Space Trajectory Analysis "ARCHEAN" User's Manual

STA-UM-20091110

Issue: 1.0

Date: November 2009

Space.Trajectory.Analysis@gmail.com

CONTENTS

1. INTRODUCTION

1.1 Scope of the Document

This document describes the User's Manual of the Space Trajectory Analysis STA "ARCHEAN" version. The STA web site can be found at the following URL: STA Web site

1.2 Applicable and Reference Documents

1.1.1 Applicable Documents

The following documents contain requirements applicable to the activity:

[A1] Space Trajectory Analysis Developer's Guide, G. Ortega, TEC-ECM-GO-20070227, February 27th, 2007

[A2] STA Source Code Licensing and Copyright, G. Ortega, TEC-ECM-GO-20071109, April 9, 2007

[A3] ESA software requirement standard ECSS-E40 tailored to small software projects. Web site as per the following URL: http://www.ecss.nl/

1.1.2 Reference Documents

The following documents contain references applicable to the activity:

[R1] Celestia User's Guide, Rev 1.4.1 and 1.4.0, February 2006

[R2] Celestia Web Site

[R3] Celestia Motherlode Web Site

2. STA BACKGROUND AND HISTORY

The European Space Agency (ESA) initiated in 2005 an internal activity to develop an open source astrodynamics software suite involving university science departments and research institutions. This project is called the "Space Trajectory Analysis" or STA.

The objectives of STA are threefold: first, to build an education environment in areas like astrodynamics, space sciences, mathematics, and informatics where teachers and students can interact towards Master, PhD, and postdoctoral degrees. Second, to promote the exchange of technical ideas, and raise knowledge and competence in these areas at University level. Last to provide a software suite for the purposes of research. In this sense, STA is conceived as test bed for the testing, verification and validation of new ideas, technologies, and techniques.

Conceived as a education and research tool to support the analysis phase of a space mission, STA is able to visualize a wide range of space trajectories. These include among others re-entry, descent and landing trajectories, orbits around planets and moons, rendezvous trajectories, etc.

The STA project is an original idea of the Technical Directorate of ESA. It was born in August 2005 to provide a framework in astrodynamics research at University level. As research and education software applicable to Academia, a number of Universities support this development partnering with ESA. STA partners are organized around the STA Steering Board. The STA Steering Board is composed of: European Space Agency, University of Delft, University of Bremen, University of Kent, University of Coimbra, Complutense University of Madrid, University of Birmingham, Instituto Superior Técnico Lisbon, University of Southampton, Technical University of Madrid, Politecnico di Milano, University of Wuerzburg, Celestia

The STA development is open source and it is based on the state of the art astrodynamics routines that are grouped into modules. The modules are programmed using the C++ language. The different STA modules are designed, developed, tested and verified by the different Universities. Software integration and overall validation is performed by ESA. Students are chosen to work in STA modules as part of their Master or PhD thesis programs. As part of their growing experience, the students learn how to write documentation for a space project using European Coorperation on Space Standardization (ECSS) standards, how to test and verify the software modules they write and, how to interact with ESA and each other in this process.

The STA project allows a strong link among applied mathematics, space engineering, and informatics disciplines by reinforcing the academic community with requirements and needs coming from space agencies and industry real needs and missions.

2.1 STA Partners and Team

Universities and entities partnering with ESA to date are as follows:

- Technical University of Delft (Netherlands)
- University of Bremen (Germany)
- University of Wuerzburg (Germany)

- University of Kent (United Kingdom)
- University of Coimbra (Portugal)
- Complutense University (Spain)
- University of Birmingham (United Kingdom)
- Instituto Superior Technico de Lisboa (Portugal)
- University of Southampton (United Kingdom)
- Technical University of Madrid (Spain)
- Politechnico di Milano (Italy)
- Celestia team

2.2 STA Users

The target user community of STA is listed as follows:

- **Space Engineering Student:** The student user is in the process of obtaining the knowledge on astrodynamics and orbital mechanics that is necessary to fully understand the models and properties of the software.
- **Space Engineering Teacher**: interested in the use of STA in the classroom as a tool for education. His/her key objective is to use the tool as a mean to explain the principles and applications of flight dynamics, orbital analysis, applied mathematics, informatics, etc.
- **Astrodynamics Specialist**: The Astrodynamics Engineer user has knowledge in the area of astrodynamics and orbital mechanics.
- **Mathematical analyst**: utilises the STA software suite as a test bench to proof-concept new ideas and algorithms in the area of the applied mathematics for space problems (propagators, integrators, interpolators, etc).

3. STA BINARIES

The STA versioning system names are taken from the Earth geological EONS. "HADEAN" corresponds to version 1. "ARCHEAN" corresponds to version 2. The vresion 3 will be named "CAMBRIAN". Intermediate versions are released every Quartal in between major releases. The present user's manual explains the features available for the ARCHEAN version of STA only.

STA comes in three flavours: Windows, Linux and MAC OS X (see figure 3.1).

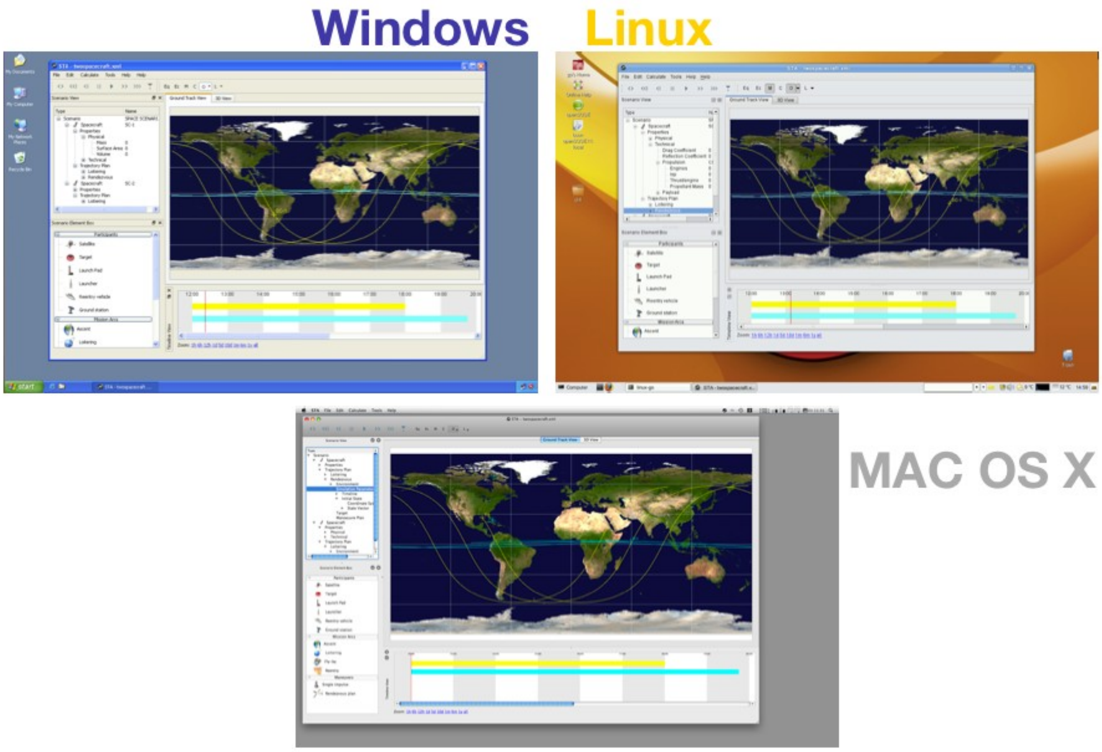

Figure 3.1: STA on three operating systems

Identical functionality is provided in all operating systems supported. The widgets are operating system native and provide the user the same look and feel of the rest of the applications of the operating system used.

3.1 Requirements to run STA

STA has been tested to work on the following variants of the 3 operating system families supported:

- Windows XP, Windows VISTA, and Windows 7
- Linux Ubuntu 8.10, Linux OpenSUSE 11.1, Linux Fedora 10, Linux OpenSUSE 11.2, Ubuntu 9.1, Linux Mint KDE 7 Gloria
- MAC OS X 10.4, MAC OS X 10.5, MAC OS X 10.6

All versions of STA require that the user has OpenGL installed. Windows and MAC OSX come with OpenGL drivers pre-installed. For linux, more and more distributions are shipping with OpenGL, so if you've recently installed or re-installed Linux, there's a good chance that you already have a working OpenGL configuration.

For a full fucntionlity of STA, the binaries need to have LUA libraries installed. Those come with the binary installtions of STA in Windows, MAC, and Linux.

In general, to run STA, the computer should have a CPU (processor) that has a speed of at least 800 million cycles per second (800 MHz). Typically, computers bought new within the last 4 years have adequate CPU speeds. If your computer is more than 4 years old or is a Pentium III or equivalent with a processor speed slower than 800 MHz, STA 3D visuals may run slowly.

The 3D visuals of STA require an OpenGL compatible graphical card to run the 3D visualisation graphics. It is required to have a computer with a minimum of 128 Mb RAM and 500 Mb hard disk.

STA ARCHEAN is a 32-bit application. However, STA ARCHEAN will run on 64-bits operating systems. Support for 64-bits is currently the target of the CAMBRIAN version.

STA ARCHEAN only runs on Intel-i386 compatible computers. At this moment, STA has not been compiled on PPC, or Solaris computers. However, the MAC OS X version is "Universal Binary" and hence it will run on PPC computers as well.

3.2 Installing STA

The web site of STA allows the download of both the source code and the current binaries.

The STA binary install packages allow the user to perform basic installations quickly and without hassle. Just run the installer package and follow its instructions (see figure 3.2).

The installation of the STA from source code requires the use of a SVN (subversion) client in the user's machine and a suitable compiler to compile the code:

- In Windows it is recommended o use TortoiseSVN.
- Most Linux distributions have SVN client install by default.
- MAC OS X 10.4, 10.5, and 10.6 has a SVN client install by default.

To download the source, from the SVN it will be just required to type in the following command in the desired folder that will accommodate the files:

```
svn co http://sta.svn.sourceforge.net/svnroot/sta STA
```

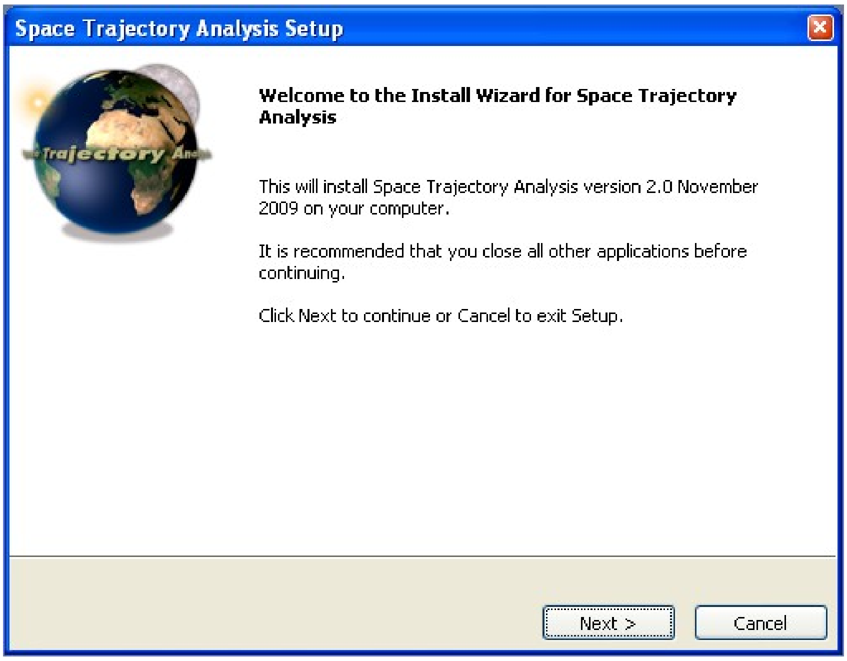

Figure 3.2: Install shield of STA in Windows XP

Once the source code has been checked out of the SVN STA repository, the user needs to compile it. STA uses Qt technology to allow fast compilation in the three operating system families supported. The Qt open source edition can be downloaded from the Qt web site. Qt will install in the three families of operating system supported by STA.

If the installation of Qt was successful, the user needs to run a simple command to compile the entire source code. The command has to be issued in the folder “sta-src” of the STA source code hierarchy:

```
qmake -config release; make
```

3.3 Understanding where STA files are located in your system after installation

After the installation has taken place, the STA installer created a set of files inside some folder as follows:

The binaries of STA along with required pre-compiled libraries are stored under any folder the user may choose. For example:

- In Windows, the installer offers the user to install under “`C:\Program Files\STA`”
- In Linux, the installer installs STA under “`/usr/local/STA`”
- In MAC OS X, the installer offers the user to install under “`/Applications`”

To function properly, STA requires some extra data files that are grouped in a particular folder. This folder contains start-up scripts of STA, textures of planets and moons, spacecraft 3D models, etc.

- In Windows, the installer will install the data folder under “`C:\Program Files\STA\sta-data`”

- In Linux, the installer will install the data folder under "`/usr/local/STA/sta-data`"
- In MAC OS X, the binary application contains already the required data inside the bundle.

While you can change the location of the STA binaries, you need to change the location of the data files with the binary. Otherwise, STA will report and error at run time.

4. USING STA

Once, STA is run for the first time, the screen of figure 4.1 appears to the user.

Figure 4.1: STA screen after run

The main screen contains 5 areas:

- The **SCENARIO VIEW** area. This area allows to add and remove all the elements that form part of a given scenario. The sum of all elements for the current scenario view.
- The **SCENARIO ELEMENT** area. This area contains all allowable elements of a given scenario, like spacecraft, ground stations,
- The **RENDERING** area. This area displays 2D and 2D trajectory visualisations of the elements that contain the scenario.
- The **TIME-LINE** area. This area displays and controls the time of live of the elements of the scenario. It allows to play back the movement of the spacecraft in the scenario.
- The **CONTROLS** area controls certain visualisation features of the 2D and 3D rendering area.

The figure 4.2 shows the same screenshot of figure 4.1 but with the different areas labelled.

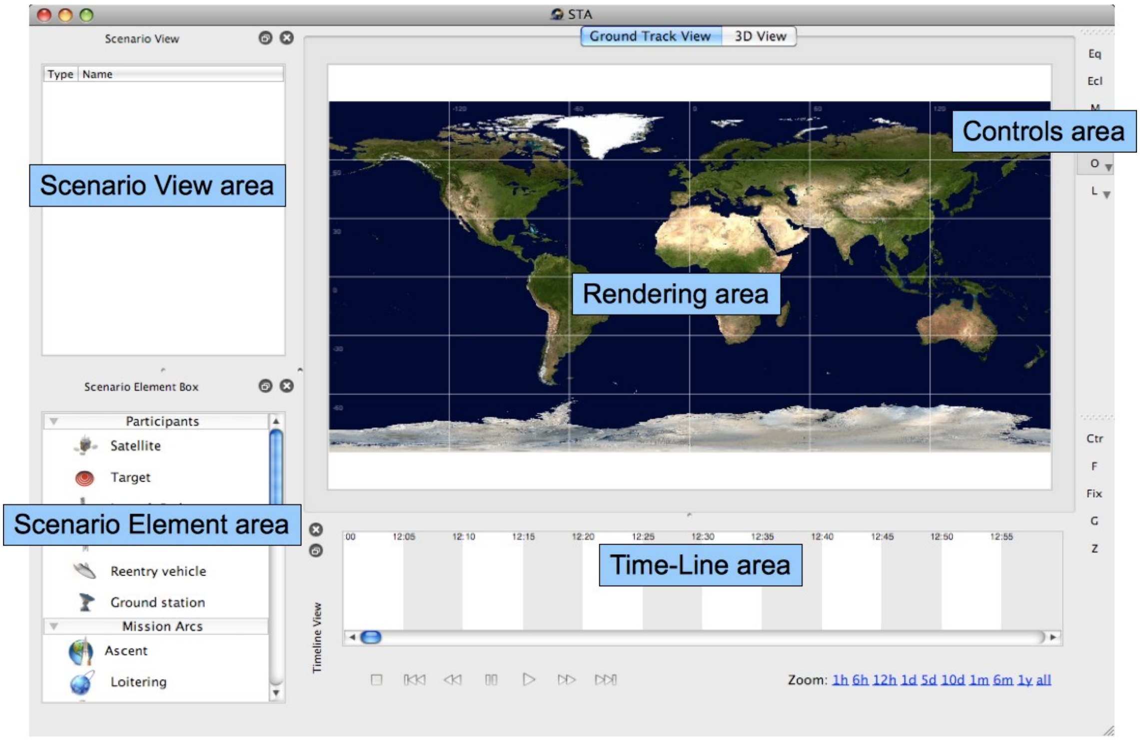

Figure 4.2: STA working areas

4.1 Docking and Un-docking Windows

In STA all working areas mentioned earlier can be docked and un-docked from the main rendering area (see figure 4.3). This feature allows the use to maximise the screen real state when concentrating on a particular area or type of work.

To undock an area from the main rendering area it is enough to click twice on the upper frame (banner) of the area the user wishes to undock. It is also possible to undock an area by clicking on the undock icon. 

To dock back the undocked area to its original position, it is sufficient to click twice on the upper frame (banner) of the area the user wishes to dock back.

The rendering area undocks in a different manner: the user must explicitly click on the pull-down menu “**Undock Ground Track View**”. This produces the undock of the 2D view window from the 3D view window. To dock back the window, the user must click again on the pull down menu “**Dock Ground Track View**”.

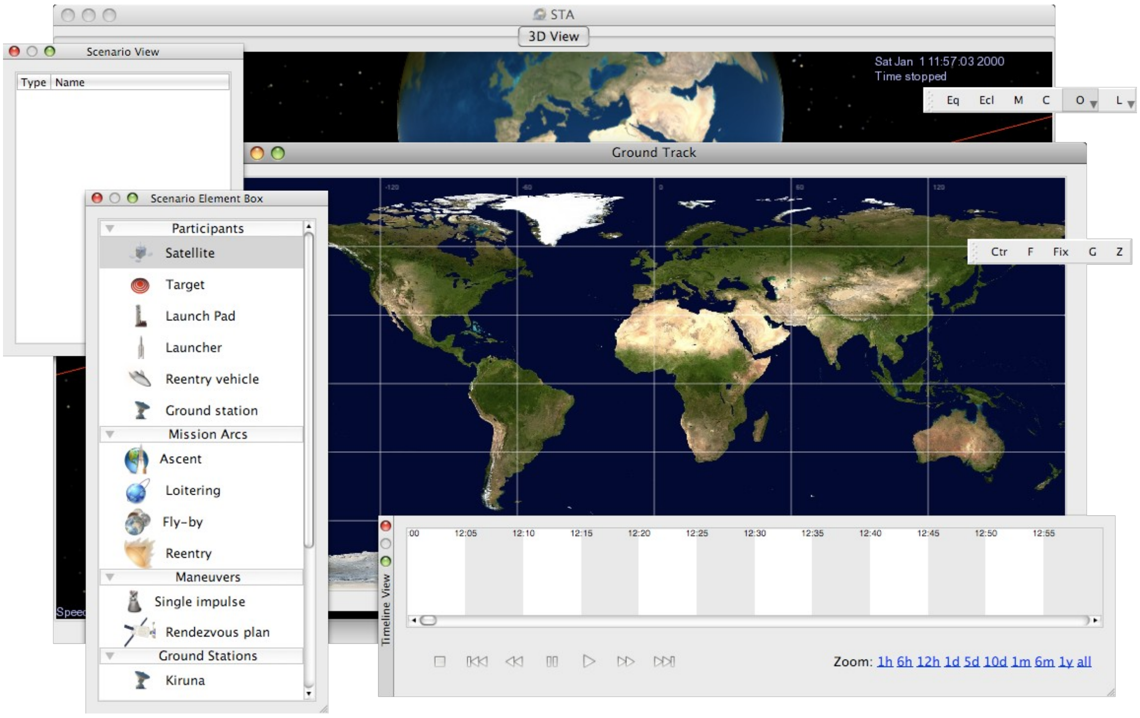

Figure 4.3: Un-docking all areas on STA

5. WORKING WITH SPACE SCENARIOS

5.1 Understanding the Space Scenario Concept in STA

STA implements the concept of Space Scenario Paradigm (SSP). The SSP of the STA software suite implements the central idea of scenario in which elements can interact to allow the user detailed analysis of these elements.

