NASA JOHNSON SPACE CENTER INTERNSHIP

Flight Mechanics Project

Work Experience - Summer 2009

Daniel Steck

8/7/2009

Abstract: This report documents the generation of an outbound Earth to Moon transfer preliminary database consisting of four cases calculated twice a day for a 19 year period. The database was desired as the first step in order for NASA to rapidly generate Earth to Moon trajectories for the Constellation Program using the Mission Assessment Post Processor. The completed database was created running a flight trajectory and optimization program, called Copernicus, in batch mode with the use of newly created Matlab functions. The database is accurate and has high data resolution. The techniques and scripts developed to generate the trajectory information will also be directly used in generating a comprehensive database.

Table of Contents

List of Figures

List of Tables

Introduction

At NASA's Johnson Space Center, I worked in the Flight Mechanics and Trajectory Design branch (EG5) which is currently focused on the Constellation program. The Constellation program is an effort to once again send man to the Moon and eventually Mars. One of the initial missions of the program is to travel to the Moon as Figure 1 illustrates.

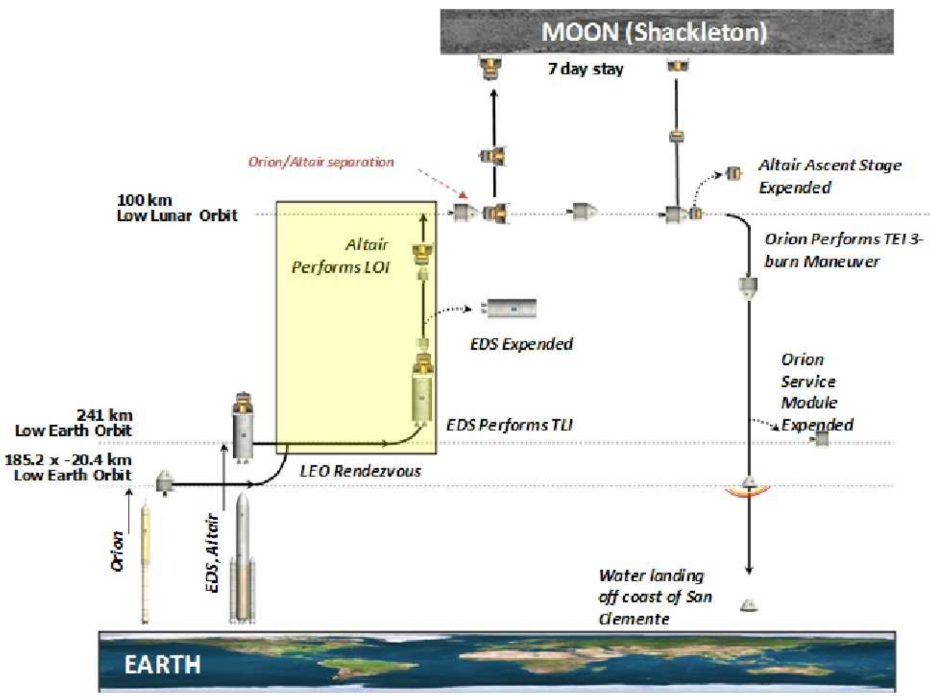

Figure 1: Diagram illustrating a mission of the Constellation program and the focus of my project

The mission consists of the Ares I rocket launching the Crew Exploration Vehicle (CEV), also known as Orion, first followed by the Ares V rocket launching the Earth Departure Stage (EDS) and the Lunar Lander, also known as Altair, from Earth (when Altair and the EDS are connected they are called the Altair-EDS stack). Once the Altair-EDS stack is in a Low Earth Orbit (LEO), Orion will rendezvous with the stack. The EDS performs the Trans-Lunar Injection

(TLI) burn which puts the vehicle on a four day transit to the Moon. After the TLI maneuver, the EDS is expended. Once the Altair-Orion stack has reached a desired position at the Moon, Altair performs the Lunar Orbit Insertion (LOI) burn which puts the stack into a Low Lunar Orbit (LLO) around the Moon. When this LLO is achieved, the stack will remain in that orbit for a period of one day for the crew to perform needed tasks and to rest. After this period, Altair and Orion will separate and Altair will descend to the surface of the Moon while Orion stays in a LLO for Altair's seven day stay on the Moon. After this time, the upper part of Altair (the Ascent Stage) will take the crew back to Orion and will subsequently be expended. Orion performs the Trans-Earth Injection (TEI) three burn maneuver which will transfer the vehicle from an orbit around the Moon to the Earth's surface. Prior to Earth entry, Orion expends the service module which allows only the Orion crew module to make a water landing off the coast of San Clemente, California.

Project description

The ultimate objective for my project is to create a database of flight trajectory information for the portion of the mission from TLI to LOI which is illustrated in Figure 2 and the highlighted region of Figure 1. A portion of the desired database was created using an older program called Earth Orbit to Lunar Orbit (EOLO), but it is more desirable to use a newer program called Copernicus to generate the database because small problems have been found in the EOLO generated database. Also, Copernicus can generate higher fidelity trajectory calculations and is a more flexible tool. The database will eventually be used to determine if it is cheaper to account for errors in the orbit around the Earth; specifically a Right Ascension of the Ascending Node (RAAN) error, an inaccuracy in "the point where the [shuttle] crosses through the [Earth's equatorial] plane in a northerly direction" (Bate, Mueller, & White, 1971), by way of the TLI maneuver or by making a plane change.

