NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MINIMIZING STAR TRACKER OCCULTATIONS FOR NASA'S LUNAR RECONNAISSANCE ORBITER IN SUN-SAFE MODE

by

Mitchell A. Kempisty

December 2021

Thesis Advisor: Mark Karpenko
Co-Advisor: Isaac M. Ross

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
MINIMIZING STAR TRACKER OCCULTATIONS FOR NASA'S LUNAR RECONNAISSANCE ORBITER IN SUN-SAFE MODE

5. FUNDING NUMBERS

6. AUTHOR(S) Mitchell A. Kempisty

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Due to degradation of the NASA's Lunar Reconnaissance Orbiter's (LRO) Inertial Measurement Unit, the LRO relies solely on its star trackers to maintain gyroless attitude control. In the event of an anomaly, the LRO is placed in sun-safe mode in order to reestablish normal operation, which constrains its attitude. While in sun-safe mode, the LRO's star trackers experience occultations from local orbiting bodies and cannot maintain an attitude solution during these periods. This poses the risk of total loss of the spacecraft due to tumbling or depleted power supply. This thesis provides mission operators with a software-based tool for determining alternate sun-safe attitudes that reduce the occultation time, minimizing operational risk. Ephemeris data from orbiting bodies and the LRO are utilized to investigate occultation occurrences. Periods of star tracker occultations for any given time frame are determined based on the LRO's fixed attitude. The goal of this thesis is to iterate alternate attitudes to define the ideal attitude that minimizes occultation occurrences. Additionally, data analysis is conducted to determine the ideal attitude update frequency for sun-safe mode based on operational constraints. The design of this software-based tool yields appropriate results for acquiring an ideal attitude solution for minimizing star tracker occultations, giving mission operators the freedom to choose attitude constraints, simulation fidelity, and attitude update frequency.

14. SUBJECT TERMS
aeronautics, space, NASA, satellite, lunar, reconnaissance, orbiter, LRO, position, navigation, attitude, star tracker, obstruction, obscuration, sun, sensor

15. NUMBER OF PAGES
97

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

MINIMIZING STAR TRACKER OCCULTATIONS FOR NASA'S LUNAR RECONNAISSANCE ORBITER IN SUN-SAFE MODE

Mitchell A. Kempisty
Lieutenant, United States Navy
BS, United States Naval Academy, 2014

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2021

Approved by: Mark Karpenko
Advisor

Isaac M. Ross
Co-Advisor

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Due to degradation of the NASA's Lunar Reconnaissance Orbiter's (LRO) Inertial Measurement Unit, the LRO relies solely on its star trackers to maintain gyroless attitude control. In the event of an anomaly, the LRO is placed in sun-safe mode in order to reestablish normal operation, which constrains its attitude. While in sun-safe mode, the LRO's star trackers experience occultations from local orbiting bodies and cannot maintain an attitude solution during these periods. This poses the risk of total loss of the spacecraft due to tumbling or depleted power supply. This thesis provides mission operators with a software-based tool for determining alternate sun-safe attitudes that reduce the occultation time, minimizing operational risk. Ephemeris data from orbiting bodies and the LRO are utilized to investigate occultation occurrences. Periods of star tracker occultations for any given time frame are determined based on the LRO's fixed attitude. The goal of this thesis is to iterate alternate attitudes to define the ideal attitude that minimizes occultation occurrences. Additionally, data analysis is conducted to determine the ideal attitude update frequency for sun-safe mode based on operational constraints. The design of this software-based tool yields appropriate results for acquiring an ideal attitude solution for minimizing star tracker occultations, giving mission operators the freedom to choose attitude constraints, simulation fidelity, and attitude update frequency.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AGI Analytical Graphics, Inc.

BCS body coordinate system

CRaTER Cosmic Ray Telescope for the Effects of Radiation

CSV comma separated value

DLRE Diviner Lunar Radiometer Experiment

ECI earth centered inertial

EPROM erasable programmable read-only memory

FOR field of regard

FOV field of view

ICRF International Celestial Reference Frame

ICRS International Celestial Reference System

IERS International Earth Rotation and Reference Systems Service

IMU Inertial Measurement Unit

JPL Jet Propulsion Laboratory

LAMP Lyman-Alpha Mapping Project

LCROSS Lunar Crater Observation and Sensing Satellite

LEND Lunar Exploration Neutron Detector

LOLA Lunar Orbiter Laser Altimeter

LRO Lunar Reconnaissance Orbiter

LROC	Lunar Reconnaissance Orbiter Camera
NASA	National Aeronautics and Space Administration
NPS	Naval Postgraduate School
OCS	orbital coordinate system
SC	spacecraft
ST	star tracker
ST1	star tracker 1
ST2	star tracker 2
STK	Systems Tool Kit
ULA	United Launch Alliance

Acknowledgments

I would like to thank Dr. Mark Karpenko for the friendly guidance and encouragement he gave me throughout this journey. From the moment I first met with him to discuss this project, I knew we would be a good team. The most important thing for me was the creative engineering freedom he granted me to find a solution to the problem. With that freedom, this thesis was not a cookie-cutter academic requirement. Instead, it became an adventure of discovery, originality, and creation of my own hands with the use of all the knowledge I gained at the Naval Postgraduate School. That is the greatest kind of engineering experience of all, and I will be forever thankful for Dr. Karpenko's ability to help me find that.

I would also like to thank Sean Boyer (Systems Engineer), Valerie Lesser (Systems Engineer), and Anna Garcia (Corporate Engineer) at Analytical Graphics, Inc. (AGI) for painstakingly working with me on particular problems I ran into while integrating AGI related tools for this thesis. Each of them spent countless hours brainstorming with their team on how to resolve issues when I hit technical road blocks. In order for them to help, they had to work to clearly understand the problem and abstract approach I was using. Their patience and attention to this effort will always be appreciated.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 History of the LRO

NASA's Lunar Reconnaissance Orbiter (LRO) was launched along with the Lunar Crater Observation and Sensing Satellite (LCROSS) onboard a United Launch Alliance (ULA) Atlas V 401 variant rocket from Cape Canaveral, Florida, on 19JUN09, as the first U.S. Moon missions in over a decade [1]. Beginning with the LRO's Exploration Mission on 15SEP09, the objective was centered around facilitating the expansion of human presence and activity beyond Earth. The LRO is a key contributor to the identification of resource heavy and terrain friendly landing sites for autonomous and manned missions to the Moon in the future. The Exploration Mission concluded on 15SEP10, and the LRO's Science Mission began, which aimed to improve the overall understanding of the Moon through the use of many technically innovative scientific instruments [2]. The scientific instruments onboard the LRO include the Cosmic Ray Telescope for the Effects of Radiation (CRaTER), the Diviner Lunar Radiometer Experiment (DLRE), the Lyman-Alpha Mapping Project (LAMP), the Lunar Exploration Neutron Detector (LEND), the Lunar Orbiter Laser Altimeter (LOLA), the Lunar Reconnaissance Orbiter Camera (LROC), and the Mini-RF Miniature Radio Frequency Radar [1]. These instruments have proven valuable in providing scientific data to be studied for decades to come. A notable achievement of the science gathered from these instruments is the discovery of water ice at the Moon's poles, primarily concentrated in permanently shadowed areas such as craters. The LRO currently maintains a polar orbit of 20 km at perilune, and 165 km at apolune, to provide a close perspective for the onboard scientific instruments. The LRO's Inertial Measurement Unit (IMU) was shut down in MAY18 due to evidence of degradation to the component, which renders the LRO dependent on its star trackers (STs) alone for obtaining an attitude solution [1].

1.2 Thesis Objectives and Scope

The objective of this thesis is to develop a software-based tool that will determine a fixed attitude for the LRO while in Sun-safe mode, that results in the least amount of total Moon occultation time of both STs simultaneously. In determining the ideal attitude within this scope, it is assumed that the LRO's solar panel is limited to a maximum offset of up to 15° perpendicular to the Sun vector. This allows the LRO to receive adequate solar energy during its orbit while in Sun-safe mode. Since the solar panel faces along the $-y$-axis face of the spacecraft (SC), the $-y$-axis becomes the vector of interest. With the combined use of Systems Tool Kit (STK) and original Matlab code, the LRO's ST Moon occultation instances and durations are simulated over a period of one year, from 03JUN21, to 03JUN22. These simulations are iterated at various fixed $-y$-axis offsets from the Sun vector and various rotation angles about the $-y$-axis. The extent of iterations used depends on the desired solution fidelity. In the case of this thesis, 2,880 iterations of different attitudes are simulated within the allowable 30° cone of freedom to determine the ideal attitude that results in the minimum amount of simultaneous ST occultations by the Moon.

It is true that the Sun and Earth also periodically induce ST occultations. However, the motion of the Sun and Earth with respect to the LRO is slow relative to that of the Moon. Also, the Moon is much closer to the LRO, such that the ST occultation severity is much higher than that of the Sun and Earth. Therefore, Moon occultations are the primary concern of this thesis.

In addition to determining an ideal Sun-safe attitude for the LRO, the necessary amount of periodic attitude updates are explored. First, the Sun-safe attitude is determined quarterly, which means that the LRO will alter its flight software stored Sun-safe attitude four times each year. Given the changing beta angle of the LRO's orbit throughout the year, the periodicity of attitude updates can be a point of concern. The total ST occultation time can theoretically be minimized further by updating the ideal attitude more frequently. With this point considered, the same simulation is run eight times throughout the year-long time frame to determine how update frequency effects results.

1.3 Thesis Outline

In Chapter 2, the relevant orbital mechanics and LRO technical information are explained. This lays the foundation of knowledge required to understand and follow the methodology and approach used to develop this tool and resolve the problem. In chapter 2, definitions of useful coordinate systems, ST configurations, and orbital mechanics relevant to the solution, are briefly explained. In addition, the online database used to acquire and integrate ephemeris data of solar system objects, named Jet Propulsion Laboratory (JPL) Horizons, is explored.

In Chapter 3, the entirety of the methodology and rationale to the determined solution is explained. First, the technical environment for the solution is defined, where the LRO's real-world specifications and requirements are considered in order to properly integrate between JPL Horizons, STK and Matlab. Then, a method of automation is described. The solution utilizes a form of simulation automation that is achieved by applying engineering mathematics to systematically organize an iteration scheme. Finally, multiple attitude update frequency schemes are explored and rationalized. It is important to consider the frequency in which the attitude is updated throughout the year, due to its considerable effect on the results.

In Chapter 4, the results following the methodology are presented and discussed. Furthermore, even greater improvement in results are presented under the theoretical condition in which some technical restrictions are broadened. Concluding remarks and suggestions for future work are presented in Chapter 5.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background

In this chapter, the background knowledge of orbital mechanics and engineering required to understand the methodology behind the development of this thesis is discussed. Relevant coordinate systems, an overview of the LRO's STs, a basic understanding of Solar beta angle, and an introduction to JPL Horizons, covers the necessary information that drives the conceptual approach to the solution.

2.1 Coordinate Systems

It is necessary to define two coordinate systems in order to reference the attitude of the LRO and determine the position of the STs relative to the SC.

The International Celestial Reference Frame (ICRF) coordinate system is a realization of the International Celestial Reference System (ICRS), managed by the International Earth Rotation and Reference Systems Service (IERS). An inertial coordinate system referenced from the Solar System barycenter, the triad of axes is based on a collection of 4,536 extra-galactic radio sources, mostly associated with the shock region of quasars [3]. This coordinate system is the inertial frame used to orient the LRO in this thesis. The LRO's ephemeris data, exported from JPL Horizons, uses this coordinate system to simulate the SC's orbit in STK. Also, the LRO's attitude relative to the Sun vector references this coordinate system. A depiction of the ICRF is displayed in Figure 2.1.

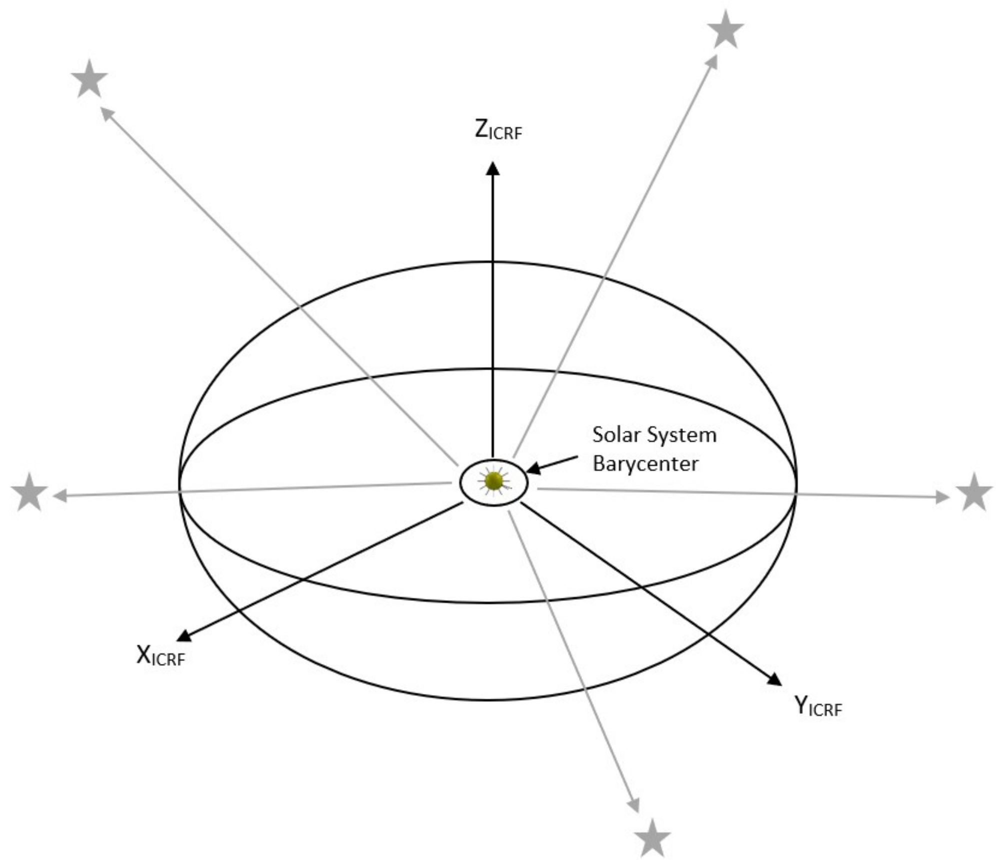


Figure 2.1. International Celestial Reference Frame. Adapted from [4].

The body coordinate system (BCS) is the coordinate system that defines the non-inertial axes of the SC. As presented in Figure 2.2, the x-axis is defined as normal to the separation plane, extending through the SC. The z-axis is defined as normal to the SC's Nadir face, and the y-axis is defined using the right hand rule based on the z and x-axis [5]. It is important to note that the $-y$-axis is normal to the SC's solar panel face when it is retracted in Sun-safe mode. The $-y$-axis is therefore the axis of primary concern. The STs are fixed relative to the BCS, as detailed in Section 2.2.

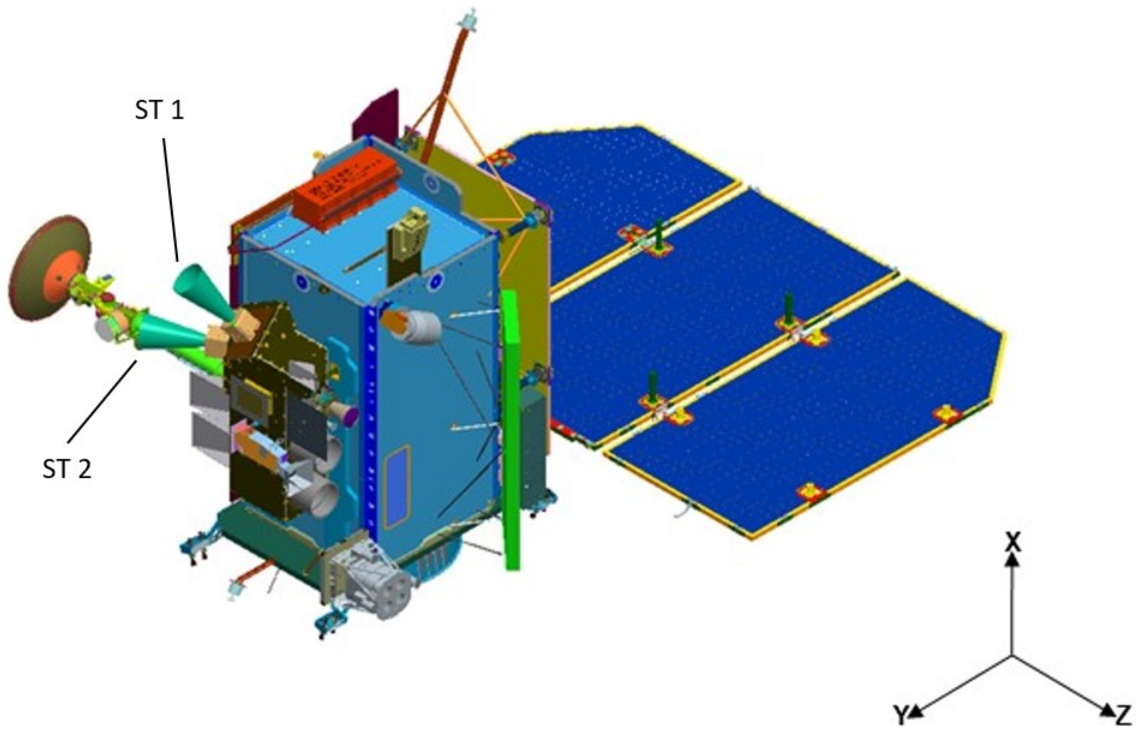


Figure 2.2. Body coordinate system. Source: [5].

2.2 Star Tracker Configuration

The LRO utilizes two Galileo-Avionica STs for SC attitude determination. Since degradation to the IMU has caused it to be effectively inoperable, the two STs onboard the LRO are currently the only sensors available for definitive attitude determination. Each ST has an 8.2° half angle field of view (FOV), but uses a 15° half angle field of regard (FOR) for additional standoff from the Moon. The FOR for the Earth is 15° as well, while the FOR for the Sun is 25°. The FOR with respect to the Earth and Sun are not considered since the scope of this thesis focuses on Moon occultations only. The LRO's STs are aligned to the SC with reference to the BCS. The 213 Euler rotation sequence that references the position of the BCS to the boresight of star tracker 1 (ST1) and star tracker 2 (ST2) are listed in Table 2.1 [5]. Figure 2.3 graphically demonstrates the positioning of the STs according to the BCS. This information is used in this thesis to establish sensor parameters for each ST in STK, coded in Matlab. The STs are treated as sensors fixed to the LRO according to its BCS. Each ST is intended to report access data when both STs become simultaneously occulted by the Lunar surface.

Table 2.1. LRO star tracker rotation sequence. Adapted from [5].