STA defines the concept of Space Scenario (SS) as “a set of elements that play a role in the analysis of trajectory path in a given Universe”.

Following the previous definition, the components of a Space Scenario could be active participants (e.g. spacecraft, ground stations, sensors etc.) and also planets, aerodynamics database of a vehicle, co-ordinate systems, etc.

Every participant is composed of active and non-active elements and they are divided in two big groups:

- Ground elements (like launch pads, ground stations, points, etc)
- Space Vehicles (like satellites, rockets, re-entry vehicles, etc)

Ground elements are characterised by a ground position and a central body.

Space Vehicles are characterised by:

- Properties (both physical and technical).
- Additional components (optional, like for example sensors).
- Trajectory plan, a set of trajectories which model the entire life of the element. In every trajectory the user needs to specify all the information needed to define (or optimise) its evolution uniquely in space and time.

The Space Scenario Paradigm is implemented using XML descriptions. XML is an extensible markup language which easily permits to describe the tree structure of the scenario. Every XML file has to be validated by an XML Schema which define the constraints on the structure and content of documents of that type.

5.2 Creating, Saving, and Loading Scenarios

To create from scratch a new scenario, the user must click on the pull-down menu “File -> New Scenario”.

STA responds by creating a new scenario with an empty scenario view (figure 5.1).

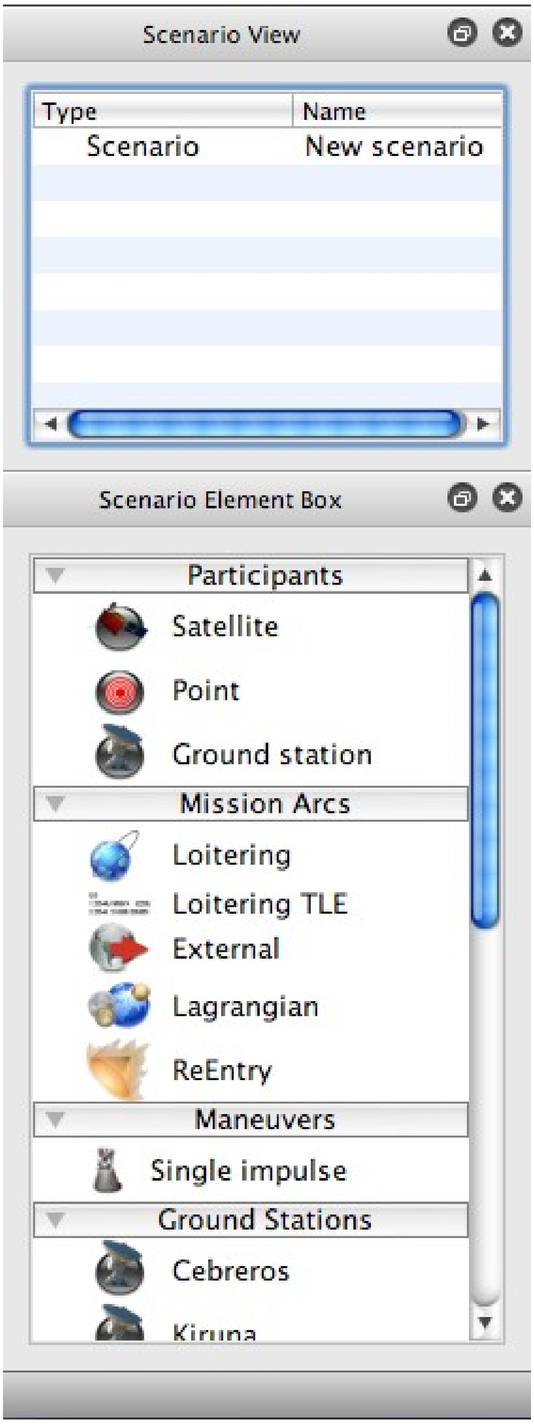

Figure 5.1: New scenario

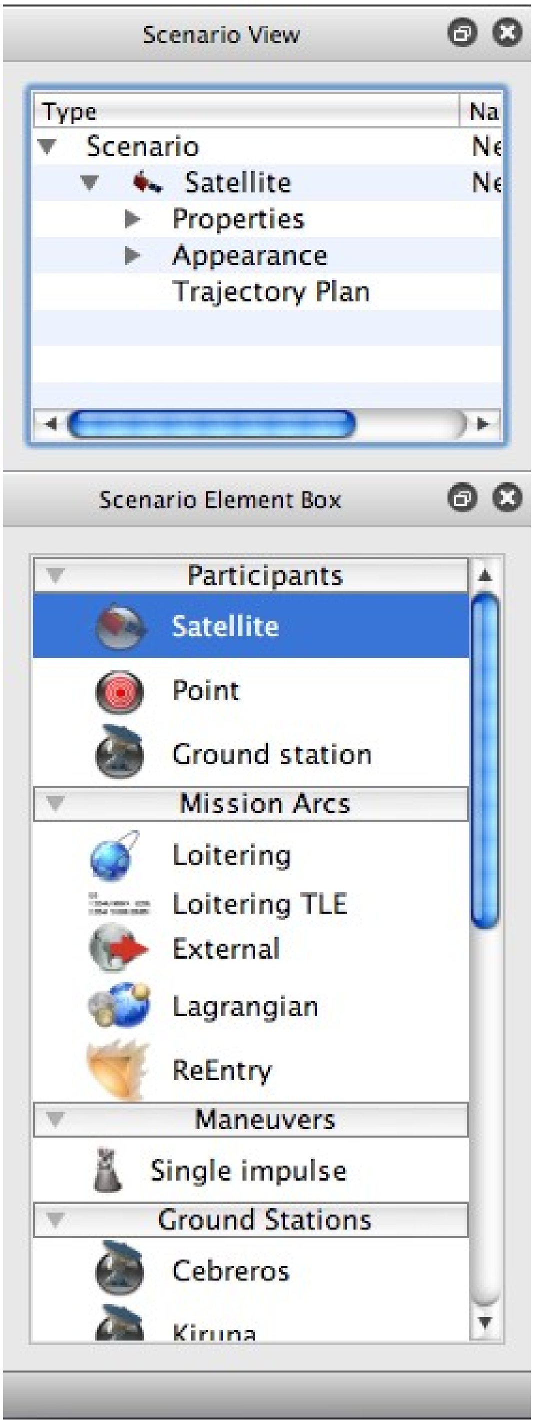

Figure 5.2: New satellite

Next, the user needs to place some participants in our new scenario. To do this, the user needs to drag a participant from the Scenario Element area (box) and drop it on top of the word "Scenario" in the Scenario View area. STA responds by creating a new participant. In the case of figure 5.2, the new participant is a satellite.

Our scenario contains now a single participant. To be able to calculate a trajectory the user needs to drag and drop a mission arc into our participant. In the example of figure

5.2, we have dragged the mission arc "Loitering" into the words "Trajectory Plan" of the participant satellite.

At this point in time, the user is now ready to edit the properties of the Loitering arc. Clicking twice on the word "Loitering" in the Scenario View area, the user can edit the properties of this mission arc (see figure 5.3).

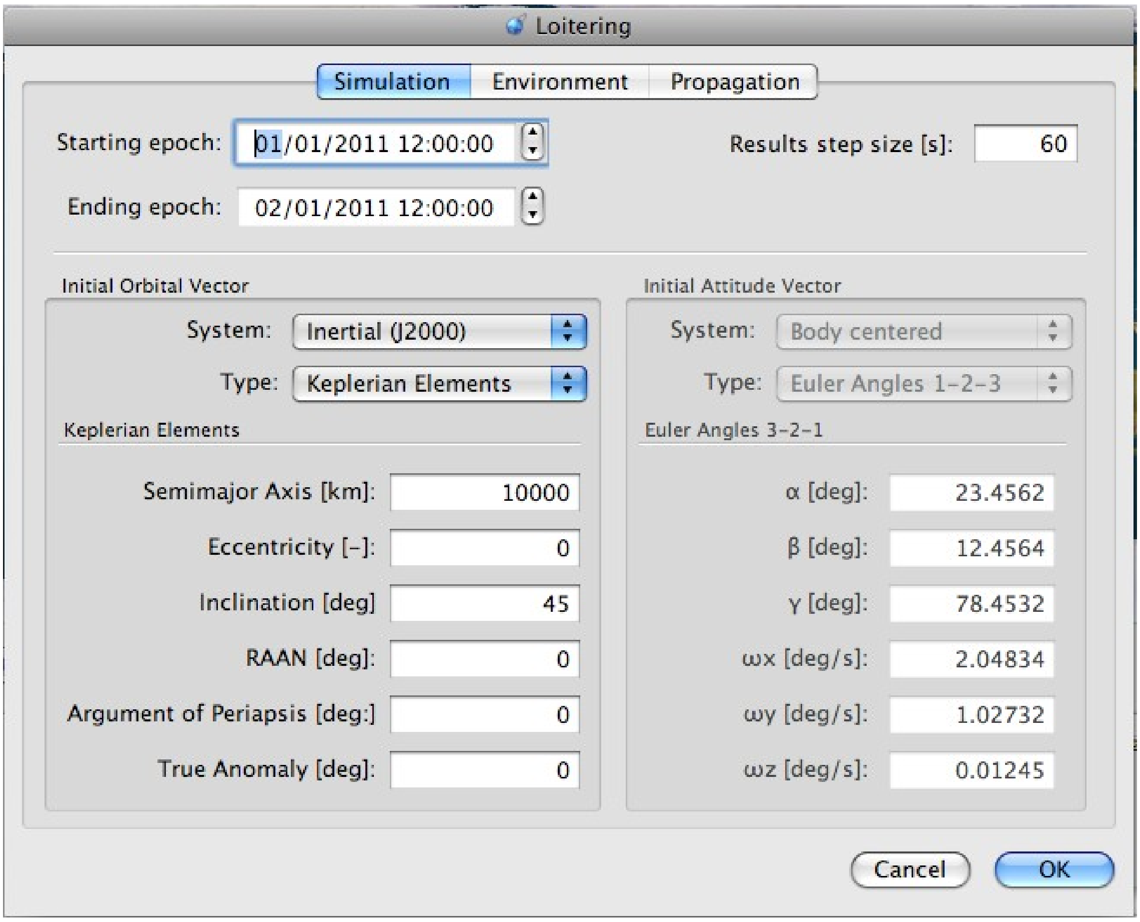

Figure 5.3: Editing the "Loitering" arc of a trajectory plan

More participants can be introduced now in our scenario. Simply drag and drop any of the participants icons on the word "Scenario".

Saving an scenario is easily done using the STA pull-down menus: File -> Save as

All STA scenarios must have the file extension *.stas

STA will not recognise any file with a different file extension.

Loading a previously saved scenario is also easily done using the STA pull-down menus: File -> Open scenario

STA comes with several examples prepared for the user. Look into the STA data folder for a sub-folder name "scenario-examples".

5.3 Other Mission Arcs in ARCHEAN

The re-entry and Lagrangian mission arcs graphical user interface are enabled in ARCHEAN. The corresponding implementation code has been enabled in this version. See here after for more details on this arcs.

5.4 Implementation of manoeuvres in ARCHEAN

The implementation of manoeuvres in ARCHEAN is not enabled yet. However as preview, the user can select one type of manoeuvres ("single impulse") by dragging and drooping them onto the trajectory plan of a participant. If this action is triggered, STA ARCHEAN does not yet calculate the corresponding manoeuvre.

5.5 Pre-defined ground stations

Among all possible ARCHEAN participants, STA allows ground stations to be placed on a given scenario. ARCHEAN however, allows the user to select a particular set of ground stations: the ESTRACK network. The ESTRACK network is the set of tracking stations of the European Space Agency. The location of these stations on Earth has been predefined in STA so the user does not need to introduce this information when creating a new participant ground station.

The user can include a new ESTRACK ground station on a given scenario by dragging and dropping the station icon on the word "Scenario". After the user has placed the new station it will render on the 2D view window once the user run a "propagation" (see more on propagating on the next section).

5.6 Propagating Trajectories in STA

Once the trajectory plan data of a particular space vehicle has been entered into STA, the user must "propagate" the trajectory. This is performed by means of the STA pull-down menu "`Propagate`" of by entering the keystrokes <`Ctr+P`> (Windows and Linux) or <`Command+P`> (MAC).

STA will respond by propagating the trajectory and creating the necessary files that will contain the history of the state vectors of the space vehicle concerned.

The ephemeris files generated after the propagation are placed on the root of the STA data folder.

5.7 STA 2D View: Ground Tracks

Any propagated trajectory will be immediately displayed by STA on the rendering area in 2D view. The 2D view is a Miller projection of the ground track of the calculated trajectory.

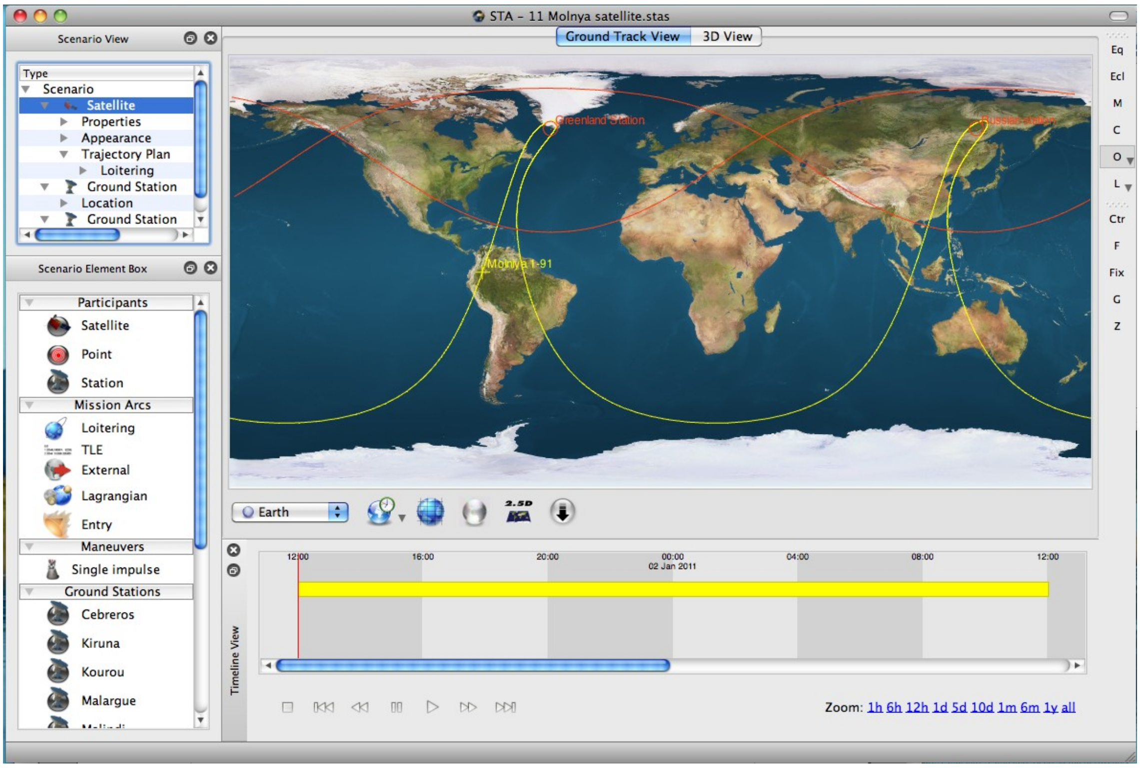

Figure 5.4: Ground track of a Molnya orbit

The figure 5.4 shows a ground track of a Molnya orbit that is provided as example in the “scenario-examples” folder of the STA installation.

5.8 STA 3D View: Orthogonal Trajectories

The 3D visuals of STA uses the rendering engine of Celestia [R1][R2][R3]. This means that almost all features of this rendering engine available in Celestia are available to STA as well.

When the user calculates a trajectory and this is rendered on the 2D view as a ground track, STA also displays the trajectory on the 3D view immediately as a orthogonal view.

The figure 5.5 shows the rendered 3D view of the Molnya trajectory.

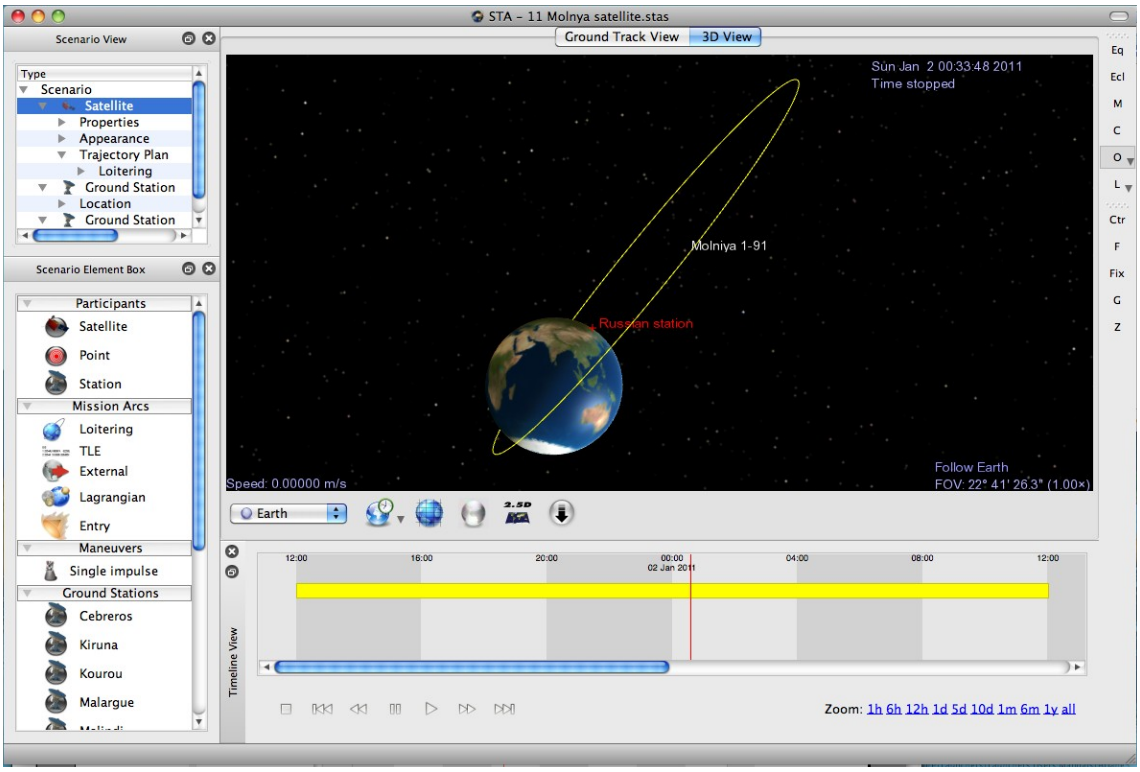

Figure 5.5: Molnya orbit in 3D

The 3D window accepts the following <u>mouse</u> functions:

Left – click on object: select object
Right drag: orbit the selected object in any direction
Left drag: orient scene in any direction
Rotate Mouse Wheel: adjust distance to selection
Right + Left drag: adjust distance to selection
Ctrl + Left drag: adjust distance to selection
Shift + Left drag: change field of view (FOV) (e.g. => telescopic view)
Wheel (middle button) click: toggle field of view between 45 degrees and the previous field
Left double click **centre** selection
Right - click bring up context/select menu