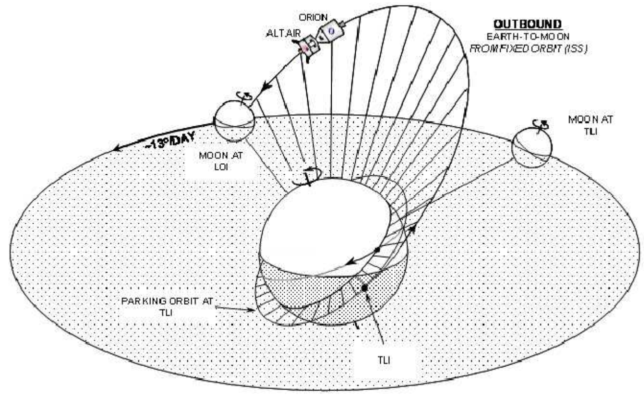

Figure 2: View of mission from Earth Orbit Departure to Lunar Orbit Insertion

The reason that there is the possibility for a RAAN error in the Earth orbit is that after Ares I launches, Ares V should launch approximately 90 minutes after; however, if Ares V is unable to launch on time there are four alternate launch times with the last one being approximately 72 hours after the nominal launch time. This delayed Ares V launch time causes Orion to be at the lower LEO altitude for a longer amount of time, to experience more effects of drag which will cause Orion's altitude to decrease which in turn causes Orion's orbit regression rate (rotation rate of the intersection line, which is called the line of nodes, between the Earth's equatorial plane and the vehicle's orbital plane) to increase. Because a plane change during rendezvous is very expensive, Ares V will be launched in order to put the Altair-EDS stack in the same plane as Orion. In the end, Orion's RAAN before rendezvous translates to an off nominal RAAN at TLI and causes the error.

The database will ultimately contain all the flight data required to reproduce the trajectory easily and will be used in another program called the Mission Assessment Post Processor (MAPP) which will use the data to rapidly create many flight trajectories. The trajectories that are desired are a nominal mission and its corresponding second, third, fourth, and fifth opportunities as well as the off-nominal (having a RAAN error) mission and its corresponding second, third, fourth, and fifth opportunities. All of these ten cases have four different trajectories because there are two possible Earth orbit RAANs and the possibility for a northerly or southerly arrival at the Moon. The database covers 19 years worth of trajectory data because the Earth-Moon geometry repeats every 18.6 years which is the length of the lunar nodal cycle. Because of the magnitude of the project, my goal for the semester was to complete the first portion of the database consisting of all four nominal first opportunity trajectories computed twice a day for a 19 year period. This data has already been generated in the EOLO database; however, it is necessary to re-generate this information in Copernicus in order to correct the EOLO errors and to ultimately generate the entire database using Copernicus.

Execution

Set up

I began the project by gaining introductory knowledge of orbital mechanics through reading selected chapters of <u>Fundamentals of Astrodynamics</u>, watching a presentation provided by my mentor Jerry Condon, and asking questions of others in my office including Tim Dawn, Dr. Juan Senent, Bob Merriam, Ellen Braden, and Ronald Sostaric. I then learned how to use and run Copernicus, which is a flight trajectory design and optimization tool. Through tutorials, I began to understand how to interpret the output data and determine if its solutions are reasonable or not. It was also necessary to get a better understanding of the precise problem formulation ground rules and constraints.

The nominal first opportunity Earth orbit to lunar orbit trajectory optimization problem formulation can be found in Figure 3.

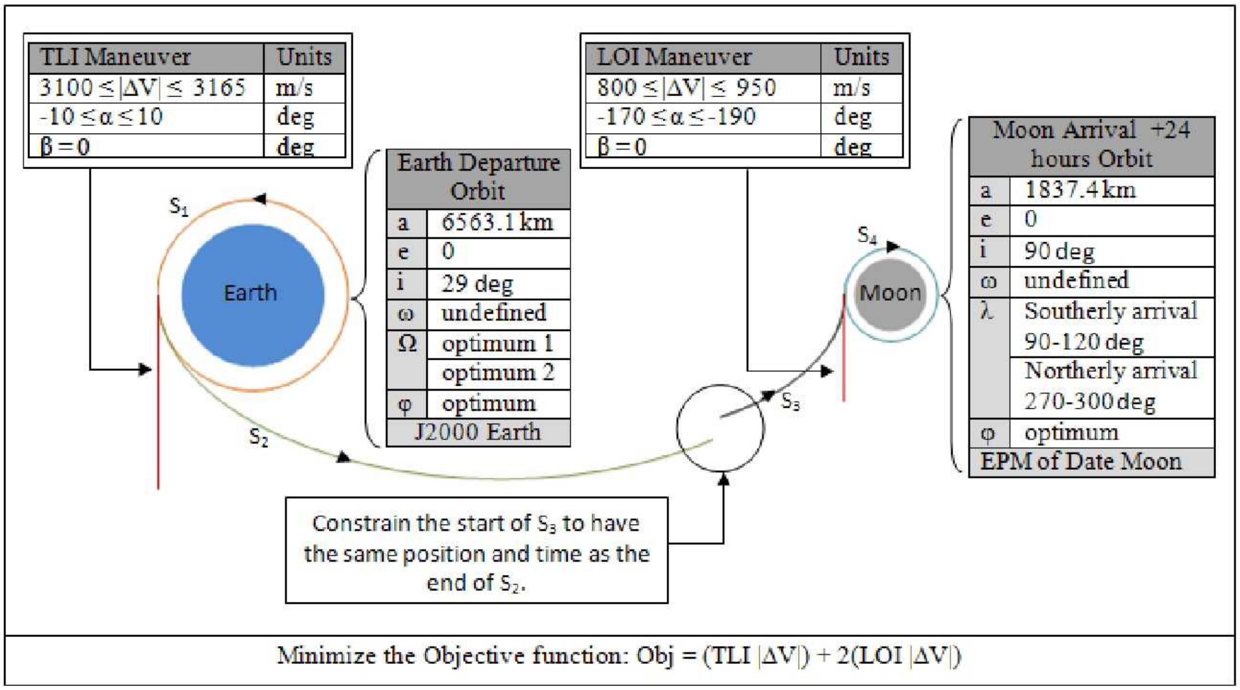

Figure 3: This figure defines the Earth orbit to Moon orbit trajectory optimization problem.

