHANGE
 73  0.         RDB OUTPUT INTERVAL 3RD CHANGE

  8  0.0        ABSOLUTE TIME FOR  LAST INTEGRATION STEPSIZE CHANGE
 12  0.0000       STEPSIZE INVOKED AT LAST CHANGE
 74  0.         RDB OUTPUT INTERVAL LAST CHANGE

C REFERENCE DATE (WHICH CORRESPONDS TO ZERO SIMULATION TIME FOR THE RUN)
287  0.0        YEAR, FOR EXAMPLE 1998.0

288  0.0        MONTH, FOR EXAMPLE   5.0 (BETWEEN 1.0 AND 12.0)
```

```
289 0.0     DAY,  FOR EXAMPLE  17.0 (DEPENDS ON THE MONTH)
290 0.0     HOUR, FOR EXAMPLE   8.0 (BETWEEN 0.0 AND 24.0)
291 0.0     MIN,  FOR EXAMPLE  42.0 (BETWEEN 0.0 AND 60.0)
292 0.0     SEC,  FOR EXAMPLE  23.0 (BETWEEN 0.0 AND 60.0)

C LATE START SNAP SHOT INITIALIZATION CONTROL
398 0.0     =1.0 IF SNAP SHOTS DATA DUMPS ARE TO BE TAKEN
387 0.0     TIME FOR  1ST SNAP SHOT TO BE TAKEN
388 0.0     TIME FOR  2ND SNAP SHOT TO BE TAKEN
396 0.0     TIME FOR 10TH SNAP SHOT TO BE TAKEN

C QUICK LOOK PAGE FORMAT CONTROL
117 0.      SELECT QUICK LOOK PAGE FORMAT
381 2.0     CHOOSE TOSS OBJ FOR QUICK LOOK (IF APPROP)
382 0.      "    "
421 1.0     CHOOSE TOSS TETHER FOR QUICK LOOK (IF APPROP)
442 0.0     BEAD NUMBER TO DISPLAY FOR QUICK LOOK (IF APPROP)
443 0.      "   "
444 0.      "   "
435 0.0     CHOOSE FINITE SOLN FOR QUICK LOOK (IF APPROP)

C REF PT/GTOSS LOCAL EXECUTION CONTROL DATA
 92 0.      = 1. TO ACTIVATE GRAV GRAD TORQUE ON REF PT OBJECT
110 0.      = 1. TO INHIBIT READING TOSS DATA AND EXECUTING TOSS
111 0.      = 1. TO INHIBIT ROTATIONAL DYNAMICS (PARTICLE OPT)
112 0.      REF PT SPECIFIC EULER ANG TYPE FOR INPUT (DEFAULTS 1)
113 2.      NUMBER OF LAST TOSS OBJECT BEING SIMULATED
114 1.      NUMBER OF ATTACH PTS ON THE REF POINT


C-----------------------------------------------
C REF PT/TOSS GLOBAL PLANETARY ENVIRONMENT MODEL OPTIONS
C-----------------------------------------------
C REF PT/TOSS GLOBAL PLANETARY ENVIR MODEL SELECTION/FIDELITY OPTIONS
481 0.      GRAVITY MODEL (0.0=SPHERICAL; 1.0=SIMPLE OBLATE,ETC)
482 0.      ATMOSPHERIC DENSITY MODEL (0.0=EARTH ARDC 1976)
483 0.      MAGNETIC FIELD (0.0=EARTH TILTED/SHIFTED DIPOLE)
484 0.      PLANET GLOBE SHAPE (0.0 =SPHERICAL EARTH)
485 0.      ATMOSPHERIC WINDS MODEL (0.0=ROTATING EARTH,ETC)
488 0.      ATMOSPHERIC PLASMA MODEL (0.0=IRI95)
486 0.      INER. FRAME MODEL (0.0=FOR INITIALLY-EARTH-ALIGNED)
487 0.      GLOBAL EULER ANGLE INPUT TYPE (NON-PREEMPTIVE IF =0)
490 0.      CONST PLANET RADIUS (DEFAULT = GEOD VALUE AT T=0)

C REF PT/TOSS GLOBAL PLANETARY ENVIRONMENT MODEL EVALUATION FREQUENCY
561 0.      =1. PREEMPTS ALL ENVIRON'S TO EVAL EVERY RP TIME STEP

C SPECIFY ACTUAL LAPSED TIME BETWEEN ENVIRONMENT EVALUATIONS
C (TIMES LESS THAN REF PT DELTA-T WILL FORCE EVAL EVERY REF PT STEP)
563 0.000   TIME BETWEEN ATMOS EVAL  (=0. DEFAULTS TO EVERY STEP)
564 0.000   TIME BETWEEN MAGNET EVAL (=0. DEFAULTS TO EVERY STEP)
565 0.000   TIME BETWEEN WIND EVAL   (=0. DEFAULTS TO EVERY STEP)
566 0.000   TIME BETWEEN PLASMA EVAL (=0. DEFAULTS TO EVERY STEP)
```

```
C BASIC REF PT GEOMETRY
C-----------------
20 49.68    REFERENCE PT MASS (SLUGS)
21 121.     IXX = -INTEGRAL (X*X) (SLUG-FT**2)
22 121.     IYY
23 121.     IZZ

24 0.0      XCG: REF PT CENTER OF MASS LOC (FT)
25 0.       YCG
26 0.       ZCG

27 0.0      IXY = -INTEGRAL (X*Y) (SLUG-FT**2)
28 0.       IXZ
29 0.       IYZ

30 0.       INER VARIATION W/MASS OPTION (1.NONE,2.LINEAR,3.QUAD)