The 3D window accepts the following <u>keyboard</u> functions:

LABEL FUNCTIONS

R	Decrease texture resolution
Shift+R	Increase texture resolution
P	Toggle (turns on or off) planet labels
M	Toggle moon labels
E	Toggle galaxy and nebula labels
B	Toggle star labels
W	Toggle asteroid and comet labels
N	Toggle spacecraft labels
=	Toggle constellation labels
Shift + &	Toggle Location labels
Ctrl + K	Toggle Markers
Ctrl + P	Set a Marker

V	Toggle info text

RENDER FUNCTIONS

U	Toggle galaxy rendering
O	Toggle planet orbits
I	Toggle clouds
Ctrl+A	Toggle atmospheres
Ctrl+B	Toggle constellation boundaries
Ctrl+E	Toggle eclipse shadows
Ctrl+L	Toggle nightside lights
Ctrl+S	Toggle stars as points, discs or fuzzy points
Ctrl+T	Toggle comet tails
Ctrl+V	Cycle through vertex shading options
Ctrl+X	Toggle antialias lines mode
Ctrl+Y	Toggle autoMag = auto adaptation of star visibility to field of view
Shift + ^	Toggle Nebula on or off
Shift + {	Decrease ambient light
Shift + }	Increase ambient light
Shift + (	Decrease galaxy brightness
Shift +)	Increase galaxy brightness
Shift + %	Toggle Star Color highlights
Shift + "+"	Toggle Limit of Knowledge textures
Alt+Enter	Toggle full screen display mode on or off
Esc	cancels command, cancels script, cancels movement or lock commands

NAVIGATION FUNCTIONS

H	Select the Sun (Home)
C	Center on selected object
G	Goto selected object
Ctrl+G	Goto surface of the object
F	Follow selected object
ENTER	Select a star or planet by typing its name, then press **Enter** again
Y	Orbit the selected object at a rate synchronised to its rotation
Shift + :	Lock on selected object. Point at 2nd object and press again to pair the two.
Shift + "	Chase selected object (orientation is based on selection's velocity)
T	Track selected object (keep selected object centred in view)
HOME	Move closer to object
END	Move farther from object
Up arrow	Your view pitches downward (also see # **8** key on numerical keypad)
Down arrow	Your view pitches upward (also see # **2** key on numerical keypad)
Left arrow	Your view rolls counter-clockwise (also see # **7** key on numerical keypad)
Right arrow	Your view rolls clockwise (also see # **9** on numerical keypad)
Shift+arrow keys	Orbit around the object automatically
Shift + *	Look back view (rear view)
1-9	Select planets around nearby Sun
Esc	Cancel hold on object, cancel command or script action
Backspace	Cancel current selection

TIME FUNCTIONS

Spacebar	Stop or pause time (or if paused, resume time)
L	Time 10x faster (repeat for faster time)
K	Time 10x slower (repeat for slower time)
J	Reverse time (it flows backward)

MULTIVIEW FUNCTIONS

Ctrl+R	Split view vertically
Ctrl+U	Split view horizontally
TAB	Cycle active view
DEL	Delete active view
Ctrl+D	Delete all views except active one

5.9 STA Time-Lines

The "Time-Line" area allows the user to control the movement of the space vehicles on a given scenario by playing-back a previously propagated session with space vehicles, ground stations, etc.

The time-line displays a set of bars in different colours. Each bar corresponds to a space vehicle (see figure 5.6).

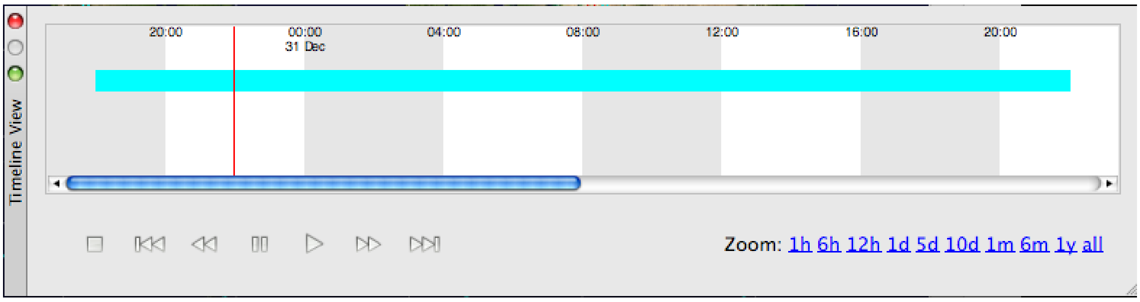

Figure 5.6: Time-Line area

Orthogonal to the time-liner bars of each space vehicle, STA shows a red line crossing all of them. This red line represents the "time now" line of a given scenario playback. The user can drag the bar left and right to accelerate of backward the time of a simulation session.

The lower left part of the time-line area contains the playback controls of the simulation (see figure 5.7).

Figure 5.7: Playback controls of the time-line area

The controls allow the user to "play", "accelerate", "retard", or "play backwards" the movement of the space vehicles in the scenario along the time-line simulation.

The lower right part of the time-line area contains the zoom controls of the simulation time-line (see figure ??).

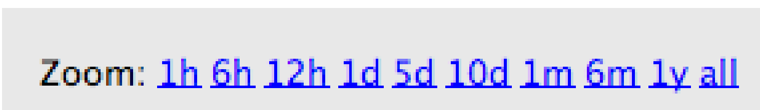

Figure 5.8: Zoom control of a time-line

The zoom allows the user to enlarge or shorten a given time line by effectibly zoom it in and out. This provides a good tool to observe the granularity of a given simulation time-line.

5.10 STA Controls area

The STA controls area is initially located on the right most part of the STA main window. The user, however, can move the toolbar and locate it on top, down or right of the main STA window. Once the change has been committed, the next time STA starts the toolbar will be located at that particular position.

Figure 5.9: Display features control

The control area is composed of two distinctive toolbars (see figure 5.9 and 5.10).

The “Display Features Control” toolbar in figure 5.9 allows the user to modify certain attributes of the 3D visualisation rendering window:

- **Eq** allows to show and hide the equatorial co-ordinates grid
- **Ecl** allows to show and hide the ecliptic co-ordinates grid. STA places an equatorial co-ordinate grid on the screen showing the celestial position and direction you are facing in space. The grid follows the standard convention of Right Ascension and Declination used in Astronomy.

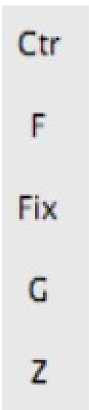

Figure 5.10: Participant features control

- **M** allows to mark and un-mark participants objects with 4 red small coned shaped arrows located in the four cardinal points of the participant. When turned-on, this feature allows the user to see and follow clearly marked participants.
- **C** allows to show and hide constellations. Since STA is based on the rendering engine of Celestia, STA is able to display constellations boundaries from its star catalogue. The STA star catalogue is based on the Hipparcos ESA star catalogue. STA will display both constellation borders [Ctrl

+ B] and their "asterism shapes" [/] for each of the 88 constellations in the sky. In the second Options menu, you can also turn on constellation labels [=].

- **O** this is a click-in small menu with that allows to show or hide the **Orbits** of objects: star, planets, asteroids, comets, moons, spacecraft, etc.
- **L** this is a click-in small menu with that allows to show or hide the **Locations** (e.g. names or labels) of objects: star, planets, asteroids, comets, moons, spacecraft, etc.

The "Participant Features Control" toolbar in figure 5.10 allows the user to modify move around viewpoints of a participant in the 3D visualisation rendering window:

- **Ctr** allows to centre the 3D window in the current follow up object (can be a planet, spacecraft, etc).
- **F** allows to focus on the selected object. This is equivalent to the keyboard key "F".
- **Fix** allow to fix the view on the selected object.
- **G** orders STA to go to a previously selected object
- **Z** orders STA to zoom into a previously selected object

6. 3D Window Visuals

STA inherits 3D visuals from Celestia. One of the most interesting features of Celestia is the ability to navigate in the Universe at a speed faster than the speed of light!

6.1 Understanding 3D views

3D view of the universe are displayed on the 3D rendering window. As example, the figure 6.1 shows a screenshot of a 3D view of Saturn.

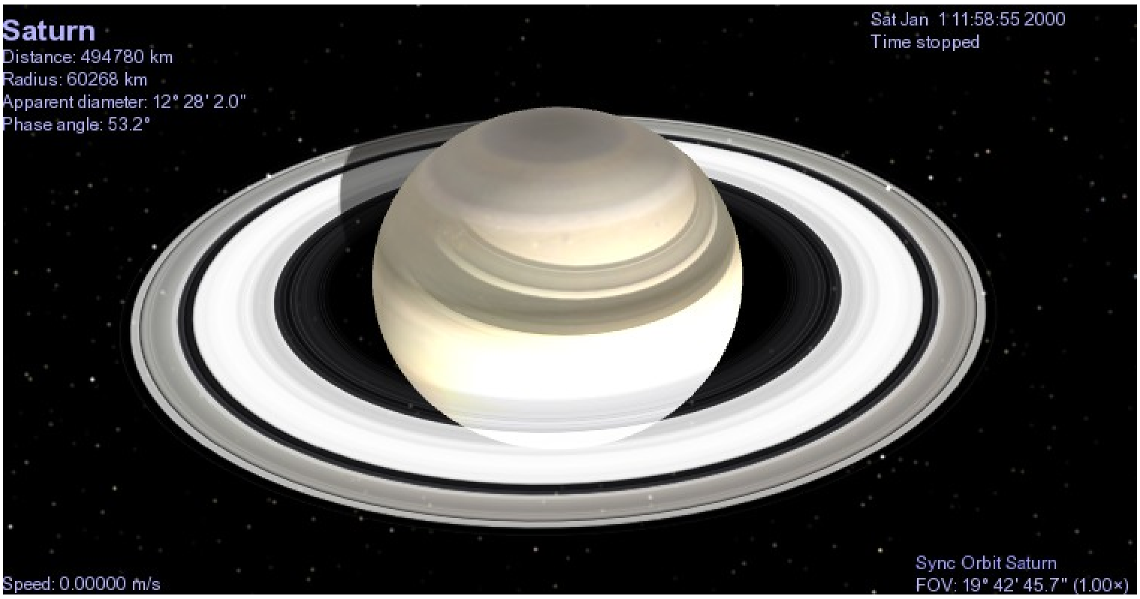

Figure 6.1: Rendering Saturn 3D view

In the top left corner STA displays information about our target (Saturn) (if you see no text, press the [V] key once on your keyboard to turn on your text display). "Distance" gives the distance from the surface of the object to our viewpoint. The Radius of the object will be listed in kilometres. The Apparent Diameter will be a value in degrees representing the size of the object in front of us as seen from our current viewpoint. If we move closer or farther away from the object, its apparent diameter will get larger or smaller respectively.

In the lower left corner STA shows our speed through space. At the moment, our view is stopped (relative to Saturn) so our speed is zero. Later, we will begin to fly on our own and our speed will be listed here (see next section on travelling through space).

In the top right corner STA displays the current date and time. In Astronomy, times are given in Universal Time (UTC) (commonly referred to as Greenwich Mean Time), and this is what STA uses by default. The date and time are in the format (Year, Month, Day, Hours: Minutes: Secs). Thus, 2005 10 19 14:10:06 UTC means that the time is October (the 10th month) 19 (19th day), 2005 at 2:10:06 PM. If the user finds it more convenient, it is possible to access the Time menu at the top of the program screen and command STA to show our local time zone instead. STA can also speed up, slow down time, or travel forward or backward in time with the touch of a button. Now, you are experiencing the program in "Stopped" mode.

6.2 Travelling the Universe

STA can travel from the vicinity of one participant to another in matter of milliseconds. This is particularly useful when dealing with complex scenarios in which the user requires to quickly view for example a planet and then move to the proximity of a spacecraft, then go to another planet, etc.

As an example, the figure 6.2 shows how to move from the Earth quickly to Saturn.

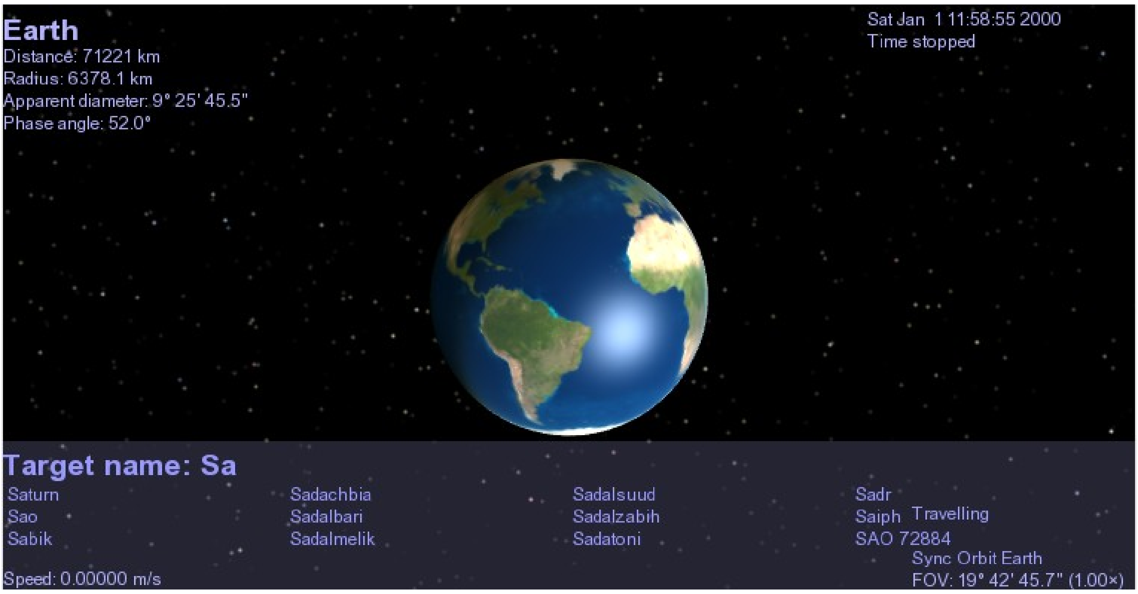

Figure 6.2: From the Earth to Saturn

The user needs to enter "Return" then write the name of the participant wished to visit (called here "Target name") and then press the key "G" from go or click on the control "G" of the controls area.

6.3 High resolution rendering

STA contains a sophisticated graphic drawing and rendering engine that draws objects using "models" from a database. To display the models dressed up with textures, STA wraps various graphic textures and images around those models.
There are three levels of textures STA can use:

- High-Resolution Textures (known as "hires" textures),
- Medium Resolution Textures (known as "medres" textures)
- Low Resolution Textures (known as "lores" textures).

The user can find those graphic files in the STA data folder, inside a sub-folder named "textures". The program model patterns can be found in a STA folder named "models".

6.4 Obtaining more from participants in 3D view

More options can be obtained by right-clicking on a given participant. The figure 6.3 shows an example of right-clicking on the participant "Saturn". This new appearing pull down menu provides the user with some similar features as the controls area but new ones are also available:

- The user can bring up the moons of a given planet ("Moons").
- The user can visualise reference systems centred on the participant concerned (very useful in the case of spacecraft attitude).
- The user can "Mark" the participant with a given mark.

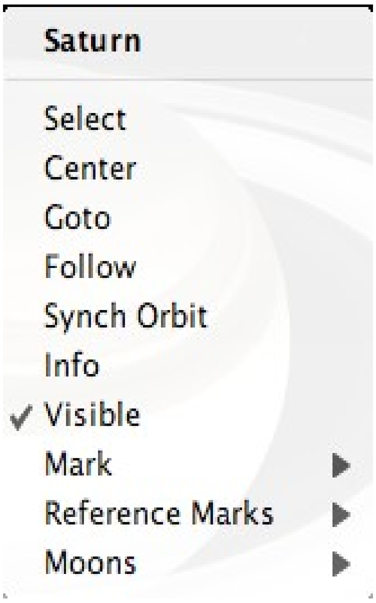

Figure 6.3: Right click on "Saturn"

7. Entry

This chapter describes all the functionalities of STA Entry (some times called Re-Entry) Module as an integrated STA module; as due to its integration in STA some features of the module have been modified, this document aims to help the user simulating re-entry trajectories.

The procedure that must be followed by the user when trying to simulate a re-entry trajectory is the following:

1. Open a new scenario;
2. Add a satellite/re-entry vehicle to the scenario by dragging and dropping the correspondent icon to the scenario tree item;
3. Add a re-entry trajectory to the satellite/re-entry vehicle by dragging the correspondent icon to the trajectory plan item;
4. Supply the GUI with the proper parameters;
5. Click the propagate button in Calculate menu.

Further comments and recommendations should be made about point 3, i.e. the inputs that the user must specify in the GUI.

In the first instance, inputs in the re-entry GUI are not the only values that intervene in the trajectory simulation; the user should also specify the following values in the (aerodynamic and physical) properties GUI:

- Physical/geometrical properties GUI:
  - S/C Mass value : is the vehicle mass value; a default value of 1000 kg is already set in the S/C Mass box. A mass equal to zero will generate an error message when propagating;
  - Radius base : a radius base equal to zero will generate an error message when propagating
  - Radius nose : this value should be set when the user wants to perform a thermal analysis of the re-entry trajectory; a radius nose equal to zero will not generate an error message and will not influence the trajectory.
- Aerodynamic/thermal properties GUI:
  - CD coefficients: the user shall write the name of the file that contains the vehicle CD coefficients (without considering the parachute if existing); the user can also find the file in the system by clicking on the button “Load file”. The file name should not include the path. If the user wants to load an external file, he shall include that file in the STA folder *\sta-data\data\aerodynamics*.

    A Cd coefficients file name not existing (i.e. not in the *\sta-data\data\aerodynamics* folder) or an empty case will generate an error message when propagating.

    The Cd coefficients file should imperatively be a .stad file with the following structure:

First column	Second column
Altitude [km]	cd coefficients [-]

The following is an example:

```
1     1.375
2     1.375
3     1.375
5     1.375
7     1.375
10    1.375
15    1.375
20    1.375
```

- G load limit: the user shall insert the maximum value of the G Load; as the simulation will stop when this value is reached, the user shall take into account the possible influence of a low value on the trajectory propagation. A G load limit value equal to zero will generate an error message.
- SP heat rate limit : the user shall insert the maximum value of the SP heat rate; as the simulation will stop when this value is reached, the user shall take into account the possible influence of a low value on the trajectory propagation. A SP heat rate equal to zero will generate an error message.
- CD parachute coefficients: the user shall write the name of the file that contains the vehicle CD only parachute coefficients. The user can also find the file in the system by clicking on the button "Load file". The file name should not include the path. If the user wants to load an external file, he shall include that file in the STA folder *\sta-data\data\aerodynamics*.

  A Cd parachute coefficients file name not existing (i.e. not in the *\sta-data\data\aerodynamics* folder) or an empty case will generate an error message when propagating.

  The Cd parachute coefficients file should imperatively be a .stad file with the same structure of the Cd coefficients file.

  Even if the vehicle is not supplied with a parachute, the user shall specify a file name for the Cd parachute coefficients (in order to avoid an error message); to nullify its influence on the propagation it is enough to set the parachute surface value and/or the deployment Mach number value equal to zero.
- Parachute surface: the user shall insert the parachute surface value; a value equal to zero will not generate an error message.
- Deployment Mach number: the user shall insert the parachute deployment Mach number value; a value equal to zero will not generate an error message.

As regards with the re-entry GUI, the values the user shall insert are the following:

- Simulation Tab (see Figure 1):

- Starting epoch: the user shall specify the initial time of the simulation; default time is J2000 reference time.
- Coordinate system type : the user shall choose between 3 types of coordinates systems; the initial orbital state window will automatically change accordingly to his choice. The default type is keplerian coordinates;
- Initial orbital state: the user shall specify the initial state of the spacecraft; depending on the inserted values the trajectory could be a re-entry trajectory or the simulation could stop immediately or after some time.