The orbital information is presented in classical orbital element notation:

"a" is the semi-major axis

"e" is the eccentricity

"i" is the inclination

"ω" is the argument of periapsis

"Ω" is the right ascension of the ascending node

"λ" is the longitude of the ascending node

"φ" is the true anomaly

The Earth departure orbit is defined in the J2000 Earth centered frame and the Moon arrival orbit (at LOI time plus 24 hours) is defined in the EPM of Date Moon centered frame. The ω is undefined because the orbit is circular which implies that there is no periapsis defined which is necessary to obtain an argument of periapsis. Although Segment four (S_4) is desired to end 24 hours after LOI, a simplification was made in order to allow Copernicus to obtain solutions rapidly. The simplification was to eliminate the 24 hour period which means the final lunar orbit is defined at LOI. This is a reasonable simplification for a polar or near polar mission because Altair is capable of performing up to a four degrees plane change to descend to the lunar surface. The maneuver information is presented in terms of its ΔV magnitude ($|\Delta V|$) or velocity increment and its α (pitch) and β (yaw) angles. The change in velocity magnitude is constrained to be between 3100 and 3165 m/s. 3165 m/s is the current upper limit based on the EDS fuel tank size estimates. The LOI velocity magnitude bounds are based on known maximum and minimums for a single burn LOI maneuver. When possible, it is beneficial to constrain an optimization variable (a value that is able to vary in order to find a solution) while still keeping generality over the study period so that the numerical solver does not get far off track which allows for a faster solution process. Alpha is the angle between the velocity vector of the spacecraft prior to the maneuver and the ΔV vector projected onto the EPO plane, this angle can be seen as the "in-plane" angle. The angle α is constrained between -10 and 10 degrees for the TLI maneuver and between -170 and -190 degrees for the LOI maneuver because it is known that the optimal solution will fall within these ranges and so that while solving the problem numerically the solver does not get so far off that it will not be able to find the solution. Beta, on the other hand, is the angle between the ΔV vector (illustrated by the red line in Figure 3) and the plane of the EPO, which can be thought of as the "out-of-plane" angle. It is constrained to zero because again we know that the best solutions (solutions requiring the least fuel) will have an out of plane component of velocity equal to zero because out of plane components are costly in fuel consumption generally. S_1, S_2, S_3, and S_4 are labels specifying different segments of the trajectory, while the arrows on each segment represent the direction of increasing time. As the figure illustrates, the spacecraft will start out at a LEO (S_1 in Figure 3) that is totally specified except for its RAAN, which can be one of two relative optima. The craft will then perform a single burn TLI maneuver which in my study will be approximated by an instantaneous burn (a maneuver which changes the velocity of a craft in an infinitesimal amount of time) that will propel the spaceship out of the LEO and into the first leg of the transfer orbit to the Moon (S_2). The state at the LLO (S_4) is completely specified except for the longitude of the ascending node which is an optimization variable with bounds that specify a northerly or southerly arrival to the Moon. The time of flight from TLI to LOI is also fixed to be four days for this study. A key that should be seen from the LLO is that it is a polar orbit (an orbit which travels over both poles of the Moon). Segment one, two, and four are all propagated forward in time; however, segment three is propagated backwards in time to make it easier for Copernicus to solve the problem by minimizing the number of constraints, moving the point of constraint away from any perturbing bodies, and allowing for trajectory calculations for each segment to take into account a gravity model that is more accurate than a point mass gravity model for the body whose non-spherical shape would have most influence on it. For example, Copernicus models segment three with an 8x8 Moon gravity model (a model which takes into account some of the non-spherical shape attributes of different planets) and a point mass Earth gravity model for the trajectory calculation because the Moon is close and the Earth is far in S_3 which causes the Moon's non-spherical attributes to have a larger effect than the Earth's. Table 1 details which gravity models are to be used in each segment for the Earth and the Moon. At the end of S_3 (the second leg of the transfer orbit) is the LOI maneuver (again an instantaneous burn) which takes the craft from the transfer orbit to the LLO. The only constraints in the problem are at the beginning of segment three which are that its position and the time at that point must be equal to the position and time of the point at the end of segment two. Once the problem is solved, the gap between segment two and three will be closed and they will form one smooth transfer trajectory. The solution to the problem is not just finding a trajectory that is feasible, but finding a solution that is optimal. An optimal solution is one which requires the least fuel consumption which is proportional to the change in the velocity magnitude. In this case, it was chosen to minimize the objective function Obj=(TLI $|\Delta V|$)+2(LOI $|\Delta V|$) which essentially maximizes the TLI $|\Delta V|$ and minimizes the LOI $|\Delta V|$. The problem is solved when all of these conditions are within a desired accuracy.