C REF PT TRANSLATION STATE INITIALIZATION
C-----------------------------------
C =0. WR/T TOPO FRAME;  =1. WR/T INER FRAME;   =2 CIRCULAR ORBIT
100 2.      SELECT REF PT TRANSLATION STATE IC OPTION

C FOR TRANSLATION IC OPTION = 1.0
81 0.0      XIO REF PT POSITION (FT) [INER FRAME]  (OPT 1)
82 0.       YIO "     "                 (OPT 1)
83 0.       ZIO "     "                 (OPT 1)
84 0.0      XIDO REF PT RATE (FT/SEC) [INER FRAME] (OPT 1)
85 0.       YIDO "     "                (OPT 1)
86 0.       ZIDO "     "                (OPT 1)

C FOR TRANSLATION IC OPTION = 0.0 AND 2.0
101 1312336.   REF PT  ALT (FT)         (OPT 0 AND OPT 2)
102 -80.6      REF PT  TOPO LONGITUDE (DEG)  (OPT 0 AND OPT 2)
103  28.5      REF PT  TOPO LATITUDE  (DEG)  (OPT 0 AND OPT 2)
107 0.0        VXTO INER VEL TOPO COMP(FPS)  (OPT 0)
108 0.         VYTO  "      "        (OPT 0)
109 0.         VZTO  "      "        (OPT 0)
80 0.0         CIRC VEL AZIM WR/T TOPO FRAME (DEG,EAST=0) (OPT=2)


C REF PT BODY ATTITUDE STATE INITIALIZATION
C------------------------------------
C =0. EULER TO ORBIT FRM;  =1. EULER TO TOPO FRM;  =2.EULER TO INER FRM
90 0.0      SELECT BODY ATTITIDE OPTION

104 -30.    EULER PITCH ANGLE (DEG)
105 0.      EULER ROLL  ANGLE
106 0.      EULER YAW   ANGLE

C REF PT BODY RATE STATE INITIALIZATION
C---------------------------------
C =0.BODY COMP;  =1.ORB RATE;  =2.EUL RATE/INER FRM, =3.EUL RATE/ORB FRM
93 1.0      SELECT BODY RATE OPTION
```

```
87 0.0     BODY-X ANG VEL, CAN BE EULER ROLL  RATE (DEG/SEC)
88 0.      BODY-Y ANG VEL, CAN BE EULER PITCH RATE
89 0.      BODY-Z ANG VEL, CAN BE EULER YAW   RATE


C ATTACH PT COORDS (1 SHOWN, 8 POSSIBLE)
134 0.0    PXBT1: REF PT (FT) X COORD OF ATTACH PT
142 0.0    PYBT1: REF PT (FT) Y COORD OF ATTACH PT
150 1.64   PZBT1: REF PT (FT) Z COORD OF ATTACH PT

C SIMPLE REF POINT AERO DRAG OPT DATA
455 0.     =1. TO ACTIVATE SIMPLE DRAG OPTION ON REF PT
456 0.0    AERO REFERENCE AREA FOR REF PT OBJECT (FT**2)
457 0.0    SIMPLE DRAG COEFFICIENT (NON-DIMENS)


C--------------------------------------------
C ACTIVATE ATTITUDE CONTROL ALGORITHMS ON HOST OBJECT
C--------------------------------------------
C (NOTE SPECIFYING GREATER THAN 2. RESULTS IN ADDITIONAL OPTIONS)
807 1.     0.=OFF, 1.=WR/T ORB FRM, 2.=WR/T INER FRM, 3.=ETC

C RATE AND ERROR FEEDBACK GAINS FOR OPTION ACTIVATED BY ITEM 807
808  5.0    RATE FEEDBACK GAIN (X-AXIS)
809  5.0    RATE FEEDBACK GAIN (Y-AXIS)
810  5.0    RATE FEEDBACK GAIN (Z-AXIS)
811  10.0   ATT ERR FEEDBACK GAIN (X-AXIS)
812  10.0   ATT ERR FEEDBACK GAIN (Y-AXIS)
813  10.0   ATT ERR FEEDBACK GAIN (Z-AXIS)


C--------------------------------
C FLAG TO ACTIVATE ARBITRARY BODY FORCES
C--------------------------------
817 1.     0.=OFF, 1.=TABLE LOOKUP, 2.=ALGORITHM-A, 3.=ALGOR-B

C TABLE DATA DEFINING ARBITRARY HOST X-BODY AXIS FORCES VS TIME
819 0.0    MAX TIME COVERED BY THIS TABLE
820 0.0    VALUE OF X-BODY FORCE AT MAX TIME

821 0.0    FIRST TIME VALUE
822 0.0    FIRST VALUE OF X BODY FORCE
C ...........ETC...........UP TO 20 TIME/FORCE POINT SETS ALLOWED
859 0.0    TWENTIETH TIME VALUE
860 0.0    TWENTIETH VALUE OF X BODY FORCE


C TABLE DATA DEFINING ARBITRARY HOST Y-BODY AXIS FORCES VS TIME
862 0.0    MAX TIME COVERED BY THIS TABLE
863 0.0    VALUE OF Y-BODY FORCE AT MAX TIME

864 0.0    FIRST TIME VALUE
865 0.0    FIRST VALUE OF Y BODY FORCE
C ...........ETC...........UP TO 20 TIME/FORCE POINT SETS ALLOWED