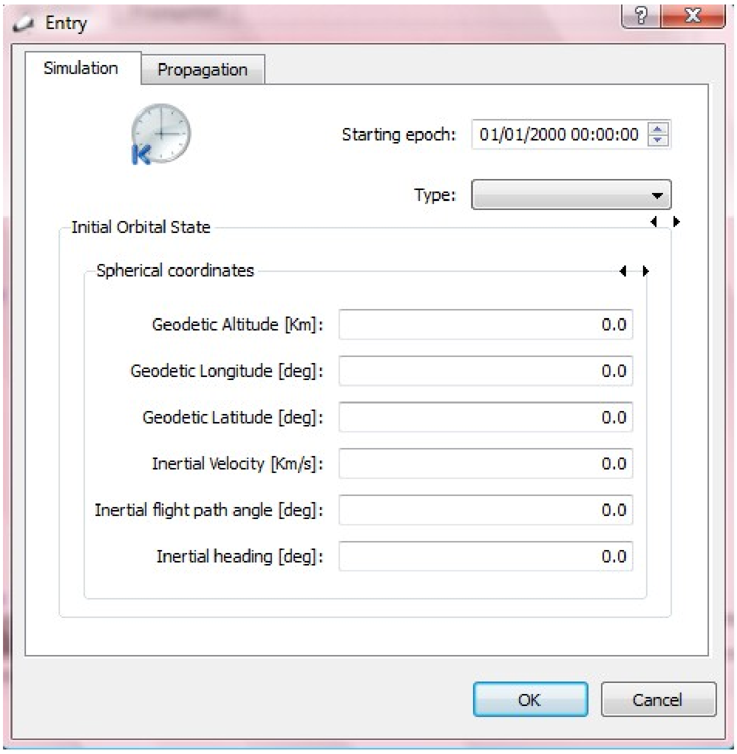

Figure 7.1: Simulation tab on the Entry GUI

- Propagation tab (see Figure 2):
  - Central body: the user shall select the central body between Earth and Mars; default body is Earth.
  - Atmosphere Model: the user shall select from the list the atmosphere model he intends to use; the default model is gram-99.
  - Gravity model, zonal and tesseral boxes: these inputs are not active yet (they will be after the perturbations layer development)
  - Integrator type: this mode is not active (it will be after the integrators box development)
  - Integration step: the user shall select the integration step value; a value equal to zero will generate a message error.

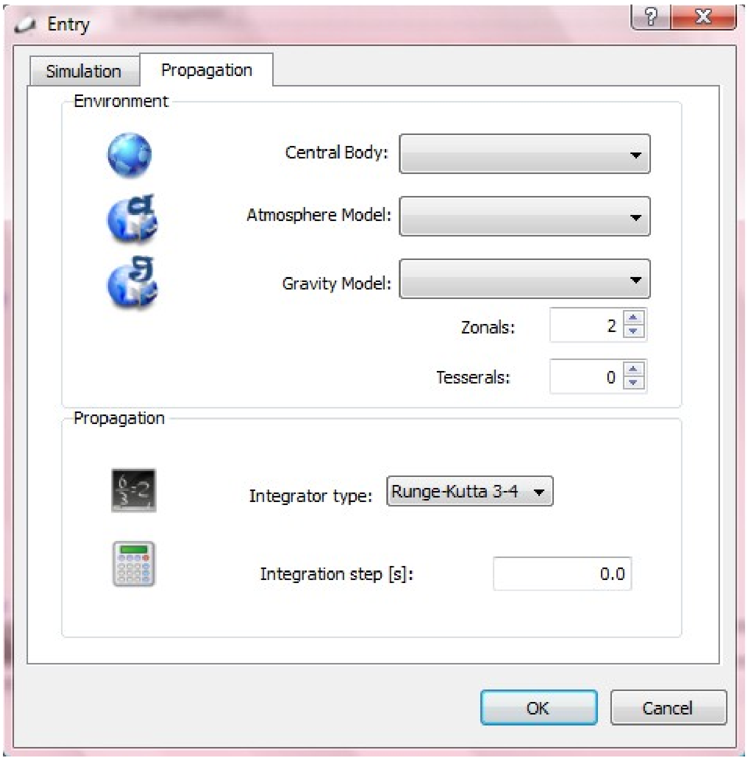

Figure 7.2: Propagation tab on the Entry GUI

8. System's Engineering

This module allows the user to conduct a preliminary system engineering analysis out of a given satellite participant.

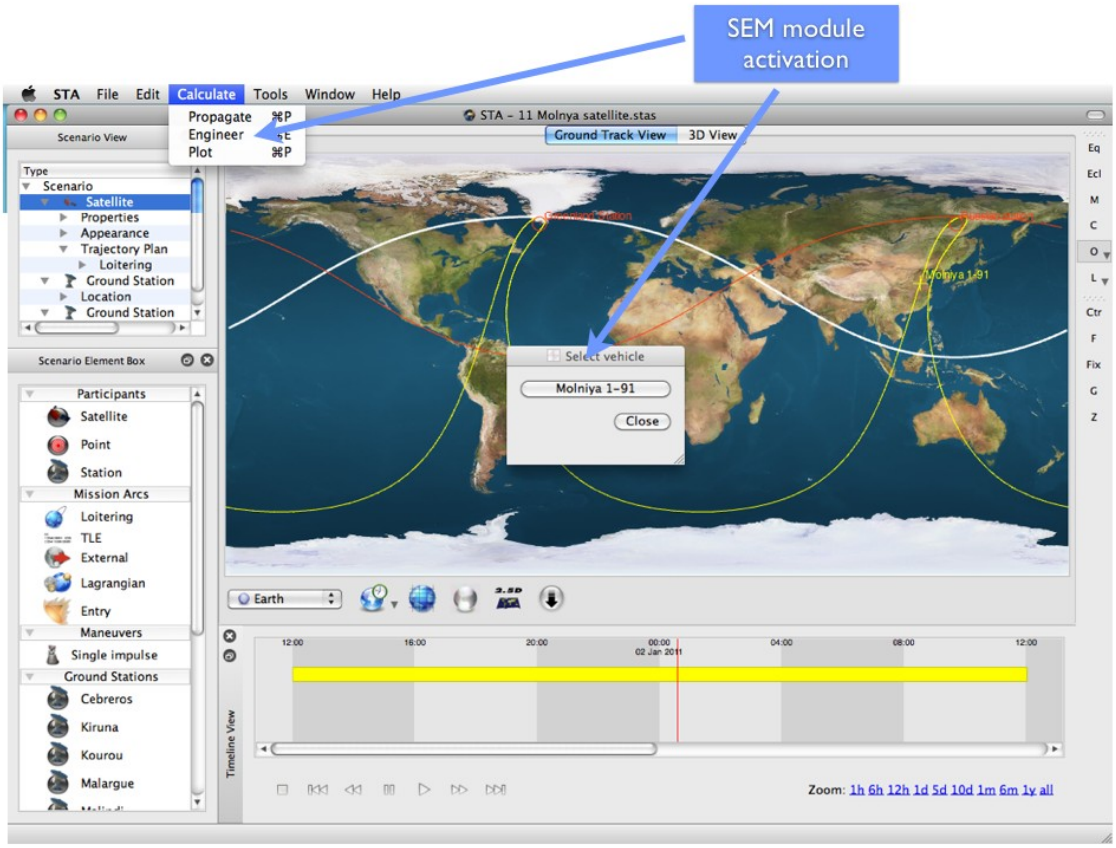

Figure 8.1: Activation of the SEM module

The procedure that must be followed by the user when trying to perform a system engineering analysis is the following:

1. Open a new scenario;
2. Add one or more satellite to the scenario by dragging and dropping the correspondent icon to the scenario tree item;
3. Add, if wanted, some inputs to the satellite properties GUI;
4. Click the system engineering button in Calculate menu.
5. Choose in the list in the new dialog window the satellite for whom the system engineering analysis will be performed.

The inputs added by the user in satellites properties GUI will be recorded and displayed by the SEM GUI when it will be opened. The user will have also the possibility to change them in the SEM GUI.

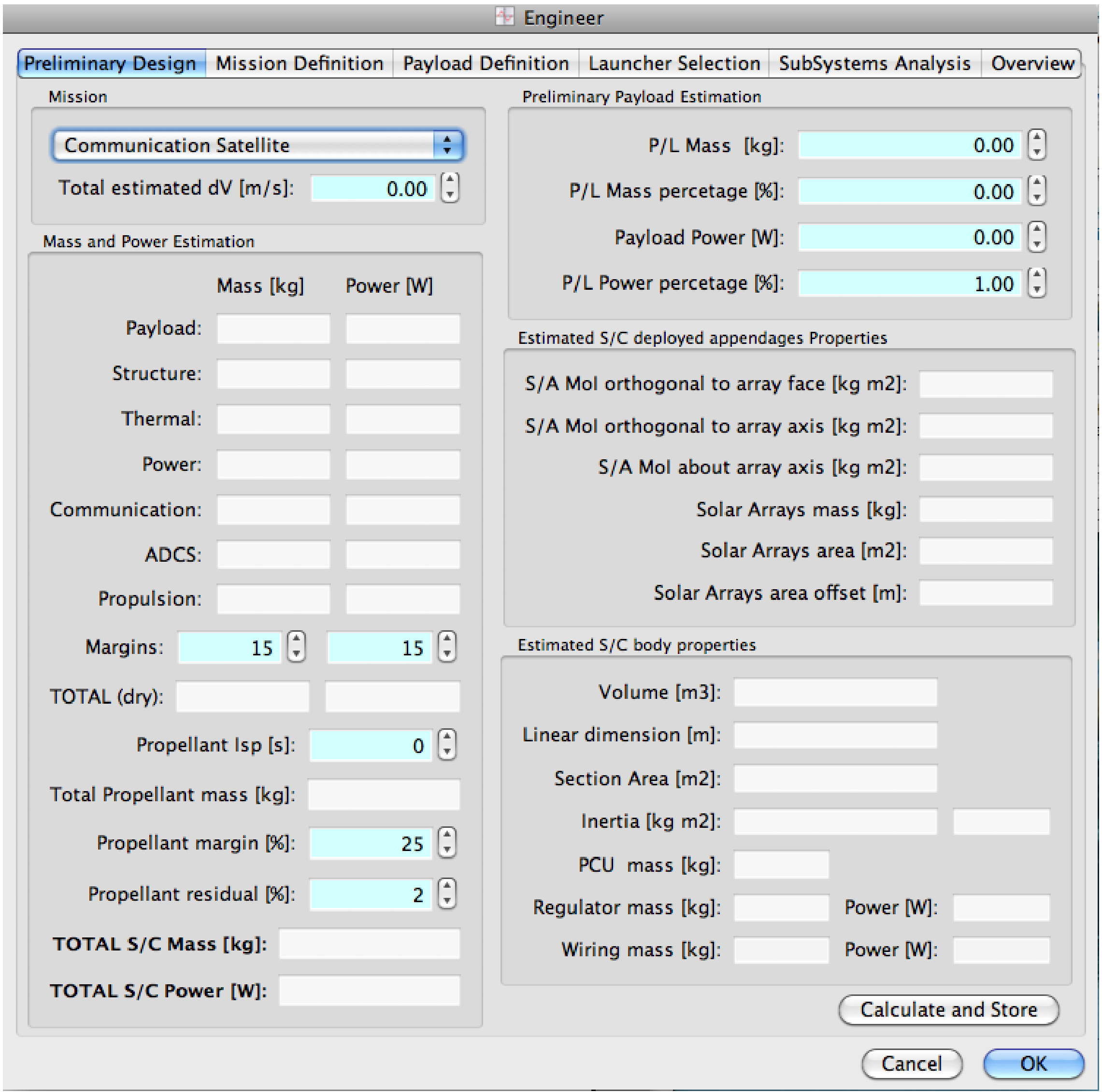

Figure 8.2: Preliminary design tab of the SEM module

The first tab of the SEM module is used to gather data of the given satellite from the scenario. The user needs to input which kind of satellite is the one being pre-designed. Hence, STA will automatically fill out the most common figure of merit of the given satellite.

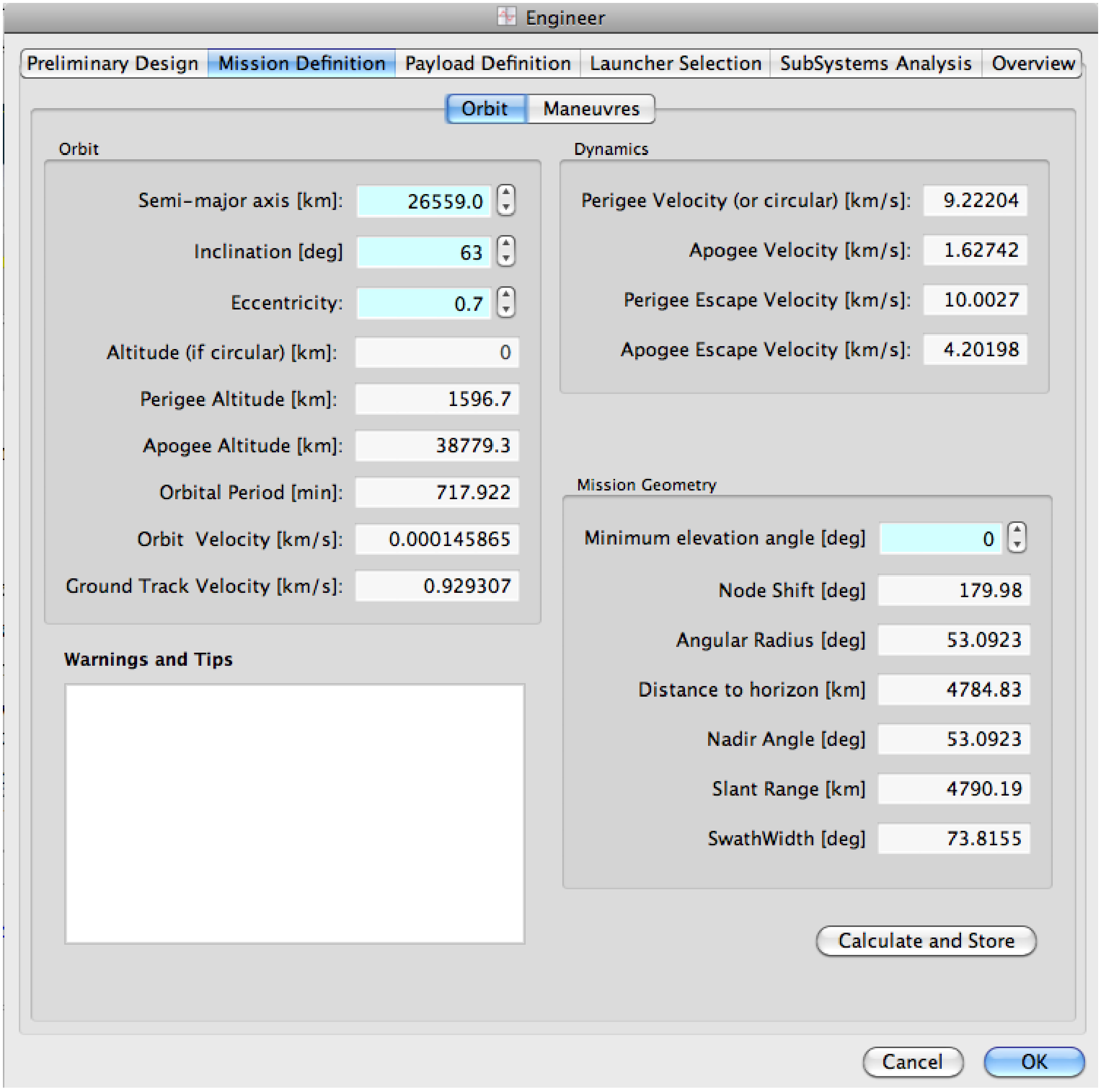

Figure 8.3: Mission Definition tab of the SEM GUI

The second tab is the definition of the mission. In this tab, the user can obtain the most common orbital parameters and relation of the trajectory of the satellite under design.

It contains also the mission geometry and the mission dynamics.

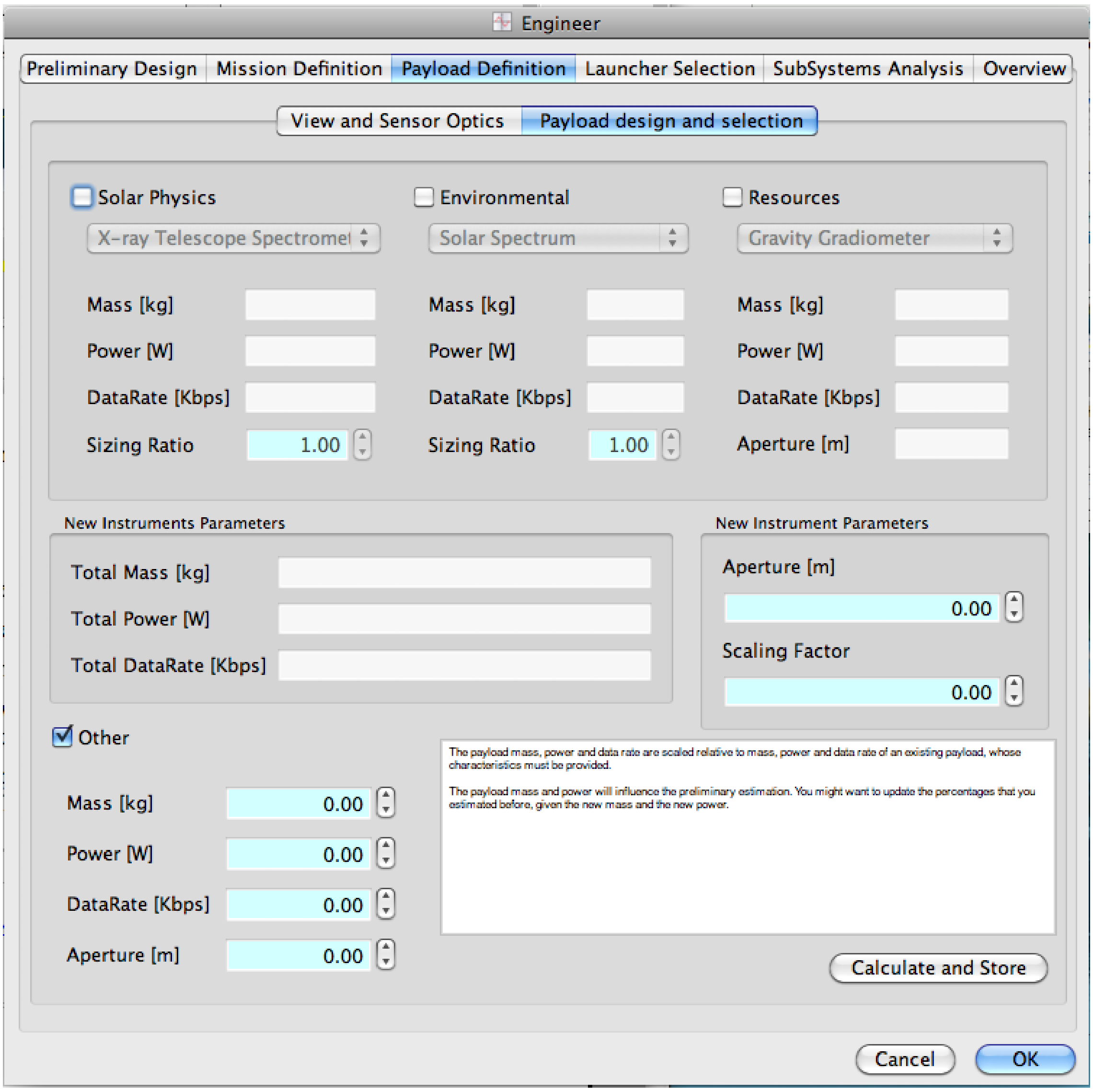

Figure 8.4: Payload definition tab of the SEM module

The payload definition tab allows the user to establish a preliminary design of the payload of the satellite. The user s asked to fill in which kind of payload mission the satellite carries. Based on this input, STA will provide the most common design parameters for the desired payload.

