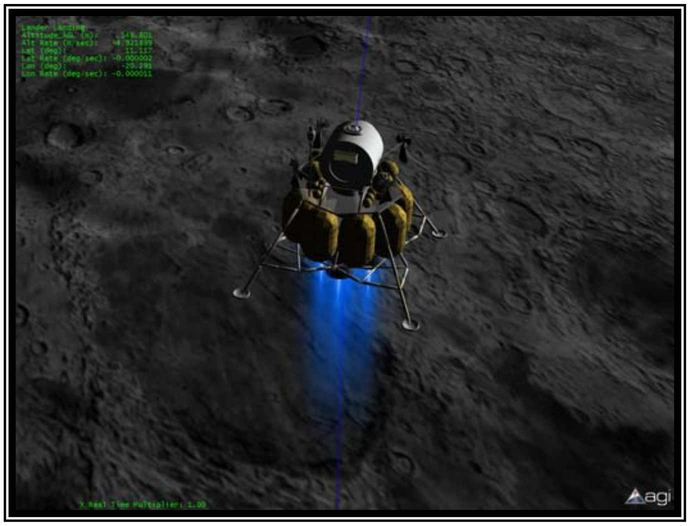
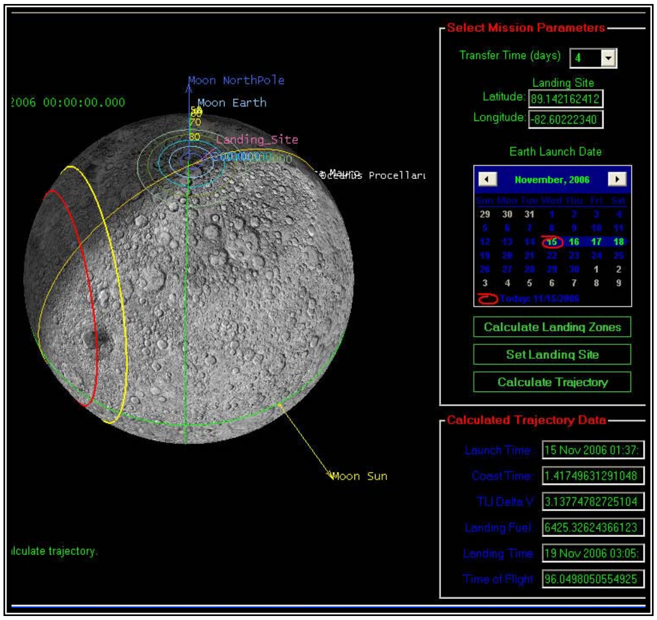
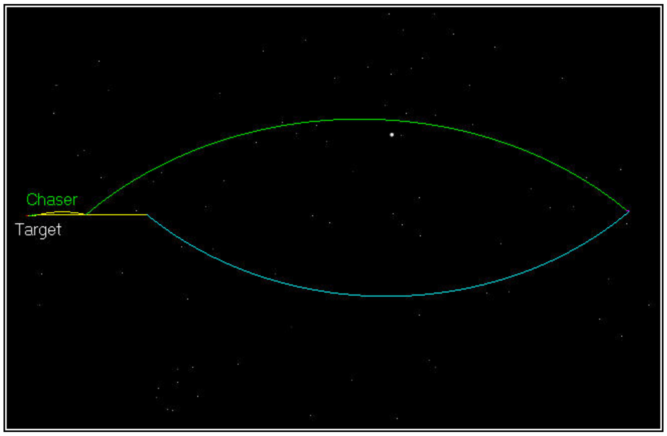
Lunar Trajectory Design with Astrogator (Day 4)







Objectives

At the conclusion of this class, you will have:

- Seen demonstrations of 3 different levels of automation within STK
    - Autosequences
    - Plug-In Scripts
    - Single Segment Propagation
- Targeted a Surveyor-type Direct Lunar Landing to the Surface of the Moon
    - Created a custom set of Axes, coordinate system, and calculation object
    - Targeted a landing site
    - Seen a 4Dx application that demonstrates a lunar application that this trajectory
    - Seen VBscript engine model plugins
- Target an Apollo-Type Lunar Landing from Orbit
    - Used an in-line VBscript calc object
    - Used nested Targeting
    - Landed on the Moon!

NOTES

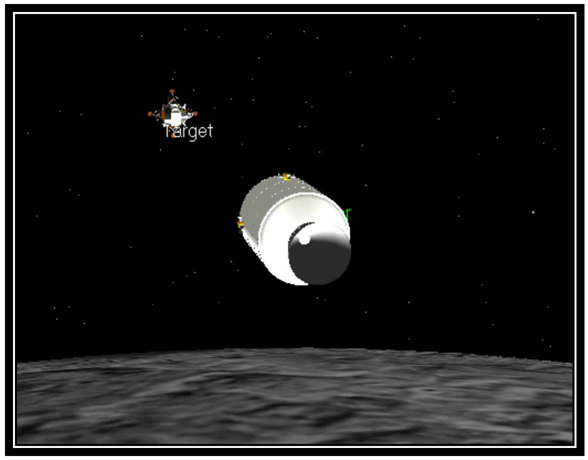
Table of Contents

NOTES

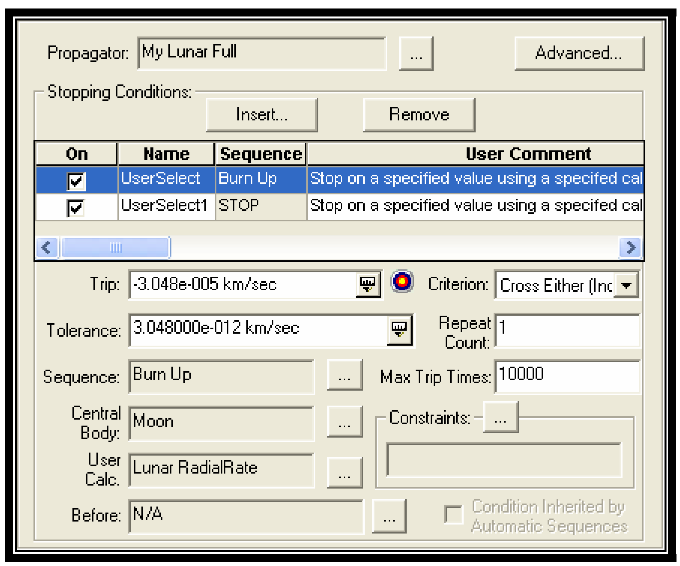
Trajectory Examples

Autosequences: Proximity Operations

1. Open the 'Lunar Rendezvous' scenario from the Astrogator Training folder..

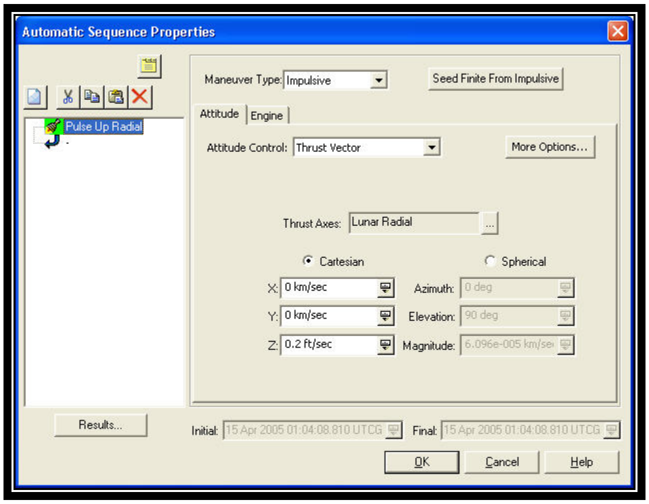


2. Open the 'Chaser' Satellite, and then locate the 'Propagate' segment within the 'Forced to -200' target sequence. Note that the 'UserSelect' stopping condition triggers an autosequence called "Burn Up"