	Rotation Angles (deg)			Boresight Unit Vector (BCS)		
	1st (2)	2nd (1)	3rd (3)	X	Y	Z
ST1	120.0	-30.0	150.0	0.75	0.5	-0.43301270
ST2	180.0	-30.0	-9.0	0.0	0.5	-0.86602540

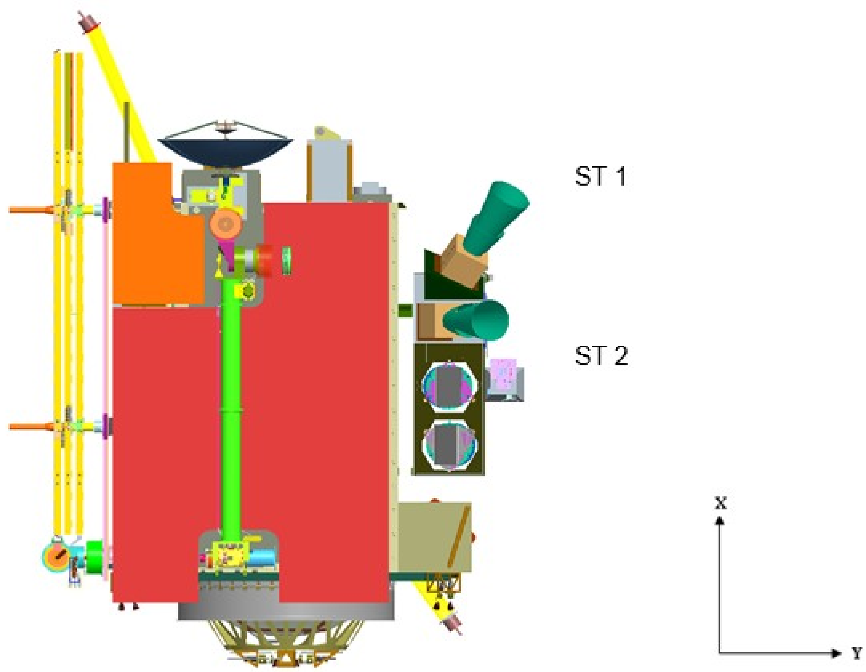


Figure 2.3. LRO ST locations. Source: [5].

2.3 Beta Angle

The beta angle (β) is a term used in orbital mechanics to describe the smallest angle between a satellite's orbital plane and the Sun vector, where the origin of the Sun vector points from the Sun's center of mass [6]. A beta angle can range from -90° to +90° . This angle describes the amount of time a satellite spends in sunlight throughout its orbit, where an increasing beta angle (absolute value) corresponds to an increasing amount of sunlight exposure to the satellite. A positive beta angle occurs when the orbit velocity vector is counterclockwise from the Sun's position, and a negative beta angle occurs when the orbit velocity vector is clockwise from the Sun's position [7]. A visual representation of Solar beta angle (β) can be seen in Figure 2.4, where L represents the angular momentum vector.

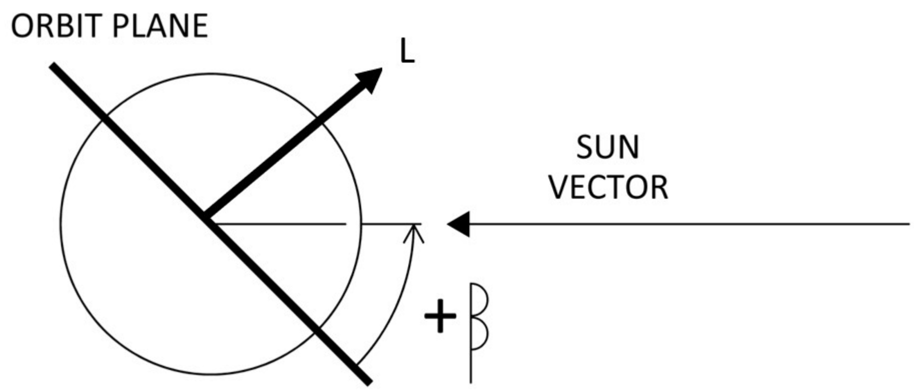


Figure 2.4. Beta angle definition

The LRO's STs are positioned approximately perpendicular to its body $-y$-axis which is the axis that points toward the Sun in Sun-safe mode. This is why Sun occultations need not be considered in this thesis. Also, the LRO's orbit precession is negligible and can be ignored for the purposes of this thesis. With these concepts, it is important to understand how the beta angle of the LRO's orbit affects the amount of total time and total instances that both STs become occulted simultaneously by the Moon. During high beta angle periods throughout the year, the STs will experience occultations every orbit. During low beta angle periods throughout the year, the STs have a larger chance of not being occulted by the Moon. Figure 2.5 demonstrates the relationship between beta angle and ST occultations.

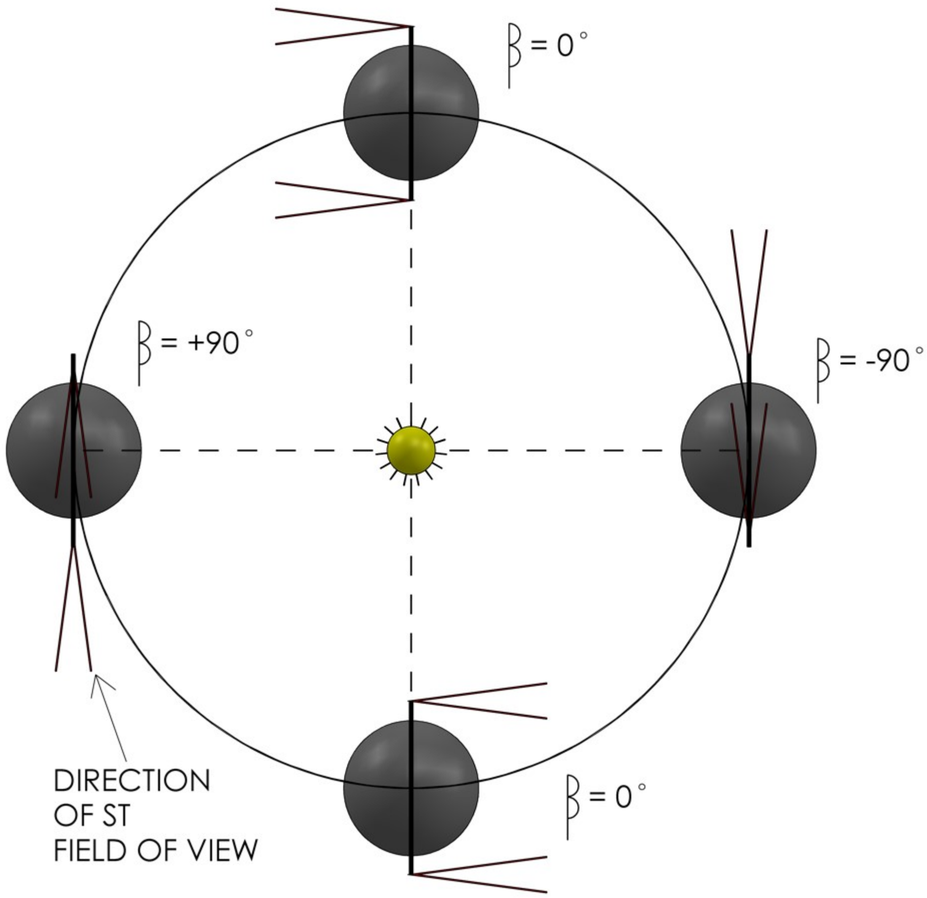


Figure 2.5. Relationship between beta angle (β) and star tracker occultations

2.4 JPL Horizons

Developed and maintained by the Solar System Dynamics Group at California Institute of Technology JPL, the JPL Horizons application is an open source web based platform that provides ephemeris computation services for solar system objects, to include but not limited to planets, asteroids, and satellites [8]. The program is designed to collect "location, motion, and observability of solar system objects as a function of time, as seen from locations within the solar system" [8] to create exportable ephemeris data in multiple formats to be used in other offline programs, such as STK and Matlab. Historic ephemeris data is available as well as future predicted data. The historic and future time ranges available are specific to each solar system object.

In this thesis, JPL Horizons is used to acquire ephemeris data for the position of the LRO's orbit with reference to the ICRF. This data is exported in comma separated value (CSV) format from JPL Horizons and imported into STK, to build a simulation of the LRO's orbit that is used for calculations. Directions for configuring JPL Horizons for the purpose of this thesis are detailed in Appendix A.

The understanding of relevant coordinate systems, the LRO's ST configuration, the concept of Solar beta angles, and the utilization of JPL Horizons is key to resolving the problem at hand. Given the background knowledge required for understanding the underlying aspects of the problem, an approach to the development of a solution can be discussed. The methodology behind is approach is explored and justified in Chapter 3.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3: Methodology

In this chapter, the methodology of determining the LRO's ideal attitude for reducing ST occultation time in Sun-safe mode is developed. The rationale for how a solution is determined in finding an ideal fixed attitude is necessary for understanding how STK can integrate with Matlab scripts to assist in resolving the problem. Once the path to a solution is defined, a summary of the developed Matlab scripts to automate this process is explained. The frequency in which the attitude is updated is then explored, to further reduce occultations.

3.1 Defining the Workspace

To solve the problem, a method of determining when and where the LRO is located in space, as well as when and where the Moon and Sun are located in space at any given time is required. In addition, it is necessary to be able to simultaneously coordinate and manipulate the attitude of the LRO along its orbital path around the Moon, with reference to an inertial reference frame. In this case, the ICRF is the reference frame that is used to correlate the orientation of the SC. This is achieved through the use of STK. The orbital path of local orbiting bodies are accounted for in STK to include the Moon and the Sun at any time past, present, and future. The orbital path of the LRO is acquired through JPL Horizons, introduced in Section 2.4. After a default satellite is created in STK, the ephemeris data obtained from JPL Horizons in CSV format is input into STK, which defines its orbit for the selected time frame.

The LRO's STs are fixed to its body in the BCS frame, utilizing the quaternion values introduced in Section 2.2. Throughout any time frame, it must be determined when, and for how long, both STs are simultaneously occulted by the Moon throughout the LRO's flight path. This is achieved by recording when and for how long both STs simultaneously access the Moon.

STK utilizes an "access" function to calculate time frames in which an object can "see" another object. For this purpose, the Moon is an object to be accessed by the LRO's STs (though the real goal is to avoid such occultations). Constraints to define an access can be considered, where in this case, both STs simultaneously accessing the Moon constitutes an undesirable access period. In Figure 3.1, an example of what constitutes an access of both STs simultaneously accessing the Moon is shown, where ST1's FOR in red is partially accessing the Moon, and ST2's FOR is fully accessing the Moon. Any partial coverage of the ST's FOR constitutes an occultation, therefore is considered as an access.

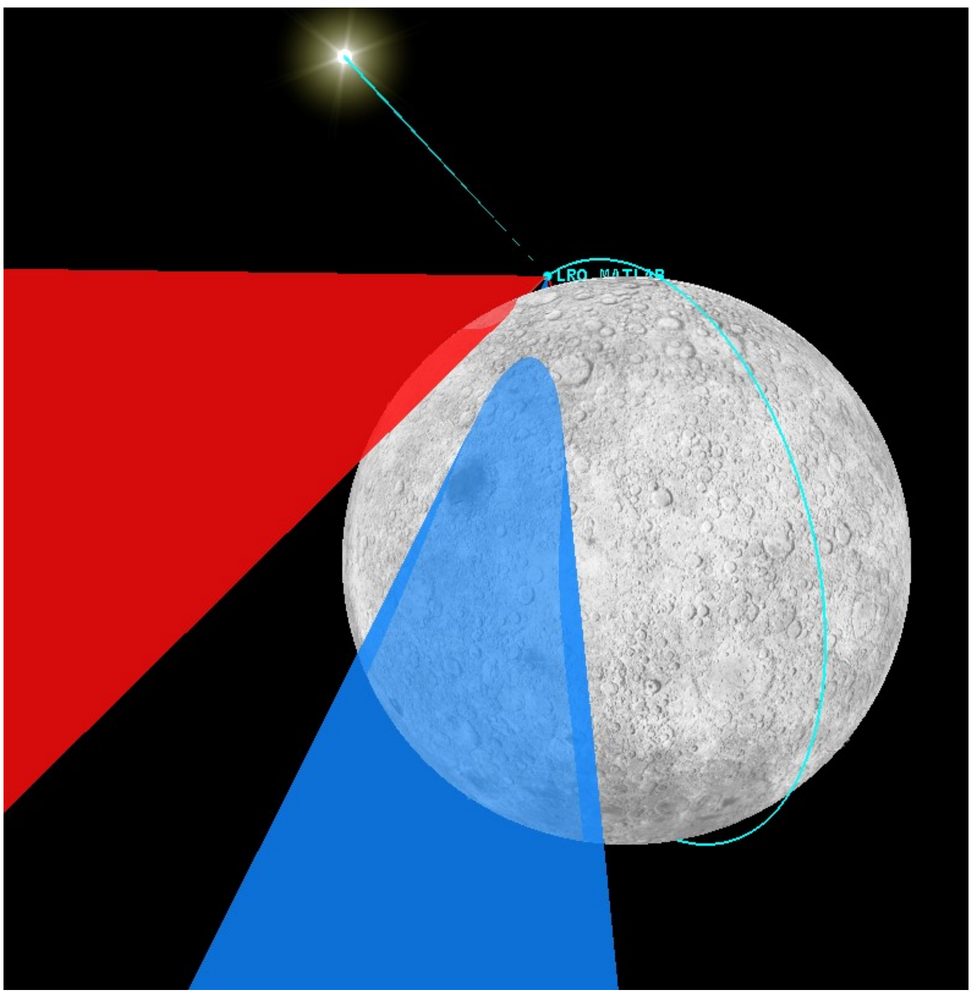


Figure 3.1. Simultaneous ST occultations represented as an access in STK (red FOR = ST1; blue FOR = ST2)

In order for ST occultations to be accounted for in STK, the STs must be treated as sensors, and the Moon must be treated as an area target to be accessed by the sensors. Due to a limitation of STK, where only one hemisphere of the Moon can be created as an area target at a time, two area targets must be created to encompass the Moon. While this introduces complexity to the solution, it is resolvable through the use of "constellations" in STK. Both area targets representing each hemisphere of the Moon are combined into one constellation, which causes them to be treated as one. Considering that both STs need to be occulted simultaneously to trigger a concern for the purposes of this thesis, both STs are combined into one constellation as well.

To set up an environment in STK, where the accesses of both STs to both hemispheres of the Moon are accounted for, a chain is created to link the access path of the STs and the Moon together. With this, an access report could be completed for any possible selected attitude for the LRO. The amount of accesses, when, and for how long they were, could be determined without the use of a Matlab program. However, the objective is to determine what attitude allows the minimum total occultation time for a given time frame. If this was done in STK alone, the time it would take and the amount of user intervention would render this effort unobtainable.

3.2 Concept of Automation

To run access simulations on the entire range of allowable LRO attitudes for Sun-safe mode, iteration automation is required. To iterate these simulations, Matlab is interfaced with STK to run a Monte-Carlo nested *for* loop on all scenarios, and to record data. The LRO's BCS $-y$-axis must be fixed to the Sun vector at all times, but may be offset with an alternative fixed attitude within 15° in all directions from the Sun vector. Therefore, a 30° cone of freedom is considered for allowable attitude options for the BCS $-y$-axis to point, offset from the Sun vector. Iterations are constructed through Matlab using the polar equation, which allows the Monte-Carlo nested *for* loop to iterate throughout the entire 30° cone of freedom systematically. The BCS $-y$-axis radius (r) from the Sun vector, the polar angle (θ) from origin with respect to the ICRF, and the angle of rotation about the BCS $-y$-axis (γ) from origin with respect to the ICRF, are the three iterated variables in the Monte-Carlo nested *for* loop. The $-y$-axis radius (r) from the Sun vector is to be considered in the units of degrees offset in any direction from the Sun vector. For context, an example of the relative axes and their rotations as related to the radius (r), the polar angle (θ), and angle of rotation about the BCS $-y$-axis (γ) is shown in Figure 3.2. In this Figure, the LRO's BCS $-y$-axis is oriented at 15° of radius from the Sun vector, 30° of polar angle with respect to the ICRF, and 0° of rotation about the BCS $-y$-axis with respect to the ICRF. The ICRF is inertial, and the Sun vector in the BCS is non-inertial. Therefore, the polar angle (θ), and the angle of rotation about the BCS $-y$-axis (γ), is referenced at 0° from the x and y plane of the ICRF, and increases from 0° in the counter-clockwise direction facing the BCS $-y$-axis.

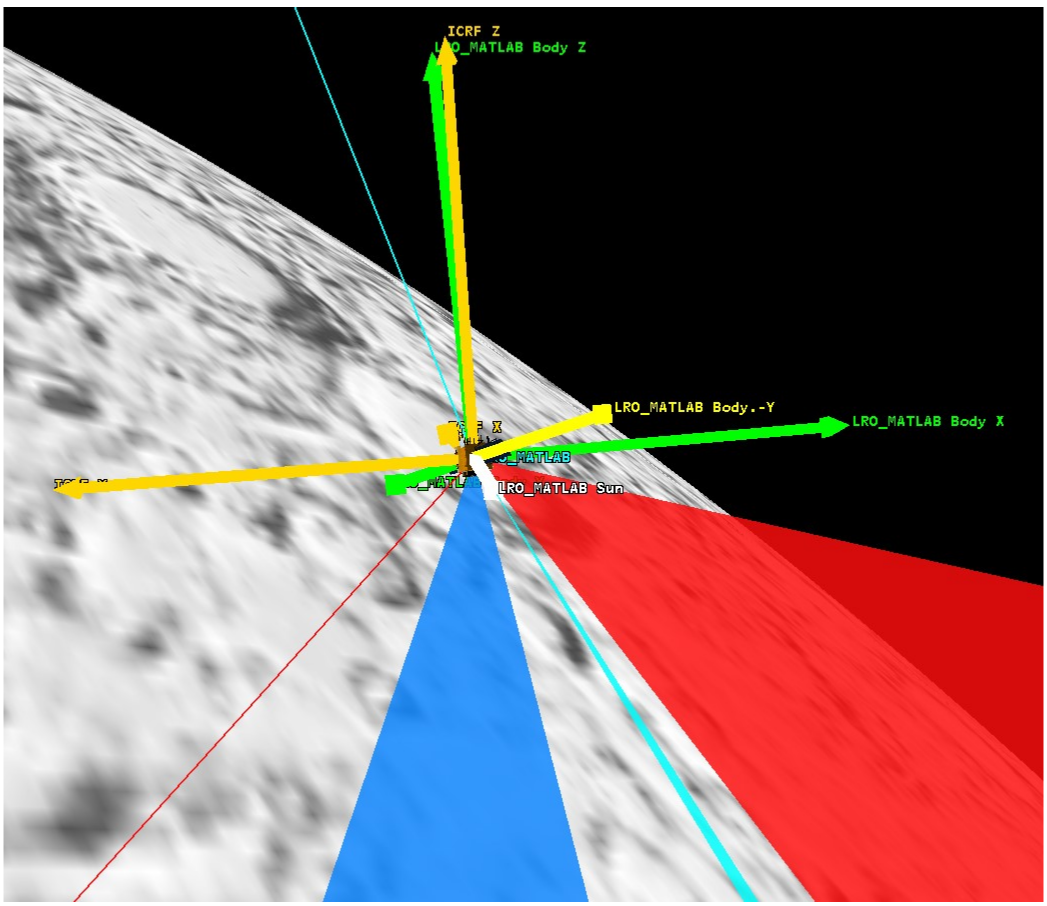


Figure 3.2. Relationship between BCS and ICRF in STK

The polar equation is used to assign offset angles to the BCS $-y$-axis, relative to the Sun vector in reference to the ICRF, for the Monte-Carlo nested *for* loop, which allows the simulation to iterate through 360° of θ and also iterate from 0° to 15° of radius from the Sun vector. The rotation angle about the BCS $-y$-axis can be iterated through 360° in order to rotate the SC about the BCS $-y$-axis, which also changes the orientation of the SC with respect to the ICRF. A graphical depiction of the relationship between polar coordinates, the BCS, and the Sun vector is shown in Figure 3.3 below.