Table 1: Gravity model specification for each segment

Segment	Earth	Moon
S_1	8x8	1x0 (Point)
S_2	8x8	1x0 (Point)
S_3	1x0 (Point)	8x8
S_4	1x0 (Point)	8x8

From the problem formulation, it is clear that there will be four 19 year databases constructed namely the optimal RAAN 1 Southerly arrival case, the optimal RAAN 2 Southerly arrival case, the optimal RAAN 1 Northerly arrival case, and the optimal RAAN 2 Northerly arrival case (a Northerly arrival, Southerly arrival, RAAN 1, and RAAN 2 can be seen in Figure 6 through Figure 9 on page 21 in the appendix). The outlined problem formulation will be kept for all of the generated data points except for the few cases when the Moon is at its highest declination during the ~19 year lunar cycle. At these times, it will sometimes be necessary to either just free the TLI β and the Earth parking orbit RAAN or free both of those and the TLI magnitude in order to get a trajectory for those times.

Batch processing

After the base input deck was constructed containing all of the previously mentioned specifications, a program was needed to generate a large database of trajectory information containing four nominal 19 year scans with data computed every 12 hours. For this, I was able to use the newly created Matlab functions which enable a user to manipulate, run, and record Copernicus files. I first had to learn the basics of Matlab programming and learn how the Copernicus Matlab functions operate by working with the commands and looking at an example program. After learning these fundamentals, I was able to begin writing a program. Figure 4 explains in words how the main program works.

Call main program and pass in the initial epoch, final epoch, and the epoch step.

Set current epoch equal to initial epoch

While current epoch is less than or equal to the final epoch

    If current epoch is equal to initial epoch

        Convert the base input deck which is assumed to contain all the problem specifications and have a starting time equal to the initial epoch into a Matlab structure and call it current ideck

    Else

        Convert the input deck generated by the previous run through the while loop into the Matlab structure called current ideck

        Add the epoch step to the starting time in the current ideck

    End

    Update the values in current ideck in case the starting time was changed

    Generate a guess at the optimum RAAN 1 or 2 and its bounds depending on whether RAAN 1 or 2 was passed into the program by the user (see page 13 for an explanation of the RAAN guess generator)

    Update the RAAN value and bounds in the current ideck structure

    Reset parameters to be in agreement with the problem formulation incase the previous solution was one of the cases where the normal problem formulation has no solution

    Create an input deck file from the current ideck Matlab structure which can be run in Copernicus

    Solve the input deck using Copernicus through a series of trials that will be detailed in another section of this paper (see page 14 for further explanation)

    Update the Matlab structure with the values from the solved/optimized Copernicus file

    Get desired data from Matlab structure

    Print header to CSV file if current epoch is equal to initial epoch

    Print new line to CSV file with desired data

    Define the previous file as the Copernicus file that was solved in this run through the while loop

End

Figure 4: Outline of main program

There are two portions that are not described in depth in the figure and deserve more explanation; these items are the RAAN guesser and the solution portion of the main program. Principles of orbital mechanics were used extensively in order to create these portions of the desired script. The RAAN guesser generates a guess as to what RAAN 1 and 2 should be and then sets up the RAAN bounds as ± 16 degrees from the calculated RAAN guess. The RAAN guess is calculated by simplifying the problem to a two body problem and knowing a little about orbital mechanics. The idea is this: it is known that for these nominal trajectories the cheapest path is going to be one in which there is no out of plane maneuver for the TLI and LOI burns. This means that without taking into account the Moon's gravity, the optimal LEO position is where its plane contains the Moon at the desired date of LOI. Since Equation 1 gives the direction of the angular momentum vector in terms of inclination (i) and RAAN (Ω) which is a vector normal to the EPO plane, and since it is possible to obtain the position vector of the Moon at the desired time of LOI for these nominal missions, it is possible to take the dot product of these two vectors and set the resultant to zero. The dot product is set equal to zero because it is desired that the Earth parking orbit plane contain the Moon's position vector which means that the Moon's position vector is orthogonal to the LEO's angular momentum vector. Once this is solved an equation is derived with one unknown, RAAN, and two solutions (as can be seen in Equations 2 and 3) that is a function of the Moon's state and the EPO inclination. In Equations 2 and 3, RA and DEC stand for Right Ascension and Declination of the Moon at the time of LOI respectively. Figure 5 provides a visualization of how each of the two RAAN options changes the flight trajectory.

$$\boldsymbol{h} = \begin{pmatrix} sin(i) * sin(\Omega) \\ -sin(i) * cos(\Omega) \\ cos(i) \end{pmatrix} \quad 1$$