NOTES

3. The ‘Burn Up’ Autosequence is called whenever the –Z velocity (or radial down towards the Moon) is greater than 0.1 ft/sec. The spacecraft first does a burn opposite of the velocity vector to move away from the target. Without corrections, the chaser would fall below the target and loop in front of it. The autosequence keeps the chaser satellite on the Vbar of the target satellite by continually rotating the line of apsides of the chaser spacecraft’s orbit.

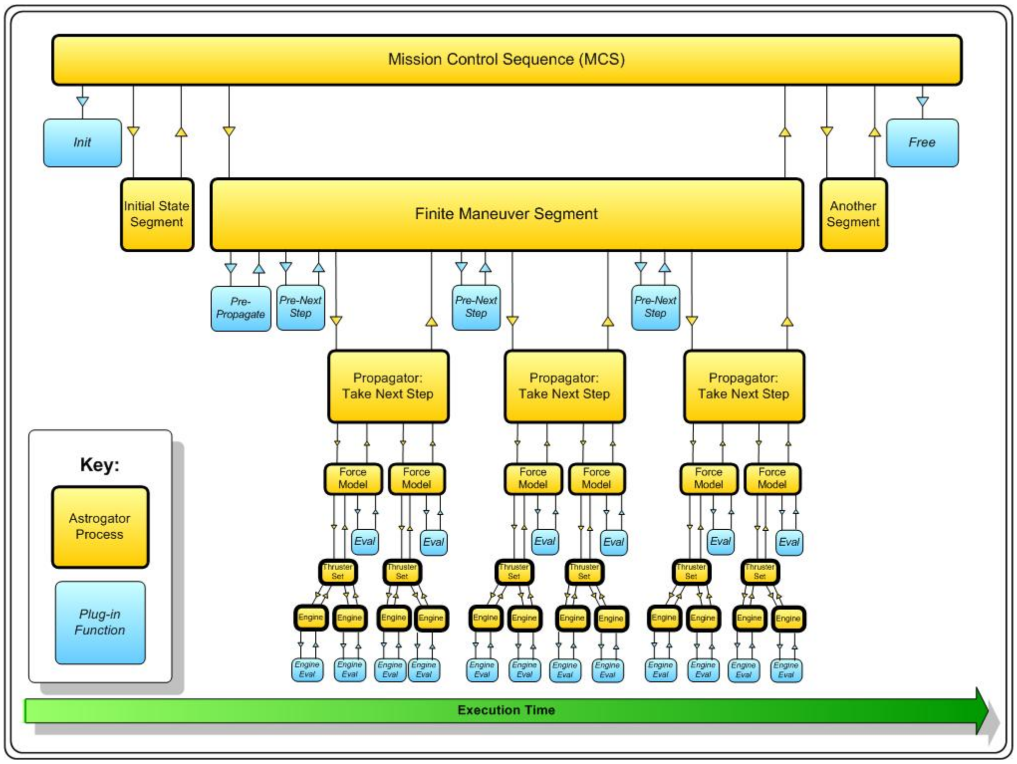


Investigate the other segments of the MCS. Look for the ‘Propagate’ segment in the ‘Forced to -0.050 m’ target sequence, which implements the logic to prevent the vehicle from going in the +Z direction.

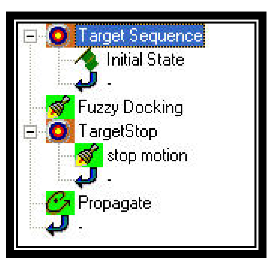
Astrogator Scripting: Plug-ins

The previous scenario demonstrates how a proximity operations scenario can be developed within the Mission Control Sequence, but also how such an MCS can easily become quite complex. Some times such a problem can be better solved outside the MCS, using an external script that can trigger forces or events in the Astrogator propagation, such as maneuvers.

NOTES



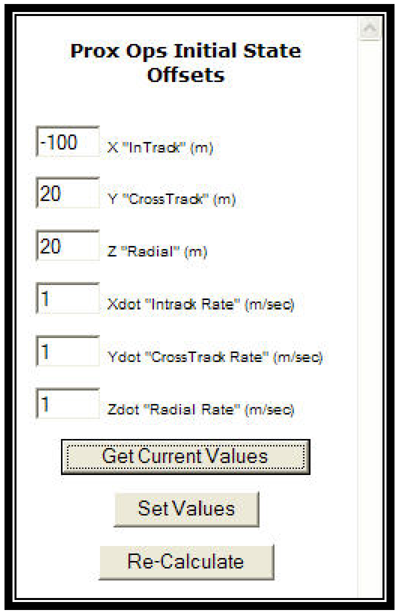
Within the Astrogator MCS are several built-in plug-in points. Each of these plug-in points allows the user to attach scripts that affect the numerical integration at every step. So, rather than creating a complex MCS that has hundreds of engine firings that perform a proximity operations sequence, we might instead set up an external control law that tells each engine model when to be on or off during the numerical integration. So, rather than the complex MCS we saw in the Lunar Rendezvous Scenario, we can have something simpler:



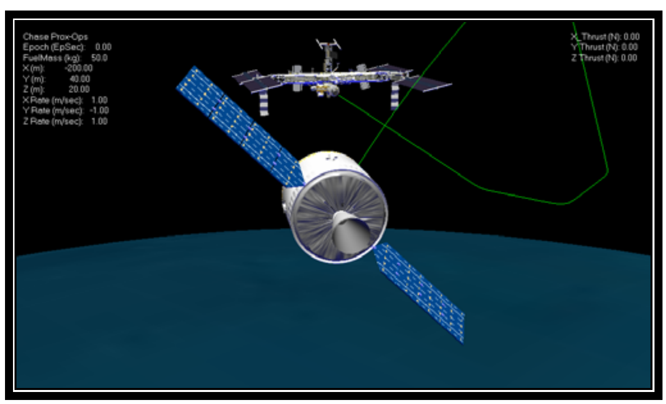
NOTES

Plug-ins: Proximity Operations and Docking

4. Open the CEV docking scenario (Don't re-run the MCS, it won't operate on your machines since it requires the registration of some DLLs.)
5. Note the HTML data utility, used to set the initial offsets of the CEV from the ISS