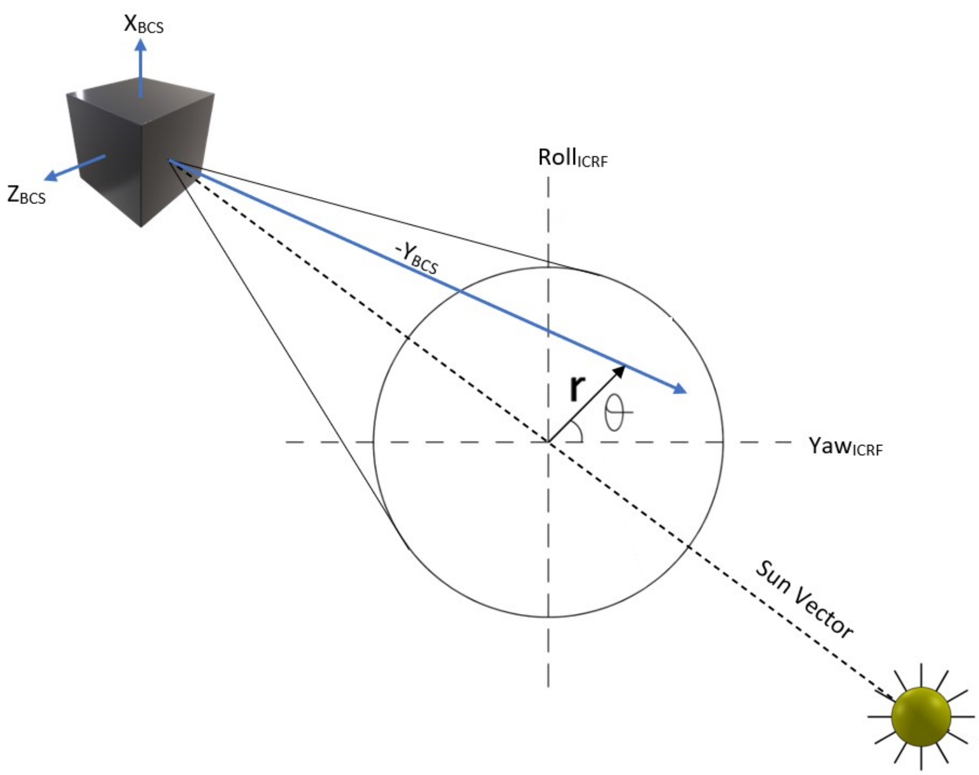


Figure 3.3. Relationship between polar coordinates, the BCS, and the Sun vector

From Figure 3.3, the BCS can be rotated to align the BCS $-y$-vector as shown by performing a 321 Euler rotation where the yaw rotation is approximated as $\theta_3 = r * cos(\theta)$ and the roll rotation is approximated as $\theta_1 = r * sin(\theta)$. Since the transformation of the direction cosine matrix is nonlinear, this leads to a small error in the pointing direction of the BCS $-y$-vector. The error is negligible from the point of view of searching for occultation reducing attitudes.

As the LRO's BCS $-y$-axis is systematically iterated through the 30° cone of freedom, from 0° to 360° for θ, and 0° to 15° for radius, the LRO must be pitched about the BCS $-y$-axis from 0° to 360° at each polar coordinate, represented as γ. The yaw/roll/pitch iteration scheme for the Monte-Carlo nested *for* loop, including θ, r, and γ is depicted in Figure 3.4 below.

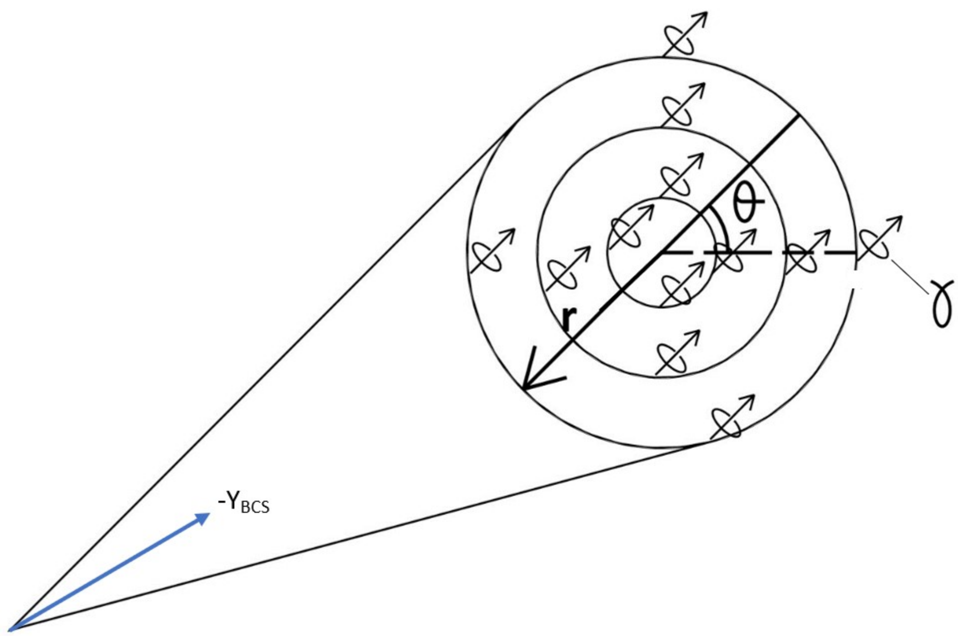


Figure 3.4. Cone of freedom for $-y$-axis pointing to Sun vector

3.3 Automating the Solution using Scripts

Utilizing scripts with Matlab and integrating with STK is the key to running large numbers of iterations to obtain a solution in a reasonable amount of time. The main features of the developed code to perform iterations entail the LRO's align and constrain parameters to the Sun, the Monte-Carlo nested *for* loop, and access data collection.

The LRO's fixed attitude relative to the Sun must be set in STK so that the $-y$-axis points toward the Sun without the ability to freely rotate about the y-axis at all times during its orbit. However, these attitude constraints must allow the orientation of the $-y$-axis to be altered in subsequent iterations to be pointed offset from the Sun, and to allow the SC to be rotated about the y-axis in subsequent iterations such that the LRO is pitched about the y-axis.

The initial attitude parameters are set by creating a custom "aligned and constrained" axis in Analysis Workbench, where the LRO is aligned to the Sun vector and constrained to the ICRF z-axis. The body reference alignment to the Sun vector is set to (0,1,0) in Cartesian coordinates, and the body reference constraint to ICRF is set to (0,0,1) in Cartesian coordinates. In addition, a custom "fixed in axes" axis must be defined in Analysis Workbench, where the allowable $-y$-axis offset from the Sun vector is defined. For this, the reference axis is defined as the custom created "aligned and constrained" axis, while the fixed orientation with respect to ICRF is set to (0,0,0) in 321 Euler Angles. The described Analysis Workbench fixed attitude parameters are coded in Matlab, shown in Figure 3.5.

```
%Create Aligned and Constrained axes in Analysis Workbench
»AxesFactory = LRO_MATLAB.Vgt.Axes.Factory;

»LRO_alignConstrain = AxesFactory.Create('LRO_alignConstrain',...
'Aligned to Sun vector and constrained to ICRF',...
'eCrdnAxesTypeAlignedAndConstrained'); %('axes type, description, syntax')

»LRO_alignConstrain.AlignmentReferenceVector.SetPath...
('Satellite/LRO_MATLAB Sun');
»LRO_alignConstrain.AlignmentDirection.AssignXYZ(0, 1, 0);
»LRO_alignConstrain.ConstraintReferenceVector.SetPath...
('Satellite/LRO_MATLAB ICRF.X');
»LRO_alignConstrain.ConstraintDirection.AssignXYZ(0, 0, 1);

%Create Fixed in Axes axis in Analysis Workbench
»LRO_fixedInAxes = AxesFactory.Create('LRO_fixedInAxes',...
'Offset -Y axis from Sun', 'eCrdnAxesTypeFixed'); %('axes type, description, syntax')
»LRO_fixedInAxes.ReferenceAxes.SetAxes(LRO_alignConstrain);
»LRO_fixedInAxes.FixedOrientation.AssignEulerAngles('e321', 0, 0, 0);

% (Angle A, Angle B, Angle C)
% Angle A - rotate about X axis from Sun vector
% Angle C - rotate about Z axis from Sun vector
```

Figure 3.5. Using Analysis Workbench to create "aligned and constrained" and "fixed in axes" axes

The Monte-Carlo nested *for* loop is initialized by setting the attitude type to aligned and constrained, while referencing the previously mentioned "fixed in axes" to serve as the aligned and constrained vectors. The aligned vector is then initialized to point and fix the $-y$-axis directly toward the Sun vector, which serves as the starting point for the systematic progression through polar coordinates within the cone of freedom. The Monte-Carlo *for* loop is initialized using the code in Figure 3.6.

```
% Define LRO_MATLAB attitude - Satellite properties\basic\attitude
»LRO_MATLAB.SetAttitudeType('eAttitudeStandard')
»LRO_MATLAB.Attitude.Basic.SetProfileType('eProfileAlignedandConstrained')
»LRO_MATLAB.Attitude.Basic.Profile.AlignedVector.set...
('ReferenceVector','Satellite/LRO_MATLAB LRO_fixedInAxes.Y')
»LRO_MATLAB.Attitude.Basic.Profile.ConstrainedVector.set...
('ReferenceVector','Satellite/LRO_MATLAB LRO_fixedInAxes.Z')
»LRO_MATLAB.Attitude.Basic.Profile.AlignedVector.Body.AssignXYZ(0,-1,0)
```

Figure 3.6. Initializing the Monte-Carlo nested *for* loop

The user may determine the start date and the end date in which to run iterations, as well as the fidelity of the Monte-Carlo iterations. In Figure 3.7, the user inputs the start and end date and time, then chooses the area and number of iterations in which to loop.

```
»dateStart = 'DD MMM YYYY 00:00:00.00';
»dateEnd = 'DD MMM YYYY 00:00:00.00';

»radius = linspace(0,15,iterations);
»theta = linspace(0,360,iterations);
»pitch = linspace(0,360,iterations);
```

Figure 3.7. User-defined parameters

Once the relevant fixed axes are defined, a Monte-Carlo nested *for* loop can systematically iterate through the $-y$-axis cone of freedom. The previously defined "fixed in axes" axis will be iterated with respect to the polar representation defined in section 3.2. Nested within the *for* loop for each polar coordinate, the angle of rotation (γ) about the $-y$-axis is iterated according to pitch angle. At the end of each loop, STK computes the access report for instances of when, and for how long both STs are simultaneously occulted by the Moon. Figure 3.8 displays the Matlab code that drives the Monte-Carlo nested *for* loop.

```
»for r = radius
»for i = theta
»for k = pitch
»LRO_fixedInAxes.FixedOrientation.AssignEulerAngles('e321',r * cosd(i), 0, r * sind(i));
»LRO_MATLAB.Attitude.Basic.Profile.ConstrainedVector.Body.AssignPR(k,0)
»computeAccess
»end
»end
»end
```

Figure 3.8. Monte-Carlo nested *for* loop

3.4 Attitude Update Frequency

The LRO has the capability of storing a pointing vector onboard its erasable programmable read-only memory (EPROM), which is where the ideal attitude for minimal ST occultations in Sun-safe mode is to be called from. This pointing vector, or chosen attitude, can be periodically updated by mission operators. The LRO's Sun-safe attitude is currently updated four times per year, to adjust for shifts between a low beta angle orbit and a high beta angle orbit throughout the year. This method of iterating through a Monte-Carlo nested *for* loop using STK and Matlab, allows this to be recreated while searching for the least total time of simultaneous ST occultations possible. Updates to the attitude using this method are broken up into four periods by date throughout the year, from 03JUN21 to 03JUN22, but can be used for any future year or general time frame. The four periods of time frames are represented in Table 3.1 and depicted in Figure 3.9 below.

Table 3.1. Quarterly attitude update scheme based on Moon position

Sequence	Start Date/Time	End Date/Time	Beta angle (β) region
1	03JUN21 00:00:00	05SEP21 00:00:00	High β
2	05SEP21 00:00:00	02DEC21 00:00:00	Low β
3	02DEC21 00:00:00	01MAR22 00:00:00	High β
4	01MAR22 00:00:00	03JUN22 00:00:00	Low β

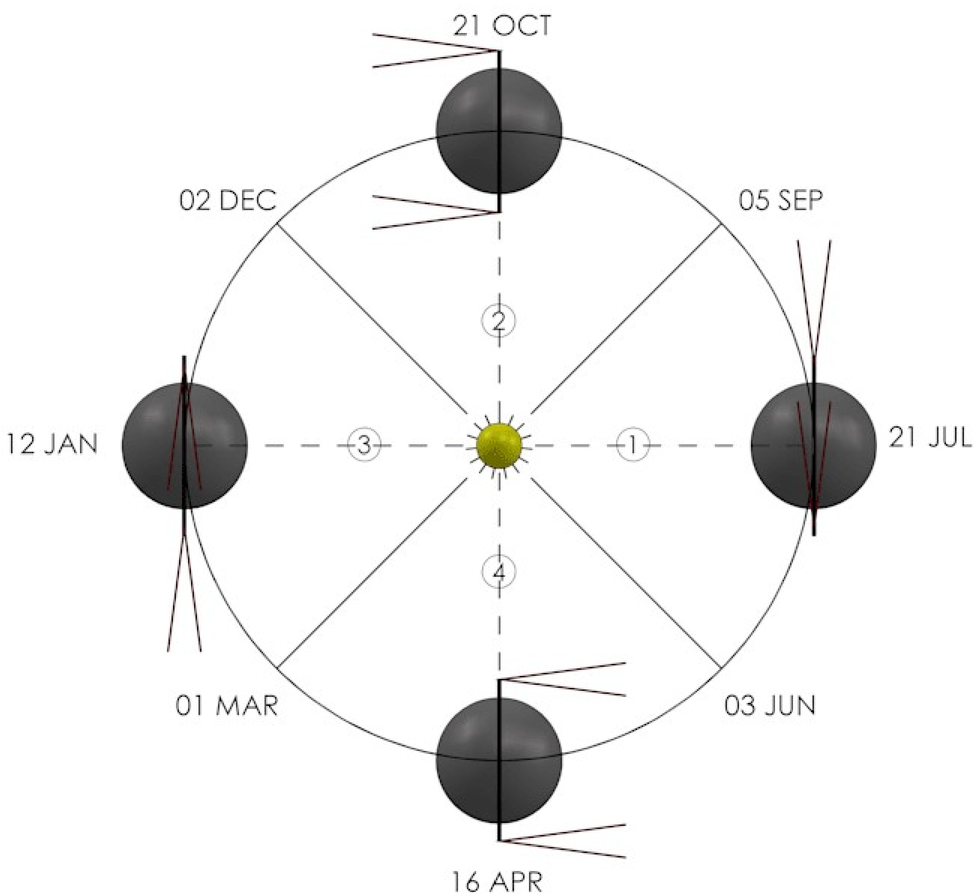


Figure 3.9. Quarterly attitude update scheme based on Moon position

Considering that the LRO's beta angle is constantly changing as the Moon orbits the Sun throughout the year, the best course of action would be to perform frequent or constant optimal fixed attitude updates using the method discussed in this thesis. This way, the LRO's fixed attitude in Sun-safe mode would always be optimized no matter the position of the Moon. However, this would be unreasonable. To further reduce ST occultations without constant intervention, optimal fixed attitude updates can be conducted bi-quarterly, rather than quarterly. Bi-quarterly updates are chosen to be performed at each Moon orbit position in time, represented in Table 3.2 and depicted in Figure 3.10 below.

Table 3.2. Quarterly attitude update scheme based on Moon position

Sequence	Start Date/Time	End Date/Time	Beta angle (β) region
1	03JUN21 00:00:00.00	21JUL21 00:00:00.00	High
2	21JUL21 00:00:00.00	05SEP21 00:00:00.00	High
3	05SEP21 00:00:00.00	21OCT21 00:00:00.00	Low
4	21OCT21 00:00:00.00	02DEC21 00:00:00.00	Low
5	02DEC21 00:00:00.00	12JAN22 00:00:00.00	High
6	12JAN22 00:00:00.00	01MAR22 00:00:00.00	High
7	01MAR22 00:00:00.00	16APR22 00:00:00.00	Low
8	16APR22 00:00:00.00	03JUN22 00:00:00.00	Low

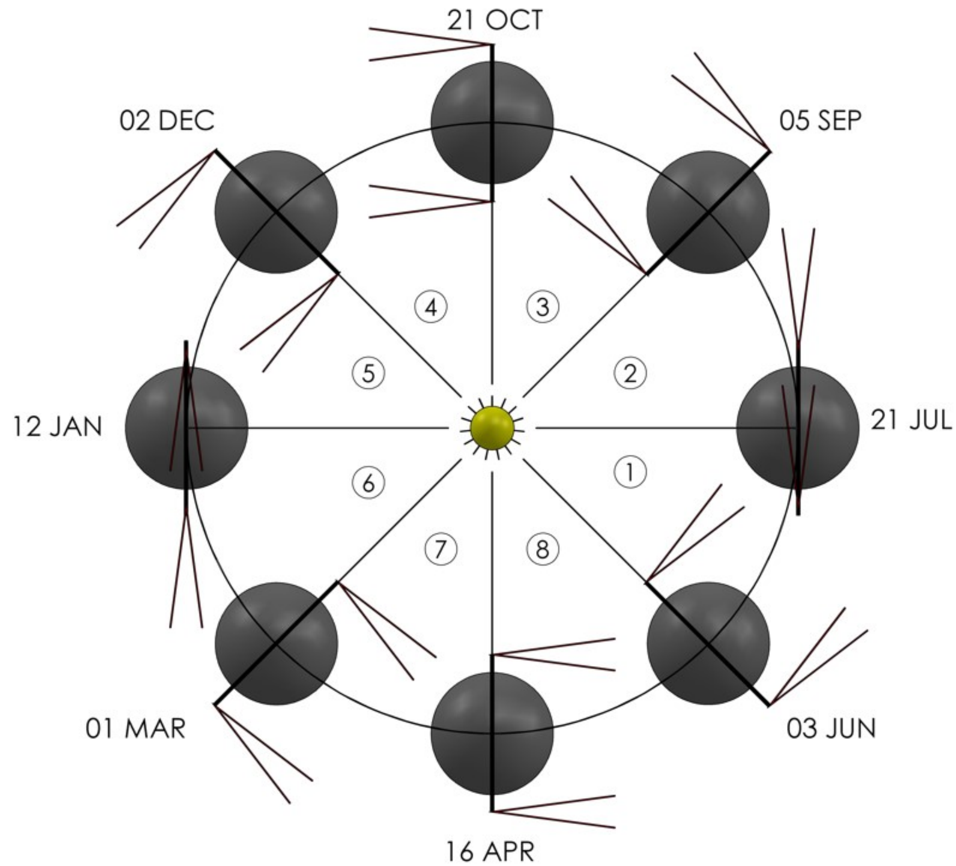


Figure 3.10. Bi-quarterly attitude update scheme based on Moon position

An automated program to determine the ideal Sun-safe attitude for the LRO for any period of time is developed using the LRO's situational parameters, and conceptualization of a systematic Monte-Carlo iteration scheme. Automation of this process uses programmatic scripts and considerations for attitude update frequency to account for beta angle changes throughout the year. This program is run and analyzed in Chapter 4 to determine how changes in attitude update frequency, maximum size of the cone of freedom, and iteration fidelity affect the results. Directions on how to set up and operate the software tool developed in this thesis for determining an ideal attitude for the LRO in Sun-safe mode for minimal ST occultations are detailed in Appendices A through D.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

Using the methodology explained in Chapter 3, various sets of simulations are run in order to analyze the results depending on attitude update frequency, maximum size of the cone of freedom, and level of iteration fidelity. In Chapter 4, a simulation is first run to analyze the results of setting parameters to a 30° cone of freedom, and updating the attitude quarterly throughout the year, using low iteration fidelity. These results are then compared with another simulation where the cone of freedom and iteration fidelity remain the same, but the update frequency is doubled to bi-quarterly. Analysis of results for bi-quarterly attitude updates with a 60° cone of freedom is compared as well. Using the resultant ideal attitude output from the bi-quarterly updated 60°cone of freedom results, a high fidelity simulation is run to determine if lower occultation times can result from finding an ideal Sun-safe attitude with higher accuracy using the iteration scheme defined in Section 3.2.