$$\Omega_1 = RA_{Moon} + arcsin(\frac{-tan(DEC_{Moon})}{tan(i)}) \quad 2$$

$$\Omega_2 = RA_{Moon} + \pi - arcsin(\frac{-tan(DEC_{Moon})}{tan(i)}) \quad 3$$

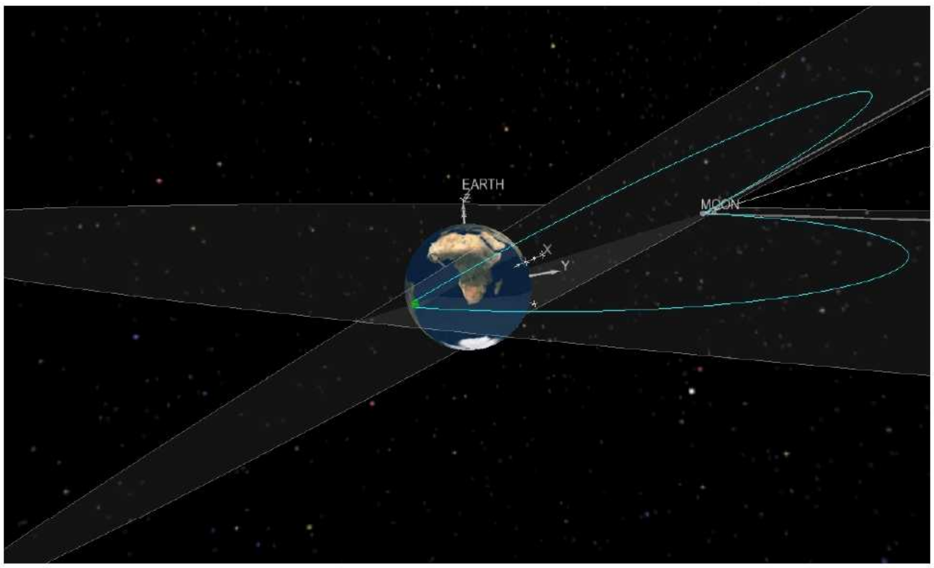

Figure 5: Visualization of two optimal RAAN options

The solution portion of my program goes through a process by which there are multiple possibilities of solution methods. Before getting into the outline of the code, it is necessary to understand that when I specify that an input deck will be attempted to be solved the code actually attempts to solve it four times. In the code of the solution process, I first try to solve the input deck in Copernicus and if it cannot be solved, then determine if the Moon is at a maximum declination (> 26.5 degrees) and the trajectory is trying to make a Northerly arrival at LOI or if the Moon is at its minimum declination (< -26.5 degrees) and the trajectory is trying to make a Southerly arrival at LOI. The reason that we check for either of these conditions is because for coplanar trajectories (as is assumed in the normal solution method) the achievable declinations are related to the inclination of the trajectory plane in a manner that an inclination of i_1 corresponds to an attainable declination of $\pm i_1$. Because the inclination of the EPO is specified in the problem description to be 29 degrees and the maximum and minimum declination of the Moon's center of mass is roughly ±28.7 degrees respectively it first seems that a coplanar trajectory to the Moon is always possible; however, this is not true because the Moon is not a point mass. For a northerly arrival and maximum Moon declination or southerly arrival and minimum Moon declination, the target point will be at a point which violates the $-i_1 \leq \text{DEC} \leq + i_i$ constraint. The previous discussion did not take into account the effects of the Moon's gravity which causes this effect to occur more often which is the reason that an assumption that anything less than -26.5 degrees or greater than 26.5 degrees is to be considered minimum or maximum declination. If the Moon is at a maximum declination and the trajectory is trying to make a Northerly arrival at LOI or the Moon is at its minimum declination and the trajectory is trying to make a Southerly arrival at LOI, then the EPO RAAN and TLI β are freed and the input deck is attempted to be solved again. If the input deck meets the criteria to free the EPO RAAN and TLI β and cannot solve the problem after making these changes, then the TLI magnitude upper bound is freed and the problem is again tried to be solved. If the reason that the first solution trial did not work is other than the aforementioned reasons, then the solving routine tries to move the beginning of S_3 away from the Moon and attempts to solve again. If after these solution techniques Copernicus is still not able to solve the input deck, there is a problem and the program will end. However, if the problem was able to be solved and the absolute value of the TLI β angle is less than one degree but not equal to zero, then the solution process will zero the TLI β and try to solve again. If it is not able to be solved by this method, then the input deck will be reverted back to what it was before trying to zero the TLI β and will be solved again; however, if this is not possible then the program will exit. If on the other hand, the TLI β is able to be zeroed and the input deck solved, then the original problem specification will be inserted back into the input deck which gives the input deck the same problem specification as the first time the input deck was attempted to be solved, however, this time the trajectory that Copernicus is starting with is close to the solution. An attempt to solve this input deck is made. If it cannot be solved, then we revert to freeing the EPO RAAN and TLI β and try to solve again. If this cannot be done, then the TLI magnitude upper bound is freed and an attempt to solve the input deck is made. If after this, the input deck cannot be solved then there is a problem and the program will exit. At the end of this entire solution process, a solved problem will come out that has been vigorously tried to be solved using the normal problem formulation, and only if a solution is not possible or too costly to find, then a converged and optimized solution to a problem specification slightly off from that described in Figure 3 will be generated.

The script I developed was necessary to generate the data for my portion of the database and will be used directly in further development of the database. Essentially this script will be used to create the entire database MAPP needs in order to rapidly produce Earth to Moon trajectories.