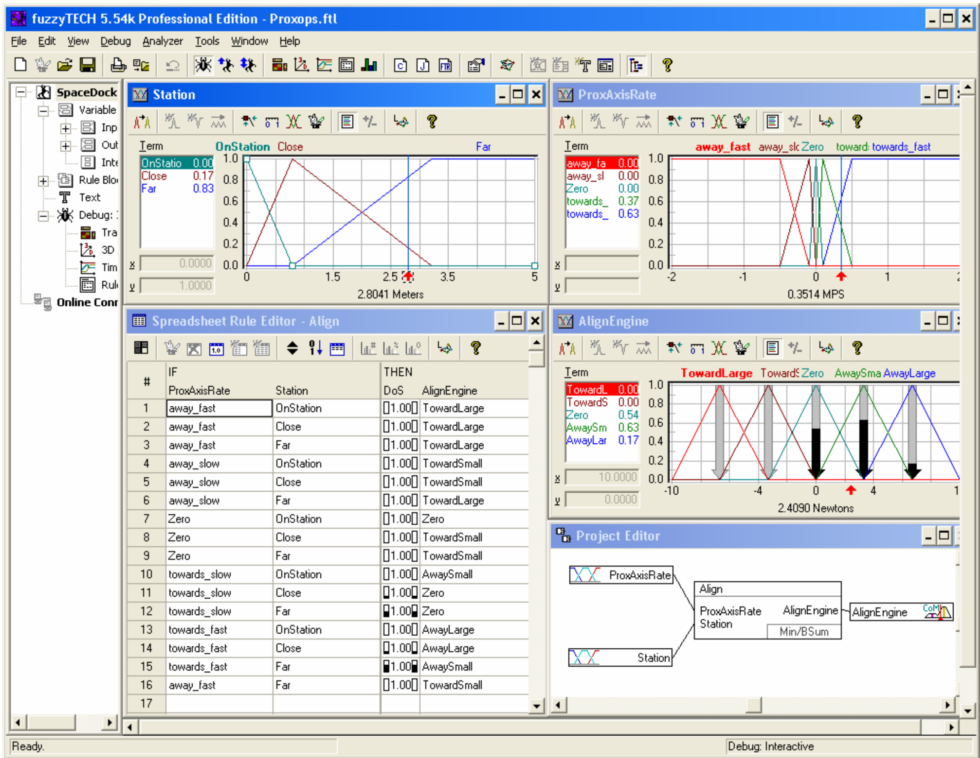


This HTML utility allows the user to enter and retrieve the offset data for the scenario, and to repropagate the MCS once the new data has been entered.



NOTES

In this example, an external script using Fuzzy Logic is used to determine when each engine should be fired. The script is run as the MCS is propagating and an external STK articulation file is also created, which allows users to visualize the firing of the engines.



Fuzzy logic provides a method for the mathematical computation of human expertise expressed in linguistic control decision terms, such as Near, Far, Slow, and Fast and the transitions between. Fuzzy logic is a heuristic method that lends itself to complex control and decision environments where multiple expertises are required. The process is deterministic, for every given set of inputs, the outputs will be the same; an important feature in fuzzy logic control for critical systems. Despite the name “Fuzzy,” the control is anything but, and provides a rigorous mathematical model based in discrete formulas, not probability.

NOTES

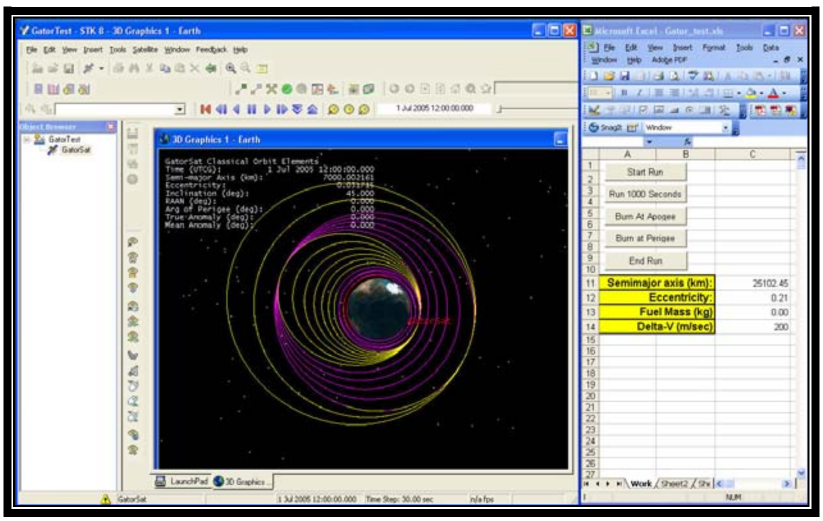
Fuzzy logic, however, is only one example of a control method. The Astrogator platform can support any set of control laws that can be coded.

Further explanations of this method and the research done with it can be found in the paper: ***Proximity Operations for Space Situational Awareness, Spacecraft Closed-Loop Maneuvering Using Numerical Simulations and Fuzzy Logic*** by Tim Carrico, et al., included with the training materials.

Astrogator Scripting: Single Segment Propagation

Finally, while scripting within a thruster segment will help you solve some problems, others require more complex logic than can be represented within a sequential MCS. In these cases it can be useful to use the MCS segments as utility functions in your own code, and build your trajectory with as logic as complex as allowed by any standard programming language..

6. Open the Single Segment Test scenario. Inspect the MCS of the "GatorSat" satellite.
7. In the scenario folder, open the "GatorTest.xls" spreadsheet. Position excel so that you can see both Excel and STK.



8. Click the 'Start run' button, and then the other buttons in any order you want until you are done. Then click the 'End Run' button. This is the equivalent of running an MCS, except you

NOTES

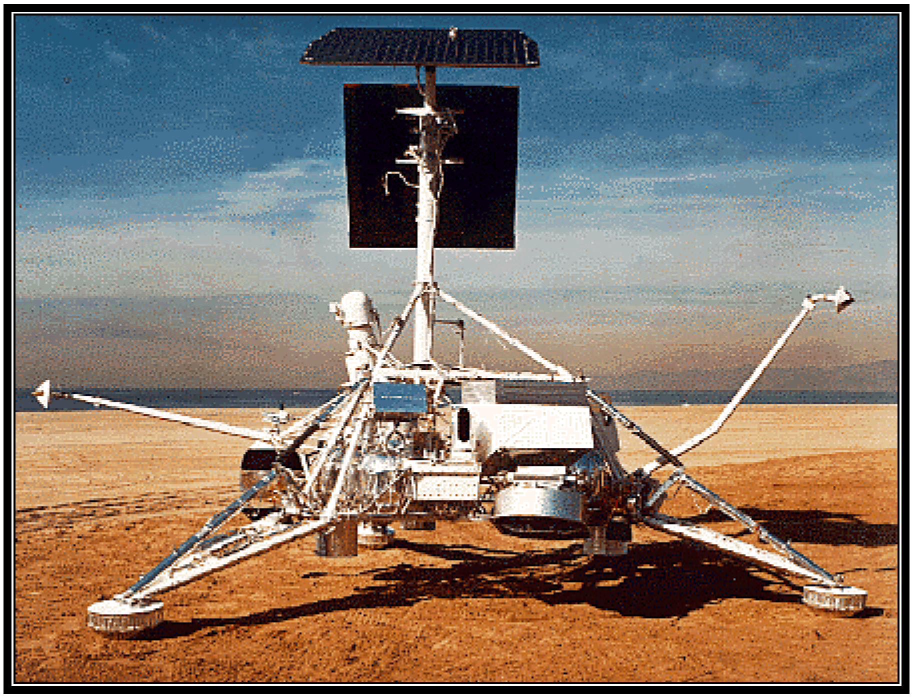
have chosen manually which order the segments have run. In a script, you other logic could be used.