902 0.0    TWENTIETH TIME VALUE
```

903 0.0 TWENTIETH VALUE OF Y BODY FORCE

C TABLE DATA DEFINING ARBITRARY HOST Z-BODY AXIS FORCES VS TIME
905 20000. MAX TIME COVERED BY THIS TABLE
906 0.0000 VALUE OF Z-BODY FORCE AT MAX TIME

907 0.0 1ST TIME VALUE
908 -0.0899 1ST VALUE OF Z BODY FORCE
909 2000. 2ND TIME VALUE
910 -0.0899 2ND VALUE OF Z BODY FORCE
911 2000.1 3RD TIME VALUE
912 0.0000 3RD VALUE OF Z BODY FORCE
913 20000. 4TH TIME VALUE
914 0.0000 4TH VALUE OF Z BODY FORCE
915 10000. 5TH TIME VALUE
916 0.0000 5TH VALUE OF Z BODY FORCE
917 10000. 6TH TIME VALUE
918 0.0000 6TH VALUE OF Z BODY FORCE
919 10000. 7TH TIME VALUE
920 0.0000 7TH VALUE OF Z BODY FORCE

C ....ETC...........UP TO 20 TIME/FORCE POINT SETS ALLOWED
945 0.0 TWENTIETH TIME VALUE
946 0.0 TWENTIETH VALUE OF Z BODY FORCE

C GTOSS DATA TERMINATOR CARD IS MANDATORY, DEFINED AS 0 IN I3 FIELD
0 END OF GTOSS AND REF POINT DATA

C***
C***
C BEGIN READ IN OF TOSS GENERAL INTEGER DATA
C***
C***
C A MANDATORY ALPHANUMERIC CARD PRECEDES EACH TOSS DATA CATEGORY
C A MANDATORY DATA TERMINATOR CARD FOLLOWS EACH TOSS DATA CATEGORY

C---
C TOSS DATA ENTERED AS: INTEGER: I4,I6,A50 REAL:I4,E16,A50
C (THE A50 FIELD ALLOWS RE-DISPLAY OF COMMENT DATA IF DESIRED)
C
C AS IT PERTAINS TO FINITE SOLNS, ALL DATA SHOWN IS FOR SOLN 1.
C TO FIND INPUT DATA ITEM NUMBER FOR SOLN 2, ALL SHOWN INPUT ID
C NUMBERS ARE INCREMENTED BY 1. (ETC FOR SOLN 3). SIMILARLY FOR
C TOSS TETHERS, EXCEPT TYPICALLY ITEM NUMBERS ARE SHOWN FOR 3.
C---

======> GENERAL <====== READ-IN LTOSQ ARRAY HERE - INTEGER CONSTANTS

C GENERAL ACTIVATION OPTIONS APPLYING TO ALL OF TOSS
C---------------------------------------
22 0 ACTIVATE WRITE-OUT OF TOSS INPUT DATA STREAM
23 0 ACTIVATE PARTICLE DYNAMICS PREEMPTS ALL OBJECTS

126 0 ACTIVATE GRAVITY GRAD BODY TORQUE PREEMPTS ALL OBJECTS

```
128    0 ACTIVATE DYNAMICS VERIFICATION CALCS (=1 ACTIVATES)
197    0 ACTIVATE DEPLOYING TETHER MASS FLOW EFFECTS
155    0 INITIALIZE PASSIVE DAMPER MODE (0=ACTIVE, 1=QUIESCENT)

C GENERAL SPECIFICATION OPTIONS APPLYING TO ALL OF TOSS
C---------------------------------------------
125    0 SPECIFY EULER ANGLE TYPE FOR DATA INPUT  PREEMPTS ALL OBJECTS
127    0 SPECIFY LIBR. ANGLE TYPE FOR DATA INPUT  PREEMPTS ALL OBJECTS


C DATA APPLYING TO ALL TETHERS
C-----------------------
 24    1 SPECIFY TOTAL NUMBER OF TOSS TETHERS (FINITE+MASSLESS)

194    0 ACTIVATE GRAVITATIONAL  FORCES FOR ALL TOSS TETHERS
195   -1 ACTIVATE AERODYNAMIC    FORCES FOR ALL TOSS TETHERS
196   -1 ACTIVATE ELECTRODYNAMIC FORCES FOR ALL TOSS TETHERS

C DEFINITION OF TOSS TETHER CONNECTIVITY
 25    1 OBJECT TO WHICH "X" END ATTACHES-TOSS TETHER #1
 50    1 ATT PT FOR "X" END OF TOSS TETHER #1

 75    2 OBJECT TO WHICH "Y" END ATTACHES-TOSS TETHER #1
100    1 ATT PT FOR "Y" END OF TOSS TETHER #1


C ASSOCIATE FINITE SOLN NUMBERS WITH TOSS TETHER NUMBERS
130    0 ASSIGN A FINITE SOLN NUMBER TO TOSS TETHER #1
284    0 SET UP STABLE GRAVITY GRADIENT TENSION IN TOSS TETHER #1
259    1 ASSIGN A DEPLOYMENT SCENARIO DATA SET TO TOSS TETHER #1
234    0 ASSIGN AN ELEC POWER SCENARIO DATA SET TO TOSS TETHER #1