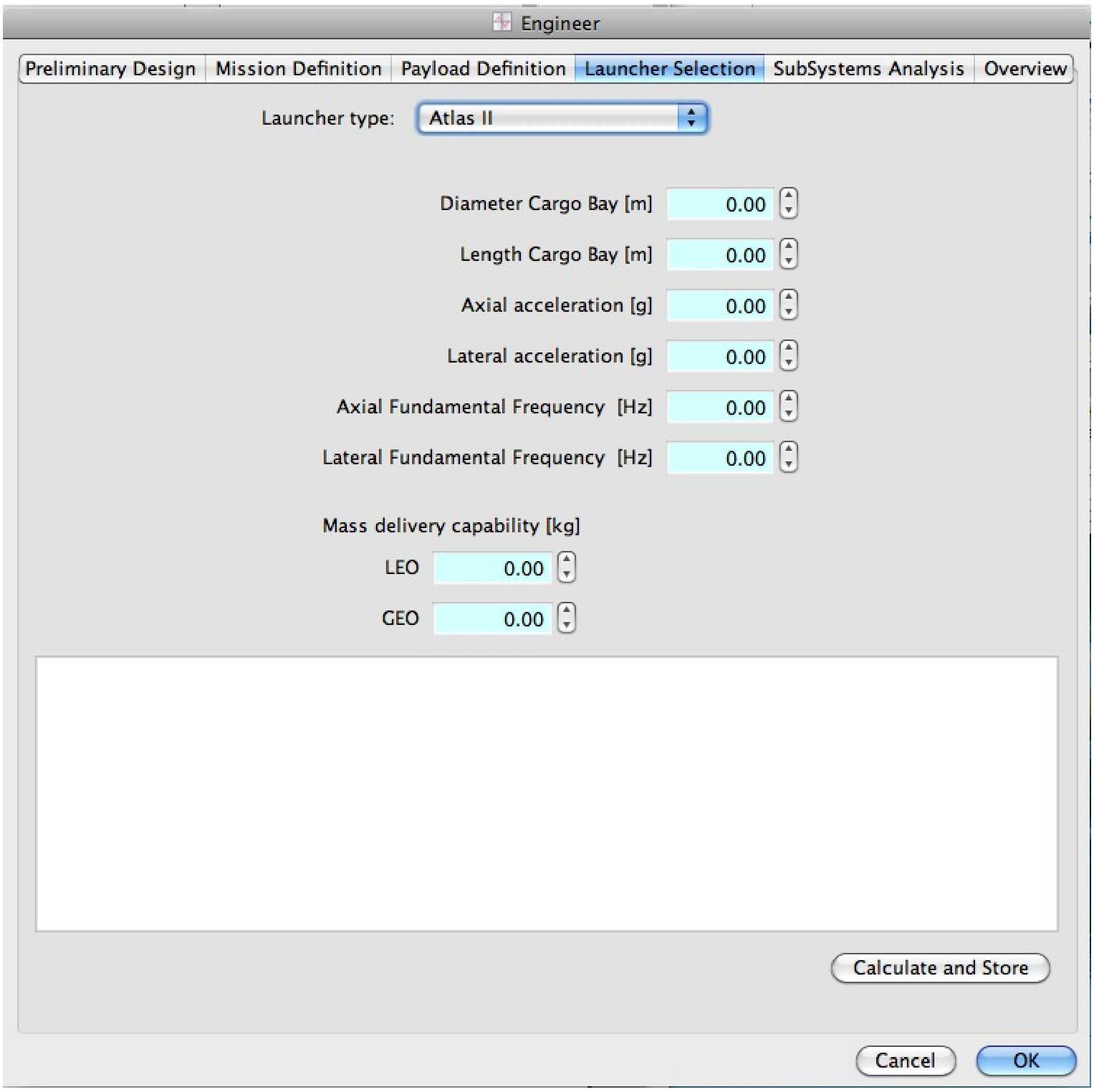

Figure 8.5: Launcher's selection tab of the SEM GUI

The launcher's selection tab allows the user to show case a possible good launcher based on geometrical and structural considerations. STA Sem includes as well restrictions like vibration and mass payload capacity of the chosen launcher.

The tab of the Subsystem Analysis has not been implemented yet in this version of STA. It will be implemented for the version 3.0 code-named CAMBRIAN.

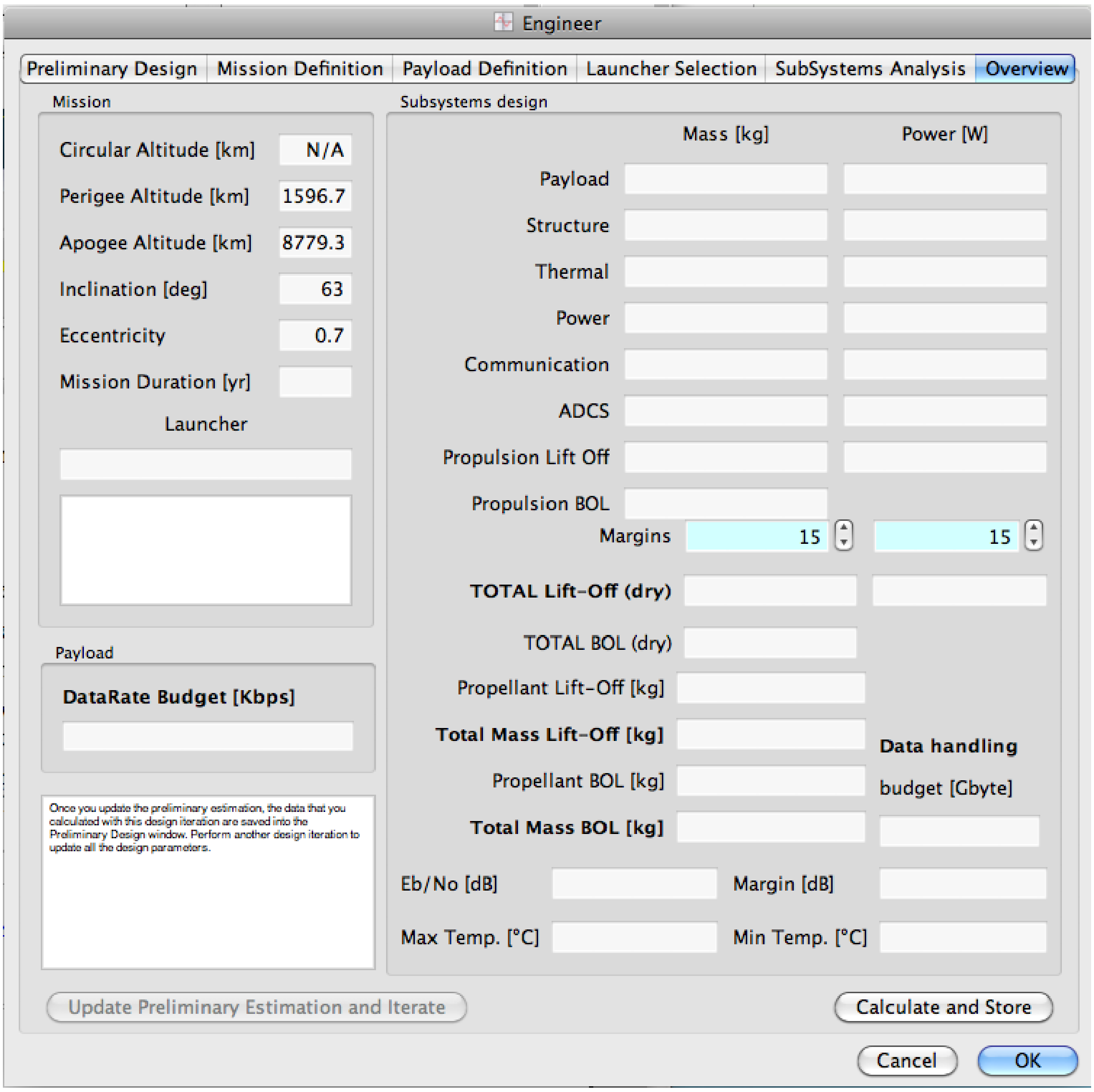

Figure 8.6: Overview tab of the SEM GUI

The last tab provides an overview of the preliminary design of the satellite concerned. It provides the main parameters of the platform as well as for the payload.

9. 3RD-BODY MODULE (3BM)

In this chapter, an overview of this module will be given in order to let the user understand the potential and possible applications for this module. Therefore a general explanation of the 3rd Body problem (3BP) will be given as well as an introduction to the main features of the 3BM.

9.1 3-BODY PROBLEM

In this paragraph, a general explanation is given of what the 3BP is and for which applications it can be used.

The most used model for trajectories design is the 2-body problem, where a satellite, with negligible mass, is gravitating around a central body. In order to improve this simple model, usually some perturbations are considered (drag, effects due to gravity field asymmetry, third body effect, radiation pressure, etc.). Amongst them, for interplanetary trajectories, the third body effect becomes very relevant. This perturbation is determined by the gravity field of another main body disturbing the motion around the first body.

In this way the third body is considered just as a perturbation, while its gravity field creates a complete different environment for the satellite.

Therefore the idea is to include this third body directly within the model in order to use the advantages given by this adjunctive mass. This model (called 3-body problem) becomes very important when a S/C moves far away its main body, actually for interplanetary trajectories.

The deep space can be considered as divided in spheres of influence. In each of them the centre is occupied by a main body and the main force acting on a S/C is given by its gravity field. As a first approximation, interplanetary trajectories can be computed by using the 2-body model, switching the central body when transiting from one sphere to another. Typically, spheres are located within other spheres and frequently the effect of the outer spheres main body is not negligible at all. It’s the case of the Earth-Moon system, where the effect of the Sun is quite relevant.

Therefore this module wants to deal with this “advanced” model, in order to improve designs of those missions interested by this problem. For this reason this module allows the user to propagate a trajectory in the 3-body problem.

Besides the use of such a model brings new fancy features, not present in the 2-body problem. The most evident is the availability of new trajectories. Amongst them the Halo orbits are the most attractive for space missions, they are quasi-stable orbits around the Lagrangian points. Actually if a S/C is sent to one of them it could remain there for many orbits requiring just very small correction manoeuvres to keep the initial orbit. It’s clear that applications for these orbits are numerous and for this reason in the last few years many missions focused on these trajectories, have been planned and already started.

Moreover these orbits have another interesting property. If the S/C is a bit perturbed from the reference orbit, it can escape from the Halo orbit and can follow a trajectory guiding the S/C towards the centre of the system (occupied by the most massive body) or outward the system. All these trajectories can be seen as a stream of trajectories (because from each point of the Halo orbit departs a different trajectory for both the directions), and they are called manifolds. On the other way around it could be possible to follow them in the opposite way, from outside the system towards the orbit itself.

It's intuitive that it could be possible to use this manifolds trajectories in order to get the Halo orbits from a parking orbit around a main body or to jump from one system to another one. For this reason the 3BM allows the user to compute these particular orbits and to optimize the transfer between a parking orbit around a main body and the Halo orbit desired, minimizing the TOF or the propellant consumption or both of them.

Finally the 3BM allows the user to design a transfer trajectory able to connect a parking orbit around the Earth and another one around the Moon by patching two different 3-body systems: Sun-Earth and Earth-Moon. In this way, manifolds are used as a highway able to link the Earth and the Moon.

9.2 STA 3BM inputs and functionalities

Four different main functionalities (modes) have been developed in this module. A brief description for each mode is here reported [A6]:

1. *Trajectory propagator* - for the simple simulation of trajectories, known the initial conditions (the three components of position and velocity) given by the user in the GUI. Moreover the user will be able to select what kinds of perturbations (solar radiation pressure, J_2, third bodies) should be included in the integration process.
2. *Halo orbit (& manifolds) computation* - for the design of Halo orbits and their relative manifolds systems. The user will be able to construct the desired Halo orbit selecting proper initial conditions. Moreover the user will be able to design also manifolds systems departing from the designed Halo orbit
3. *Transfer orbit optimization* - for the design of a complete mission from a main body of the 3body-system to a designed (by using the previous mode) Halo orbit. A GO process will be used in order to optimize the mission (TOF and/or total ΔV).
4. *Earth-Moon transfer orbit* - for the design of a transfer orbit from the Earth to the Moon, by using a heteroclinic connection between the L2 points of the Earth-Sun and the Earth-Moon systems.

Besides there is a common part for all the modes that has to be filled by the user in order to let the computations start.

9.3 General overview of the 3BM

This chapter describes all the functionalities of the STA 3BM and how the user can input the required model data and select the different options.

All the inputs and the functionalities of the 3BM will be discussed and explained by following the division in modes, because it's the same one used to design the GUI. Since the first contact for the user is actually the GUI itself, it makes sense to explain the 3BM functionalities following that structure.

Besides there is a common part for all the modes that has to be filled by the user in order to let the computations start. This part sets the simulation environment and the integration properties. Before explaining this common part and then all the different modes, a general overview on the structure of the 3BM will be given.

The 3BM GUI is divided in two different windows, the first one (Figure 9.1) is the main GUI, that allows the user to set the environment of the problem and to use the first mode. Then clicking on the button "advanced" it's possible to have access to the advanced GUI, where the other three modes are available.

It's worth emphasizing that **the choice of the mode in use is done indirectly by the user**. It depends on which GUI has been activated and on the last window used by the user. It means if the user didn't open at all the advanced GUI, the mode in use would be the first one.

On the other hand, if the advanced GUI has been opened, the mode in use is not the first one anymore, unless the user will close and reopen the main GUI.

How the user has to select the other three modes (mode 2, 3 and 4) will be shown in the apposite sections.

By the way, in order to launch the 3BM in the main GUI of STA, the user has to open a new scenario and drag on it a satellite. Hereafter the user can drag the 3BM on the trajectory plan. In this way the 3BM is ready to be launched and used.

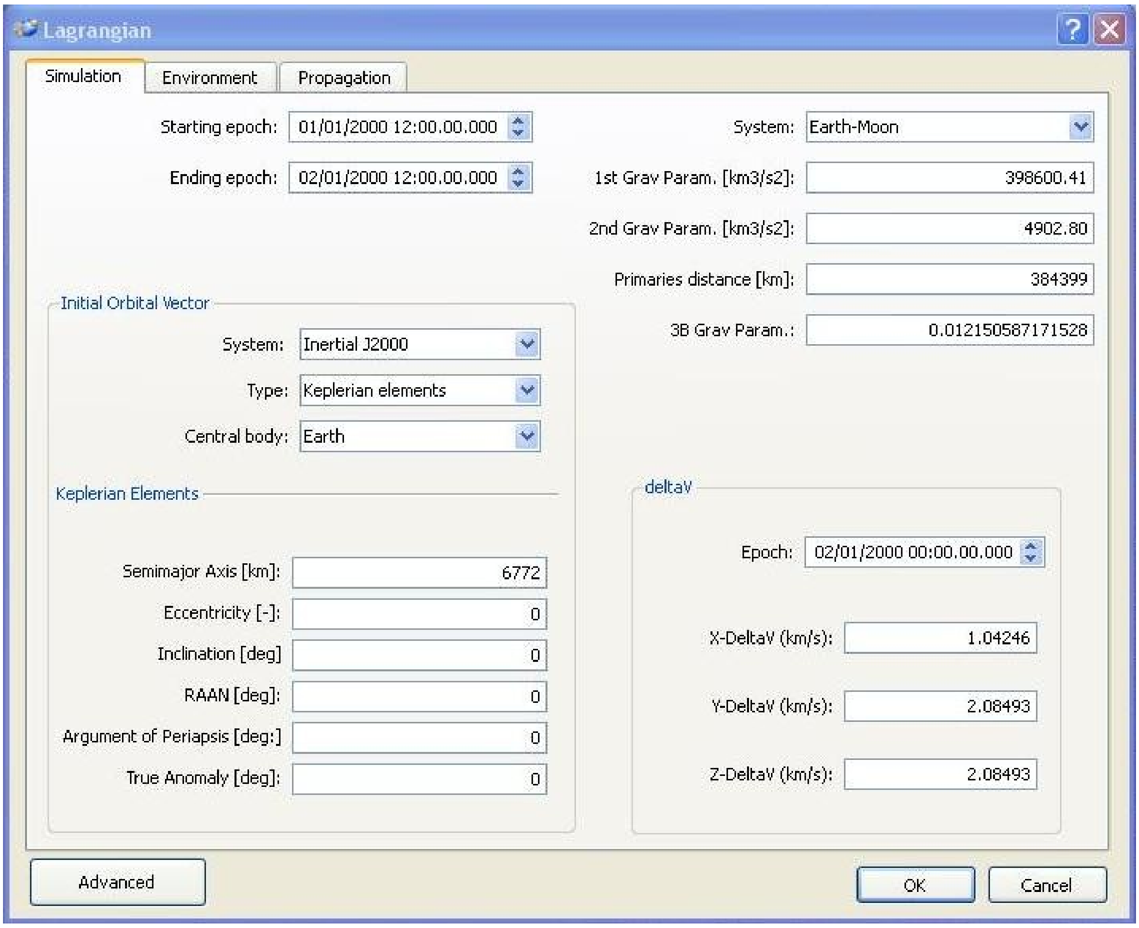

Figure 9.1 3BM main GUI

9.4 Common features of the four modes, 3BM GUI

In this paragraph, the 3BM GUI will be analyzed, considering only the common features that should be set in any case for all the modes.

The 3BM GUI is divided in three different sub-windows:

1. simulation
2. environment
3. propagation.

In the **Simulation** part, there are all the settings related to the 3-body system in use and to mode 1. Therefore this window will be partially discussed now (for the part about the general settings of the 3BM, shown in Figure 9.2) and partially in the next paragraph (for the features related to mode 1).

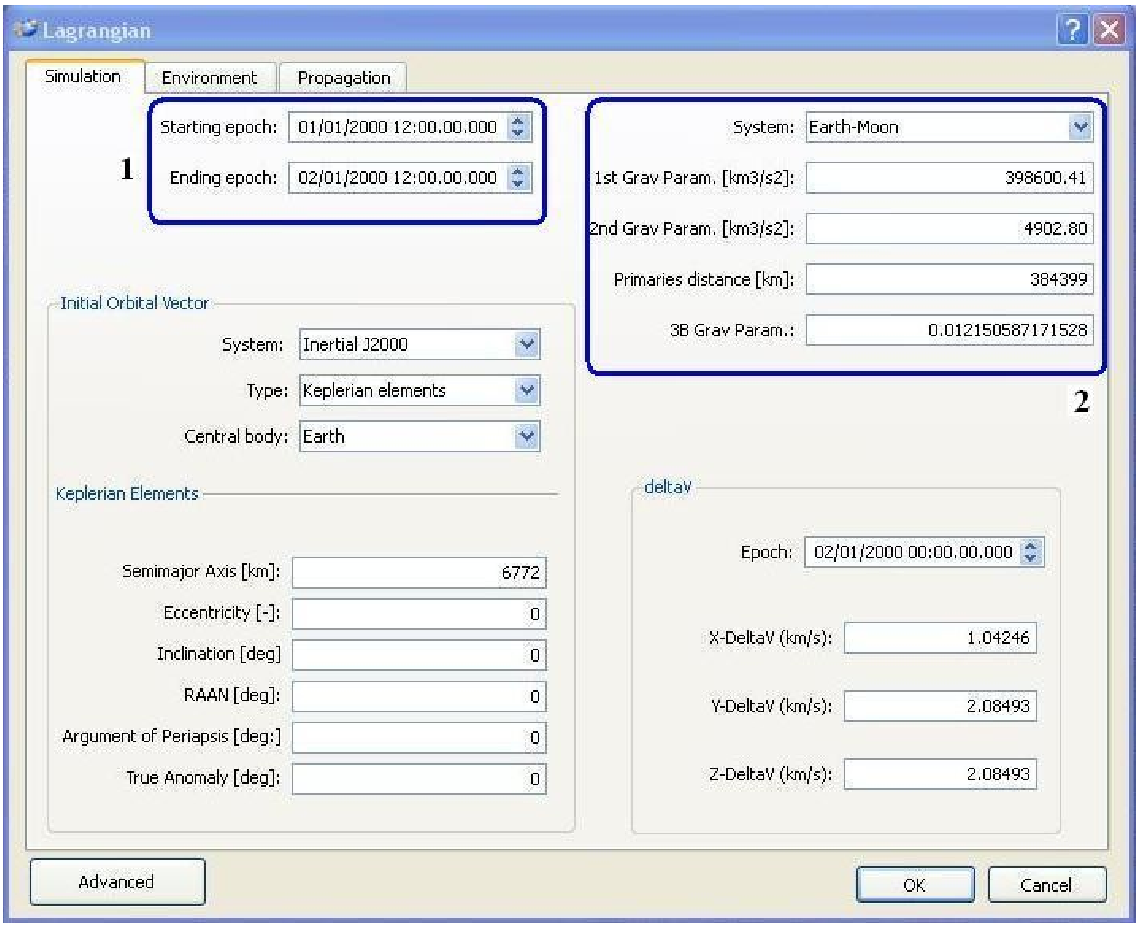

Figure 9.2 3BM main GUI: common features in Simulation window

In panel 1, the simulation timeline is defined. Here the user has to specify the starting and the final epochs of the simulation. It regards the propagator mode, but also the other ones. It means, for instance, that a Halo orbit, designed by using mode 2, is plotted in the epoch defined in this panel (starting epoch).