9. Use the Tools/Macro/Visual Basic Editor to inspect the VBA code.

NOTES

Exercise 11: Direct Lunar Landing

In this exercise, we will use Astrogator to land on the Moon. We will perform a direct landing here, i.e. we will not go first to orbit but instead will land directly on the Moon from our cislunar trajectory. This is similar to the trajectories used by the NASA Lunar Surveyor spacecraft. The Surveyor program consisted of seven unmanned lunar missions that were launched between May 1966 and January 1968. Five of these spacecraft, Surveyor 1, 3, 5, 6, and 7 successfully soft-landed on the lunar surface.



Scenario Setup

10. Re-open the Lunar Orbiter scenario from the day 3 manual.

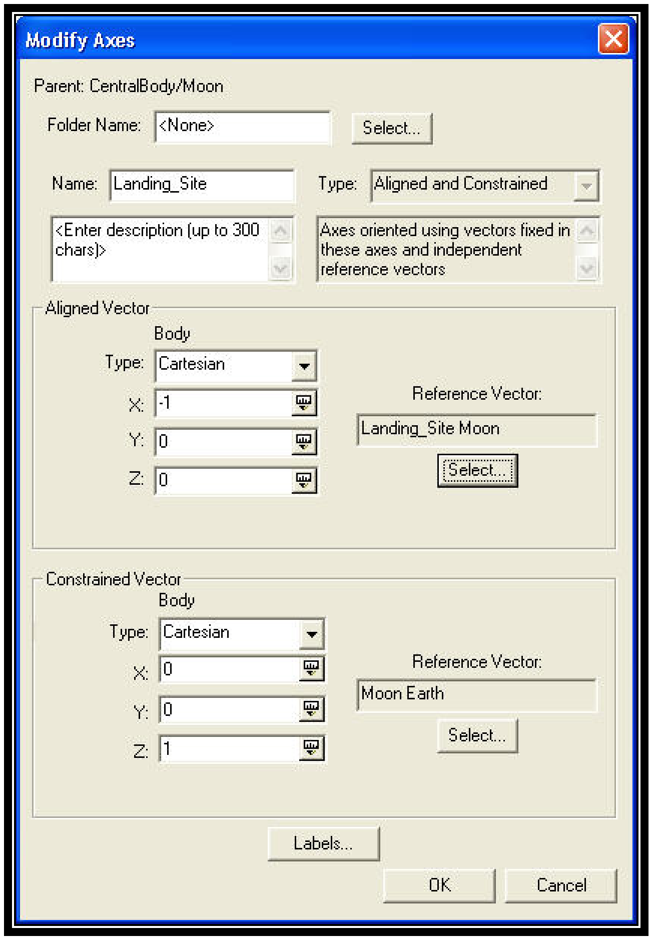
11. Save it as "Lunar Direct landing"

NOTES

12. Using the 'Insert' Menu item, change the setting for 'Central Body for New Objects' to the Moon.

13. Insert a Target in the scenario, and name it 'Landing_Site'

14. Using the Vector Geometry Tool, create a new Moon-based Axes that uses the Landing-Site Moon vector as the –X coordinate, and the Z axis constrained towards the Moon-Earth Vector.



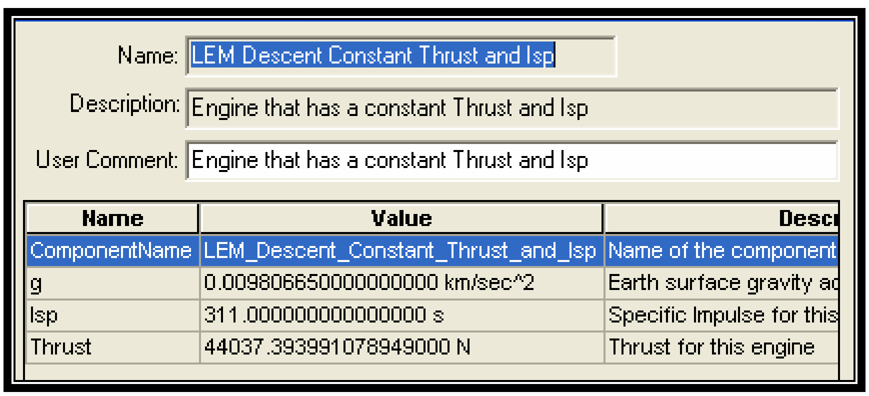
15. Then create a **Moon**-Based coordinate system (Type = Assembled) that uses this coordinate system as the Reference Axes and the Moon as a the Origin Point. Name this "Landing Site". We will use this system as a way to create a target-based Latitude and Longitude for our landing site that doesn't have the same difficulties at high latitude that a normal Moon-Fixed system would have.

NOTES

16. Deactivate the Alt-Inc profile in your 'Target Moon' Target Sequence, and place a STOP segment after this Target sequence to disable the 'Get Into Orbit' and subsequent segments.
17. Change the Bplane targets for the Bplane profile to BdotT = 0.0 and BdotR = 0.0
18. Rerun the MCS. Investigate the conditions of impact and see that your segment is stopping on your altitude stopping condition.
19. Change the stopping condition altitude from 10 km to 15,o00 ft.
20. Configure 2 new calc objects in the 'Spherical Elems' folder. Make copies of both Right Ascension and Declination. Change the coordinate system for both to the newly created 'Landing Site' system. Name the new Calc Objects 'Target RA' and 'Target Dec'.

These calc objects will be used like Latitude and Longitude would be to target the landing site. While normal Latitude and Longitude are available for the Moon, these units don't work well with the differential corrector when we want to target somewhere near the Lunar poles. Thus, we'll use calc objects that are tied to our desired landing site, and will avoid these difficulties.