C---------------------------------------
C THERMAL SIMULATION CONTROL FOR ALL TOSS TETHERS
C---------------------------------------
177    0 DIRECT SOLAR RADIATION HEATING        (=1 ACTIVATES)
178    0 PLANET ALBEDO HEATING           (=1 ACTIVATES)
179    0 PLANET BLACK BODY RADIATION HEATING    (=1 ACTIVATES)
180    0 AERODYNAMIC HEATING             (=1 ACTIVATES)
181    0 ELECTRICAL RESISTIVE HEATING         (=1 ACTIVATES)

182    0 HEAT RADIATION FROM A TETHER          (=1 ALLOWS)
183    0 HEAT CONDUCTION ALONG A TETHER          (=1 ALLOWS)
184    0 THERMAL EXPANSION CHANGES TO TETHER LENGTH  (=1 ALLOWS)


C DATA APPLYING TO FINITE SOLUTIONS (1ST IN SERIES OF 9 POSSIBLE SHOWN)
C------------------------------------------------------------
129    0 NFINIT, LARGEST FINITE SOLN NUMBER ALLOWED TO BE ACTIVE (ALL)
198    0 SPECIFY SOLN TYPE (1=STD, 2=HST, 3=HST_UN-SYM)  FINITE SOLN 1
207    0 NUMBER OF BEADS ASSIGNED TO                FINITE SOLN 1
185    0 WAVE SHAPE-VELOCITY IC OPTION               FINITE SOLN 1
343    0 SKIP ROPE IC ACTIVATOR (1 = CW, 2 = CCW)      FINITE SOLN 1

352    0 SKIP ROPE MODE SHAPE   (1, 2, 3, 4, ...)       FINITE SOLN 1
```

```
C DO NOT USE THESE NEXT TWO ITEMS UNLESS YOU KNOW WHAT YOU ARE DOING
164    0 IF=-1, INHIBITS STOPS ON HST CONSTRAINT EXCEEDANCE  ALL SOLNS
165    0 PREEPTS HST SOLN TENS REF END (X-END =1, Y =2)  FINITE SOLN 1

 0   0       END OF DATA

 ------------------ READ-IN JTOSQ ARRAY HERE - INTEGER VARIABLES
 0   0       END OF DATA


C*****************************************
C*****************************************
C BEGIN READ IN OF TOSS GENERAL REAL DATA
C*****************************************
C*****************************************

 ---------------- READ-IN FTOSQ ARRAY HERE - REAL CONSTANTS

C**************************************************************
C NOMINAL, BASIC PHYSICAL DATA APPLYING TO ALL UNIFORM TETHERS
C**************************************************************
C TOSS TETHER INITIAL DEPLOYED, UNDEFORMED LENGTH
 50      2.033 UN-STRETCHED LENGTH (FT)........TOSS TETHER #1

C LINEAL DENSITY FOR TOSS TETHERS (ASSUMED UNIFORM)
1223      15.5 LINEAL DENSITY (LBM/1000FT).....TOSS TETHER #1

C YOUNGS MODULUS FOR TOSS TETHERS (ASSUMED UNIFORM)
1248     5.0E+6 YOUNGS MODULUS (PSI)............TOSS TETHER #1

C EFFECTIVE ELASTIC DIAMETER (ASSUMED UNIFORM)
1273     0.1405 ELASTIC DIAMETER (IN)...........TOSS TETHER #1

C STRESS-PROPORTIONAL STRAIN RATE DISSIPATION COEFF (ASSUMED UNIFORM)
1298      0.010 BETA, DISSIPATION COEFF (SEC)...TOSS TETHER #1

C EFFECTIVE AERODYNAMIC DIAMETER (ASSUMED UNIFORM)
1323       0.0 AERODYNAMIC DIAMETER (IN).......TOSS TETHER #1


C NON-LINEAR STIFFNESS MODEL STRAIN BIAS CONSTANT (ASSUMED UNIFORM)
C-------------------------------------------------------
1348       0.0 NON-LINEAR STRAIN BIAS (ND).....TOSS TETHER #1

C NON-LINEAR STIFFNESS MODEL EXPONENT, >0 TO ACTIVATE (ASSUMED UNIFORM)
1373       0.0 NON-LINEAR EXPONENT (ND)........TOSS TETHER #1


C VALUE TO INITIALIZE AMPERAGE IN TOSS TETHERS (EASY CONST CURRENT OPT)
C-----------------------------------------------------------
 181    0.0     INITIAL/CONST UNIFORM CURRENT (AMPS) TOSS TETHER #1


C MULTIPLIERS FOR DEPLOY AND POWER GENERATION SCENARIOS
```

```
C--------------------------------------------------
C MULTIPLIER USED BY TOSS TETHER FOR ANY ASSIGNED POWER GEN DATA SET
 335    0.0    PWR MULTIPLIER (DEFAULTS TO 1.0) - TOSS TETHER #1

C MULTIPLIER USED BY TOSS TETHER FOR ANY ASSIGNED DEPLOYMENT DATA SET
 360    0.0    MULTIPLIER (DEFAULTS TO 1.0) - TOSS TETHER #1


C****************************************
C DATA APPLYING ONLY TO MASSLESS TETHERS
C****************************************
C ADDITIONAL ELASTIC TETHER LENGTH FROM ANCHOR PT TO TOSS ATTACH PT
C AFFECTS ONLY SPRING RATE CALCULATION (AND IS ALWAYS IN EFFECT)
C--------------------------------------------------------
1137        0.0  ANCHOR LENGTH BIAS (FT) - MASSLESS TOSS TETHER #1