4.1 Quarterly Updates with a 30° Cone of Freedom

Access data from STK is exported and post-processed in Matlab to determine the LRO's optimal fixed attitude in Sun-safe mode from 03JUN21 to 03JUN22. Simulations are first run quarterly, with a 30° cone of freedom. Given a quarterly attitude update frequency, the optimal attitude is determined, as seen in Table 4.1. The sum of occultation times of all quarters totals 6,044,615 seconds, or 69.95 days throughout the year. It is clear that the beta angle (β) affects the total occultation time in each quarter, where high β quarters yield more occultation time than low β quarters. This result assists in validating the methodology and approach to the Matlab program. An important clue to reducing occultations further is the resultant radius (r) of each reported optimal attitude. Each reported r is 15°, which indicates that the LRO's attitude is best suited somewhere along the edge of its 15° cone of freedom, and that expanding the cone of freedom may further reduce the total occultation time. Note that all reported start and stop dates are at time 00:00:00.00.

Table 4.1. Quarterly results (30° cone of freedom)

Start Date	End Date	Optimal Attitude	Occultation Time (s)	Occultation Time (days)	Beta Angle (β) Region
03JUN21	05SEP21	r-15-θ-188-γ-157	1793199.850180	20.75	High
05SEP21	02SEC21	r-15-θ-16-γ-141	1255463.732836	14.53	Low
02DEC21	01MAR22	r-15-θ-16-γ-344	1638569.295297	18.96	High
01MAR22	03JUN22	r-15-θ-0-γ-157	1357382.769269	15.71	Low
			Total Time (days)	69.95	

4.2 Bi-Quarterly Updates with a 30° Cone of Freedom

To reduce total occultation time, simulations are run bi-quarterly with a 30° cone of freedom, which adjusts the LRO's fixed attitude more frequently to account for changes in β as the Moon orbits around the Sun throughout the year. The sum of occultation times of each bi-quarter totals 3,343,408 seconds, or 38.70 days throughout the year, as seen in Table 4.2. This result is a 44.67% reduction in total occultation time from the quarterly 30° cone of freedom results. It is still the case that the high β bi-quarters yield more occultation time than low β bi-quarters. Given that the cone of freedom was still restricted to 30°, each reported r is 15°, which still indicates that reductions in occultation time will benefit from expanding the cone of freedom.

Table 4.2. Bi-quarterly results (30° cone of freedom)

Start Date	End Date	Optimal Attitude	Occultation Time (s)	Occultation Time (days)	Beta Angle (β) Region
03JUN21	21JUL22	r-15-θ-188-γ-157	517039.263590	5.98	High
21JUL21	05SEP21	r-15-θ-344-γ-313	515003.599729	5.96	High
05SEP21	21OCT21	r-15-θ-172-γ-329	347580.819788	4.02	Low
21OCT21	02DEC21	r-15-θ-63-γ-141	251166.637066	2.91	Low
02DEC21	12JAN22	r-15-θ-172-γ-141	536345.146632	6.21	High
12JAN22	01MAR22	r-15-θ-16-γ-344	515853.274868	5.97	High
01MAR22	16APR22	r-15-θ-157-γ-344	330797.955151	3.83	Low
16APR22	03JUN22	r-15-θ-297-γ-172	329622.031255	3.82	Low
			Total Time (days)	38.70	

4.3 Bi-Quarterly Updates with a 60° Cone of Freedom

Expansion of the cone of freedom from 30° to 60° is now simulated bi-quarterly to further reduce total occultation time, as can be seen in Table 4.3. The sum of occultation times of each bi-quarter totals 2,343,337 seconds, or 27.13 days throughout the year. This result is a 61.22% reduction in total occultation time from the quarterly 30° cone of freedom results. 7 of 8 reported *r* are 30°, which indicates that reduction in occultation time benefited from expanding the cone of freedom.

Table 4.3. Bi-quarterly results (60° cone of freedom)

Start Date	End Date	Optimal Attitude	Occultation Time (s)	Occultation Time (days)	Beta Angle (β) Region
03JUN21	21JUL22	r-30-θ-235-γ-188	328762.512760	3.81	High
21JUL21	05SEP21	r-30-θ-313-γ-297	324886.580515	3.76	High
05SEP21	21OCT21	r-23-θ-219-γ-313	314573.956472	3.64	Low
21OCT21	02DEC21	r-30-θ-63-γ-125	198190.153195	2.29	Low
02DEC21	12JAN22	r-30-θ-141-γ-125	274510.139662	3.18	High
12JAN22	01MAR22	r-30-θ-47-γ-0	336570.944869	3.90	High
01MAR22	16APR22	r-30-θ-125-γ-344	285383.077465	3.30	Low
16APR22	03JUN22	r-30-θ-297-γ-172	280460.033572	3.25	Low
			Total Time (days)	27.13	

4.4 High Fidelity Iterations using Bi-Quarterly Updates with a 60° Cone of Freedom

Each of the above simulations are run at a standard low fidelity Monte-Carlo nested *for* loop iteration scheme that encompasses a systematic distribution of $-y$-axis pointing vectors throughout the entirety of the cone of freedom. The coded low fidelity iteration scheme for a 30° cone of freedom is detailed in Table 4.4. In a 60° cone of freedom, the maximum r value is changed to 30°.

Table 4.4. Low fidelity Monte-Carlo nested *for* loop iteration scheme

Iteration count	2,880
Radius (r)	linspace(0,15,5)
Theta (θ)	linspace(0,360,24)
Gamma (γ)	linspace(0,360,24)

After determining the reported optimal attitudes for each selected time frame, a high fidelity iteration scheme is used to account for unexplored polar coordinates within the area of the reported low fidelity optimal $-y$-axis pointing vector, to ensure the most optimal attitude is determined with higher fidelity. The high fidelity iteration scheme runs a Monte-Carlo nested *for* loop on only the area of the cone of freedom that is 7.5° above and below the reported low fidelity r, and 15° above and below the reported fidelity θ. The simulation still pitches through 360° of γ at each $-y$-axis pointing vector. If the reported low fidelity r is less than 7.5° from the maximum radius in the cone of freedom, the value of degrees above the maximum radius in the cone of freedom is added to the value of degrees below the reported low fidelity r to define the segmented area for iterating. A depiction of the high fidelity iteration scheme is shown in Figure 4.1, and the coded high fidelity iteration scheme for a selective segmented cone of freedom is detailed in Table 4.5, where "X" represents the custom minimum and maximum values for each reported low fidelity r and θ.

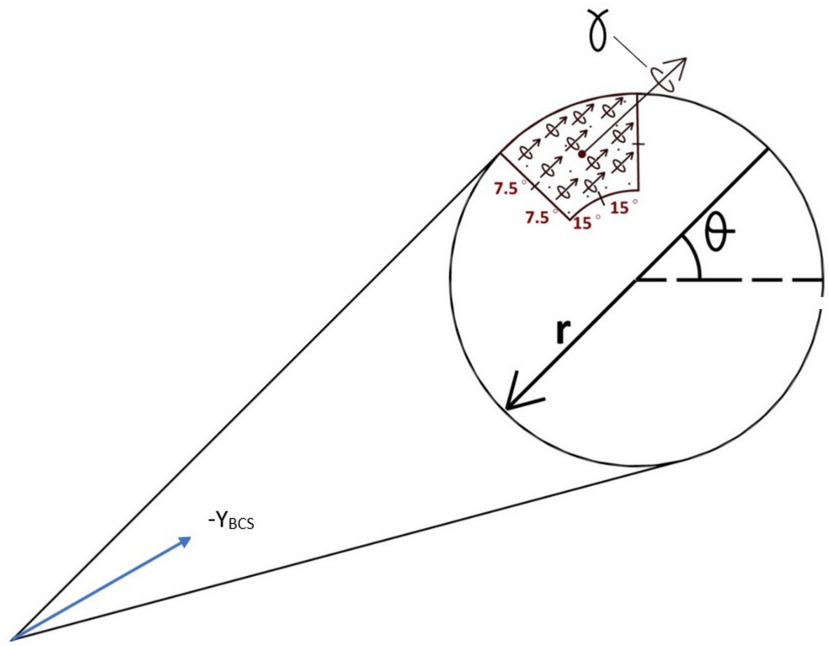


Figure 4.1. High fidelity segmented cone of freedom for $-y$-axis pointing to Sun vector

Table 4.5. High fidelity Monte-Carlo nested *for* loop iteration scheme

Iteration count	5,832
Radius (r)	linspace(X,X,18)
Theta (θ)	linspace(X,X,18)
Gamma (γ)	linspace(0,360,18)

The high fidelity iteration scheme is used to run a bi-quarterly 60° cone of freedom simulation from 03JUN21 to 03JUN22 in order to further reduce occultation time by iterating unexplored $-y$-axis pointing vectors within a segmented region around each reported low fidelity optimal $-y$-axis pointing vector. The results can be seen in Table 4.6. The sum of occultation times of each bi-quarter totals 2,322,822 seconds, or 26.88 days throughout the year. This result is a 61.57% reduction in total occultation time from the quarterly 30°

cone of freedom results, which is only a 0.35% improvement from its low fidelity counterpart. With this disappointing result of improvement from it's low fidelity counterpart, it may be reasonable to deduce that running the high fidelity iteration scheme is futile. It is also interesting to note that the reported occultation times for the 21OCT21-02DEC21, and 16APR22-03JUN22 time frames are higher than that of the low fidelity results. This is because the high fidelity iteration scheme is iterating through different polar coordinates than the low fidelity iteration scheme, and the polar coordinates simulated in the low fidelity iteration scheme happen to be better with the given parameters, even though the high fidelity iteration scheme has more iterations in a smaller area.

Table 4.6. High Fidelity Bi-quarterly results (60° cone of freedom)

Start Date	End Date	Optimal Attitude	Occultation Time (s)	Occultation Time (days)	Beta Angle (β) Region
03JUN21	21JUL22	r-30-θ-243-γ-191	316322.729433	3.66	High
21JUL21	05SEP21	r-30-θ-305-γ-296	314141.178445	3.64	High
05SEP21	21OCT21	r-30-θ-225-γ-296	314451.794687	3.64	Low
21OCT21	02DEC21	r-30-θ-67-γ-127	199661.735406	2.31	Low
02DEC21	12JAN22	r-30-θ-149-γ-127	271519.949353	3.14	High
12JAN22	01MAR22	r-30-θ-51-γ-0	331759.775829	3.84	High
01MAR22	16APR22	r-30-θ-135-γ-0	285355.577661	3.30	Low
16APR22	03JUN22	r-29-θ-291-γ-169	289609.571628	3.35	Low
			Total Time (days)	26.88	

4.5 Analyzing Time per Occultation

The results presented in Chapter 4 portray total simultaneous ST occultation time accrued throughout a time period, but this is a simplified representation. The total occultation time is not contiguous throughout any time period, as the STs pass in and out of occulted instances as the LRO orbits the Moon. Therefore, if the non-contiguous occultation time is short, the attitude control solution may still be viable even though there is an occultation. Using the results from the ideal attitude for the 21OCT21-02DEC21 time frame in Table 4.6 for high fidelity bi-quarterly results (60° cone of freedom), the non-contiguous time periods of each instance of occultation is shown in Table 4.7. This particular result yields 363 instances of occultations, though only the first 30 instances are shown in Table 4.7 for brevity. Each start and stop time stamp is represented in a custom serial time stamp format for ease of manipulation in scripts. The start and stop times of occultation are represented as the day, and fraction throughout the day, between the start and stop date of the simulation. From 21OCT21 00:00:00.00 to 02DEC21 00:00:00.00, there are exactly 42 days, so the whole number counts upwards from 0 to 42 depending on the day of the occultation instance, while the decimal represents the time of day, or the fraction of time throughout the day of the occultation instance. This detailed result is an interesting and possibly useful set of collected data, considering that some occultation instances can be up to 30 minutes long, while others can last only a matter of seconds. Small or large elapsed times of individual ST occultations can affect the ability for ST to maintain an attitude solution, or not, which may allow for refinement of how an occultation is defined. In this case, an alternate approach to the problem can be followed. For example, if the maximum occultation time frame allowed in order to maintain an attitude solution is determined to be 60 seconds, only the occultation instances that are greater than 60 seconds would constitute a valid occultation. All other occultation instances would be ignored. Therefore, the ideal attitude would then be based on the smallest number of occultation instances that are greater than 60 seconds, rather than the total occultation time within a given attitude update period. This method would utilize much of the code created in this thesis with relatively minor alterations.

Table 4.7. Start and Stop Times of Non-contiguous Occultation Instances

Instance	Start Time (Serial)	Stop Time (Serial)
1	0.0223890740890056	0.0357186862966046
2	0.103950791643001	0.117213165503927
3	0.185512803262100	0.198707567178644
4	0.267073798575439	0.280201872694306
5	0.348633225657977	0.361696155043319
6	0.430192936328240	0.443190399324521
7	0.511752879596315	0.524684565956704
8	0.593313115765341	0.606178664369509
9	0.674873647047207	0.687672764994204
10	0.756434449111112	0.769167281221598
11	0.837995480280370	0.850661763921380
12	0.919556846027263	0.932156165479682
13	1.00111841084436	1.01365052547771
14	1.08268027543090	1.09514476615004
15	1.16424244910013	1.17663898842875
16	1.24580482637975	1.25813309953082
17	1.32736752543133	1.33962717244867
18	1.40893045603298	1.42112114583142
19	1.49049358337652	1.50261508452240
20	1.57205582980532	1.58410891436506
21	1.65361623500939	1.66560270485934
22	1.73517691437155	1.74709639581852
23	1.81673772458453	1.82858999771997
24	1.89829878241289	1.91008355212398
25	1.97986008331645	1.99157702550292
26	2.06142161111347	2.07307039236184
27	2.14298334147315	2.15456370019820
28	2.22454529162496	2.23605690745171
29	2.30610747111496	2.31755005905870
30	2.38766988890711	2.39904310181737
.	.	.
.	.	.
.	.	.

After running simulations using various parameters available in the developed tool for determining an ideal attitude for minimizing LRO ST occultations, it is determined that attitude update frequency, maximum size of the cone of freedom, and level of iteration fidelity does have an impact on the results. While this program successfully determines the ideal attitude for the LRO in Sun-safe mode for any given time frame and set of parameters, the chosen parameters input into the simulation can have profound effects on results.

CHAPTER 5: Conclusion

The purpose of the development of methodology in determining an ideal attitude for the LRO in Sun-safe mode for minimal ST occultations in this thesis is to establish a software-based tool. There are numerous variables and preference-based considerations that make it unreasonable to consider the results as a single definitive solution. The results differ, depending on the optimal attitude update frequency chosen, the iteration fidelity chosen, and the maximum radius of $-y$-axis offset from the Sun.

While it is recommended that the attitude update frequency be at least bi-quarterly, the end-user has the freedom to increase or decrease the frequency depending on a subjective accepted risk. The low and high iteration fidelity used in this thesis were chosen to allow for an acceptably large sample size, but was limited based on time constraints in the development of this method for the purposes of this thesis. Low fidelity quarterly updates take 16 hours of run-time per quarter, low fidelity bi-quarterly updates take 8 hours of run-time per bi-quarter, and high fidelity bi-quarterly updates take 17 hours of run-time per bi-quarter as per the chosen iteration parameters. The end-user may choose to increase fidelity further to obtain a more accurate solution, which will increase run-time. The maximum radius of $-y$-axis offset allowed can be narrowed or expanded, depending on accepted risk to the LRO's ability to receive direct sunlight energy to its stowed solar array in Sun-safe mode. The $-y$-axis offset radii that are explored in this thesis are $15°$ and $30°$.

In addition, due to accuracy degradation of predictive ephemeris data in the far future, it is not viable to find all solutions for the remainder of the LRO's service life within this research. This research establishes a method of determining the attitude for minimal ST occultations to be used as needed, prior to an intended update period.

It is clearly determined through the results that a sound method and tool for acquiring an ideal attitude solution is achieved for minimizing ST occultations. Given the use of this tool, it is determined through the results in Chapter 4 that LRO ST occultations can be reduced by 44.67% by increasing the attitude update frequency from quarterly, to bi-quarterly. ST occultations can be reduced by 61.22% by increasing the maximum radius of $-y$-axis offset allowed from 15° to 30°, in addition to updating the Sun-safe attitude bi-quarterly.

While this tool can and should be used with ample time prior to an intended update period, it is undeniable that it does take a significant amount of time to simulate in order to find the ideal Sun-safe attitude for a given period, depending on the chosen iteration fidelity. In future work, it could be speculated that run-time could be significantly reduced through the use of numerical optimization tools as opposed to Monte-Carlo search.

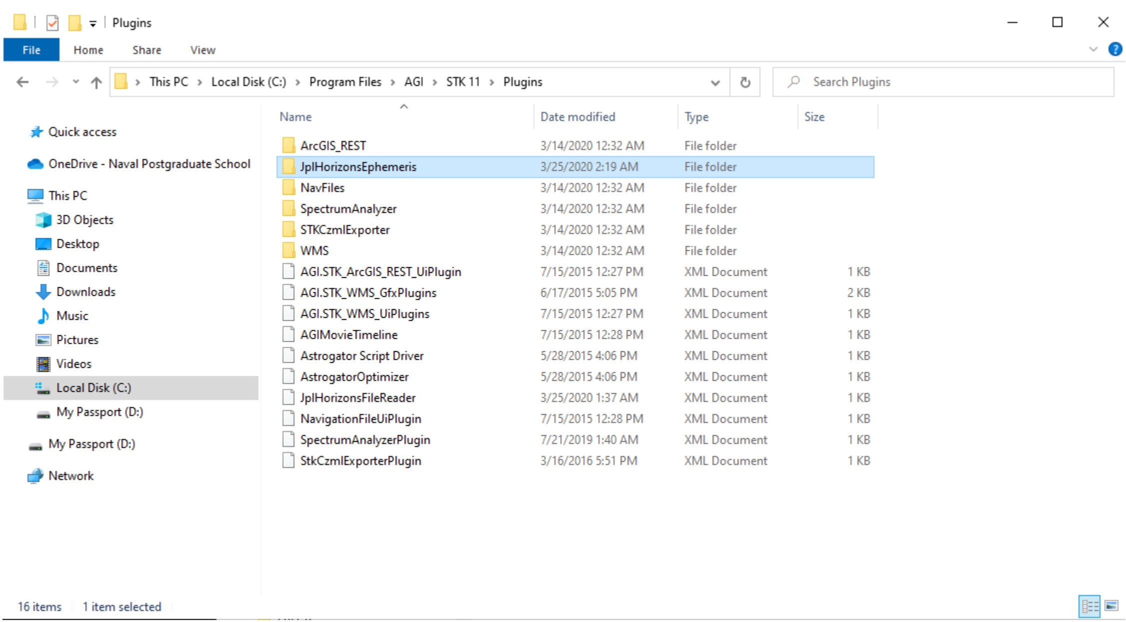
APPENDIX A:
Installing JPL Horizons Ephemeris Reader

The following programs are recommended for use with the instructions in Appendix A. Other software versions may work, but have not been tested here.