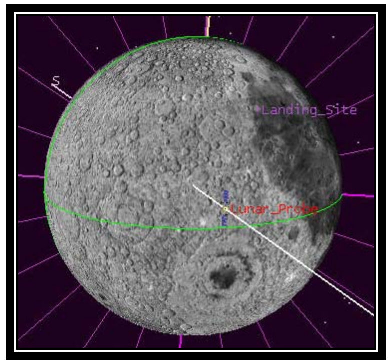
21. Add both of your new calc objects to the 'prop to Moon' segment. We will use these to force our landing site to occur at the facility we choose.
22. Configure a new engine model, copy the Constant Thrust and ISP default, and give it the following settings (9900 lbf thrust):



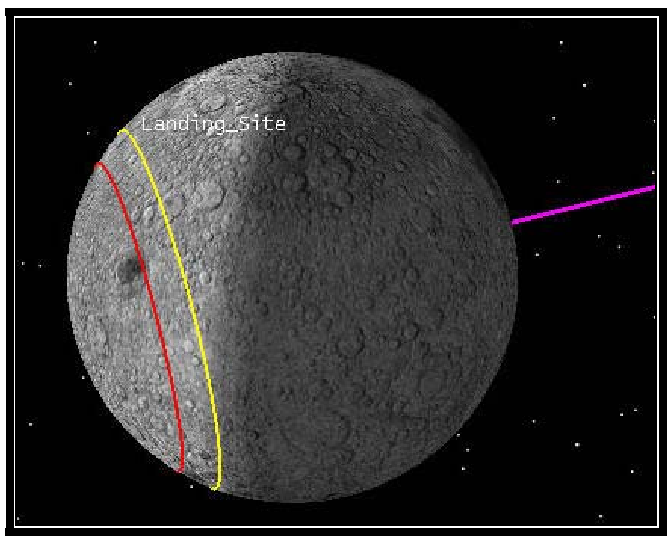
23. Make sure the 'Update Animation Time after running this segment' option is selected for the 'Prop to Moon' segment. This will allow us to see where our landing site is after targeting.
24. Check the summary report for the 'Prop to Moon' segment. The end of the report will show the Target RA and Target Dec calc object values, letting you know the relative locations of the landing site and our impact location.

NOTES

Let’s now investigate where our Impact is hitting, and the location of our Landing Site facility.



Note that our landing site is close, in this case, to our impact spot. Since we are attempting here to land directly, we are restricted from landing on certain sites. When the spacecraft uses a direct trajectory, some locations opposite on the surface from the incoming asymptote are unavailable. The figure below shows the restricted region:



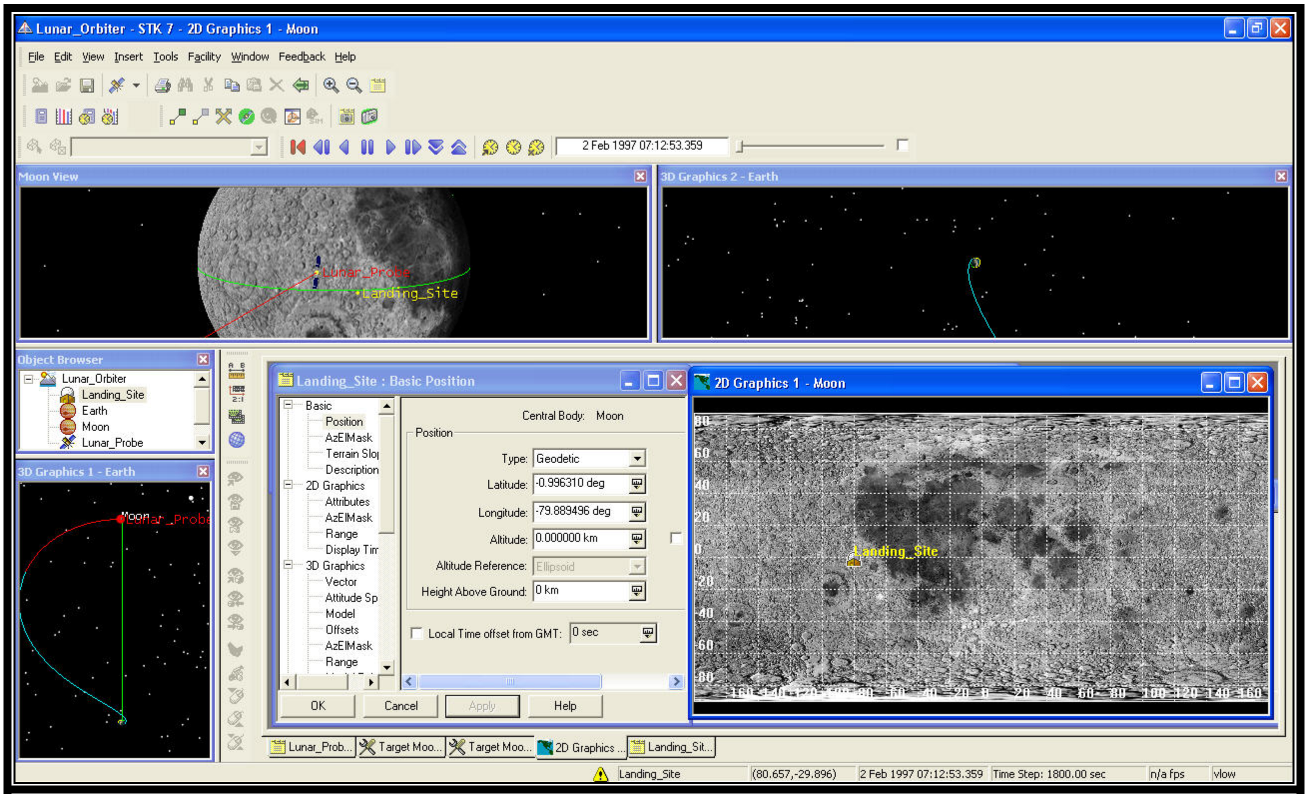
In this case, the incoming trajectory is shown in purple to the right, the exclusion zone is the red circle, and the yellow circle shows a region 10 degrees outside of the exclusion zone. The size and

NOTES

location of the exclusion zone depends on the energy of the incoming trajectory as well as the direction.

While we can't see our exact exclusion zone for our scenario, we can easily miss it by setting our Landing Site at the proper location.

25. Turn off the Bplane (3D Graphics/Bplane) for the satellite to keep the window from being too cluttered.
26. In the 'View' menu, create a new 2-D window. Change the Map Central body to be the Moon.
27. Open the properties of the Landing site, and with both windows open, click on the Map window to relocate the Landing site.



NOTES

28. Create a new differential corrector targeting profile in the 'Target Moon' target sequence called 'Target Site'. Use Launch Epoch, Coast Duration and TLI Delta-v as controls, and use Target RA, and Target Dec, and Time of flight as constraints.

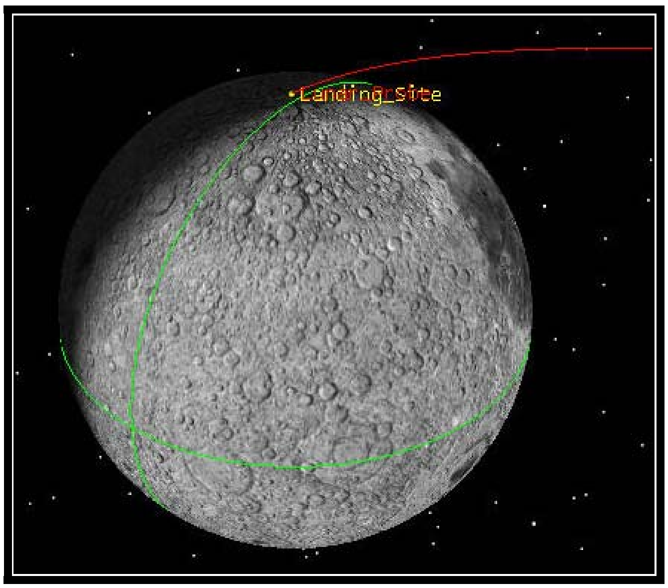
    a. Launch Epoch: Perturbation: 10 sec, Max Step: 100 sec.

    b. Coast: Perturbation: 1 sec, Max Step: 10 sec

    c. TLI Delta-V: Perturbation: 0.0001 km/sec, Max Step: 0.001 km/sec

    d. Target RA, Target DEC: Desired Value: 0.0, Tolerance: 0.01 deg.