In panel 2, the 3-body system is selected. In the first box the user can select the specific system that wants to work with (as defined in A5). Below all the main parameters are listed: the Gravitational parameters of each body, the distance between them and the resulting 3-body Gravitational parameters. These values can't be changed by the user.

The other panels relate to mode 1, and for this reason they will be discussed in the next paragraph.

Other parameters, needing to be specified for all the different modes, are collected in the page called **Propagation** (see Figure 9.3).

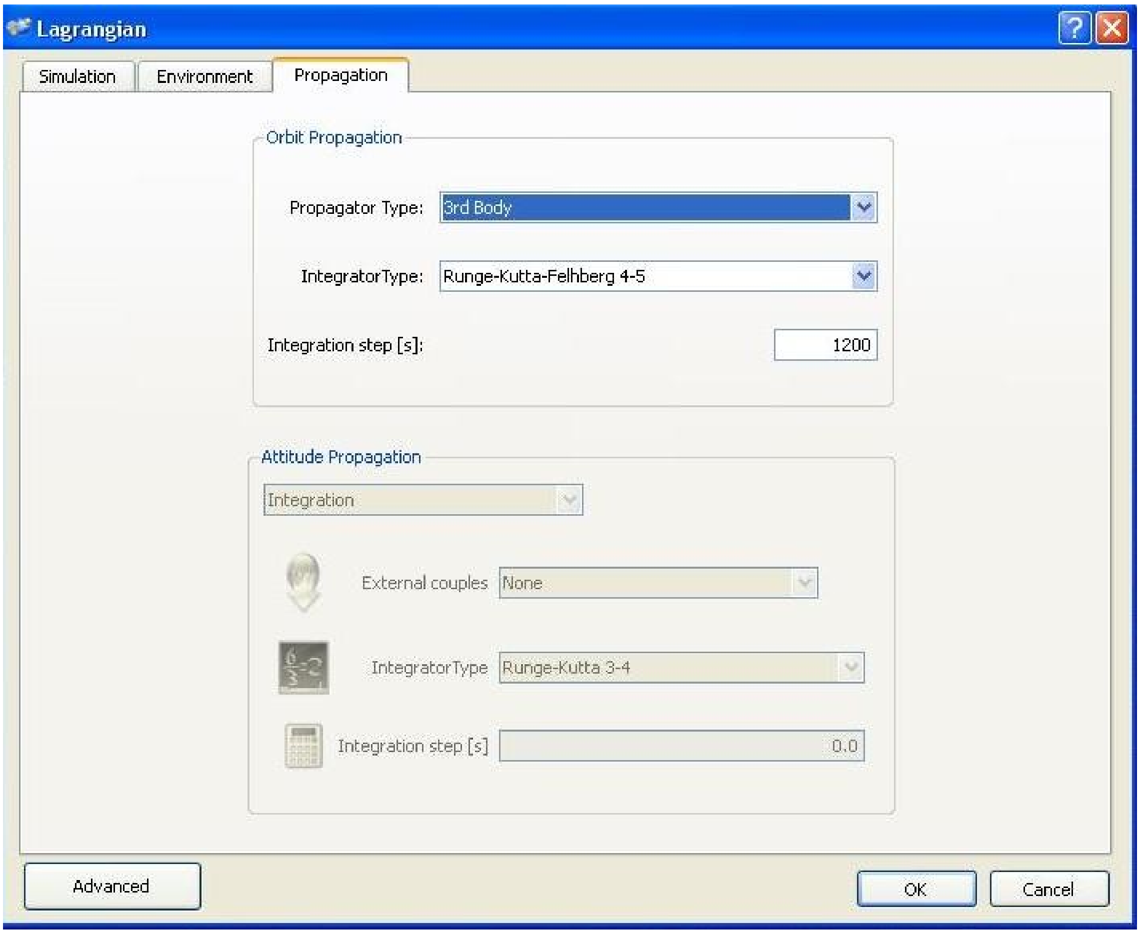

Figure 9.3 3BM main GUI: Propagation window

This register card contains two panels: orbit propagation and attitude propagation. The second one is disabled, because in the module the attitude propagation is not allowed. In the first one the user can select the integrator type (so far only the RKF45 has been implemented) and the integration step (in sec), while the propagator type obviously is the 3rd body.

The E**nvironment** window depends on inputs from another STA module, intended to add propagations in the equations of motions. Development of that module had not been completed by the time the 3BM was released ad this UM written. For this reason the discussion of this part will be tackled once that module will be available.

9.5 FEATURES OF MODE 1

This first mode allows the user to propagate a trajectory within the 3-body model after having specified the initial conditions. By using the previous parts of the GUI, the user has been able to select the system and the integration step to use. Now the initial conditions have to be provided, by using the again the **Simulation** window of the main GUI. In Figure 9.4 and the panels to use are shown.

In the Initial Orbital Vector panel, the user can select from two different sets of initial conditions that can be used to describe the initial state for the 3BM simulation:

1. keplerian elements
2. Cartesian state vector.

In the first case (Figure 9.4) a parking orbit around the main body is specified, and in the other one (Figure 9.5) a state vector, representing the initial position and velocity has to be provided.

In case the keplerian elements are given by the user, the reference frame (RF) adopted is the Ecliptic J2000. This frame is defined by the Earth equator at epoch J2000 (1st January, 2000, 11:58:55.816) as the fundamental plane. The x-axis points along the J2000 vernal equinox, the z-axis is the mean north pole, and the y-axis completes the right-handed system.

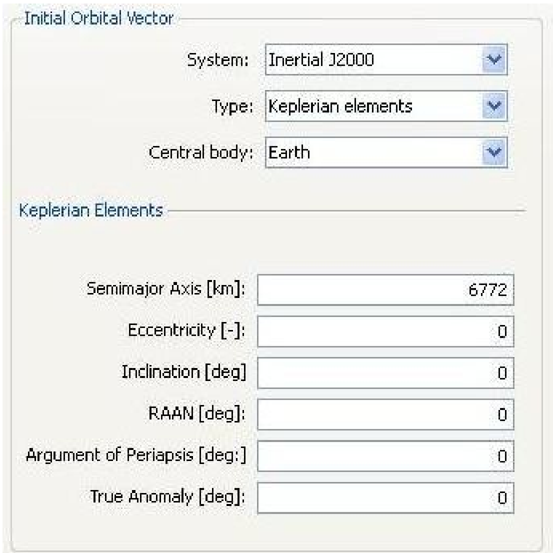

Figure 9.4 3BM main GUI: initial conditions panel for mode 1 (keplerian elements)

In the other case three different reference frames can be used:

1. Ecliptic J2000
2. Rotating normalized RF (shown in Figure 9.5)
3. Rotating RF

The rotating normalized RF is the classical RF used for the 3-body problem. It is rotating around the mass center of the system with a constant angular velocity of 2π/T (where T is the time needed for the second body to complete a revolution around the mass center). This RF can be normalized by using the distance between the primaries, the linear velocity of the second body and T*2*π. Here it’s the difference between the rotating normalized and the rotating RF.

In case a state vector is provided to the integrator as initial conditions, all the components of the position (three) and the velocity (three) have to be provided.

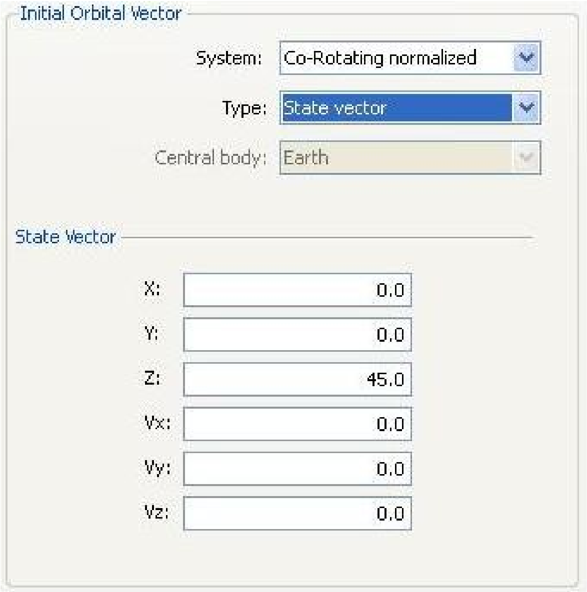

Figure 9.5 3BM main GUI: initial conditions panel for mode 1 (state vector)

The last panel of Figure 9.1 is used to execute a maneuver during the trajectory propagation. Actually during a trajectory can be useful/interesting/required to check how it changes after an impulsive ΔV is given to the S/C.

Therefore the 3BM allows the user to do such a computation. The inputs required are the epoch when the maneuver is executed, and the magnitudes of the ΔV (as shown in Figure 9.6). If in the previous panel the normalized system has been chosen, also in this case normalized magnitudes have to be provided.

Figure 9.6 3BM main GUI: deltaV panel for mode 1

9.6 ADVANCED 3BM GUI

As already explained, the 3BM main GUI has to be used to insert the common parameters for all the modes and to propagate a general trajectory in the 3-body problem (mode 1).

Besides, in order to use the other features of the 3BM, the user has to make use of the advanced GUI that is divided in four different windows:

1. Halo orbit (mode 2)
2. Manifolds (mode 2)
3. Transfer trajectory (mode 3)
4. Earth-Moon transfer (mode 4)

Each of them represents a different process of the 3BM. For this reason, depending on the actual window opened when the user pushes the button "OK", the 3BM is able to understand which functionality has been requested by the user.

It means if the user will fill all the inputs for all these four different windows, the 3BM will compute only the trajectory(ies) required by the window that is open at that time.

Therefore mode by mode (actually window by window), all the interactions required by the advanced GUI will be discussed in the next paragraphs.

9.7 FEATURES OF MODE 2

This mode aims to compute Halo orbits and trajectory manifolds. The entire procedure for these computations is reported in [A6]. Here only the inputs/outputs flows regarding the GUI will be analysed.

For the computation of Halo orbits, a first guess is required to let the computation starts. Thereto the user can insert directly this first guess by using panel 2 (see Figure 9.7). The initial guess has the following layout:

1. x-position (given by the user)
2. y-position = 0
3. z-position (given by the user)
4. x-velocity = 0
5. y-velocity (given by the user)
6. z-velocity = 0

It could happen that the user doesn't know anything about the Halo orbit. In this case, selecting the check box "research first guess values for the initial conditions", the user can design that orbit just specifying the location (in the L point selection box the user can select the Lagrangian point where the Halo orbit will take place) and the amplitudes (only in the x and z-direction, because for Halo orbits the y one is determined by the x amplitude). Therefore by using the first panel the user can get the first guess required to start the computation.

Once a first guess has been given/computed, the user should select how many positions he wants to know and the maximum computational time for the Halo orbit. Actually that orbit is obtained by using an iterative method that can last for quite a long time, depending on the precision of the first guess. In order to avoid the 3BM to run for so long, the user can select a maximum computational time.

When the computation has been completed, the results are shown in panel 3. The real initial conditions are printed as well as the real amplitudes and the period of such orbit. It could happen the orbit computed is too different from the one required, in this case an error box communicates it to the user.

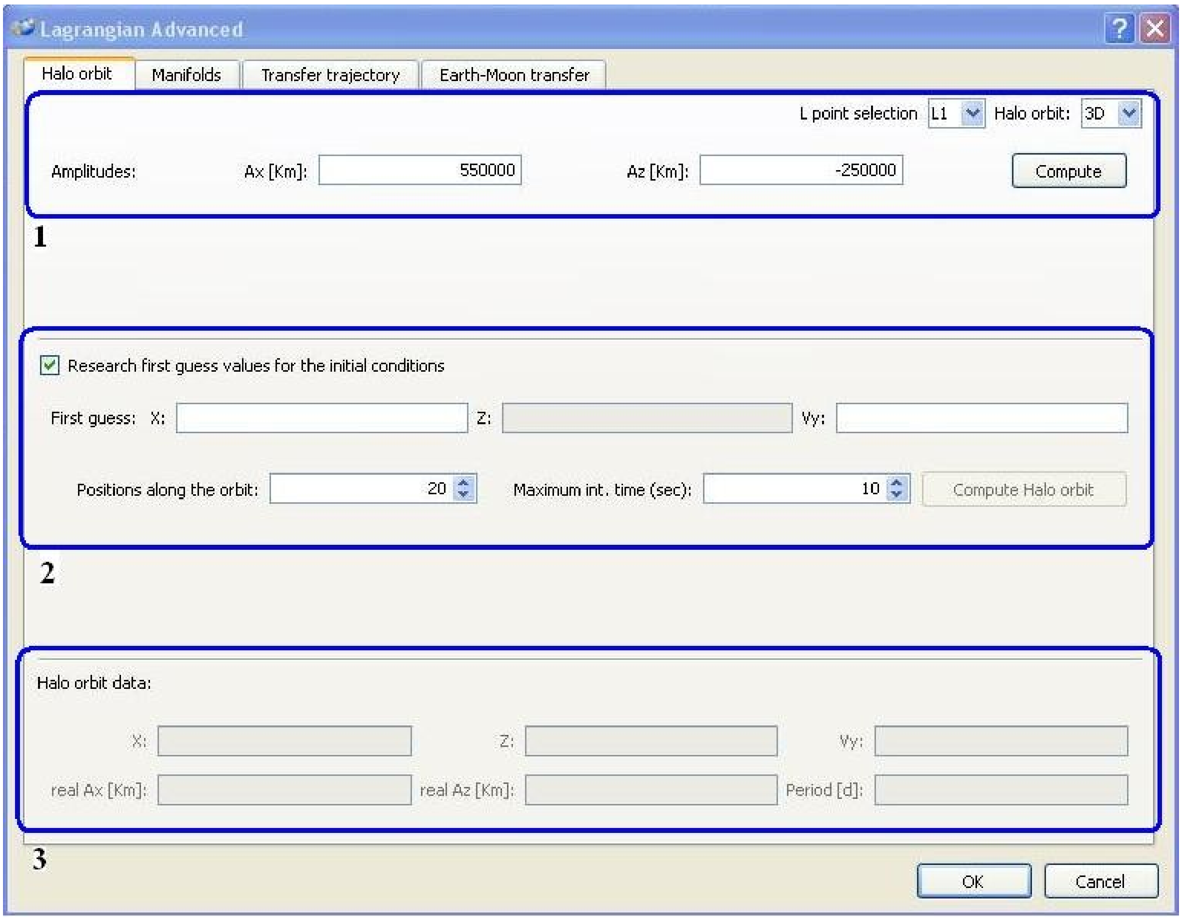

Figure 9.7 3BM advanced GUI: Halo orbit window

The second feature of mode 2 is the computation of trajectory. It's worth emphasizing that in order to use this functionality the user has to compute a Halo orbit by using the previous window. Manifolds trajectories are designed by perturbing a Halo orbit, therefore that one has to be provided to the 3BM.

In the Manifolds window of the advanced GUI (Figure 9.8), the user can select the deviation relative to the initial Halo orbit. Moreover the user has to select which manifolds he wants to compute, checking the apposite checkboxes (first column on the left). Then the final conditions for the manifold branches selected have to be determined. They can be computed until intersecting with a main body or can be propagated specifying a certain time of integration.

The first case is available only for some particular combinations, depending on the geometry of the problem:

1. L1 → right branch
2. L1 → left branch
3. L2 → left branch

In the other cases, propagating till the intersection of a main body makes no sense (for instance on the right side of L2 there are no main bodies)

In case a time of integration is specified (in normalized units as explained in 9.5), the user shall specify also how many positions he wants to know for each trajectory, while the number of trajectories is determined by the number of position known (selected in panel 2 of the Halo orbit window, Figure 9.7)

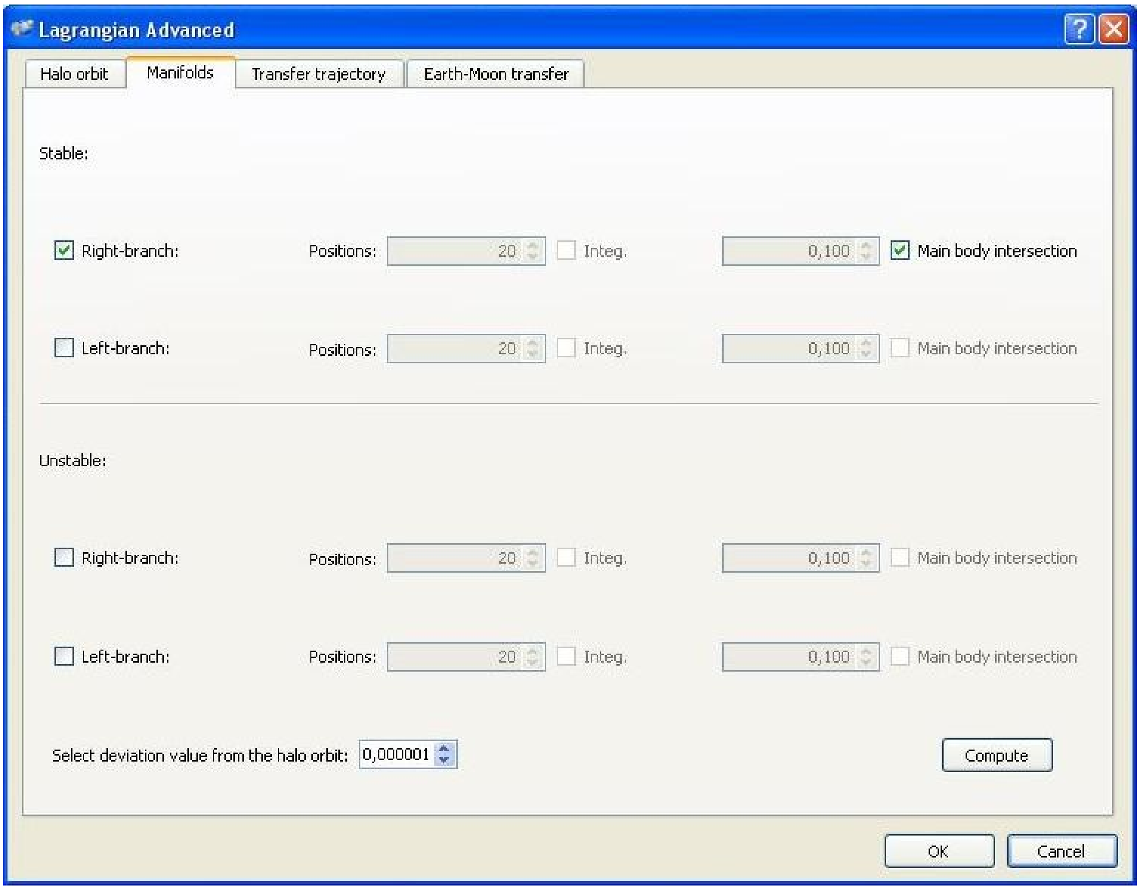

Figure 9.8 3BM advanced GUI: manifold window

9.8 FEATURES OF MODE 3

Mode 3 is intended to find the best connection between a parking orbit around a main body and a Halo orbit. It means that three different sets of inputs are required:

1. parking orbit
2. Halo orbit
3. optimization settings

Actually these three sets represent the three different panels present in the advanced GUI dedicated to this mode (Figure 9.9).