C USE CONSTANT SPRING-RATE INSIDE THIS: ANCHOR + DEPLOYED LENGTH
C SPRING-RATE CLAMPS AT A VALUE EQUIVALENT TO THIS PROXIMITY LENGTH
1162      1000.0  PROXIMITY LENGTH (FT)  - MASSLESS TOSS TETHER #1

C ALTERNATIVE MASSLESS TETHER STIFFNESS AND DISSIPATION SPECIFICATION
C--------------------------------------------------------------
C (THESE VALUES ARE BASED ON INITIAL LENGTHS, IE. ITEM 50,..., ABOVE)
C THIS IS USED ONLY IF TOSS TETHER YOUNGS MODULUS IS ZERO
 25         0.0  SPRING RATE (LB/FT) - MASSLESS TOSS TETHER #1

C THIS IS USED ONLY IF TOSS TETHER DISSIPATION COEFFICIENT IS ZERO
 75         0.0  END-END DAMPNG COEF (LB/FPS) MASSLESS TOSS TETHER #1



C THERMAL PROPERTIES FOR FINITE TETHERS
C-------------------------------
1506    0.0     PROPERTIES BASELINE TEMPERATURE        FINITE SOLN 1
1398    0.0     TETH HEAT CONDUCTIVITY /UNIT AREA      FINITE SOLN 1
1407    0.0     THERMAL LINEAR EXPANSION COEFF (/K)    FINITE SOLN 1
1416    0.0     ABSORPTIVITY IN SOLAR SPECTRUM         FINITE SOLN 1
1425    0.0     EMISSIVITY IN RADIANT SPECTRUM         FINITE SOLN 1
1434    0.0     SPECIFIC HEAT/UNIT MASS J/(KG-K)       FINITE SOLN 1
1443    0.0     SLOPE OF SPECIFIC HEAT WR/T TEMP       FINITE SOLN 1
1470    0.0     CONDUCTOR RESISTIVITY /UNIT LENGTH     FINITE SOLN 1
1479    0.0     SLOPE OF RESISTIVITY WR/T TEMP         FINITE SOLN 1

C THERMAL EFFECTS IMPLIED BELOW ARE NOT CURRENTLY IMPLEMENTED
1452    0.0     SLOPE OF YOUNGS MODULUS WR/T TEMP      FINITE SOLN 1
1461    0.0     SLOPE OF DAMPING WR/T TEMP             FINITE SOLN 1
1488    0.0     HEAT CONDUCTIVITY AT X-END TETH AP     FINITE SOLN 1
1497    0.0     HEAT CONDUCTIVITY AT Y-END TETH AP     FINITE SOLN 1


C FRACTIONAL SEG LENGTH ERROR AT WHICH HST SOLN WILL NOTIFY USER
C----------------------------------------------------
C (GENERALLY NOT USED UNLESS YOU KNOW WHAT YOU ARE DOING)
207     0.0     FRACTONAL ERROR LIMIT (DEFAULT = 0.01) FINITE SOLN 1
```

```
C HST CONSTRAINT STABILIZATION FEEDBACK GAINS FOR SEG/SEG-RATE ERRORS
C (GENERALLY NOT USED UNLESS YOU KNOW WHAT YOU ARE DOING)
216   0.0    ERROR FDBK GAIN MULTIPLIER (DEFAULT =1.0)  ALL SOLNS
217   0.0    RATE  FDBK GAIN MULTIPLIER (DEFAULT =1.0)  ALL SOLNS


C-----------------------------------
C DATA FOR DEPLOYMENT DATA SETS 1 THRU 10
C-----------------------------------
C (SEE REF MANUAL FOR DEFINITIONS OF SCENARIO TYPES)

C DEFINE DEPLOYMENT DATA SET NUMBER 1
C-------------------------------
520   1.0    DEPLOYMENT SCENARIO TYPE   FOR DATA SET 1
530   0.0    ABSOLUTE TIME TO START DEPLOY  DATA SET 1

540  500.0     PERIOD DURATION       - PERIOD 1, DATA SET 1
550    0.000   BEGINING DEPLOY LEVEL - PERIOD 1, DATA SET 1
560    1.457   ENDING   DEPLOY LEVEL - PERIOD 1, DATA SET 1
570    0.00    LIMIT CRITERIA  VALUE - PERIOD 1, DATA SET 1

580  500.0     PERIOD DURATION       - PERIOD 2, DATA SET 1
590    1.457   BEGINING DEPLOY LEVEL - PERIOD 2, DATA SET 1
600    2.914   ENDING   DEPLOY LEVEL - PERIOD 2, DATA SET 1
610    0.00    LIMIT CRITERIA  VALUE - PERIOD 2, DATA SET 1

620  500.0     PERIOD DURATION       - PERIOD 3, DATA SET 1
630    2.914   BEGINING DEPLOY LEVEL - PERIOD 3, DATA SET 1
640    4.372   ENDING   DEPLOY LEVEL - PERIOD 3, DATA SET 1
650    0.00    LIMIT CRITERIA  VALUE - PERIOD 3, DATA SET 1

660  500.0     PERIOD DURATION       - PERIOD 4, DATA SET 1
670    4.372   BEGINING DEPLOY LEVEL - PERIOD 4, DATA SET 1
680    5.829   ENDING   DEPLOY LEVEL - PERIOD 4, DATA SET 1
690    0.00    LIMIT CRITERIA  VALUE - PERIOD 4, DATA SET 1