29. Move your landing site to a non-excluded location and re-run the MCS.



30. Add a target sequence after the 'Prop to Moon' segment called 'Solid Stage'.

    a. Insert an impulsive maneuver segment into the new target Sequence called 'burnout'

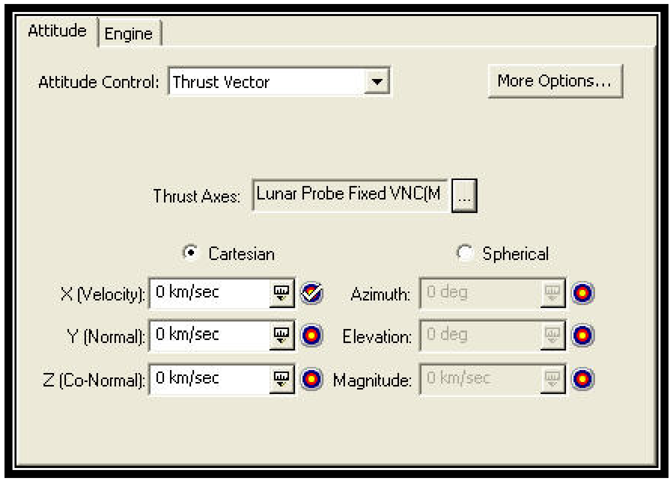
        i. Set the Attitude control to 'Thrust Vector'

        ii. Set the Thrust Axes to satellite Moon VNC (fixed) *[You may have to create this system in the VGT. Use the satellite object and copy the VNC(Moon) axes]*

        iii. Set the velocity component as a control

NOTES

iv. Add Moon-based altitude rate (Geodetic) as a result



b. Set the profile to use the Velocity component of the maneuver to set the Lunar Altitude Rate to -400 ft/sec.

c. Set the 'Solid Stage' Target Sequence to 'Run Active Profiles'

31. Add an update segment after the 'Solid Stage' Target Sequence called 'Stage Solid'

    a. Set dry mass to a value of 6531 kg

    b. Set the fuel mass to a value of 8165 kg.

32. Add a finite maneuver segment, called 'Landing Burn'

    a. Set the Attitude Control to 'Thrust Vector'

    b. Set the Attitude update to 'update during burn'

    c. Set the thrust axes to the Fixed VNC (Moon) axes you created earlier.

    d. Set the X(Velocity) component to -1 (Note: This is a unit vector, **don't enter units**.)

    e. Set the Engine_Model to LEM Descent Constant Thrust and ISP

    f. Set the "Thrust efficiency" to 1, and set this as a control. (This will act as a throttle)

    g. Change the Finite Maneuver Propagator to 'Lunar'

    h. Using the 'More Options' button, change the stopping condition to User Select, and use a Lunar Altitude Rate of -4 ft/sec.

NOTES

33. Add a new targeting profile to 'Target Moon' (named 'Thrust Level') that adjusts the efficiency of 'Landing Burn' to achieve an altitude of 25 ft at the end of the finite maneuver segment(0.01 perturbation, 0.1 max step size, tolerance = 1 ft)

34. Experiment with profiles that adjust the final landing site to hit landing site.

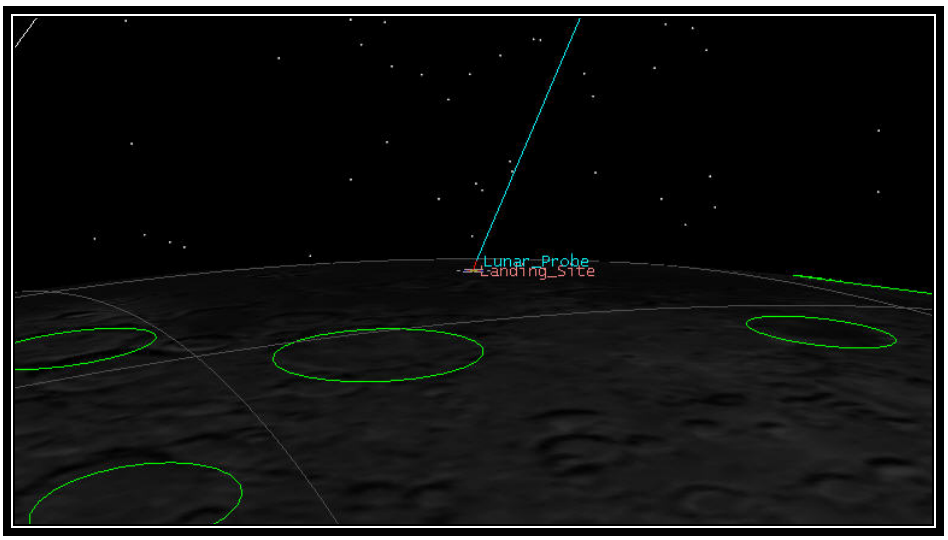
    a. Add Target Dec, Target RA as results to 'Land'. Modify Thrust Level profile to include Launch, Coast, Burn as controls, and Target RA, Target Dec and TOF as constraints.

    b. Add a return segment before the 'Solid Stage' segment, along with profiles that toggle it on and off.

    c. Retarget.

    d. Add an Altitude rate result to the final segment and inspect it in the summary report.

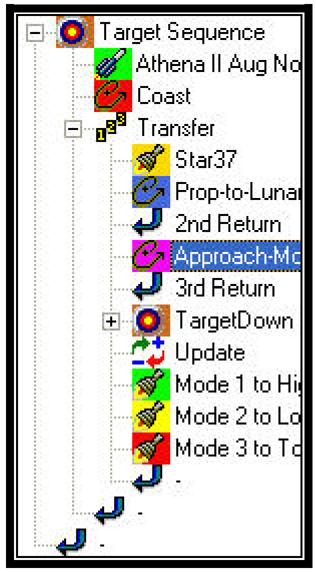
35. Close the scenario.



Trajectory Examples: Lunar Lander Application

1. Open the 'Direct Surveyor-Type Kinetic' scenario from the Day 3 folder.
2. Open the properties of the satellite and inspect the MCS.

NOTES



This MCS has many features that we've seen in previous Lunar Mission Control Sequences, including returns, and a Surveyor-like staged targeter "Target Down" and an "Update" segment. After the Update segment we have 3 finite maneuver segments:

Mode 1 to High gate: This segment throttles via an engine model plug-in. The Throttle logic calculates at each point, the constant thrust level that would get the spacecraft from it's current altitude and altitude rate, to an altitude of 512 feet with an altitude rate of -16 ft/sec. The acceleration is calculated via:

$$a_d = (V_0^2 - V_f^2) / (2 (H_0 - H_f))$$

where H0 and V0 are the current altitude and velocity, respectively. Hf is the altitude at "high gate "(512 ft), and Vf the desired rate at that point (-16 ft/s).