In the first panel a parking orbit has to be specified, by using keplerian elements. Moreover the central body has to be selected between the two bodies of the system selected in the main GUI. Finally a launch window has to be selected. So far the optimization process is able to optimize only the daily launch window of the day selected as the starting epoch (indeed it's possible to find the optimized RAAN), but the 3BM has been created in such a way that setting the optimizer in the correct way allows the optimization of the annually launch window (in order to optimize the relative inclination between the Earth rotational axis and the Halo orbit, that varies over a period of one year).

In the second panel the initial conditions for the Halo orbit have to be provided. It's possible to import them from the previous window of the advanced GUI just selecting the apposite checkbox. Anyway the 3BM runs again the computation of the Halo orbit in order to work with the real Halo orbit. Moreover the user shall specify the deviation from that orbit in order to compute transfer orbits. This value is left to the user, because it depends on the mission requirements. Actually this mode guides the S/C till the perturbed position, therefore the S/C should execute some mid-course manoeuvres in order to get the Halo orbit (not considered in this module).

Finally in the third panel all the settings about the optimization process are collected. The user has to specify if the total ΔV, the ToF or both of them have to be optimized. In any case, also the maximum allowable ToF has to be selected, in order to exclude those solutions not acceptable.

Then the user shall select the number of populations for the optimizer. The size of them is by default equal to 200, but the user can decide how many times the optimizer is launched.

Finally the user can select also what kind of optimizer he wants to use. For the mono-objective problem, the optimizer is by default the NSGA-II (Non-Sorting Genetic Algorithm), because for that problem gave the best performances. For the multi-objective problem, the user can use either the NSGA-II or the DG-MOPSO (Double-Grid Multi-Objective Particle Swarm Optimisation).

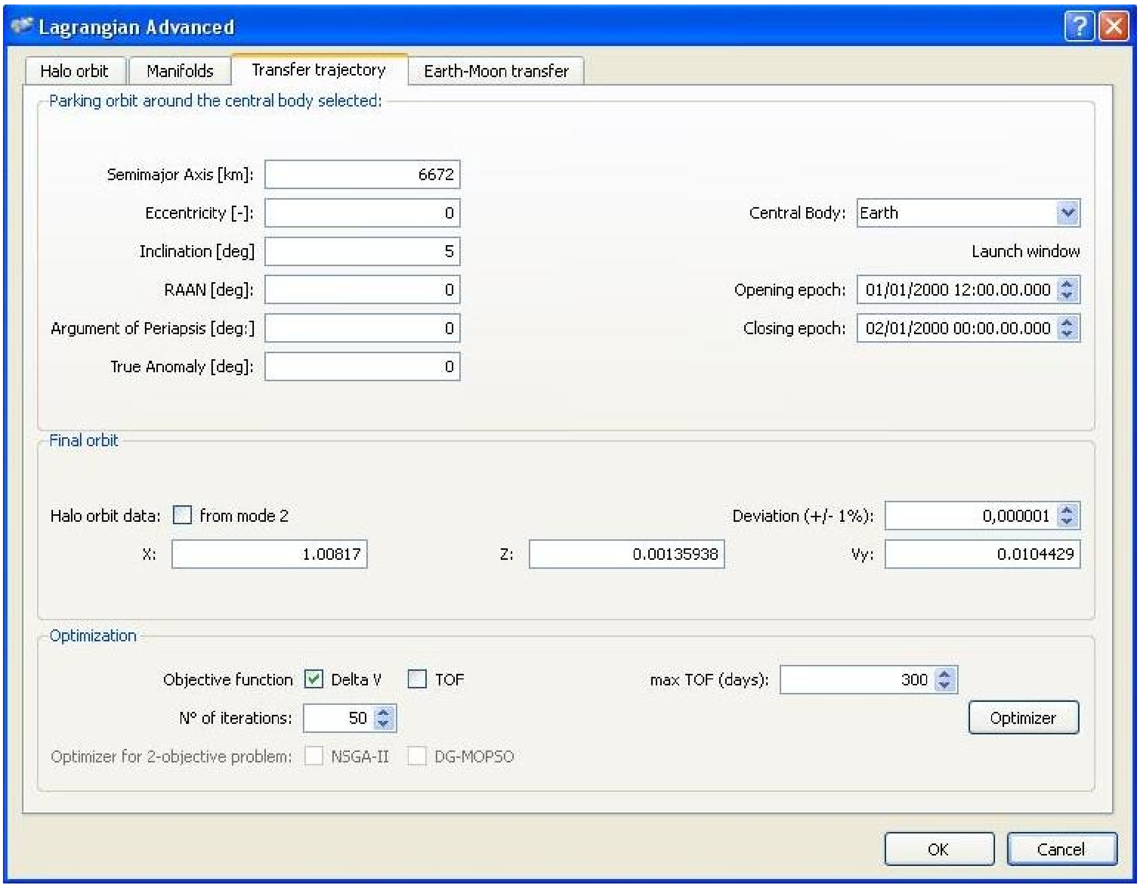

Figure 9.9 3BM advanced GUI: Transfer trajectory window

9.9 FEATURES OF MODE 4

This mode is not yet implemented. It will be implemented in the version 3 of STA code-named CAMBRIAN.

9.10 STA 3BM OBTENTION OF RESULTS

After having explaining all the 3BM GUI structure and indeed having let the user be able to interact with it, some examples of what results the 3BM is able to deliver will be shown in this section. An example for each mode will be given by using some screenshots of the 3BM.

9.11 RESULTS OF MODE 1

For this mode a trajectory around the Earth has been propagated. After some hours an impulse shot has been given to the S/C in order to show how the trajectory is changed.

In the next two figures the inputs given (Figure 9.10) and the trajectory obtained (Figure 9.11) are shown:

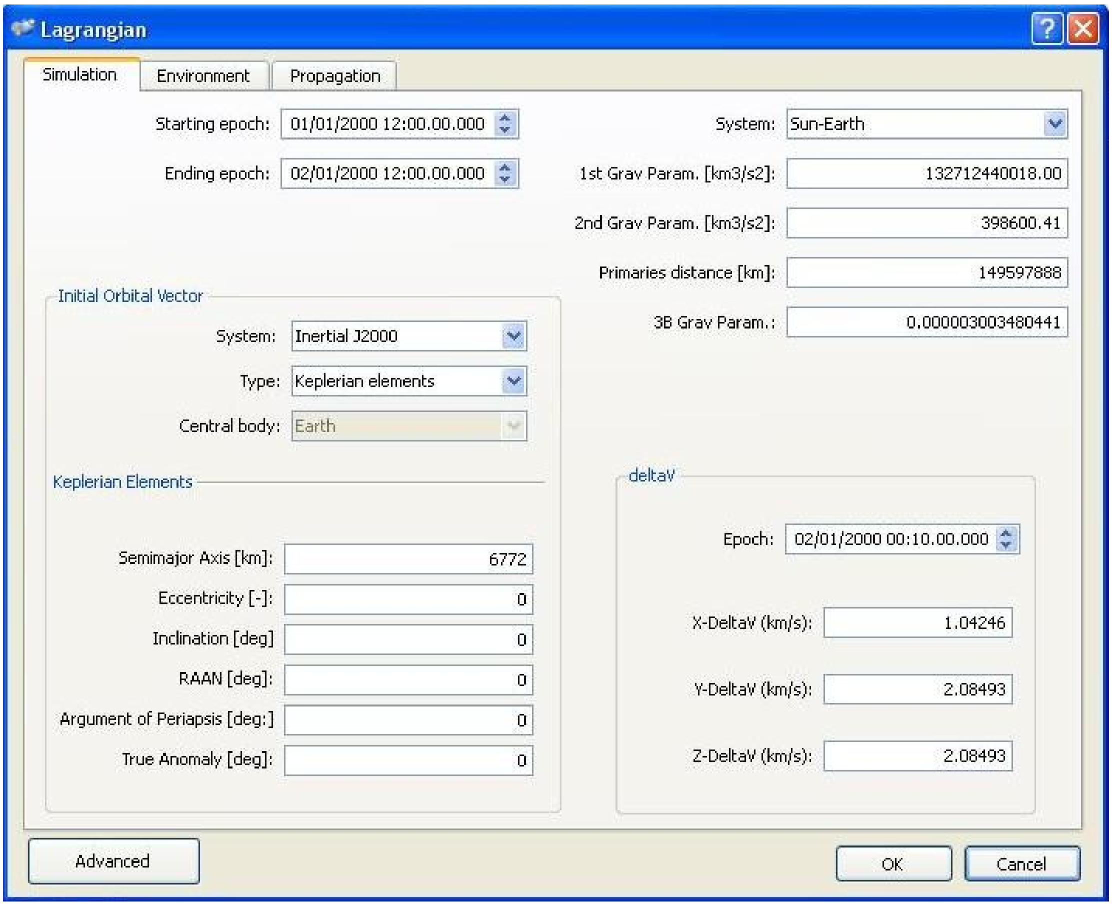

Figure 9.10 GUI screenshot for an example case for Mode 1

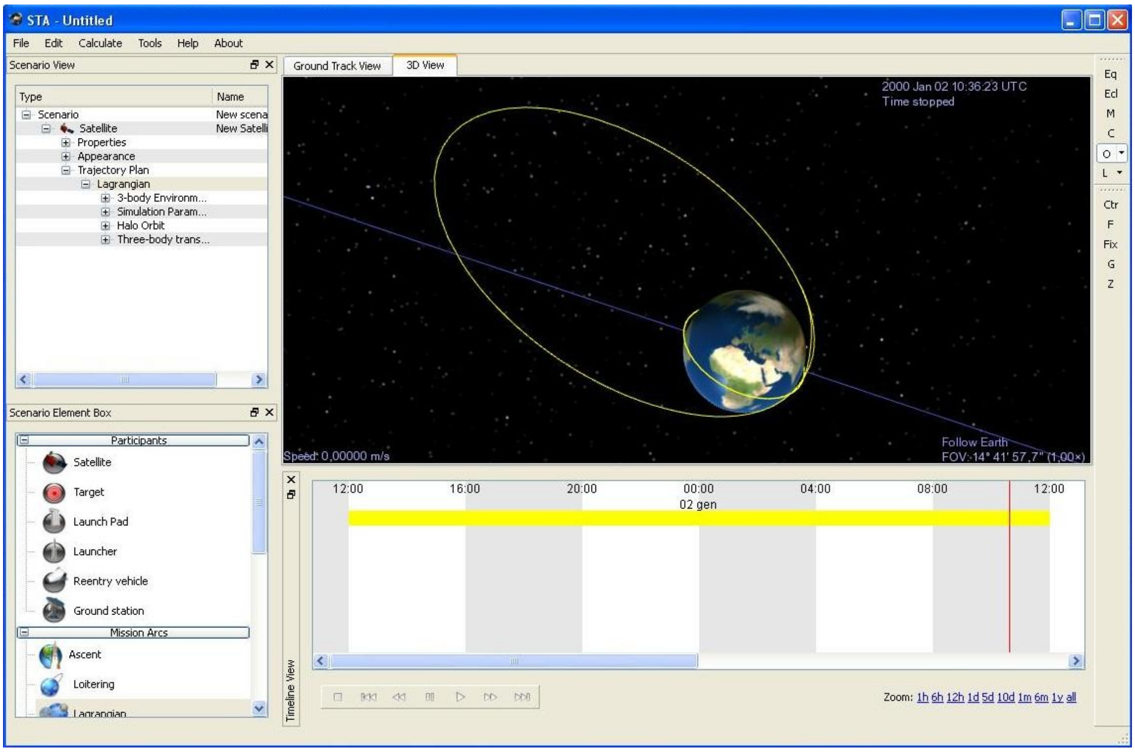

Figure 9.11 Trajectory screenshot for an example case for Mode 1

9.12 Results of mode 2

Two different results are shown here. The first one is about the computation of a Halo orbit around L1 of the Sun-Earth system (Figure 9.13). Hereafter the manifolds trajectories departing from this orbit are shown as well (Figure 9.15). Again the inputs given are reported respectively in Figure 9.12 and Figure 9.14.

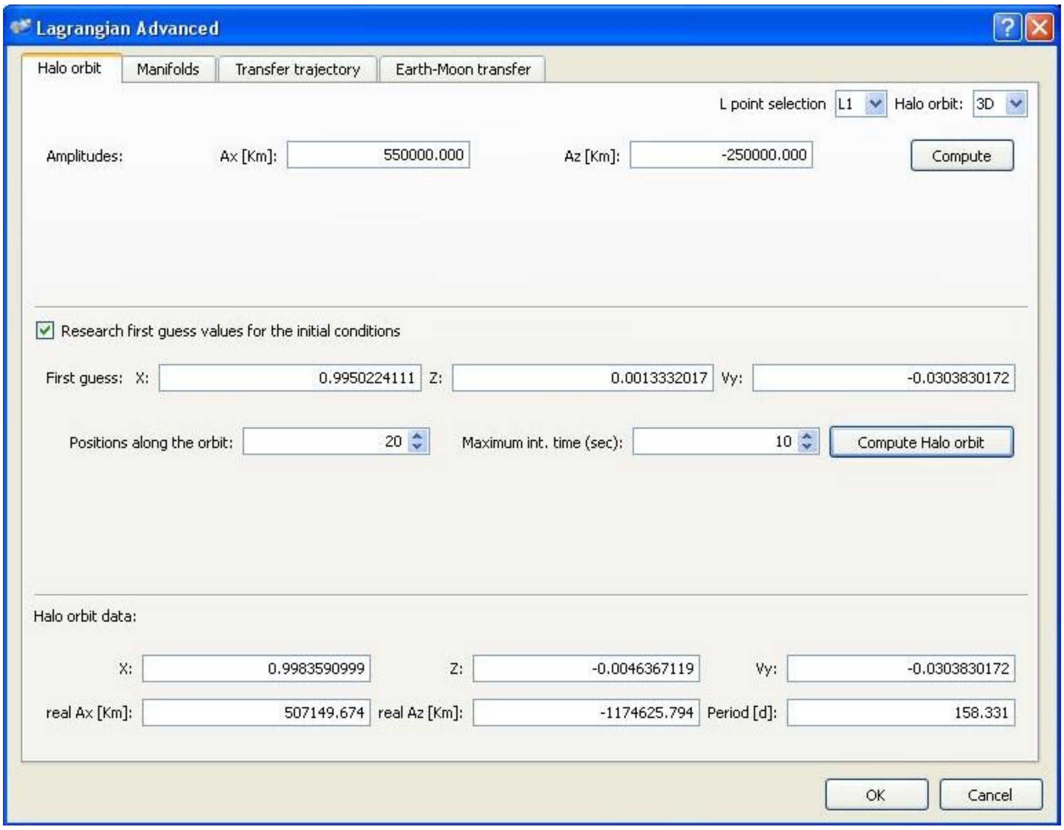

Figure 9.12 GUI screenshot for an example case of Halo orbit for Mode 2

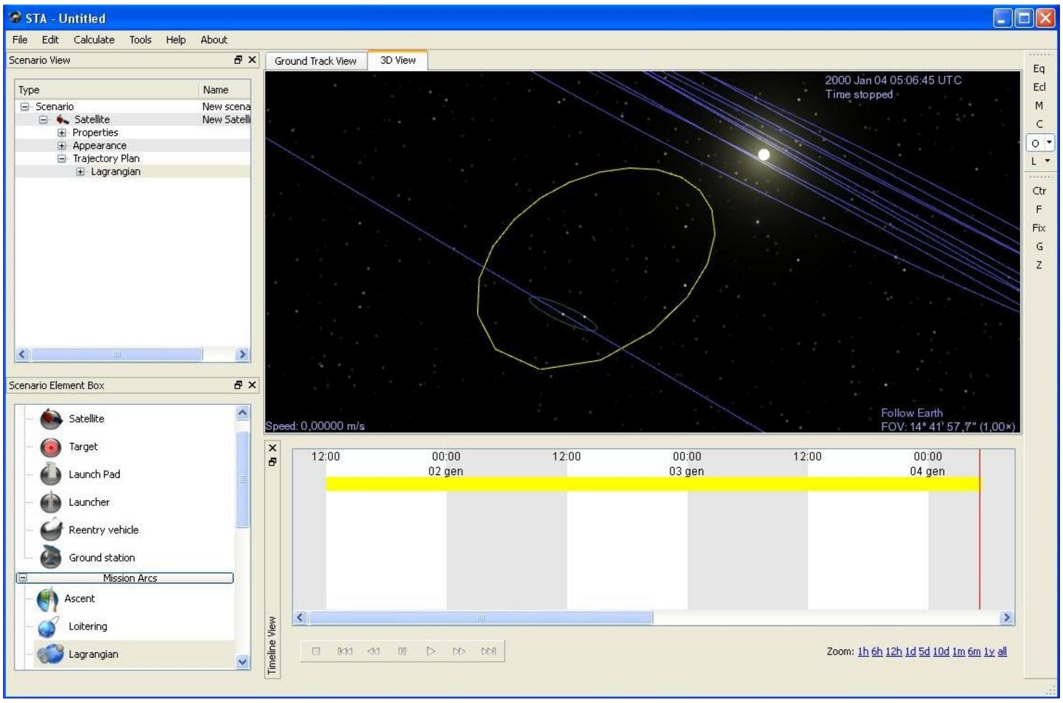

Figure 9.13 Trajectory screenshot for an example case of Halo orbit for Mode 2

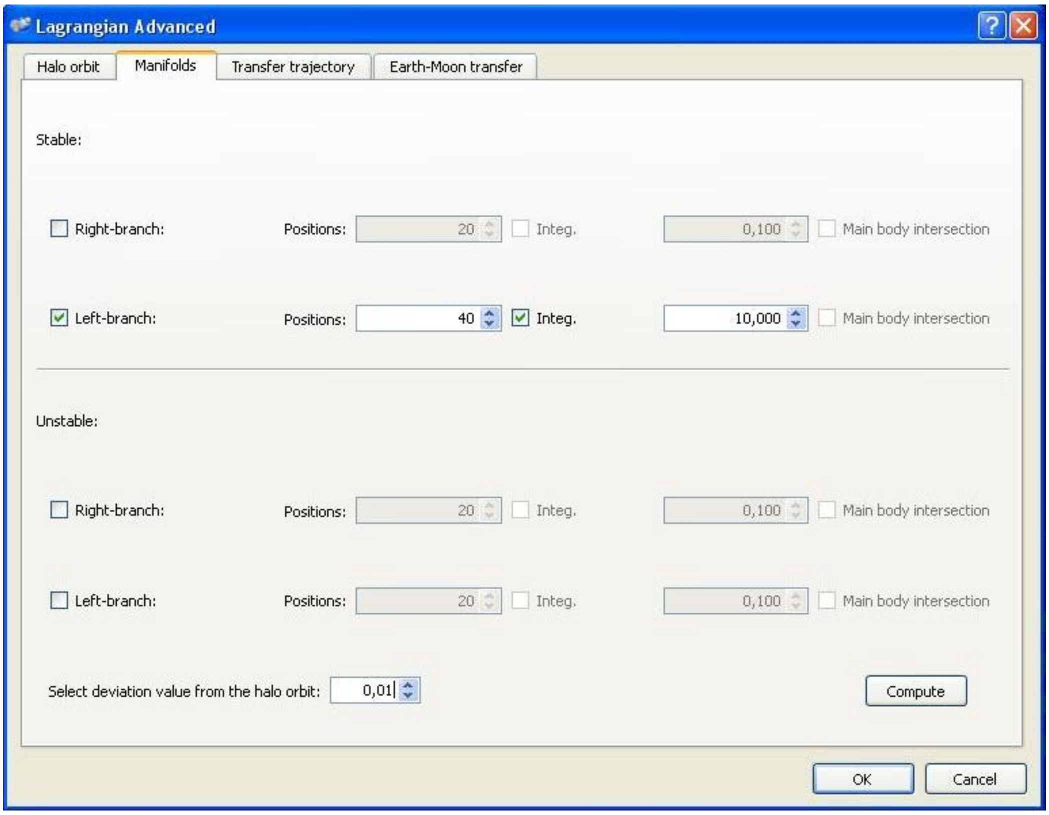

Figure 9.14 GUI screenshot for an example case of manifolds trajectories for Mode 2