700 20000.0    PERIOD DURATION       - PERIOD 5, DATA SET 1
710    5.829   BEGINING DEPLOY LEVEL - PERIOD 5, DATA SET 1
720    5.829   ENDING   DEPLOY LEVEL - PERIOD 5, DATA SET 1
730    0.00    LIMIT CRITERIA  VALUE - PERIOD 5, DATA SET 1

740  20000.0    ABSOLUTE TIME TO STOP  DEPLOYMENT DATA SET 1


C FIXED SUN ANGLE OPTION (FOR DAY/NITE DETERMINATION)
C---------------------------------------------
222   0.0    SUN LONGITUDE (DEG) WR/T GREENWICH-EQUATOR (INER)
223   0.0    SUN LATITUDE  (DEG) WR/T GREENWICH-EQUATOR (INER)
224   0.0    SOLAR CONSTANT IN VICINITY OF PLANET (W/M**2)
225   0.0    PLANET ALBEDO (FRAC OF SOLAR RAD BEING REFLECTED)

 0        0.0    END DATA

 ---------------- READ-IN DTOSQ ARRAY HERE - REAL VARIABLES

 0        0.0    END DATA
```

```
C*****************************************
C BEGIN READ IN DATA FOR EACH TOSS OBJECT
C*****************************************

C-----------------------------------------
C TOSS OBJECT 2 DATA (INTEGER FOLLOWED BY REAL)
C-----------------------------------------
=====> OBJECT 2 <===== READ-IN LTOS2 ARRAY HERE - INTEGER CONSTANTS
 18   1 NUMBER OF ATTACH POINTS ON THIS OBJECT
 19   0 AERO CALCULATION OPTION  (=1 ACTIVATES)
 20   0 CONTROL SYS CALC OPTION  (.GE. 1 ACTIVATES CONTROL)
 21   3 TRANSLATIONAL STATE IC OPTION
 22   3 ATTITUDE STATE     IC OPTION
 29   1 ATTITUDE RATE STATE IC OPTION
 23   0 SPECIAL          IC OPTIONS (NOT CURRENTLY USED)
 24   0 OBJECT-SPECIFIC PARTICLE DYN OPTION (=1 INHIBITS ROT DYN)
 25   0 OBJECT-SPECIFIC EULER ANGLE TYPE FOR INPUT (DEFAULTS TYPE 1)
 26   0 TOSS OBJECT GRAVITY INHIBIT OPTION ON (=-1 INHIBITS GRAV)
 27   0 OBJECT-SPECIFIC GRAVITY GRADIENT BODY TORQUE (=1 ACTIVATES)
 28   0 OBJECT-SPECIFIC LIBRATION ANGLE TYPE FOR INPUT (DEFAULT TO 1)
 30   0 INERTIA VARIATION WR/T MASS (1=NONE, 2=LINEAR, 3=QUADRATIC)

C INTEGERS APPLYING TO CONTROL OPTION = 4
 37   0 OTHER TOSS OBJ (WR/T WHICH INLINE-THRUST/LIB CONTROL IS DONE)
 38   0 ATTACH PT NUMBER ON OTHER TOSS OBJ (WR/T WHICH INLINE-......)
 39   0 ATTACH PT NUMBER ON CONTROLLED OBJ (I.E. THIS OBJECT)
 40   0 TOSS TETHER NUMBER (FOR LENGTH ACTIVATION OF IN-LINES)

  0   0       END OF DATA

  --------------- READ-IN JTOS2 ARRAY HERE - INTEGER VARIABLES
  0   0       END OF  DATA

  --------------- READ-IN FTOS2 ARRAY HERE - REAL CONSTANTS
  3        20.9 INITIAL MASS FOR THIS OBJECT (SLUGS)
  4        51.  INITIAL IXX FOR OBJ = -INTEGRAL (X*X) (SLUG-FT**2)
  5        51.   "  IYY    "     "
  6        51.   "  IZZ    "     "

  7         0.0 INITIAL IXY FOR OBJ = -INTEGRAL (X*Y) (SLUG-FT**2)
  8         0.   "  IXZ    "     "
  9         0.   "  IYZ    "     "

 10          0.0 INITIAL X CG POS WITHIN THIS OBJECT (FT)
 11          0.   "  Y CG    "   "   "
 12          0.   "  Z CG    "   "   "

C ATTACH POINT LOCATIONS IN BODY REF FRAME
 13          0.0 X COORD FOR ATTACH PT # 1 ON THIS OBJECT (FT)
 21          0.0 Y COORD FOR ATTACH PT # 1 ON THIS OBJECT (FT)
 29         -1.64 Z COORD FOR ATTACH PT # 1 ON THIS OBJECT (FT)

 37          0.0 X COORD OF AERO REF PT FOR TOSS OBJECT (FT)
```

```
38        0. Y  "     " " "    "  "
39        0. Z  "     " " "    "  "

C TRANSLATION STATE INITIALIZATION DATA FOR OBJECT (FOR MOST IC OPTIONS)
40        0.0 INITIAL POSITION X COORD (OF OBJECT WR/T RP) (FT)
41        0.   "    "   Y  "     "   "
42        4.92  "   "   Z  "     "   "
43        0.0 INITIAL RATE X COORD (OF OBJECT WR/T RP) (FPS)
44        0.   "   " Y  "     "  "
45        0.   "   " Z  "     "  "

C TRANSLATION STATE INITIALIZATION FOR TRANSL IC OPTION = 5
170       0.0 INITIAL IN-PLANE  LIB     (DEG) CAN ALSO BE ELEV
171       0.  "  OUT-PLANE LIB       "  CAN ALSO BE AZIM