Mode 2 to Low gate: This segment throttles via an engine model plug-in. The Throttle logic calculates at each point, the constant thrust level that would get the spacecraft from it's current altitude and altitude rate, to an altitude of 12 feet with an altitude rate of -4 ft/sec. The acceleration is calculated via the same equation.

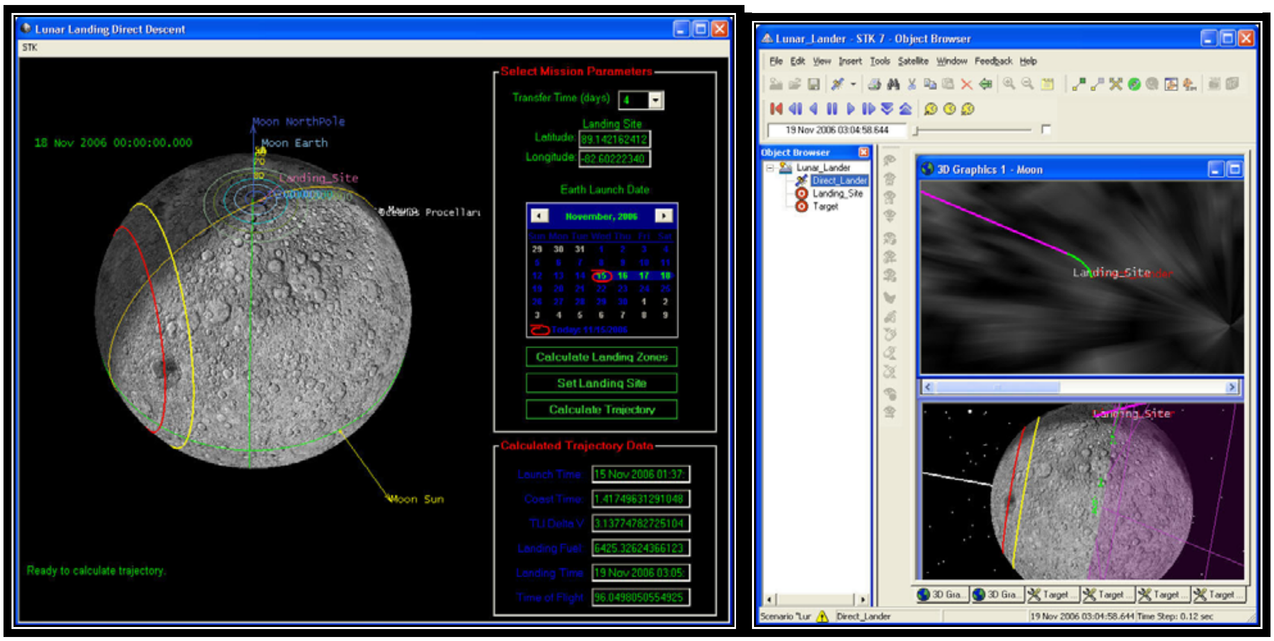
Mode 3 to Touchdown: This segment throttles via an engine model plug-in. The Throttle logic calculates at each point, the constant thrust level required to maintain a constant altitude rate of -4 ft/sec until touchdown.

NOTES

All 3 scripts are coded in VBscript, and are included in the scenario folder.

The scenario targets the ‘approach moon’ segment to be directly above the selected landing site (for speed) and once that target is achieved, the last 3 segments are turned on, to perform the landing phase.

The scenario can be run from the VB.net Lunar Tool application, as will be demonstrated by the instructor:



This tool, coded in VB.net, allows the user to select the landing site and epoch, and choose it based on lighting conditions and landing restrictions inherent in the direct landing approach.

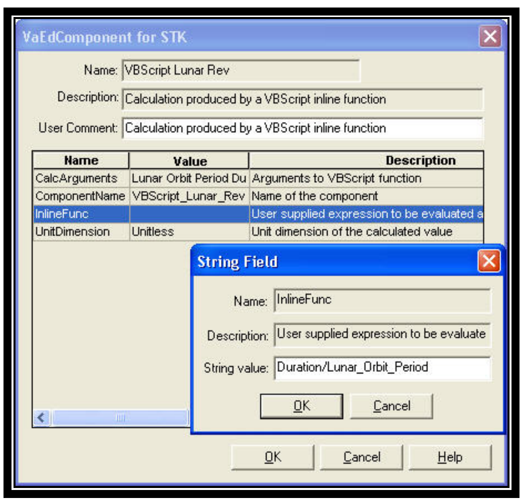
This tool takes advantage of the STK 4DX functionality, which allows the user to embed STK technology in their own applications. In our example, 4DX is being used ALONG with a running instance of STK for debugging purposes. In a released tool, the STK instance would not be required.

Several other lunar landing throttling techniques were explored in: A Comparison of Lunar landing Trajectory Strategies Using Numerical Simulations by Loucks, Carrico et al. which is included as a PDF file in the electronic materials.