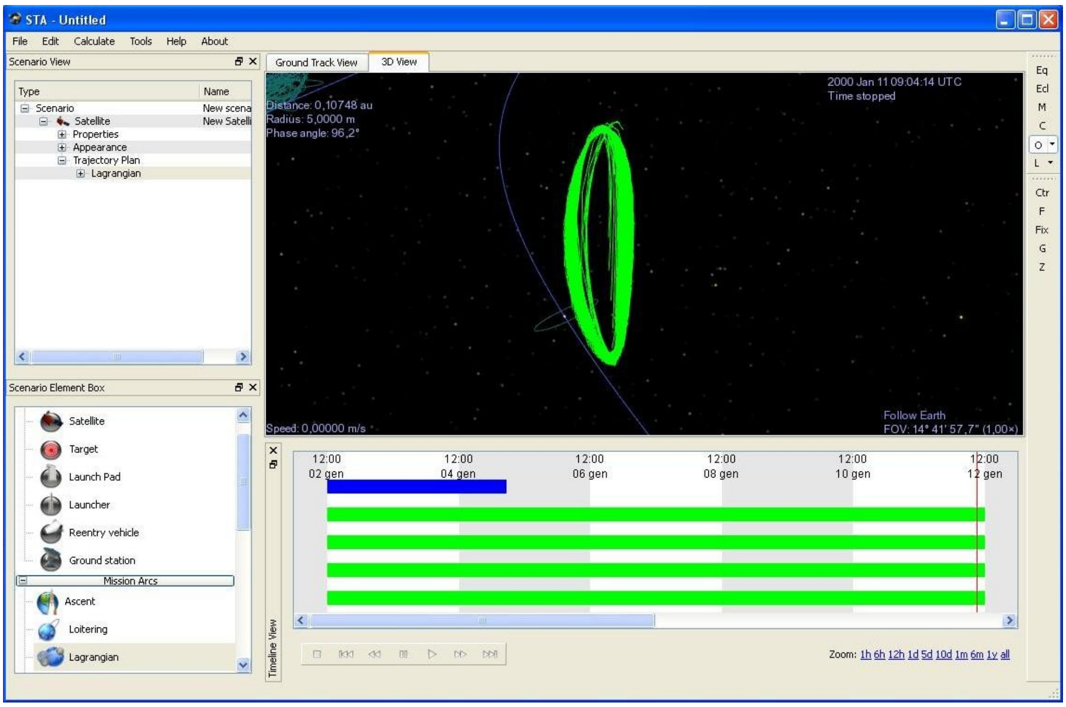

Figure 9.15 Trajectories screenshot for an example case of manifolds for Mode 2

10. STA Perturbations Service

In this chapter an overview of this service layer will be given in order to let the user understand the potential and possible applications for this STA components. Therefore a general explanation of the Perturbations Layer will be given as well as an introduction to the main features.

The most used model for trajectories design is the 2-body problem, where a satellite, with negligible mass, is gravitating around a central body.

The propagation of this kind of orbit, when only the central body's influence is considered, can be easily done by incrementing the true anomaly at each time step when keeping constant all the other orbital parameters, obtaining the exact solution. This operation is done by the Loitering Module.

However this kind of trajectories are only theoretical, because this simple model is complicated by the effects of orbital perturbations (generated by different causes), that exert a perturbing acceleration on the spacecraft (the magnitude of these accelerations is some degree smaller with respect to the acceleration due to central body attraction).

Introducing the orbital perturbation implies the necessity to solve a differential equation for the calculation of the spacecraft position and velocity at each time step.

The orbital perturbations that have been treated by the layer are :

- Aerodynamic drag due to the central body's atmosphere;
- Solar pressure radiation perturbation;
- Gravity of the third body perturbation;
- Central body's gravity field perturbation;
- Debris and micrometeorids collisions perturbation.

Adding one or more of these perturbations allow solving the equations of motion taking into account these perturbing factors.

Not only the Loitering trajectory can be influenced by perturbations: theoretically all kinds of trajectory, from re-entry to rendez-vous, are affected by perturbations, but in certain cases the perturbing effects are so small compared to the main forces acting on the spacecraft that they can be ignored without committing significant errors.

This is the reason why, while in the Loitering module it will be possible to add all the perturbations listed above, in the REM only the central body's gravity field perturbation can be added; moreover, in the REM the aerodynamic drag will not be treated as a perturbation but as the main force acting on the spacecraft.

10.1 How to use the Perturbations Layer

As explained in the previous paragraph, the user shall use the Perturbations Layer when he wants to simulate a trajectory considering the effects of orbital perturbations; the inputs related to perturbations will be added in a dedicate tab contained in the GUI of the specific trajectory he wants to propagate.

In this paragraph the user will be guided in the process of properly filling the section of the GUI dedicated to perturbations. It should be noted that this procedures will concern only the perturbation problem definition, but as perturbations is only an aspect of the astrodynamics problem, they will not be sufficient to properly propagate a trajectory.

The distinction between loitering and re-entry trajectory propagation will be done.

10.1.1 Loitering trajectory

The information related to perturbations has to be added in the loitering GUI.

The user must open the tab "Environment".

He can firstly select the central body from the menu and then pass to the "Perturbations" section.

Here the selection of the perturbations wanted to be taken into account in the propagation shall be made by clicking the radio button associated to every perturbation;

when selecting it, the corresponding field enables and the user can insert in it some other values, when possible.

- Aerodynamic drag due to the central body's atmosphere;

  The button to be clicked is "Atmospheric Drag". The inputs to be inserted are the following:

  - *atmospheric model* : it refers to a file containing a list of densities as function of the altitudes, necessary for the calculation of the atmosphere effect on the spacecraft. The user can choose from a list of models; so far the models are:
    - gram99 (Global Reference Atmospheric Model),
    - exponential model, based on the simple exponential formula for density calculation;
    - USSA1976 (U.S. Standard Atmosphere 1976)
    - mars_emcd (European Mars Climate Database).
- Solar pressure radiation perturbation;

  The button to be clicked is "Solar Pressure". The inputs to be inserted are the following:

  - *reflectivity* : it refers to the reflectivity constant of the surface of the spacecraft; the user shall insert a number from 0 to 1 (0 : no reflectivity, 1 : completely reflective).
  - Albedo : the user shall select this case if he wants to simulate the albedo radiation from the central body.
  - IR : the user shall select this case if he wants to simulate the infra-red radiation from the central body.
- Central body's gravity field perturbation;

The button to be clicked is "Gravity Field". The inputs to be inserted are the following:

- *gravity model* : it refers to a file containing a list of harmonic coefficients representing the potential of the gravity field (depending on the actual shape of the celestial body). Every file refers to a particular central body; the available models are :
  - EGM2008 (Earth Gravitational Model 2008);
  - GTM090 (Mars Gravity Model deducted from Mars Global Surveyor mission),
  - SHGJ180 (Venus Gravity Model deducted from Magellan mission),
  - JGL165 (Moon Gravity Model deducted from Lunar Prospector mission, see).
- *gravity zonals* : it represents the number of zonal harmonics the user wants to take into account in the simulation: higher the number, better the approximation of the gravity perturbation. The user can put a integer from 2 to 20.
- *gravity tesserals* : it represents the number of tesseral harmonics the user wants to take into account in the simulation: higher the number, better the approximation of the gravity perturbation. The user can put a integer from 2 to the current value of the gravity zonals.

- Gravity of the third body perturbation;

  The button to be clicked is "Third Body". The inputs to be inserted are the following:

  - Once decided which celestial bodies will be considered as exerting an influence on the spacecraft, select them from the list on the left and put them in the list on the right by double-clicking them.

- Debris and micrometeoroids collisions perturbation.

  The button to be clicked is "MMOD". There are no inputs to be added.

10.1.2 Entry trajectories

The information related to perturbations has to be added in the re-entry GUI.

The user must open the tab "Propagation".

He can firstly select the central body from the menu and then pass to the "Perturbations" section.

Just one perturbation is available, and its selection shall be made by clicking the radio button associated to it; when selecting it, the corresponding field enables and the user can insert in it some other value.

- Central body's gravity field perturbation;

The button to be clicked is "Gravity Field". The inputs to be inserted are the following:

- *gravity model* : it refers to a file containing a list of harmonic coefficients representing the potential of the gravity field (depending on the actual shape of the celestial body). Every file refers to a particular central body; the available models are :
  - EGM2008 (Earth Gravitational Model 2008);
  - GTM090 (Mars Gravity Model deducted from Mars Global Surveyor mission).
- *gravity zonals* : it represents the number of zonal harmonics the user wants to take into account in the simulation: higher the number, better the approximation of the gravity perturbation. The user can put a integer from 2 to 20.
- *gravity tesserals* : it represents the number of tesseral harmonics the user wants to take into account in the simulation: higher the number, better the approximation of the gravity perturbation. The user can put a integer from 2 to the current value of the gravity zonals.

10.1.3 Spacecraft properties parameters

When selecting a perturbation, other parameters shall be imperatively added to allow the perturbing accelerations being calculated and consequently the trajectory being simulated; these parameters shall not be inserted in the Loitering / Re-Entry GUI but in the spacecraft properties GUI.

These inputs will be briefly listed :

- Aerodynamic drag due to the central body's atmosphere;
  - mass of the spacecraft;
  - surface of the spacecraft;
  - spacecraft CD coefficients file;
- Solar pressure radiation perturbation;
  - mass of the spacecraft;
  - surface of the spacecraft;
- Gravity of the third body perturbation;

  none
- Central body's gravity field perturbation;

  none
- Debris and micrometeoroids collisions perturbation.
  - mass of the spacecraft;
  - surface of the spacecraft;

10.2 Perturbations layer GUI

Figure 10.1 shows the tab “environment” of the Loitering GUI. The perturbations information shall be added in the proper section.

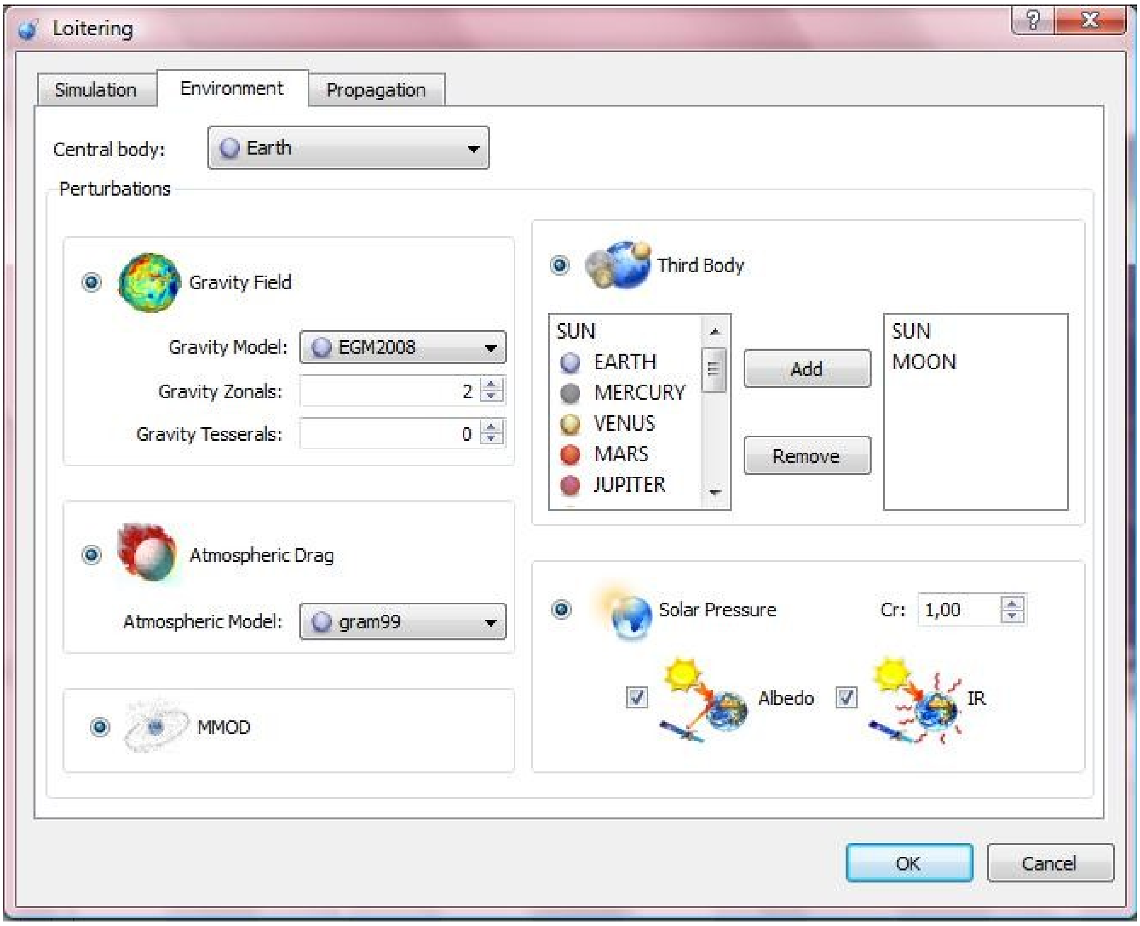

Figure 10.1: Perturbations section in Loitering GUI

Figure 10.2 shows the tab “Propagation” of the REM GUI; the perturbation section is reduced with respect to the Loitering one, and containing only Gravity Field perturbation.

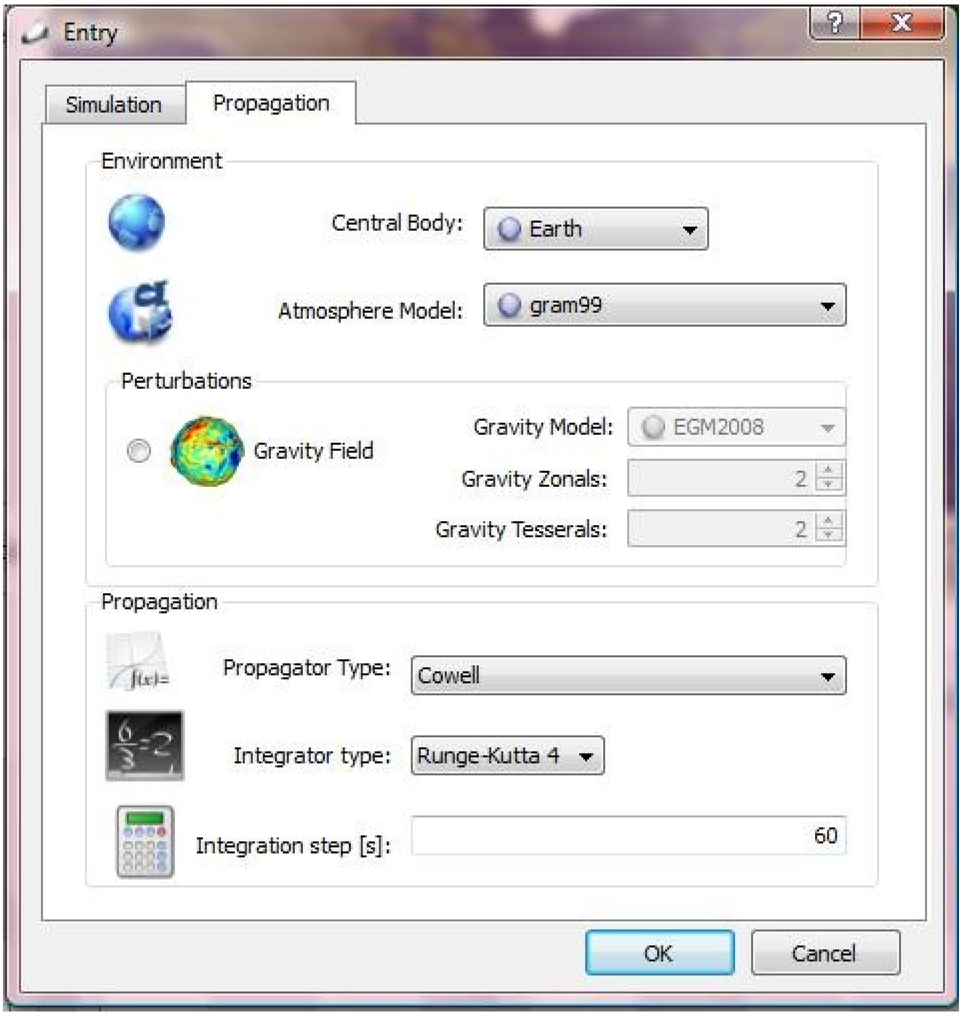

Figure 10.2: Perturbations section in Re-Entry GUI

11. STA LICENCING AND COPYRIGHT

STA is distributed under the GNU public license (found below) in the spirit that it will be useful to academic and industrial communities of space research.

In no way the STA Steering Board or any of its members shall be liable for any use of STA or its source code.

STA code is open source. The code in the STA project is license under the GENERAL PUBLIC LICENSE, Version 2, June 1991 of the Free Software Foundation, commonly known as GNU GPL.

The STA code, logo, and web site are copyright of the STA Steering Board.

In the STA project, all Intellectual Property Rights (IPRs) are assigned to the STA Steering Board.

11.1 GNU GENERAL PUBLIC LICENSE, Version 2, June 1991

For details about this license see the corresponding URL here

11.1.1 How to Apply These Terms to New Programs based on STA

If you develop a new program based on STA, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

This General Public License does not permit incorporating STA in its totality or its parts into proprietary programs without making the source code available.

12. How to Participate on STA development

The STA development team is composed of a group of enthusiast in astrodynamics, informatics, and applied mathematics. Developers are teaming up from Universities or Academia (Master thesis, PhDs students) to enthusiast volunteers and space agencies officers that want simply to help.

The organization of the STA project is lead by the STA Steering Board (SB).

The STA project allows the partner-shipping with space Agencies, Academia, Industry and free volunteers.

Individuals wishing to contribute to STA can do it in several ways:

- By sending a note to the central STA e-mail address requesting participation on the STA project: Space.Trajectory.Analysis@gmail.com
- By contacting a Steering Board member from one of the Institutions listed in our site and establishing a link with this person. The list of the requested modules for development is made available to all Board members.

Developers are persons that participate actively in STA code by creating, modifying, and maintaining code.

Contributors are occasionally participants on the forums and discussion list that have been following the developers email discussion list, or have downloaded the source code found in the nightly build repository. Contributors can become developers by writing patches for some features of the code that they think needs tweaking, etc.

13. LIST OF ACRONYMS

DoF	Degrees of Freedom
ESA	European Space Agency
ICD	Interface Control Document
SB	Steering Board
STA	Space Trajectory Analysis

* End of Document *