Results

The generated database consisting of optimal RAAN 1 Southerly arrival, optimal RAAN 2 Southerly arrival, optimal RAAN 1 Northerly arrival, and optimal RAAN 2 Northerly arrival trajectories is the true result of this project. The recorded parameters for each trajectory can be seen in Table 2. The chosen parameters allow for easy reproduction of any trajectory, the desired information that MAPP will use (namely the V infinity data), and the solution method. The database also showed that if a space craft is going to get into an orbit around the Moon for a nominal mission it will be at either a LAN between 95 and 116 degrees for a Southerly arrival or between 275 and 296 degrees for a Northerly arrival (as illustrated in Figure 10). Figure 11 is a plot of the objective function for the full 19 year period for all four cases. This style of plot is divided into quadrants with the top left being the Option 1 (RAAN 1) Southerly arrival case, the top right being Option 2 (RAAN 2) Southerly arrival case, the bottom left being the Option 1 Northerly arrival case, and the bottom right being the Option 2 Northerly arrival case. Figure 11 shows blips (points which go off the chart) in the data in the 2025 time period. This is expected because this is the time period when the Moon is at its maximum inclination and the Moon is intermittently at its maximum and minimum declination, which can cause coplanar trajectories not to be achievable. In those instances, the necessary variables were freed according to the solution method and solved. These data points which go off the chart area correspond to actual trajectories but are not solved by the normal problem formulation detailed in Figure 3. It is also interesting to note that for Option 1 trajectories (with either northerly or southerly arrivals) the times when the Moon is at its maximum or minimum declination corresponds to times in which the optimization function is least variable. However, for Option 2 trajectories the opposite effect is seen. Figure 12 shows the objective function for only 2018. This figure has very high resolution because two trajectories were calculated per day and the cyclic variation of the objective function from maximum to minimum is quite rapid. Figure 13 is a plot of the TLI ΔV magnitude which shows that for the recorded trajectories solved by the normal problem formulation the ΔV magnitudes range from 3128 to 3148 m/s which is well below the 3165 m/s maximum value. The LOI ΔV magnitude plot is shown in Figure 14 where the same maximum/minimum inclination consistency or inconsistency can be seen which accounts for the same trend seen in Figure 11. Figure 14 also shows that LOI ΔV magnitudes range between 810 and 880 m/s. The trajectory calculations of Copernicus and EOLO are compared by their TLI and LOI magnitudes as seen in Figure 15 which shows great similarity between the two. The difference in the LOI magnitudes is generally between -20 and 3 cm/s while the TLI magnitude differences are between -13 and 1 cm/s. The negative points correspond to times when EOLO is giving a more optimistic result than Copernicus and the positive points correspond to times when Copernicus is giving a more optimistic result than EOLO. The exact reason for the differences has not been determined; however, one of the proposed reasons is that EOLO and Copernicus have different gravity models where EOLO uses a 2X0 gravity model for all bodies and Copernicus uses an 8X8 gravity model on the closest body and a 1X0 gravity model on distant bodies. Another proposed reason is that Copernicus uses a more accurate ephemeris than EOLO.

Table 2: Details trajectory parameters recorded in database

Database Parameter(s)	Coordinate Frame
TLI and LOI Epoch	------
Time of Flight	------
preTLI state vector	J2000 Earth
preTLI RAAN and both RAAN guesses	J2000 Earth
TLI maneuver	ijk J2000 Earth
	vuw J2000 Earth
Vinf Mag, RA & DEC	J2000 Moon
	EPM of Date Moon
preLOI state vector	j2000 Moon
postLOI LAN	EPM of Date Moon
LOI maneuver	ijk J2000 Moon
	vuw J2000 Moon
Solution method	------

Conclusion

The objective of my project were met; I created a preliminary database, using Copernicus, for a four day TLI to LOI transfer with the data closely matching previous results generated in EOLO. The techniques developed during my summer Co-Op set the stage for the development of a comprehensive database to include multiple TLI opportunities as well as their corresponding off-nominal trajectories. Specifically, the Matlab script will be directly used while the input deck can be used with minor modification. The RAAN guesser was found to work very well at keeping each case within the proper RAAN family and the provided RAAN guess was at a maximum of 16 degrees off while the usual difference was significantly less. I learned many things during my Co-Op including scripting in Matlab, how to run Copernicus and interpret its data, how to submit jobs to a cluster of computers (distributed processing), many new Excel capabilities, the basics of orbital mechanics, and how difficult it is to solve an Earth to Moon trajectory problem even with the sophisticated technology available to us today.

APA Right Now

Bibliography

Bate, R. R., Mueller, D. D., & White, J. E. (1971). *Fundamentals of Astrodynamics.* New York: Dover Publications, Inc.

Appendix

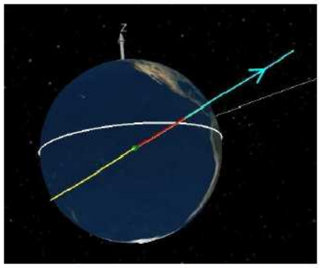
Figure 6: Illustrates EPO RAAN 1

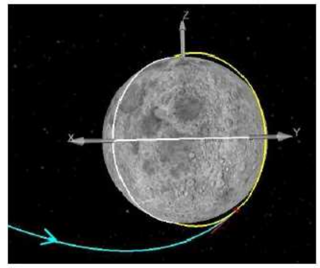
Figure 7: Illustrates Southerly arrival at Moon

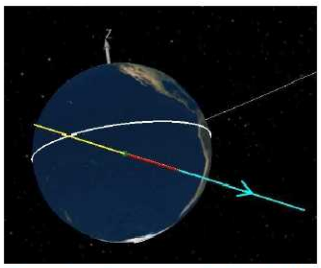
Figure 8: Illustrates EPO RAAN 2

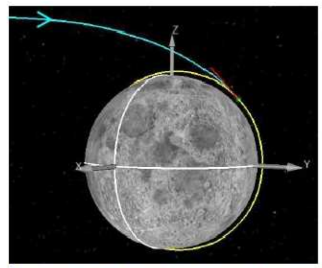
Figure 9: Illustrates Northerly arrival at Moon