NOTES

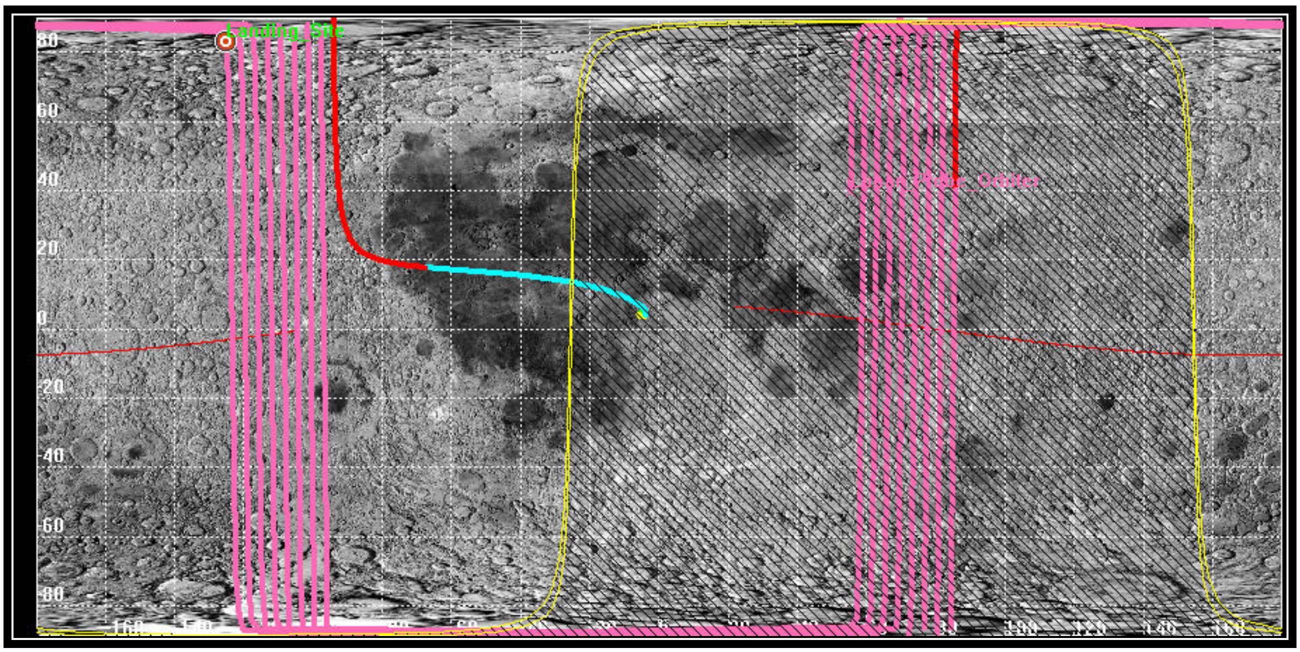
Exercise 12: Orbital Lunar Landing

1. Re-open the Lunar Orbiter scenario from the day 3 manual.
2. Make sure the 'Orbit Capture' targeter puts the satellite into a circular orbit.
3. Save the scenario in a new folder called "Orbital Landing"
4. Change the Satellite to be Moon Centered. (Copy and paste the MCS segments into a new satellite that you create after setting the Central Body for New Objects to the Moon). This is required if we want to have a Lunar 2D Ground track.
5. Create a new Moon-centered 2D window that allows you to observe the ground track.
6. Create a new Moon-centered landing site called "LandingSite".
7. Create new Calc Object "InLine" script called "Lunar Rev"
    a. Create new Calc object "Lunar Orbit Period" that is a copy of Keplerian Elems/Orbit Period but uses the Moon as CB.
    b. Duplicate the "Vbscript" calc object from the Calc Objects/Scripts folder and name it "VBscript Lunar Rev"
    c. Populate the "VBscript Lunar Rev" script with the "Lunar Orbit Period" calc object and the 'duration' calc object from the 'Time' folder.
    d. Enter the following in the "Inline Function" field:



NOTES

8. Change the stopping condition on the “Prop Twice Around” segment to use Lunar Rev.
    a. Insert the ‘User Select’ calc object
    b. Change the ‘User Calc’ to be the “VBscript Lunar Rev” calc object.
    c. Change the ‘Trip’ value for this stopping condition to 8.
    d. With the ‘Run only changed segments’ option selected, re-run the MCS.
9. In the properties of the 2D window, using the ‘Lighting’ page, turn on the ‘Fill’ for both the Penumbra and the Umbra.
10. Using the 2D window, adjust the date of launch until your ground tracks are near the landing site or adjust the landing site to be near the ground tracks. Move your facility to fall under the last ground track.



(Of course with a real mission you’d have to adjust the timing of your transfer to do this, and the number of revs, but we’ll cheat here)

11. Remove 3 revs from the ‘Prop Twice Around’ segment’s stopping condition trip.
12. Add a target sequence at the end of the MCS called ‘Target Down’

NOTES

a. Add a copy of the 'prop twice around' segment into the Targeter, name it 'phase', set the trip value to 1, and select it a control.

b. Add another targeter inside 'Target Down' called 'Target DOI'

   i. Add a impulsive Maneuver segment, called 'DOI'. Configure the burn to be Moon VNC, set the X as a control

   ii. Add a result to DOI of 'Lunar Periapsis Altitude'.

   iii. Set the profile of 'Target DOI' to target a periapsis altitude of -50 km with default tolerances.

   iv. Set up the targeter to 'Run Active Profiles'

c. Add a propagate segment after the Target DOI targeter called "Prop to Altitude" that propagates to a Lunar Altitude of 50 km.

d. Copy the update segment from the direct landing scenario before the Target Down targeter, and use the same engine model you used from that segment.

13. Duplicate the Finite maneuver segment from the direct landing scenario into a Targeter named 'Target Landing' that targets the maneuver efficiency to achieve an altitude of 4 feet at the end of the maneuver, and place this targeter inside the 'target down' targeter after the 'DOI' targeter'. Use a stopping condition (as before) of -4 ft/sec in Lunar altitude rate.

14. Set the 'Target landing' targeter to run active profiles.

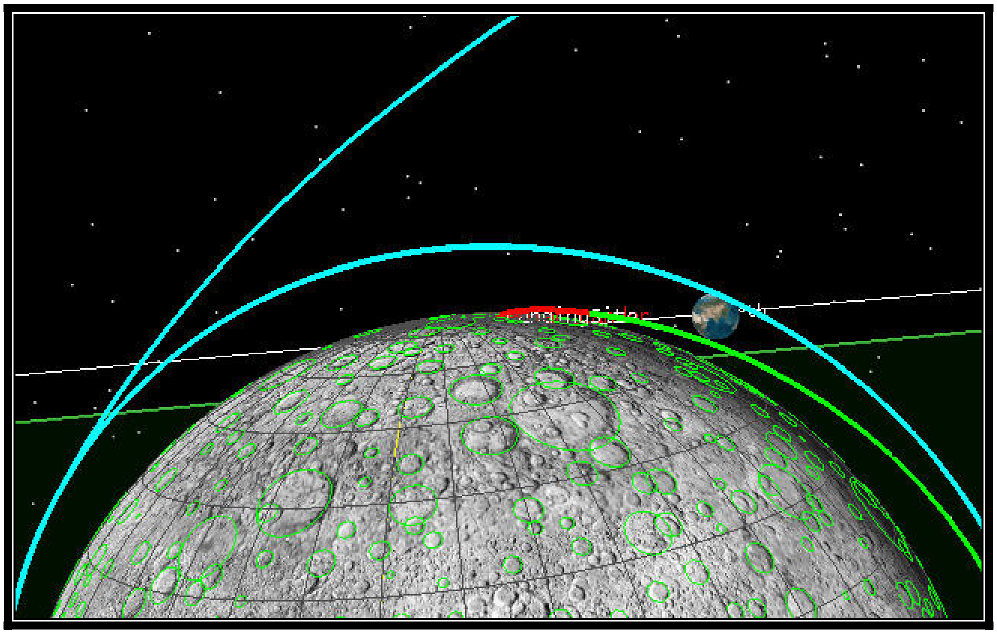
15. Add the results of Target RA and Target DEC to the 'Target Landing' targeter.

16. Configure the Target Down targeter with a profile that controls the trip value of the 'Phase' segment (0.1 rev perturbation, 1 rev max step) to achieve the Target RA and Target Dec constraints (0.01 deg tolerance) on the 'Target Landing' targeter. On the 'Convergence' tab of this profile, change the 'Convergence Criteria "Either Equality Constraints or Last Control parameter updates within tolerance'.

17. Add another Differential Corrector profile that uses the Phase segment again, along with the y component of a maneuver you create (0.0001 perturbation, 0.001 max step) at the beginning of the 'Target Down' target sequence to achieve the Target RA and Target Dec to the tolerances listed above.

18. Run the MCS.

NOTES



Now that you've Landed on the Moon, are you ready for induction into the Astrogator's Guild? Maybe.....

Find out more at: www.astrogatorsguild.com

Finally, if you need more help: www.see.com for all of your advanced Astrogator requirements.

NOTES