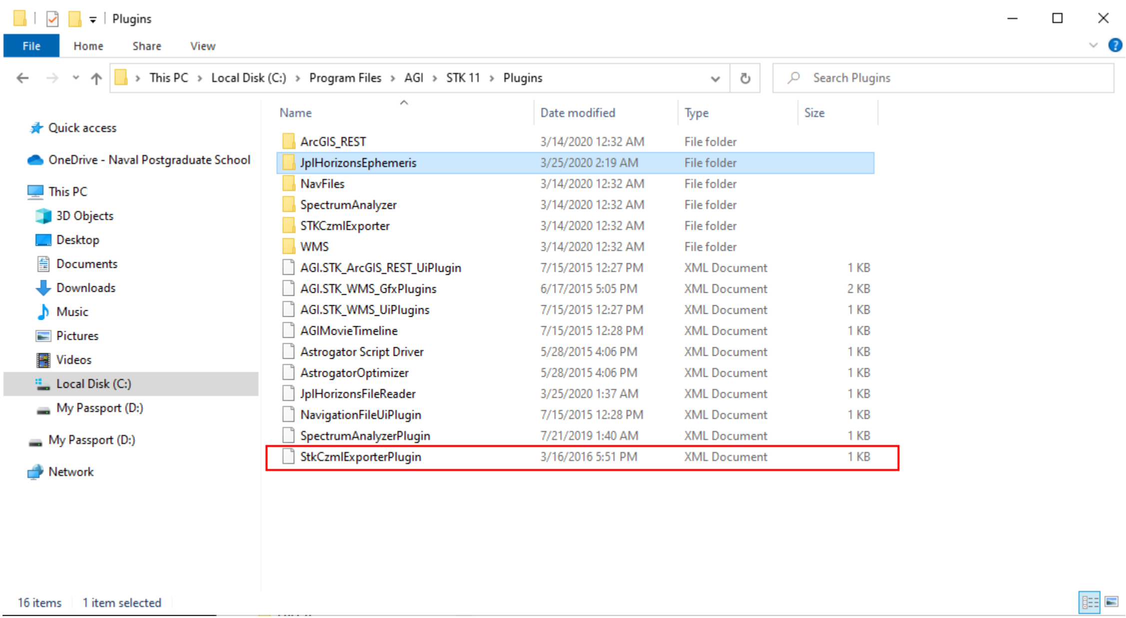
- Windows 10 OS
- STK 12.0.0

- From the AGI website below, download Resource File 1: JplHorizonsEphemeris.zip, at https://agiweb.secure.force.com/faqs/articles/HowTo/interplanetary-ephemeris-summary

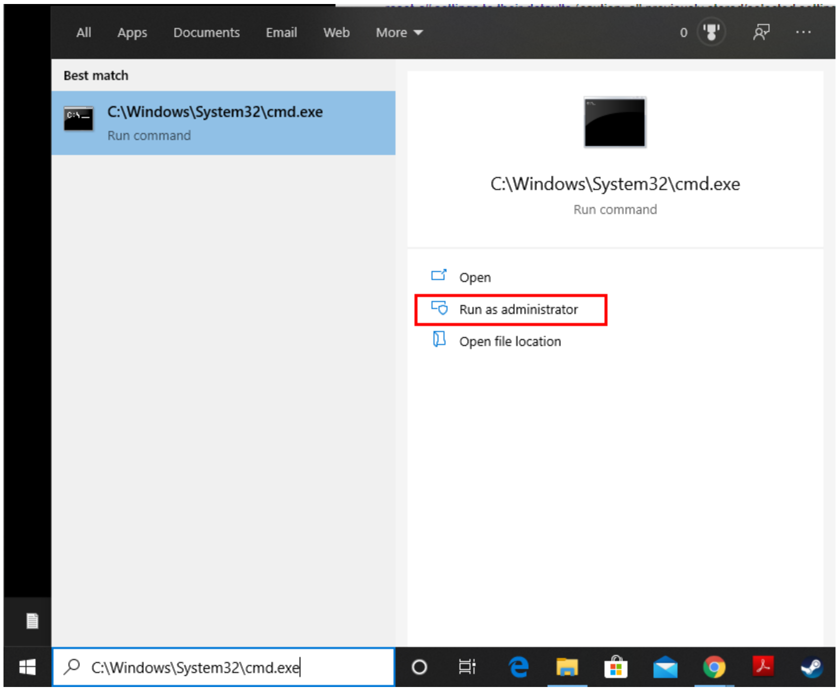
- Extract all files, and place the whole folder into the “Plugins” folder of STK in the following directory. (C:\Program Files\AGI\STK 12\Plugins)



- In the "JplHorizonsEphemeris" folder, move the file "JplHorizonsFileReader.xml" to the "Plugins" folder at the same directory as above.



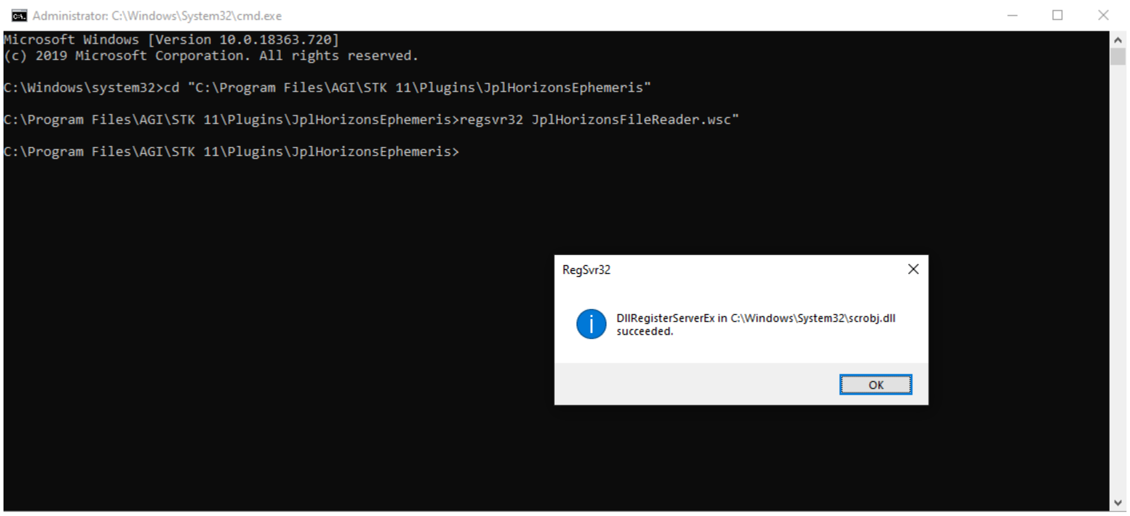
- Open the 64-bit command prompt (C:\Windows\System32\cmd.exe) to register a plugin for the 64-bit version of STK. The command prompt must be run as administrator.



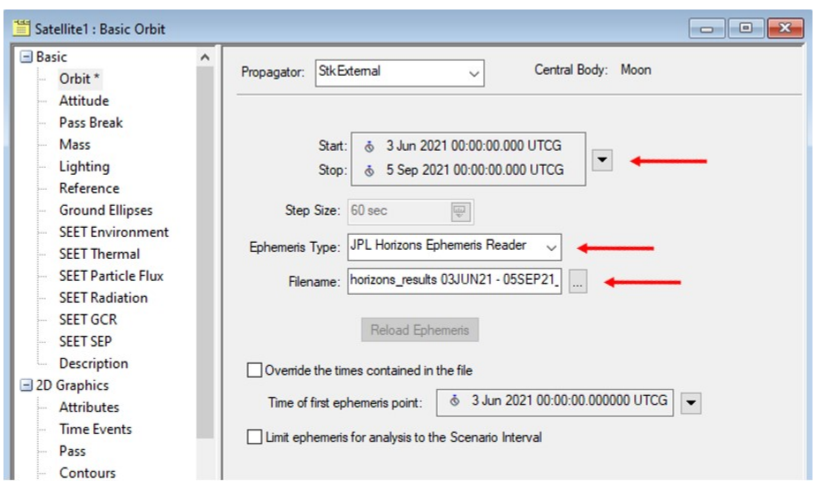
- Once the administrator 64-bit command prompt is open, change the directory to the “JplHorizonsEphemeris” folder inside the aforementioned "Plugins" folder. Then, register the “JplHorizonsFileReader.wsc” file with Windows in the folder “JplHorizonsEphemeris” folder using the >regsvr32 command, as follows.

```
>regsvr32 JplHorizonsFileReader.wsc
```

- Successful registration is noted with a Windows notification pop-up.



- The option to use JPL Horizons Ephemeris Reader should now be available in STK's satellite orbit options under the StkExternal propagator.



APPENDIX B:
JPL Horizons Parameters for LRO Integration with STK

The following programs are recommended for use with the instructions in Appendix B. Other software versions may work, but have not been tested here.

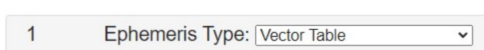
- Windows 10 OS
- STK 12.0.0

- Go to JPL Horizons website, at https://ssd.jpl.nasa.gov/horizons/app.html#/

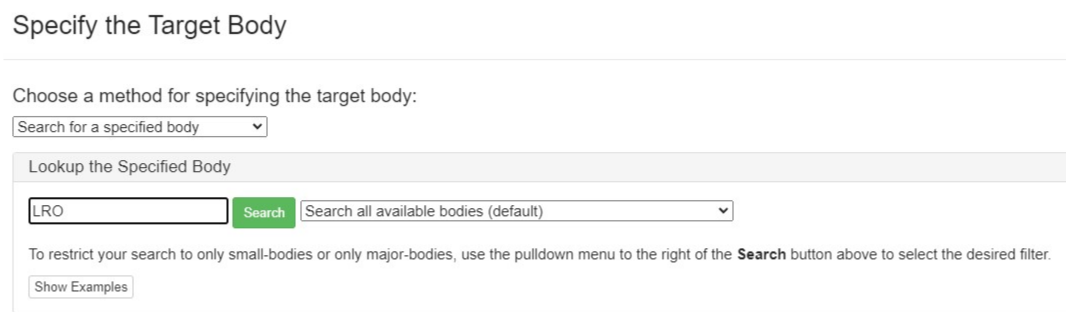
- Note: In order for STK to propagate the orbit correctly for the LRO from Horizons specifically, the settings must be set exactly like this. The only thing that can be altered is the Time Span, but keep in mind that Horizons is limited to outputting 90024 lines generated in the .txt file, so if the step size and time is too large, the .txt file will not be generated. Make sure the Time Span selected is within the STK Scenario parameters as well. To acquire a Time Span of 1 year with 1 minute intervals, create ephemeris files in 2 month spans. Copy and paste each set of intervals into one file, ensuring that the rows of duplicate start times are deleted for continuity.

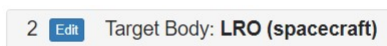
- Input the following parameters

1) Ephemeris Type: Vector Table



2) Target Body: LRO (spacecraft) (-85)



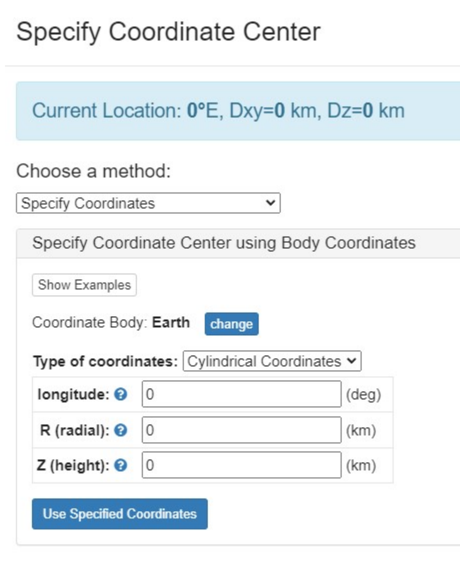


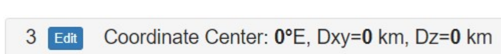
3) Coordinate Center: Cylindrical Coordinates

longitude: 0 (deg)

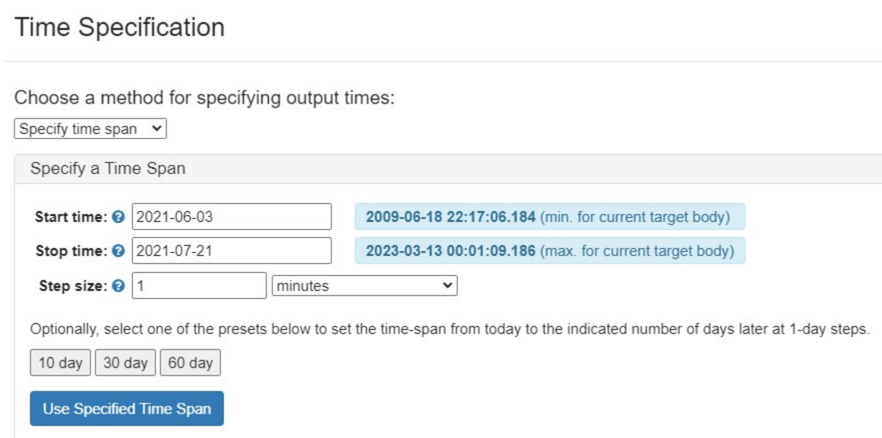
R (radial): 0 (km)

Z (height): 0 (km)





4) Time Specification
Start time: YYYY-MM-DD
Stop time: YYYY-MM-DD
Step size: 1 (minutes)



5) Table Settings

Select Output Quantities: 2 State vector x,y,z,Vx,Vy,Vz

Statistical Uncertainties: Leave all unchecked

Reference frame: ICRF

Reference plane: x-y axes of reference frame (equatorial/equatorial-aligned, inertial)

Vector correction: geometric states

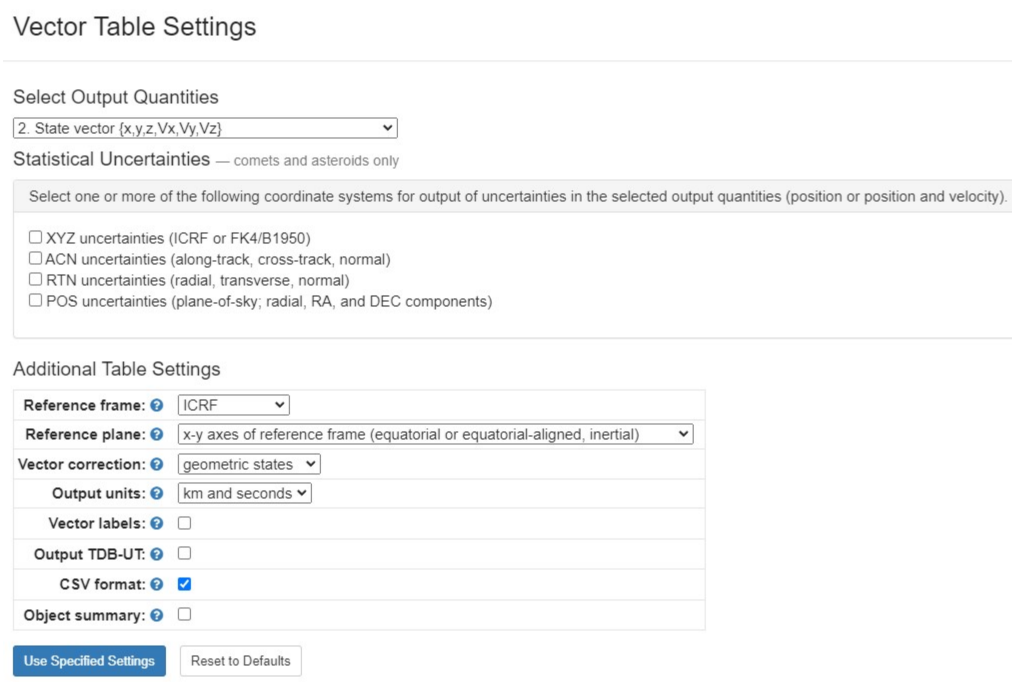
Output units: km and seconds

vector labels: Unchecked

Output TDB-UT: Unchecked

CSV format: Checked

Object page: Unchecked



- Using the specified settings above, select "Generate Ephemeris," then select "download results" in .txt format. It is important to note that the generated output is limited to 90024 lines. For large time spans, generate multiple results and paste them together in a text editor.

```
*******************************************************************************
Ephemeris / WWW_USER Tue Oct 26 17:10:25 2021 Pasadena, USA      / Horizons
*******************************************************************************
Target body name: LRO (spacecraft) (-85)          {source: LRO_merged}
Center body name: Earth (399)                     {source: DE441}
Center-site name: BODY CENTER
*******************************************************************************
Start time      : A.D. 2021-Jun-03 00:00:00.0000 TDB
Stop  time      : A.D. 2021-Jul-21 00:00:00.0000 TDB
Step-size       : 1 minutes
*******************************************************************************
Center geodetic : 0.00000000,0.00000000,0.0000000 {E-lon(deg),Lat(deg),Alt(km)}
Center cylindric: 0.00000000,0.00000000,0.0000000 {E-lon(deg),Dxy(km),Dz(km)}
Center radii    : 6378.1 x 6378.1 x 6356.8 km     {Equator, meridian, pole}
Output units    : KM-S
Output type     : GEOMETRIC cartesian states
Output format   : 2 (position and velocity)
EOP file        : eop.211025.p220118
EOP coverage    : DATA-BASED 1962-JAN-20 TO 2021-OCT-25. PREDICTS-> 2022-JAN-17
Reference frame : ICRF
*******************************************************************************
```

THIS PAGE INTENTIONALLY LEFT BLANK

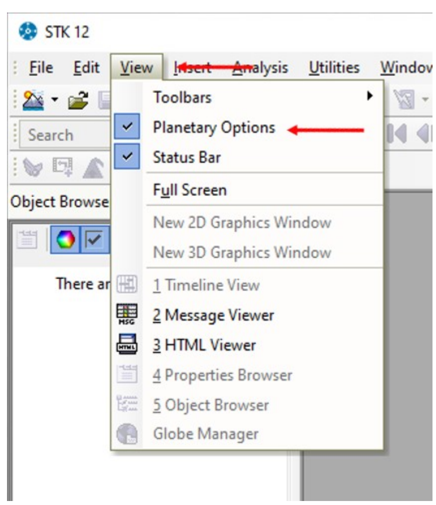
APPENDIX C:
LRO Scenario Setup in STK

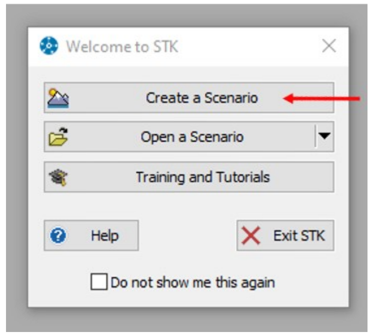
The following programs are recommended for use with the instructions in Appendix C. Other software versions may work, but have not been tested here.

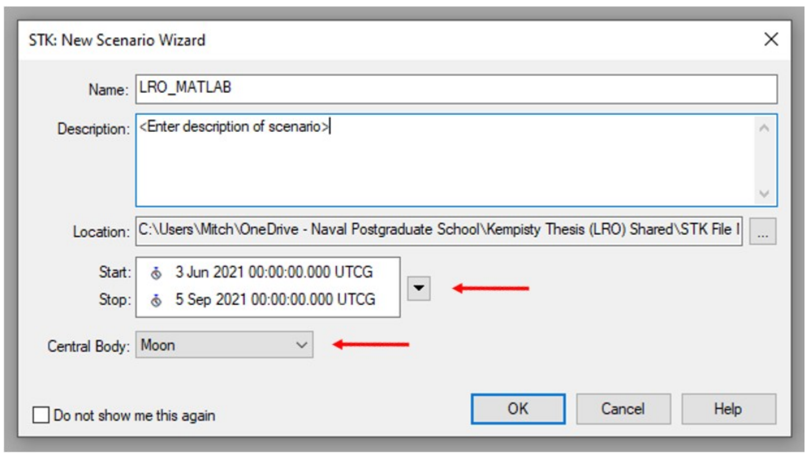
- Windows 10 OS
- STK 12.0.0
- Note: The Matlab program created for the purposes of this thesis automatically establish the STK parameters explained in Appendix C. Appendix C should not be performed. The purpose of Appendix C is to generate an understanding of how the parameters defined in Matlab alter the parameters in the STK user interface.