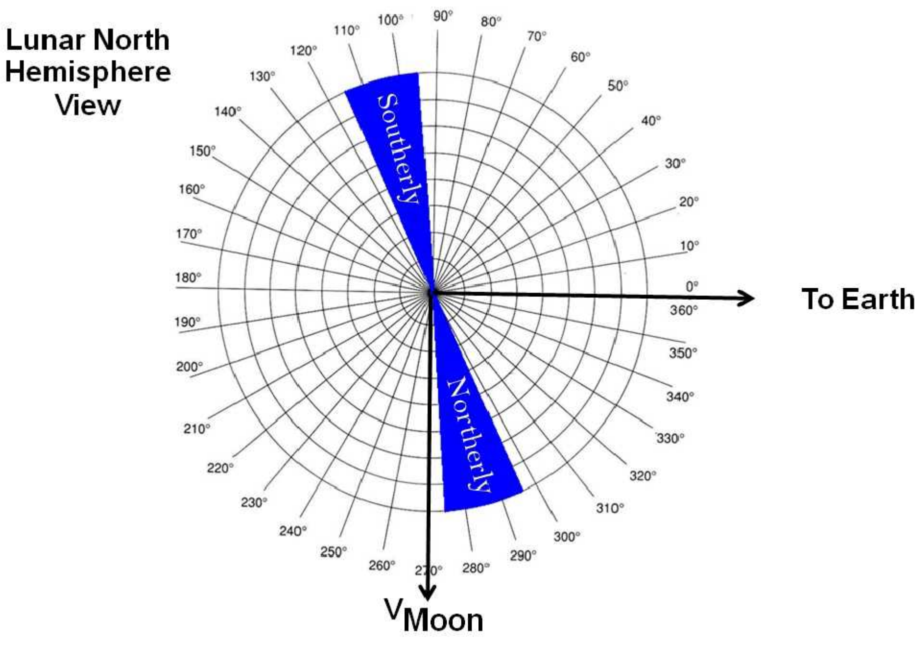

Figure 10: Illustrates the Lunar parking orbit LAN for the generated database for Southerly and Northerly arrivals