172       0.0 INITIAL IN-PLANE  LIB RATE  (D/S) CAN ALSO BE ELEV
173       0.  "   OUT-PLANE LIB RATE,  "  CAN ALSO BE AZIM

174       0.0 INITIAL RANGE    TO TOSS RP (FT)
175       0.0  "   RNG-RATE TO TOSS RP (F/S)

C ROTATIONAL STATE INITIALIZATION DATA
46        0.0 BODY-X ANG VEL, CAN BE EULER ROLL  RATE (DEG/SEC)
47        0.  BODY-Y  "  " , CAN BE EULER PITCH RATE
48        0.  BODY-Z  "  " , CAN BE EULER YAW   RATE

49        0.0 EULER PITCH ANGLE OF OBJECT (DEG)
50        0.  EULER ROLL  ANGLE
51        0.  EULER YAW   ANGLE

C DATA FOR SIMPLE AERO DRAG ON OBJECT
102       0.0 AERO REF AREA (FT)
103       0.  SIMPLE DRAG COEFF FOR OBJECT


C-------------------------------------------------------------
C DATA APPLYING TO CONTROL SYSTEM "OPTION 2" (SPECIAL FORCES/MOMENTS)
C-------------------------------------------------------------
C (ALL THE EFFECTS FOR THIS OPTION ARE ACCUMULATIVE ON THE OBJECT)
62        0.0 CONSTANT BODY AXIS MOMENT, X-AXIS (FT-LB)
63        0.   "    "     "   Y-AXIS
64        0.   "    "     "   Z-AXIS

C SEVEN PERIODS AVAILABLE, ONLY 1ST TWO ARE SHOWN.....
65        0.0 TIME TO START CONSTANT FORCE, 1ST PERIOD (SEC)
66        0.  CONSTANT BODY X-AXIS FORCE,    1ST PERIOD (LB)
67        0.   "    " Y "   "      1ST PERIOD
68        0.   "    " Z "   "      1ST PERIOD
69        0.0 TIME TO START CONSTANT FORCE, 2ND PERIOD
70        0.  CONSTANT BODY X-AXIS FORCE,    2ND PERIOD
71        0.   "    " Y "   "      2ND PERIOD
72        0.   "    " Z "   "      2ND PERIOD
C              ...................
```

```
C-----------------------------------------------------------------
C DATA APPLYING TO OBJECT CONTROL SYS OPTION 4 (LIB CONT/INLINE THRUST)
C-----------------------------------------------------------------
 176       0.0 DEADBAND FOR LIB CONTROL (DEG)  (=0.0 DE-ACTIVATES)
 177       0.0 IN-PLANE CMD ANGLE (DEG) OF LIB TYPE USED FOR INPUT
 183       0.0 START TIME FOR LIB CONTROL (0.0 MEANS ON IMMEDIATE)
 184       0.0 STOP  TIME FOR LIB CONTROL (0.0 MEANS NO TIME STOP)
C ZERO FOR ACTIVATION TRIGGERS ITEMS BELOW CAN MEAN, IGNORE-OR-ACTIVATE
C SEE TOSS REF MANUAL FOR LOGICAL INTERACTION OF THESE ITEMS
 178       0.0 INLINE THRUSTER LEVEL (LB) (.EQ. 0.0  DE-ACTIVATES)
 179       0.0 TIME TO    ACTIVATE (.EQ. 0. MAY MEAN TURN ON)
 180       0.0 TIME TO DE-ACTIVATE (.EQ. 0. MAY MEAN NO TURN OFF)
 181       0.0 DEP LENGTH ACTIVATE (L .LT. = ON, TIMES PREEMPT)

  0       0.0     END DATA

 ------------------ READ-IN DTOS2 ARRAY HERE - REAL VARIABLES
  0       0.0     END DATA

C RUN SEQUENCE TERMINATION CONSISTS OF 2 ALPHANUMERIC CARDS (AS
C EXPECTED BY GTOSS),  PLUS A GTOSS DATA CARD WITH NEGATIVE VALUE
C IN THE I3 FIELD