C.1 Scenario

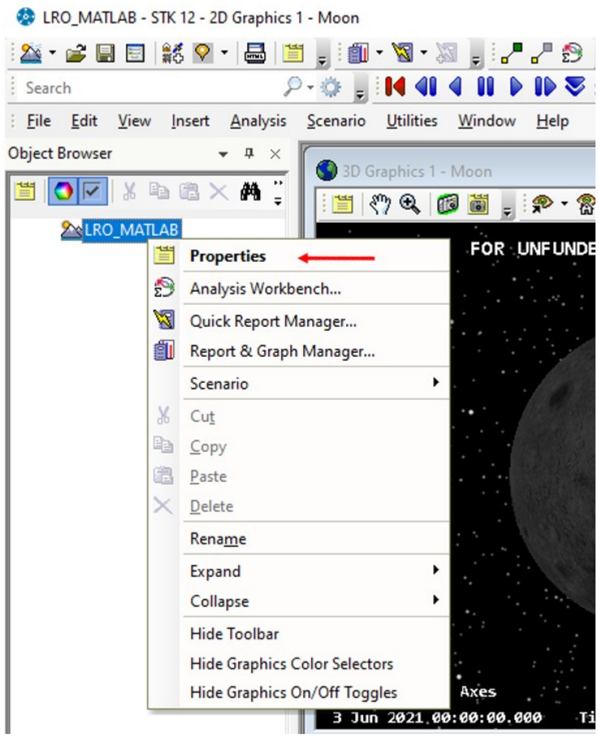
- Open STK. Before creating a new scenario, click "View," and select "Planetary Options." This will enable different options for central body than just Earth. Now click "Create a Scenario," and fill out the information as desired. Ensure the scenario start and stop time meets or exceeds the time span you plan to use from the Horizons ephemeris files. Under the "Central Body" drop down menu, select "Moon." Click "OK."

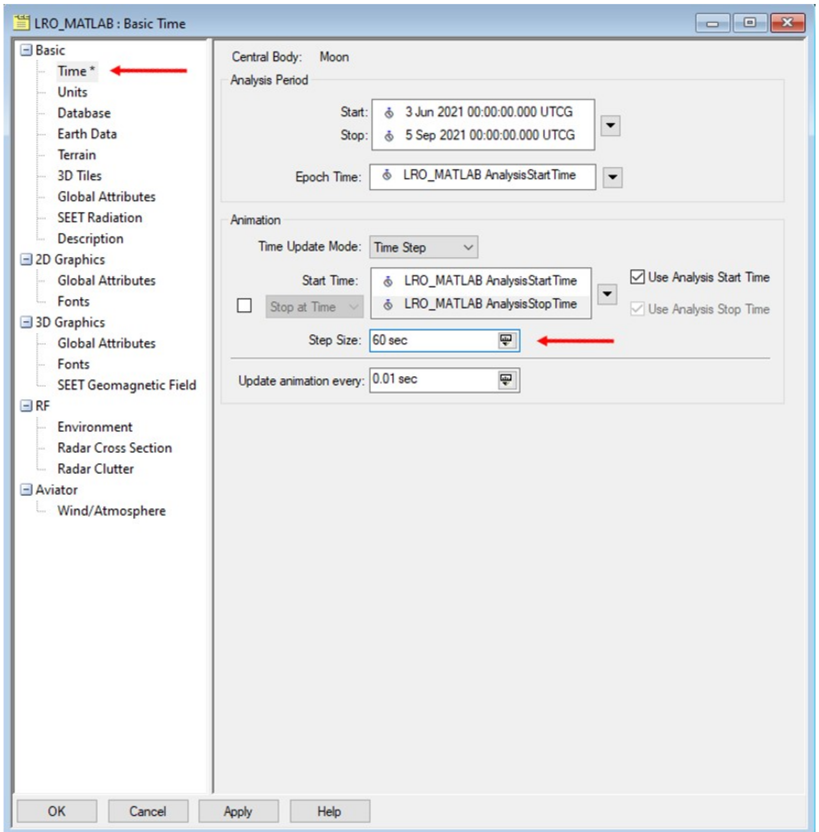


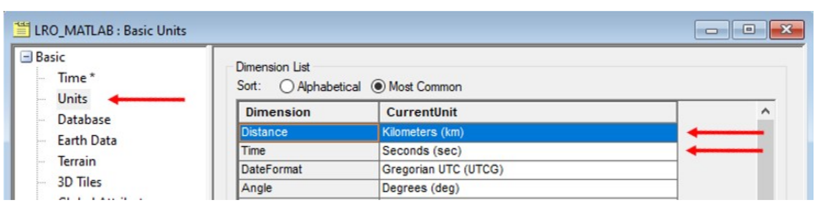




- The 2D and 3D visualization windows should display the Moon. To ensure that the Horizons settings will merge with STK properly, ensure a few settings. Right click on the scenario object and click "Properties." In "Basic"-"Time," set "Step Size" to 60 seconds In "Basic"-"Units," set distance to "km" and Time to "sec" Click "OK" to exit properties window.

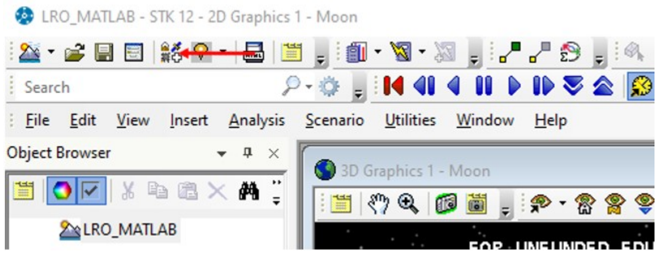

LRO_MATLAB - STK 12 - 2D Graphics 1 - Moon
Search
File Edit View Insert Analysis Scenario Utilities Window Help
Object Browser
3D Graphics 1 - Moon
LRO_MATLAB
Properties
Analysis Workbench...
Quick Report Manager...
Report & Graph Manager...
Scenario
Cut
Copy
Paste
Delete
Rename
Expand
Collapse
Hide Toolbar
Hide Graphics Color Selectors
Hide Graphics On/Off Toggles
Axes
3 Jun 2021 00:00:00.000

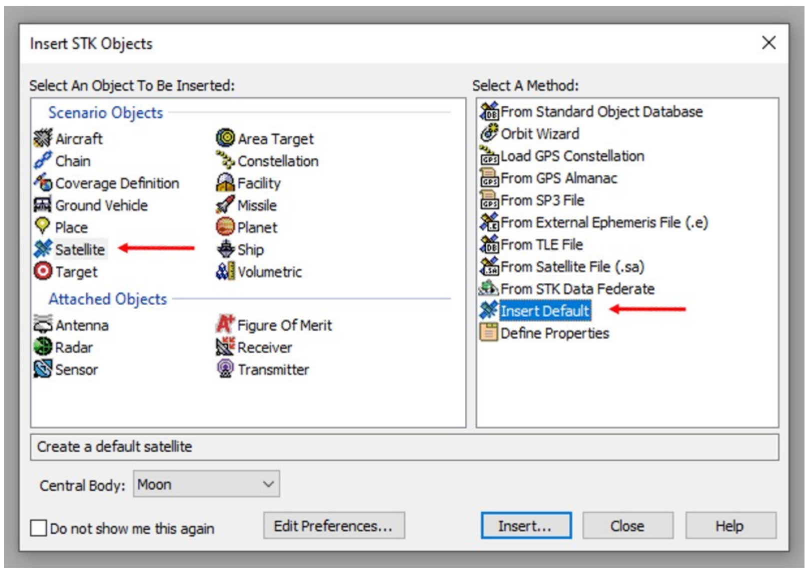




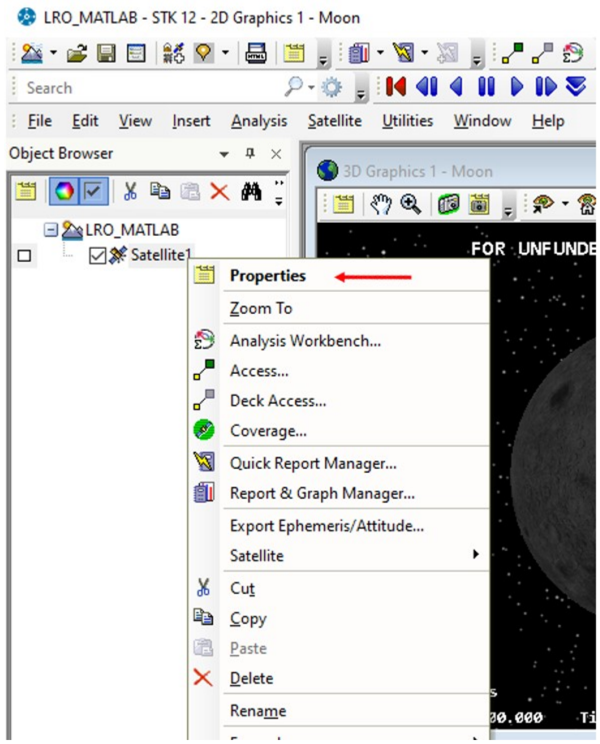
C.2 Satellite

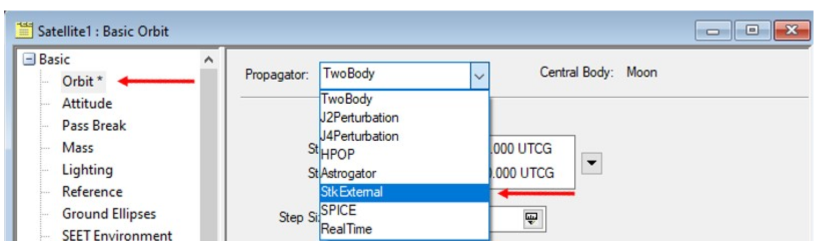
- In the menu bar, click "Insert Object," select "Satellite," then double click "Insert Default."

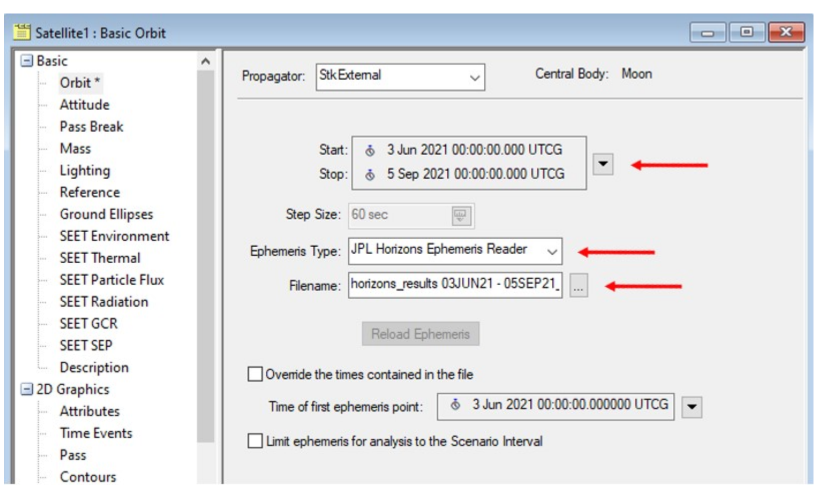




- Once the satellite object is created, right click the satellite in the left hand pane and click “Properties” to establish parameters simulating the LRO. In “Basic”-“Orbit,” select “StkExternal” as the propagator. Ensure the start and stop time is the same as the scenario time frame. Select “JPL Horizons Ephemeris Reader” as the ephemeris type, and upload the JPL Horizons text file that corresponds to the scenario time frame. Click “OK.” Note that the JPL Horizons Ephemeris Reader plugin must to be installed prior to this step. Instructions for installing this plugin are detailed in Appendix A. Instructions for creating ephemeris files are detailed in Appendix B.

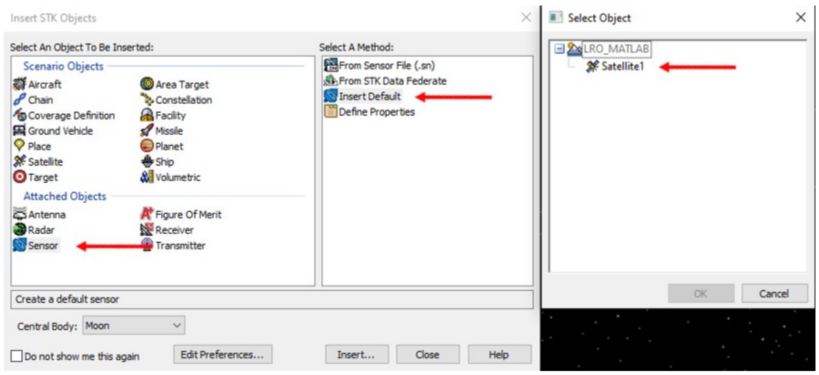

LRO_MATLAB - STK 12 - 2D Graphics 1 - Moon
Search
File Edit View Insert Analysis Satellite Utilities Window Help
Object Browser
3D Graphics 1 - Moon
LRO_MATLAB
Satellite1
Properties
Zoom To
Analysis Workbench...
Access...
Deck Access...
Coverage...
Quick Report Manager...
Report & Graph Manager...
Export Ephemeris/Attitude...
Satellite
Cut
Copy
Paste
Delete
Rename


Satellite1 : Basic Orbit
Basic
Orbit *
Attitude
Pass Break
Mass
Lighting
Reference
Ground Ellipses
SEET Environment
Propagator:
TwoBody
Central Body: Moon
TwoBody
J2Perturbation
J4Perturbation
HPOP
Astrogator
StkExternal
SPICE
RealTime

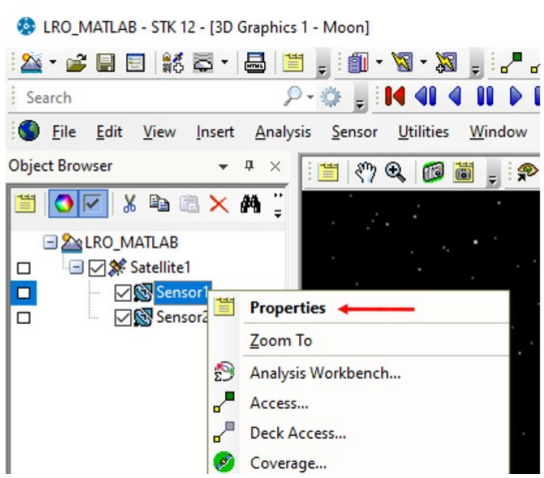

Satellite1 : Basic Orbit
Basic
Orbit *
Attitude
Pass Break
Mass
Lighting
Reference
Ground Ellipses
SEET Environment
SEET Thermal
SEET Particle Flux
SEET Radiation
SEET GCR
SEET SEP
Description
2D Graphics
Attributes
Time Events
Pass
Contours
Propagator: StkExternal
Central Body: Moon
Start: 3 Jun 2021 00:00:00.000 UTCG
Stop: 5 Sep 2021 00:00:00.000 UTCG
Step Size: 60 sec
Ephemeris Type: JPL Horizons Ephemeris Reader
Filename: horizons_results 03JUN21 - 05SEP21_
Reload Ephemeris
Override the times contained in the file
Time of first ephemeris point: 3 Jun 2021 00:00:00.000000 UTCG
Limit ephemeris for analysis to the Scenario Interval

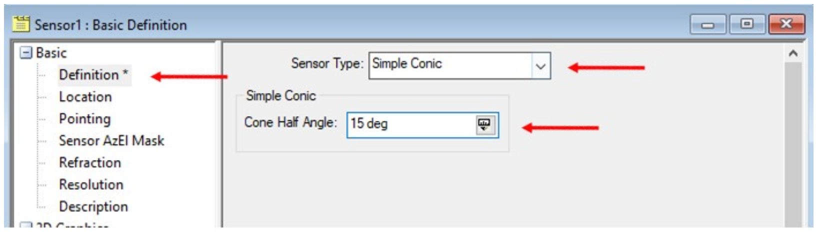
C.3 Star Trackers

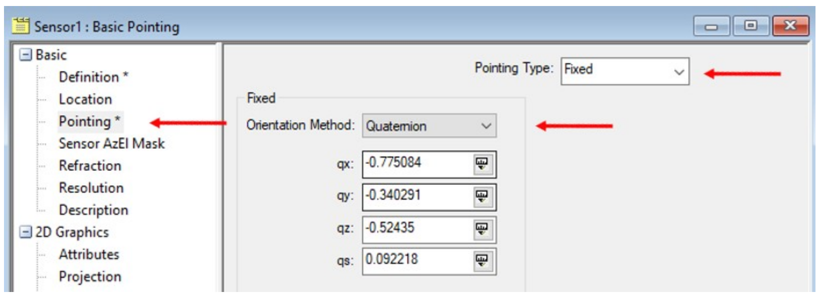
- In the menu bar, click “Insert Object,” select “Sensor,” then double click “Insert Default.” Associate the sensor with the satellite created previously and click “OK.” Repeat this step in order to create two sensors associated with the satellite.

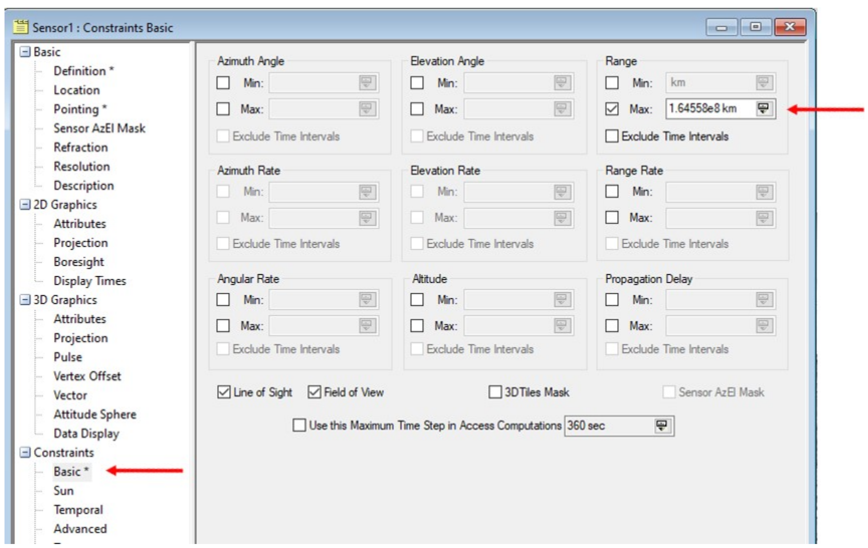


- Once the sensors are created, right click the first sensor in the left hand pane and click “Properties” to establish parameters simulating the LRO’s ST1. In “Basic”-“Definition,” select “Simple Conic” as the sensor type, and assign the cone half angle to 15°İn “Basic”-“Pointing,” set the pointing type to “Fixed” and enter the quaternions that represent the boresight of ST1 referenced in Chapter 2.2. In “Constraints”-“Basic,” set the range to a maximum of 1.64558e8 km in order to ensure the sensor reaches past the relevant local orbiting bodies. Repeat these steps for the second sensor to represent ST2. For ST2, refer to Chapter 2.2 to assign the corresponding boresight pointing vector in quaternions.