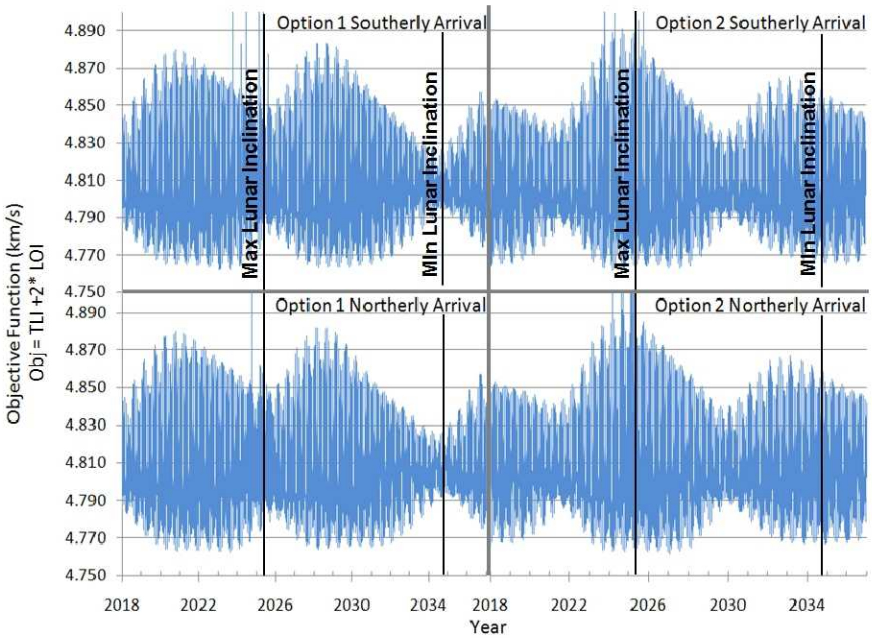

Figure 11: Plot of objective function for full 19 year cycle

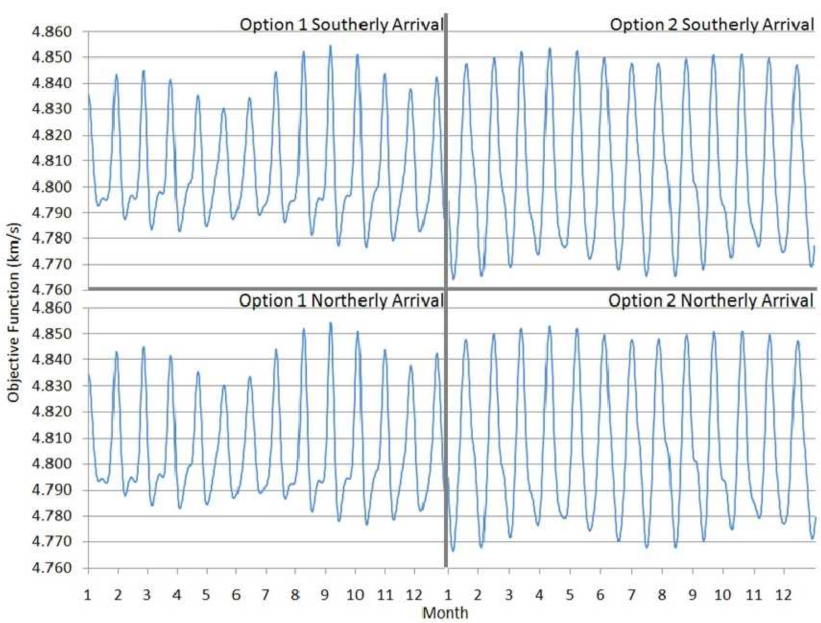

Figure 12: Plot of objective function for the year of 2018

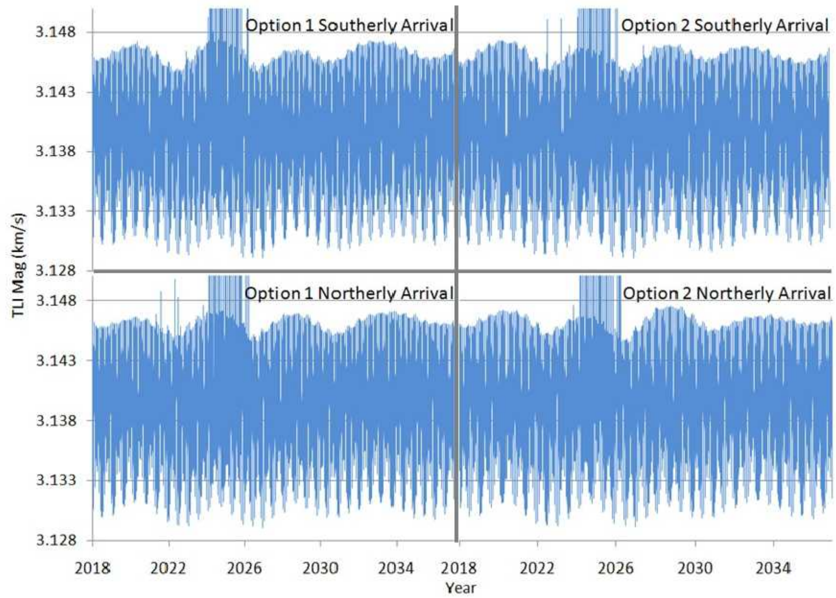

Figure 13: Plot of TLI magnitude for full 19 year cycle

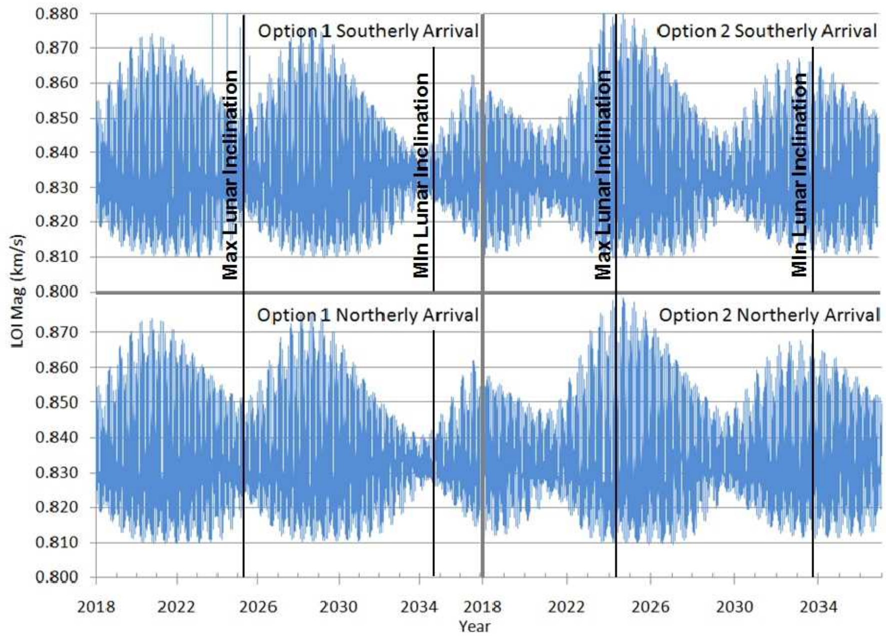

Figure 14: Plot of LOI Magnitude for full 19 year cycle

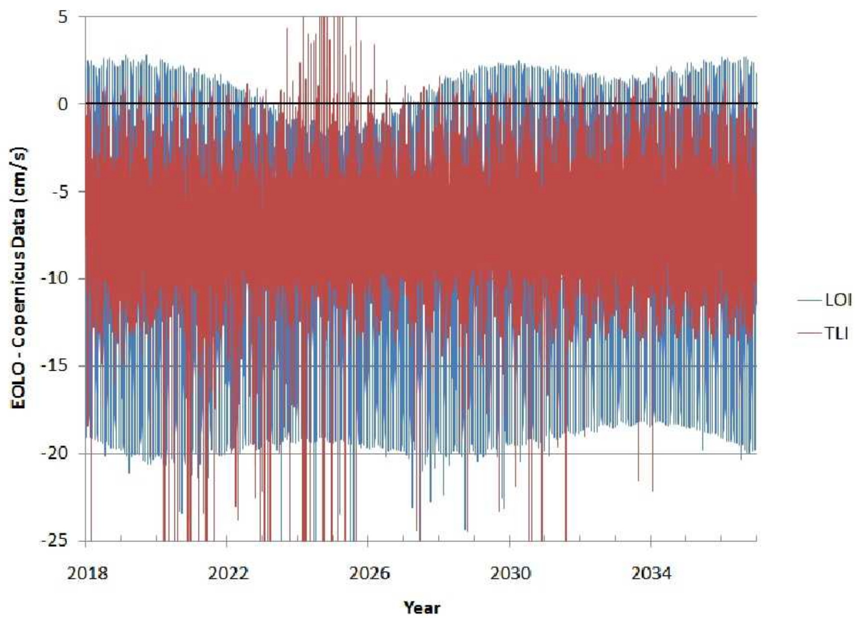

Figure 15: Details the differences between the TLI and LOI magnitudes computed by EOLO and Copernicus

Needs bottom axis label