 >>>>>>>>>>>>>> GTOSS ALPHANUMERIC CARD <<<<<<<<<<<<<<<<<<<
 >>>>>>>>>>>>>>  END OF RUN STREAM DECK <<<<<<<<<<<<<<<<<<<
 -1
```

Appendix 5: GSFC/Gradsat Trip Report

February 16, 2001

To: Randy Baggett and Ken Welzyn, MSFC/NASA

From: John Glaese, Control Dynamics Division of bd Systems

Subject: GSFC/Gradsat Trip Report

On the evening of the 13th of February, I traveled to Greenbelt, Maryland to particapte in the Gradsat Proposal planning meeting. The travel was somewhat unpleasant due to bad weather and crowded flights which made for delays. The meeting was held at the Integrated Mission Design Center (IMDC), Room N311 in building 23, at GSFC. It started at 9 am EST on Wednesday, the 14th. Patrick Taylor overviewed the activities and introduced the first speaker, Dr. Jose Merayo. Dr. Merayo described the proposed mission including instrumentation and a spacecraft configuration including a 10-20 km tether. . The separation between end bodies determines how well the gradient measurements can determine crustal field sources. The system would consist of 2 nearly identical end bodies each having a vector and a scalar magnetometer mounted on a boom of sufficient length (3-5 m) to provide magnetic isolation to the instruments from end body induced fields. The instruments must provide measurements with noise in the fractional nT range. The system is configured to provide data for making a gradient measurement of the earth magnetic field. The tether is required to provide separation for the gradient measurements. The gradient measurements in turn provide data to determine earth crustal field sources. The desired inclination of the orbit is polar (inclination = 90□). This would provide data in a region never measured from space before. In this concept each end body mass was approximately 30 kg. The tether is envisioned to be similar to the NRL Tether Physics and Survivability (TiPS) experiment and deployed by a SEDS type deployer. That tether was a nonconductor, 4 km in length and approximately 2 mm in diameter. Scaling this to 10 km in length may present a problem and needs to be looked at. Will Webster at GSFC who has been involved in the SEDS missions as well as the TSS missions suggested that perhaps 8 km might be sufficient for the mission. End body separation determines how accurately crustal field sources can be determined. An altitude of 400 km or less may be required to achieve crustal values.

Dr. Merayo's presentation was followed by a presentation by Dan Mark/Swales Aerospace. Swales is the industry team member selected by GSFC. They have taken a set of mission requirements as provided by GSFC and the draft Announcement of Opportunity. In their concept, each end body is 130 kg, although the lower end body is expected to be heavier because it will hold the deployer. The launch would occur in about the 2004 time frame. The launch vehicle would be a Pegasus or half a Delta. They would consider being a secondary payload. Their concept includes solar arrays to provide power for the housekeeping functions of the end bodies. They have included a momentum wheel and a set of magnetic torquers on each end body to provide attitude control. They also discussed using thrusters to drive the deployment possibly out of ignorance of the SEDS system which has in the past used heavy springs to provide an initial separation rate in the 1.5 m/s or greater range. Some possible launch vehicle options may drive the design toward the use of thrusters. This was not discussed in order to get through the Swales presentation in a reasonable time. They have sized the system assuming batteries on both ends and a deployer on the lower end. A propulsion system is also included in their concept. They did a study on the booms for the magnetometers and concluded they needed to be at least 20 m long for sufficient magnetic isolation. There was discussion if that couldn't be reduced by carefully maintaining magnetic cleanliness of the end bodies. Booms in the 5 m range were thought to be approaching the limit of what could be built. The momentum wheel may be a considerable source of induced onboard magnetic dipole moment and may have driven the Swales boom considerations. It is planned that after the gradient measurement phase of the mission is complete, the tether will be cut giving the upper end body an altitude boost and additional lifetime, while lowering the lower body into the denser atmosphere causing it to de-orbit sooner. This was put in to give the mission additional pizazz. No one really wanted to say this was an essential feature but merely a bonus capability.

I met with Greg Garby and we had lunch along with the principal players from GSFC including Patrick Taylor and Will Webster. In the afternoon, we split informally into two groups. The group I chose critiqued the current mission concept and suggested revisions to it so that Swales could make some sensible revisions. With all of that ,there was really no discussion of roles and missions. They are talking about a SEDS type deployer and a TiPS size tether. I haven't fully reconciled the TiPS 4 km tether with the desired 10 km tether yet. Also, what MSFC's role might be was not explicitly discussed. They welcomed our participation and it seemed implicit that they were looking for help on the tether dynamics and deployer system. They did do a brief study of effects of drag assuming a 2 mm, 10 km tether with 2 identical 100 kg end masses using a simple, drag model. That came up with a 49 day satellite lifetime from 400km altitude. It was decided that the initial orbital altitude should be in the range 475-500km depending upon launch vehicle capability. A trade study will need to be done to optimize the starting altitude. The tether design would allow sufficient time in the beginning of the mission to complete the systems checkout and allow the tether dynamics to damp, while assuring that the orbital decay will bring the system to 400km before anything fails or the tether is inadvertently cut. This was still an open issue but not felt to be an obstacle when the meeting ended. It was evident that the selling strategy was to be that this is a sound core mission concept with sufficient remaining flexibility to provide a range of potential experiments. Greg and I left after the afternoon meeting which ended about 3:15 pm.

The overall mission concept seems like it is well suited for tether applications and is probably something that could not be done easily in any other way. Cost figures for the spacecraft/tether system were put in the $26-30M range. The figure $27M was suggested as an estimate of the launch vehicle cost, although it was unclear what that represented, a Pegasus or half a Delta? It was also stated that NASA doesn't like it when the LV costs more than the spacecraft.

The preceeding is as much as my notes and my memory cover. If you have questions, please call or e-mail me.

John Glaese
256-882-2720 ext 114

Appendix 6: STANDARD FORM 298

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 19, 2001	3. REPORT TYPE AND DATES COVERED Final 21 July 2000 – 19 March 2001	
4. TITLE AND SUBTITLE Electrodynamic Tether Operations and Control			5. FUNDING NUMBERS H-32041D
6. AUTHORS John R. Glaese			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) bd Systems, Inc. Control Dynamics Division 600 Boulevard South, Suite 304 Huntsville, AL 35802			8. PERFORMING ORGANIZATION REPORT NUMBER TCD20010080A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration George C. Marshall Space Flight Center Marshall Space Flight Center, AL 35812			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT			12b. DISTRIBUTION CODE
13. ABSTRACT *(Maximum 200 words)* This final report documents and summarizes work performed under the PO # noted in item 5.			
14. SUBJECT TERMS Tether, current collection, simulation, dynamics, electrodynamic tethers, bare wire current collection, plasma contactor, skip rope observer			15. NUMBER OF PAGES 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT

NSN 7540-01-280-5500 Computer Generated STANDARD FORM 298 (Rev 2-89)
Prescribed by ANSI Std 239-18
298-102