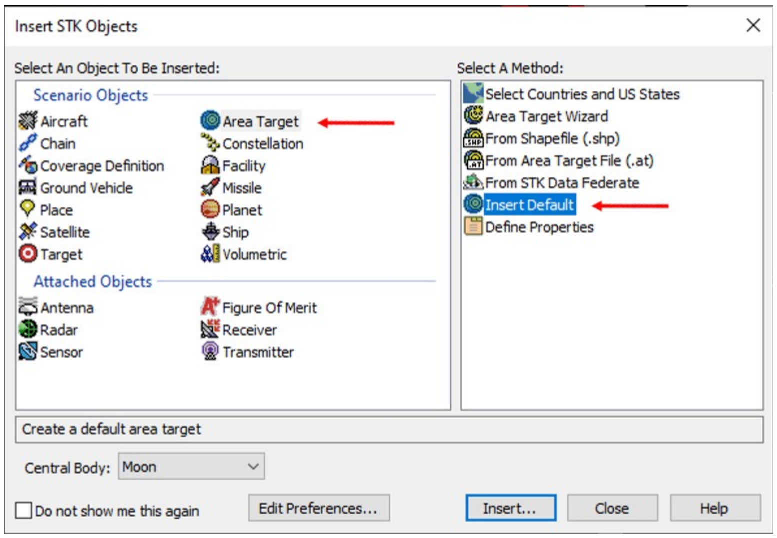


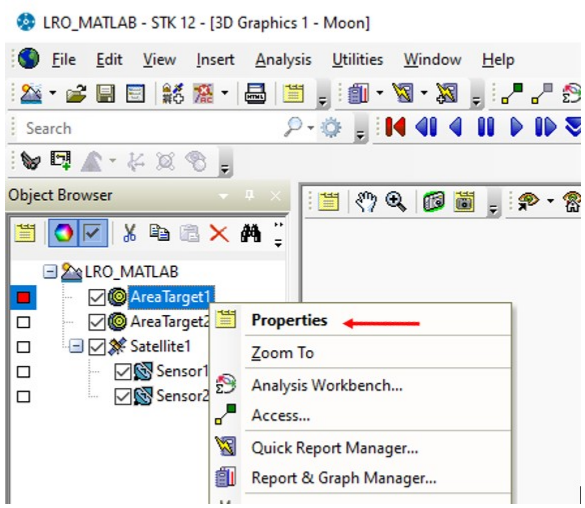


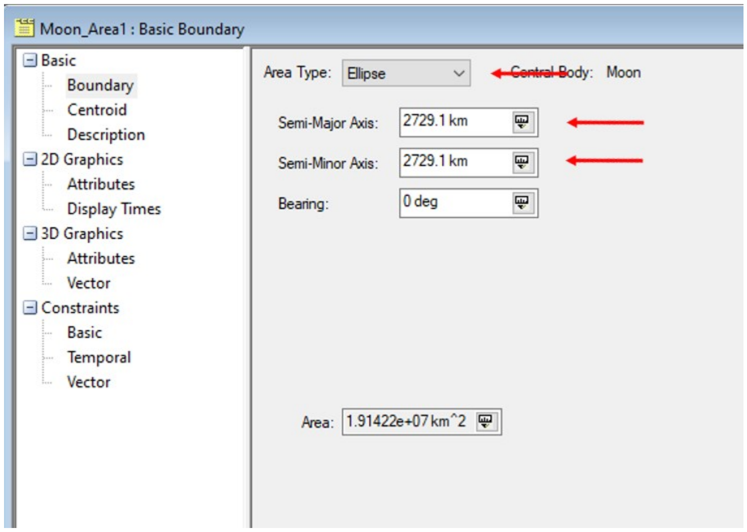
C.4 Area Targets

- As discussed in Chapter 3.1, two area targets must be used in STK to surround the Moon in order to allow the STs to access them as objects. In the menu bar, click "Insert Object," select "Area Target," then double click "Insert Default." Repeat this step in order to create two area targets that will encompass each hemisphere of the Moon.

- Once the area targets are created, right click the first area target in the left hand pane and click "Properties" to establish parameters for area targets. In "Basic"-"Boundary," select "Ellipse" as the area type, and enter 2729.1 km as the Semi-Major Axis, and 2729.1 km as the Semi-Minor Axis. Repeat these steps for the second area target, but enter 2729.2 km for the Semi-Major Axis and 2729.2 km as the Semi-Minor Axis. These parameters will fully encompass the Moon as an accessible object for the STs.

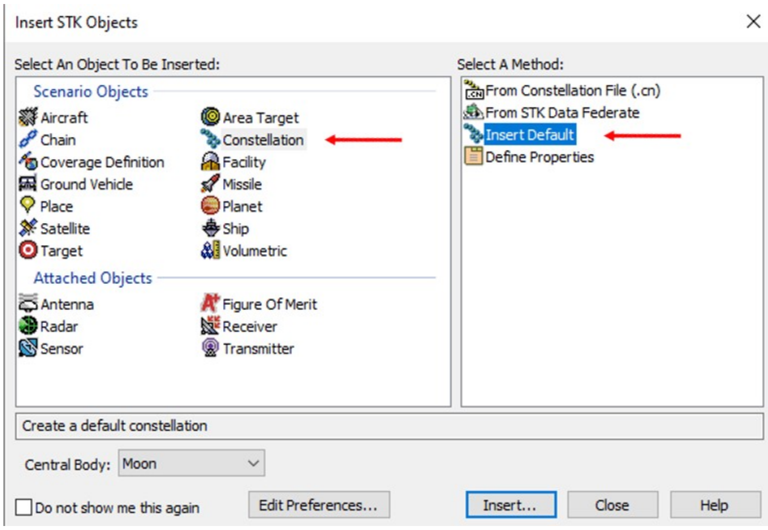


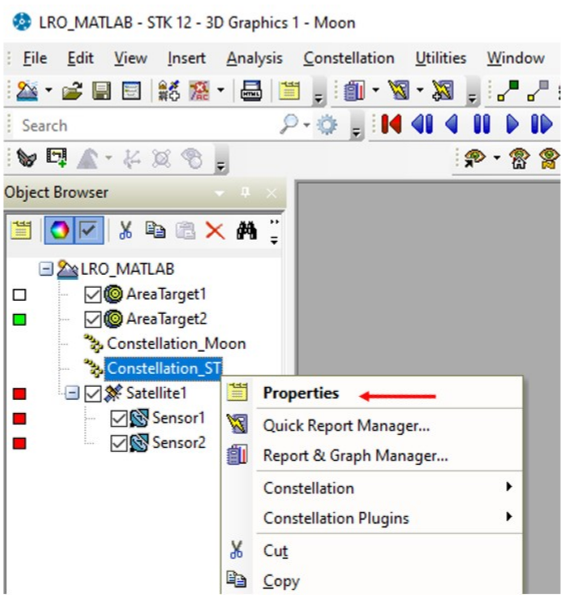


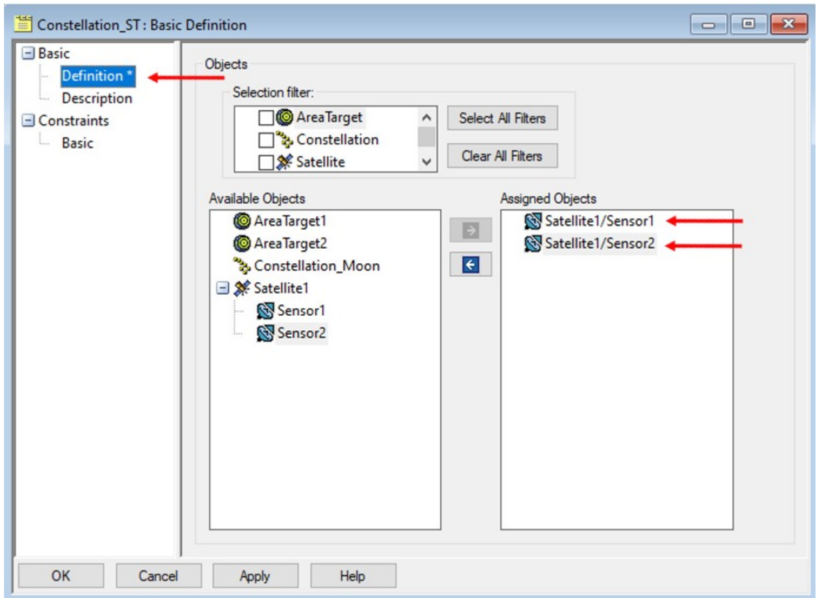


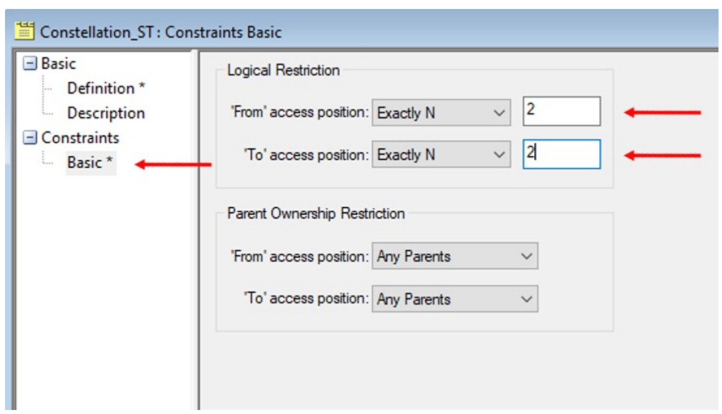
C.5 Constellations

- Both area targets must be treated as one object in order for the MATLAB code to work properly. In addition, because the condition of concern is when both STs are occulted simultaneously, both STs must be given the condition where both must access the Moon area targets to meet the condition. In order to do this, two constellations must be created. In the menu bar, click "Insert Object," select "Constellation," then double click "Insert Default." Repeat this step in order to create two constellations, and label one for the STs and one for the Moon.
- Right click the constellation labeled for the STs in the left hand pane and click "Properties." In "Basic"-"Definition," select "Sensor1" and "Sensor2" as assigned objects. In "Constraints"-"Basic," 'From' access position and 'To' access position should be set to "Exactly N" and "2," as logical restrictions. Click "OK."

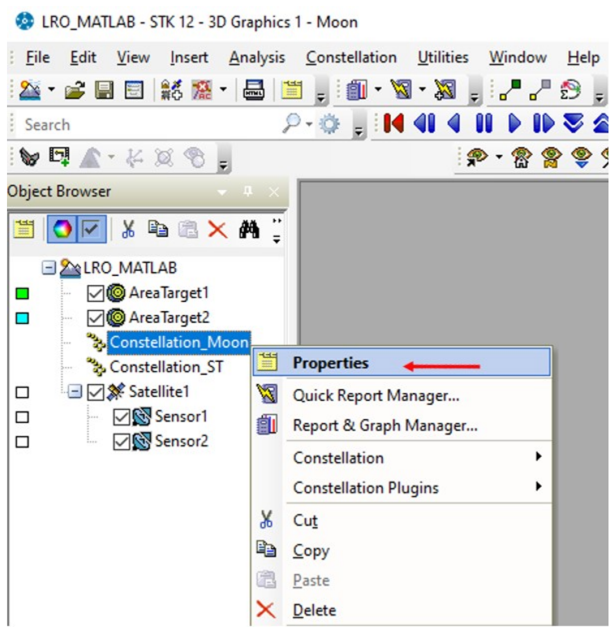

Insert STK Objects
Select An Object To Be Inserted:
Scenario Objects
Aircraft
Area Target
Chain
Constellation
Coverage Definition
Facility
Ground Vehicle
Missile
Place
Planet
Satellite
Ship
Target
Volumetric
Attached Objects
Antenna
Figure Of Merit
Radar
Receiver
Sensor
Transmitter
Select A Method:
From Constellation File (.cn)
From STK Data Federate
Insert Default
Define Properties
Create a default constellation
Central Body:
Moon
Do not show me this again
Edit Preferences...
Insert...
Close
Help

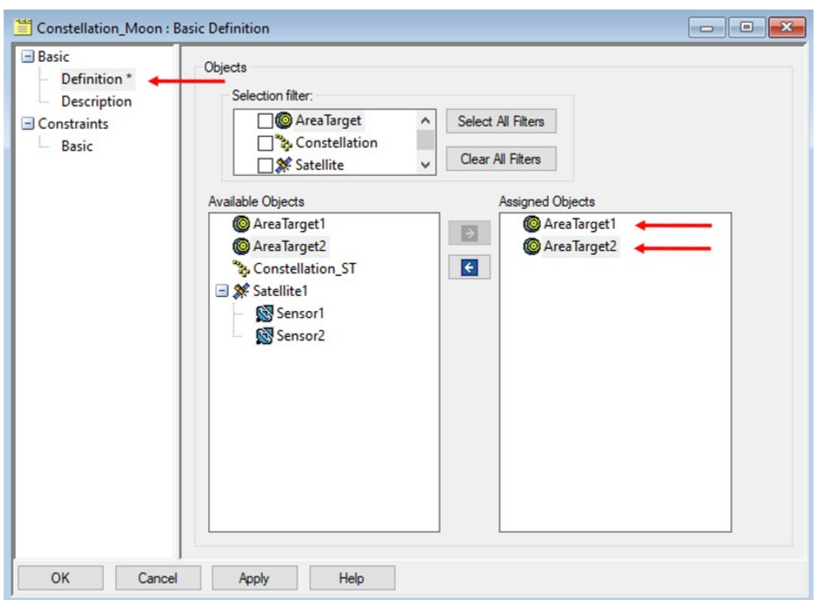

LRO_MATLAB - STK 12 - 3D Graphics 1 - Moon
File
Edit
View
Insert
Analysis
Constellation
Utilities
Window
Search
Object Browser
LRO_MATLAB
AreaTarget1
AreaTarget2
Constellation_Moon
Constellation_ST
Satellite1
Sensor1
Sensor2
Properties
Quick Report Manager...
Report & Graph Manager...
Constellation
Constellation Plugins
Cut
Copy





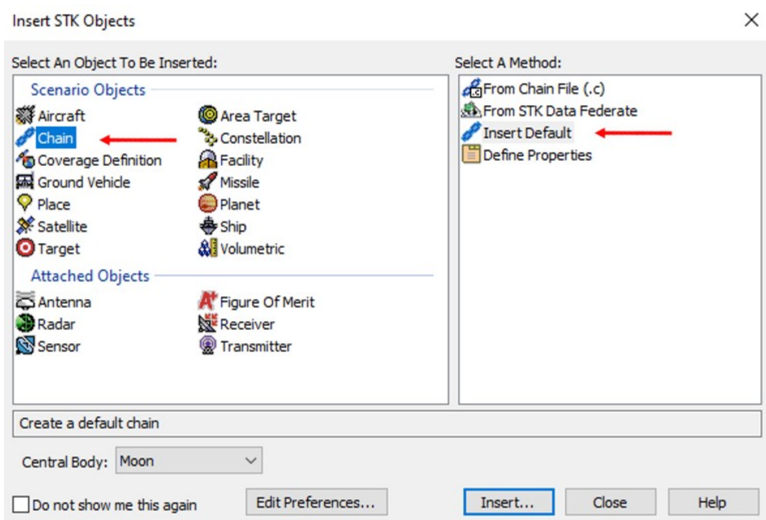
- Right click the constellation labeled for the Moon in the left hand pane and click “Properties.” In “Basic”-“Definition,” select “AreaTarget1” and “AreaTarget2” as assigned objects. Click “OK.”

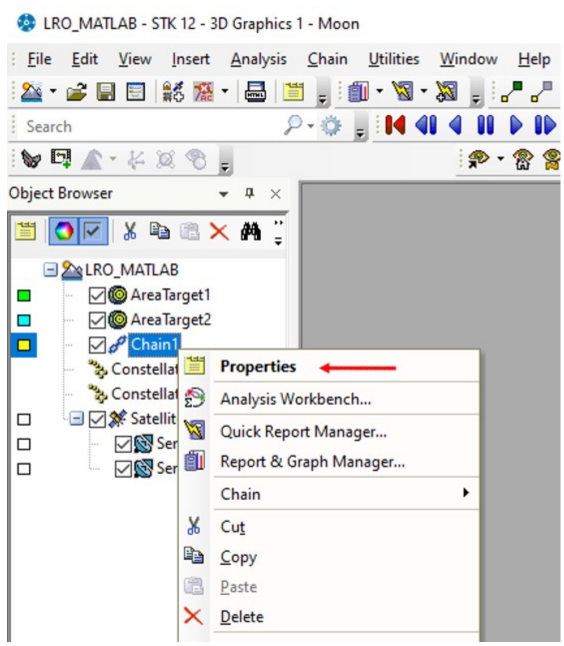


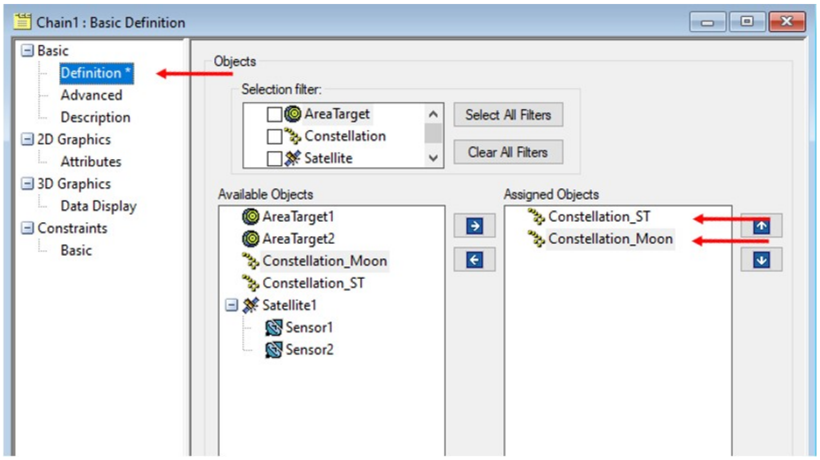


C.6 Chain

- In order for the STs to systematically access the Moon, a chain must be created that accounts for both previously created constellations. In the menu bar, click "Insert Object," select "Chain," then double click "Insert Default." Right click the chain in the left hand pane and click "Properties." In "Basic"-"Definition," select "Constellation_ST" and "Constellation_Moon" as assigned objects in that exact order. Click "OK."






Chain1 : Basic Definition
Basic
Definition *
Advanced
Description
2D Graphics
Attributes
3D Graphics
Data Display
Constraints
Basic
Objects
Selection filter:
AreaTarget
Constellation
Satellite
Select All Filters
Clear All Filters
Available Objects
AreaTarget1
AreaTarget2
Constellation_Moon
Constellation_ST
Satellite1
Sensor1
Sensor2
Assigned Objects
Constellation_ST
Constellation_Moon

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D:
Using the Matlab Program with STK

The following programs are recommended for use with the instructions in Appendix D. Other software versions may work, but have not been tested here.

- Windows 10 OS
- STK 12.0.0
- Matlab 2020a

- Open STK and Matlab, and navigate to the directory containing all relevant scripts.

D.1 Matlab_to_STK.m

- The "MATLAB_to_STK.m" script is the primary script that will will be executed. This script executes all secondary scripts.
- To first establish a connection between Matlab and the current STK scenario, run Section 1. Defining the satellite object that simulates the LRO is included in the following script as well.

```
»clear all; close all; clc;
»uiApplication = actxGetRunningServer('STK12.application');
»root = uiApplication.Personality2;
»root.CurrentScenario.InstanceName
»LRO_MATLAB = root.CurrentScenario.Children.New('eSatellite','LRO_MATLAB');
```

- After running Section 1, navigate to the STK user interface and manually input the desired JPL Horizons ephemeris data in .txt format, to the LRO's orbit definition in accordance with Appendix C.
- Run Section 2. This section establishes sensors, area targets, constellations, chains, analysis workbench, and 3D graphics, according to the LRO's configuration and orbit. "SensorSet.m," "graphicsSet.m," "areaTargets_constellations_chains.m," and "analysisWorkbench.m," are further explained in following sections in this Appendix.

```
»sensorSet
»graphicsSet
»areaTargets_constellations_chains
»analysisWorkbench
```

- Section 3 is where the Monte Carlo nested *for* loop is run, which consists of multiple aspects of the primary code. Before the loop is initiated, the following code describes the LRO's initial attitude to start the loop. The LRO is placed in an "aligned and constrained" profile, where the aligned vector and constrained vector are custom built vectors in "Analysis Workbench," which is explained in further detail in Section D.5. The LRO's aligned vector is set to begin with the $-y$-axis fixed to the Sun vector.

```
»LRO_MATLAB.SetAttitudeType('eAttitudeStandard')
»LRO_MATLAB.Attitude.Basic.SetProfileType('eProfileAlignedandConstrained')
»LRO_MATLAB.Attitude.Basic.Profile.AlignedVector.set...
('ReferenceVector','Satellite/LRO_MATLAB LRO_fixedInAxes.Y')
»LRO_MATLAB.Attitude.Basic.Profile.ConstrainedVector.set...
('ReferenceVector','Satellite/LRO_MATLAB LRO_fixedInAxes.Z')
»LRO_MATLAB.Attitude.Basic.Profile.AlignedVector.Body.AssignXYZ(0,-1,0)
```

- The following lines of code are the only user input sections included in the program, to be edited before running Section 3. The desired date range is set by the user in accordance with the ephemeris data input into STK, and the iteration fidelity is chosen. The iteration fidelity defines the number of attitude iterations as well as the radius of the cone of freedom.

```
»dateStart = '21 Jul 2021 00:00:00.00';
»dateEnd = '05 Sep 2021 00:00:00.00';

»radius = linspace(0,15,5);
»theta = linspace(0,360,24);
»pitch = linspace(0,360,24);
```

- The Monte-Carlo nested *for* loop is represented below. “r” represents the radius of the cone of freedom, “i” represents the θ as the polar angle, and “k” represents γ as the angle of $-y$-axis pitch. The loop utilizes Euler angle A and Euler angle C in STK to systematically loop through polar angles and radii within the cone of freedom. Nested in the *for* loop, the $-y$-axis is pitched at each polar coordinate.
- At the end of each loop, “computeAccess.m” and “serialConversion.m” are run. These secondary scripts are explained in further detail in Section D.6 and D.7 respectively.
- “totalTime_Moon” adds each access instance time period from the reported chain accesses together for each polar coordinate in order to represent the total simultaneous ST occultation time for each simulation run in the Monte-Carlo nested *for* loop.
- “struct_SERIAL” creates a structure to list the results of each occultation instance from each polar coordinate iteration. “struct_totalTime” creates a structure to list the sum of all occultation times for each polar coordinate iteration.
- “postProcess.m” is run after the completion of the Monte-Carlo nested *for* loop. This secondary script is explained in further detail in Section D.8.

```
»for r = radius
»for i = theta
»for k = pitch
»LRO_fixedInAxes.FixedOrientation.AssignEulerAngles('e321',r * cosd(i), 0, r * sind(i));
»LRO_MATLAB.Attitude.Basic.Profile.ConstrainedVector.Body.AssignPR(k,0)

»computeAccess
»serialConversion

»totalTime_Moon = sum(cell2mat(completeChainAccess_Chain_Both_STs_Moon(:,4)));
»struct_SERIAL.(sprintf('r_%1.0f_i_%1.0f_k_%1.0f',r,i,k)) = access_SERIAL_matrix;
»struct_totalTime.(sprintf('r_%1.0f_i_%1.0f_k_%1.0f',r,i,k)) = totalTime_Moon;
»end
»end
»end

»postProcess
```

D.2 sensorSet.m

- "sensorSet.m" is run in the primary script prior to running the Monte-Carlo nested *for* loop in order to defined the LRO's STs in the simulation. Each ST is defined separately. Both STs are established as 15° half angle simple conic, and their boresights are fixed to the SC. Both STs also are constrained to 1.64558e8 km of maximum sensor range to ensure the sensors reach past the Sun.
- The fixed boresight orientation relative to the BCS is defined for each ST separately in accordance with the quaternions listed in Table **??**.

```
% ST1
% Establish STK Child Object: Sensor (Star Tracker 1)
»ST1 = LRO_MATLAB.Children.New('eSensor','ST1');
% Sensor properties (ST1)\basic\definition
»ST1.SetPatternType('eSnSimpleConic')
»ST1.Pattern.set('ConeAngle',15)
% Sensor properties (ST1)\basic\pointing
»ST1.SetPointingType('eSnPtFixed')
»ST1.Pointing.Orientation.AssignQuaternion(-0.775084,-0.340291,-0.52435,0.092218)
% Sensor properties (ST1)\constraints\basic
»ST1_constraint_range = ST1.AccessConstraints.AddConstraint('eCstrRange');
»ST1_constraint_range.set('EnableMax',1)
»ST1_constraint_range.set('Max',1.64558e8)

% ST2
% Establish STK Child Object: Sensor (Star Tracker 2)
»ST2 = LRO_MATLAB.Children.New('eSensor','ST2');
% Sensor properties (ST2)\basic\definition
»ST2.SetPatternType('eSnSimpleConic')
»ST2.Pattern.set('ConeAngle',15)
% Sensor properties (ST2)\basic\pointing
»ST2.SetPointingType('eSnPtFixed')
»ST2.Pointing.Orientation.AssignQuaternion(-0.682597,0.683036,0.182868,0.184611)
% Sensor properties (ST1)\constraints\basic
»ST2_constraint_range = ST2.AccessConstraints.AddConstraint('eCstrRange');
»ST2_constraint_range.set('EnableMax',1)
»ST2_constraint_range.set('Max',1.64558e8)
```

D.3 graphicsSet.m

- In "graphicsSet.m," 2D and 3D graphics displayed in the STK user interface are defined for visual simplicity, but are not required. The following code removes all visual pass tracks in the 2D and 3D view except for the current orbit pass to reduce clutter.

```
% Satellite properties\2D graphics\pass
»LRO_MATLAB.Graphics.PassData.GroundTrack.SetLeadDataType('eDataFraction')
»LRO_MATLAB.Graphics.PassData.GroundTrack.LeadData.set('Fraction',0.6738)
»LRO_MATLAB.Graphics.PassData.GroundTrack.SetTrailDataType('eDataNone')
»LRO_MATLAB.Graphics.PassData.Orbit.SetLeadDataType('eDataNone')
»LRO_MATLAB.Graphics.PassData.Orbit.SetTrailDataType('eDataNone')
% Satellite properties\3D graphics\pass
»LRO_MATLAB.VO.Pass.TrackData.PassData.GroundTrack.SetLeadDataType...
('eDataNone')
»LRO_MATLAB.VO.Pass.TrackData.PassData.GroundTrack.SetTrailDataType...
('eDataNone')
»LRO_MATLAB.VO.Pass.TrackData.PassData.Orbit.SetLeadDataType('eDataFraction')
»LRO_MATLAB.VO.Pass.TrackData.PassData.Orbit.LeadData.set('Fraction',0.6738)
»LRO_MATLAB.VO.Pass.TrackData.PassData.Orbit.SetTrailDataType('eDataNone')
```

- To run smoothly, Matlab must delete RefCrdns that already exist before establishing variables and adding them. Since Matlab cannot perform operations on RefCrdns that already exist or do not exist, the following code must be executed, then vectors can be represented on the LRO. A vector pointing from the LRO fixed on the Sun allows the user to visualize how the LRO reorients its $-y$-axis about the cone of freedom relative to the Sun vector. This assists in ensuring proper operation of the program. In addition, the $-y$-axis, and standard Cartesian coordinates are represented relative to the BCS in order to further assist in observing the real-time attitude of the LRO and ensuring proper operation of the program.

```
% Satellite properties\3D graphics\vector\vectors
»LRO_MATLAB.VO.Vector.RefCrdns.RemoveAll()
»vo_vector_vector_sun = LRO_MATLAB.VO.Vector.RefCrdns.Add...
('eVectorElem','Satellite/LRO_MATLAB Sun Vector');
»vo_vector_vector_sun.Visible = 1;
»vo_vector_vector_sun.LabelVisible = 1;

»vo_vector_vector_solarpanelface = LRO_MATLAB.VO.Vector.RefCrdns.Add...
('eVectorElem','Satellite/LRO_MATLAB Body.-Y Vector');
»vo_vector_vector_solarpanelface.Visible = 1;
»vo_vector_vector_solarpanelface.LabelVisible = 1;

% Satellite properties\3D graphics\vector\Axes
»vo_vector_axes_bodyaxes = LRO_MATLAB.VO.Vector.RefCrdns.Add...
('eAxesElem','Satellite/LRO_MATLAB Body Axes');
»vo_vector_axes_bodyaxes.Visible = 1;
»vo_vector_axes_bodyaxes.LabelVisible = 1;
```

D.4 areaTargets_constellations_chains.m

- In "areaTargets_constellations_chains.m," area targets are defined to allow the Moon to be accessed as an object. Constellations are established to treat both Moon area targets as one object and to define the condition for both STs to simultaniously access the Moon in order for an access to be triggered. A chain is established to define the order of access reporting between the ST constellation and the Moon area target constellation.
- Each area target is established to encompass one hemisphere of the Moon separately. This is required due to limitations with the STK software, where only one hemisphere can be created around an orbiting body at once. The first area target is created as an ellipse with a 2,729.1 km semi-major axis and semi-minor axis. The second area target is also created as an ellipse, but with a 2,729.2 km semi-major axis and semi-minor axis. The order of the area targets do not matter, since the purpose is simply to cover the orbiting body in its entirety. There is no need for higher fidelity of the semi-major

axis and semi-minor axis in order to narrow the 0.1 km gap between the two, since the ST accesses will not be affected.

```
% Establish STK Child Object: Area Target (Moon_Area1)
»Moon_Area1 = root.CurrentScenario.Children.NewOnCentralBody...
('eAreaTarget','Moon_Area1','Moon');
»root.BeginUpdate()
»Moon_Area1.set('AreaType','eEllipse')
»Moon_Area1.AreaTypeData.SemiMajorAxis = 2729.1;
»Moon_Area1.AreaTypeData.SemiMinorAxis = 2729.1;
»Moon_Area1.VO.FillInterior = true;
»root.EndUpdate()

% Establish STK Child Object: Area Target (Moon_Area2)
»Moon_Area2 = root.CurrentScenario.Children.NewOnCentralBody...
('eAreaTarget','Moon_Area2','Moon');
»root.BeginUpdate()
»Moon_Area2.set('AreaType','eEllipse')
»Moon_Area2.AreaTypeData.SemiMajorAxis = 2729.2;
»Moon_Area2.AreaTypeData.SemiMinorAxis = 2729.2;
»Moon_Area2.VO.FillInterior = true;
»root.EndUpdate()
```

- The Moon constellation is created in order to make both hemispheres of the Moon act as one object to be accessed, and reported as such. Both Moon area targets are added to the constellation. The order in which they are added to the constellation does not matter. The ST constellation allows for the condition to be created where both STs must access the Moon constellation in order for an access to be reported. The order in which each ST is added to the constellation does not matter. A constraint is defined that requires the number of objects, being the STs, within the constellation to be exactly two, which represents both STs simultaneously accessing the Moon constellation.

```
% Constellation_Moon
»Constellation_Moon = root.CurrentScenario.Children.New...
('eConstellation','Constellation_Moon');
»Constellation_Moon.Objects.Add('*/AreaTarget/Moon_Area1');
»Constellation_Moon.Objects.Add('*/AreaTarget/Moon_Area2');

% Constellation_STs
»Constellation_STs = root.CurrentScenario.Children.New...
('eConstellation','Constellation_STs');
»Constellation_STs.Objects.AddObject(ST1);
»Constellation_STs.Objects.AddObject(ST2);
»Constellation_STs.Constraints.SetFromRestrictionType('eCnCnstrRestrictionExactlyN');
»Constellation_STs.Constraints.FromRestriction.set('NumberOfObjects',2);
»Constellation_STs.Constraints.SetToRestrictionType('eCnCnstrRestrictionExactlyN');
»Constellation_STs.Constraints.ToRestriction.set('NumberOfObjects',2);
```

- With both the Moon constellation and the ST constellation defined, a simple chain is then created. The chain collects the ST constellation and the Moon constellation into one object. For the chain, the constellations must be added in the order of the ST constellation, then the Moon constellation. This chain is the reference object that will be simulated and reported for accesses.

```
»Chain_Both_STs_Moon = root.CurrentScenario.Children.New...
('eChain','Chain_Both_STs_Moon');
»Chain_Both_STs_Moon.Objects.Add('Constellation/Constellation_STs');
»Chain_Both_STs_Moon.Objects.Add('Constellation/Constellation_Moon');
»Chain_Both_STs_Moon.SetTimePeriodType('eUseScenarioTimePeriod');
```

D.5 analysisWorkbench.m

- The purpose of "analysisWorkbench.m," is to create custom "aligned and constrained" axes, as well as "fixed in axes" axes that will constrain the LRO's $-y$-axis to one fixed attitude relative to the Sun within the cone of freedom, but also allow the LRO to pitch about its $-y$-axis rather than pitch about the Sun vector.

- A custom axis is created in analysis workbench, where the axis type is “aligned and constrained” in order to align the LRO to the Sun vector and constrain it to Body Y. The alignment reference vector is from the LRO to the Sun. The defined attitude alignment direction is along the y-axis, and the defined attitude constrained direction is along the z-axis of the ICRF.

```
»AxesFactory = LRO_MATLAB.Vgt.Axes.Factory;
»LRO_alignConstrain = AxesFactory.Create...
('LRO_alignConstrain', 'Aligned to sun vector and...
constrained to ICRF', 'eCrdnAxesTypeAlignedAndConstrained');...
%('axes type, description, syntax')
»LRO_alignConstrain.AlignmentReferenceVector.SetPath...
('Satellite/LRO_MATLAB Sun');
»LRO_alignConstrain.AlignmentDirection.AssignXYZ(0, 1, 0);
»LRO_alignConstrain.ConstraintReferenceVector.SetPath...
('Satellite/LRO_MATLAB ICRF.Z');
»LRO_alignConstrain.ConstraintDirection.AssignXYZ(0, 0, 1);
```

- A second custom axis is created in analysis workbench, where the axis type is “fixed in axes” in order to enable offsetting the $-y$-axis from the Sun. The reference axis is assigned as the previously custom created “aligned and constrained” axis, while the initial fixed orientation is defined as Euler angles (0,0,0) in the order of 321. Angle A rotates the BCS about the x-axis from the Sun vector, while angle C rotates the BCS about the z-axis from the Sun vector. These angles will be altered throughout the Monte-Carlo nested *for* loop in “Matlab_to_STK.m” in Section D.1.

```
»LRO_fixedInAxes = AxesFactory.Create...
('LRO_fixedInAxes', 'Offset -Y axis from Sun', 'eCrdnAxesTypeFixed');...
%('axes type, description, syntax')
»LRO_fixedInAxes.ReferenceAxes.SetAxes(LRO_alignConstrain);
»LRO_fixedInAxes.FixedOrientation.AssignEulerAngles('e321', 0, 0, 0);
```

D.6 computeAccess.m

- The “computeAccess.m” script directs the program to compute accesses in STK and export the access data back to Matlab. The access report returns four arrays of data, being access number, start time, stop time, and duration of each access. Once the access report is complete, all data is combined into one matrix of values for simplicity and variable reduction. Because the script will terminate execution if no accesses are found through a single simulation, the functions “try” and “catch” are used to prevent termination, and continue to the next iteration. Where there are no accesses, the four columns of the matrix for that given iteration are populated with [0;0;0;0].

```
»format long

»completeChainAccess_Chain_Both_STs_Moon = Chain_Both_STs_Moon;
»completeChainAccess_Chain_Both_STs_Moon.ComputeAccess()

»dp_Chain_Both_STs_Moon =...
completeChainAccess_Chain_Both_STs_Moon.DataProviders.Item...
('Complete    Access').Exec(root.CurrentScenario.StartTime,
root.CurrentScenario.StopTime);

»try
»accessNumber_Chain_Both_STs_Moon =...
dp_Chain_Both_STs_Moon.DataSets.GetDataSetByName("Access Number").GetValues;
»startTime_Chain_Both_STs_Moon =...
dp_Chain_Both_STs_Moon.DataSets.GetDataSetByName("Start Time").GetValues;
»stopTime_Chain_Both_STs_Moon =...
dp_Chain_Both_STs_Moon.DataSets.GetDataSetByName("Stop Time").GetValues;
»duration_Chain_Both_STs_Moon =...
dp_Chain_Both_STs_Moon.DataSets.GetDataSetByName("Duration").GetValues;

»completeChainAccess_Chain_Both_STs_Moon =...
horzcat(accessNumber_Chain_Both_STs_Moon,startTime_Chain_Both_STs_Moon,...
stopTime_Chain_Both_STs_Moon,duration_Chain_Both_STs_Moon);
»catch
»warning('No Accesses: Both STs - Moon')
»completeChainAccess_Chain_Both_STs_Moon = [0,0,0,0];
»end
```

D.7 serialConversion.m

- Start and stop times in the access data are exported to Matlab as time stamp strings, which cannot be manipulated using mathematical operators. In order to rectify this, each time stamp is converted to a serial value by subtracting the starting time stamp entered by the user in “MATLAB_to_STK.m,” explained in Section D.1, from each time stamp in the matrix. The function “datenum” is used to convert time stamp strings to serial values. The resulting serial values are represented as the number of days since the start date, with decimals representing the time of the day. The start and end serial values are then combined into one matrix of values for simplicity and variable reduction.

```
»if iscell(completeChainAccess_Chain_Both_STs_Moon) == 1
»completeChainAccess_Chain_Both_STs_Moon_SERIAL_Start =...
datenum(completeChainAccess_Chain_Both_STs_Moon(:,2))- datenum(dateStart);
»completeChainAccess_Chain_Both_STs_Moon_SERIAL_end =...
datenum(completeChainAccess_Chain_Both_STs_Moon(:,3))- datenum(dateStart);
»else
»completeChainAccess_Chain_Both_STs_Moon_SERIAL_Start = 0;
»completeChainAccess_Chain_Both_STs_Moon_SERIAL_end = 0;
»end

»access_SERIAL_matrix = [];
»access_SERIAL_matrix = horzcat(...
»completeChainAccess_Chain_Both_STs_Moon_SERIAL_Start,... % 1 Moon START
»completeChainAccess_Chain_Both_STs_Moon_SERIAL_end);     % 2 Moon END
```

D.8 postProcess.m

- In “postProcess.m” the structure “struct_totalTime” created in “MATLAB_to_STK.m,” explained in Section D.1, is converted to a cell in order to find the minimum value of all fields within the structure. This minimum value becomes the resultant minimal occultation time which corresponds to the associated LRO attitude iteration populated in the field name.

```
% convert struct_totalTime struct to cell
»array_totalTime = cell2mat(struct2cell(struct_totalTime));

% find minimum value in array_totalTime
»minimumTime = min(array_totalTime);

% findIndex_1 tells findIndex_2 to find the index of that minimum value
»findIndex_1 = array_totalTime == minimumTime;
»findIndex_2 = find(findIndex_1);

% fieldNames lists the fields of struct_totalTime and converts it to cell
»fieldNames = fields(struct_totalTime);

% fieldName_minimumTime identifies the same index to the correct field.
»fieldName_minimumTime = fieldNames(findIndex_2);
»fieldName_minimumTime = cell2str(fieldName_minimumTime);

»fieldName_minimumTime = fieldName_minimumTime(3 : end-2);
```

List of References

[1] National Aeronautics and Space Administration, "Lunar Reconnaissance Orbiter," Jul. 11, 2019 [Online]. Available: https://solarsystem.nasa.gov/missions/lro/in-depth/

[2] Goddard Space Flight Center, "The LRO mission." Accessed Nov. 3, 2021 [Online]. Available: https://lunar.gsfc.nasa.gov/about.html

[3] International Earth Rotation and Reference Systems Service, "The international celestial reference system (icrs)," 2013 [Online]. Available: https://www.iers.org/IERS/EN/Science/ICRS/ICRS.html

[4] Navigation and Ancillary Information Facility, "An overview of reference frames and coordinate systems in the SPICE context." Accessed Oct. 14, 2021 [Online]. Available: https://slideplayer.com/slide/5687394/

[5] J. Garrick, "LRO Attitude Control System (ACS) alignments and coordinate systems," Goddard Space Flight Center, Greenbelt, MD, USA, Tech. Rep., 2008.

[6] Thermal Environments JPL D-8160, "Earth's thermal environment," KK Associates, Tech. Rep., 2008 [Online]. Available: http://www.tak2000.com/data/planets/earth.htm

[7] R. M. Sumanth, "Computation of eclipse time for low-earth orbiting small satellites," *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 6, no. 5, 2019 [Online]. https://commons.erau.edu/cgi/viewcontent.cgi?article=1412context=ijaaa.

[8] J. Giorgini, D. Yeomans, A. Chamberlin, P. Chodas, R. Park, B. Jacobson, M. Brozovic, and T. N. group, *Horizons System*, ver. 4.90, Pasadena, CA, 2021 [Online]. Available: https://ssd.jpl.nasa.gov/horizons/manual.html

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California