Deportment of Electronics and Electrical Engineering

Integrated Mechanical and Electrical Engineering

MENG Final Project Report Academic year 2022/23

GNSS Main and Side lobe signals for Lunar navigation: Analysis and Visualisation

Student name: Juan Sierra

Student number: 179168899

Supervisor name: Biagio Forte

Assesor name: Hui Tang

Word count: 10,980

[This page has been left intentionally blank]

Abstract

NASA and ESA´s motivation to land humans on the Moon this decade as part of the Artemis program has added pressure on the systems used for positioning on and around the Moon. Until now, carrier tracking has been used for positioning in deep space, but this system relies on 3 ground stations which can be susceptible to weather and spoofing incidents. An alternative system is using signals from GNSS satellites available in orbit to find the position of spacecraft on the Moon. This would leverage the signals not only from the main lobe, but also from the side lobes of the antenna pattern which provide a wider field of view, important or Lunar missions.

In this project, the orbital software Freeflyer was used to build a simulation of the GPS constellation, the lunar Gateway spacecraft and a lunar ground station in the South Pole of the Moon. Main and side lobes were created for the GPS antenna and their signal availability was analysed through the course of the lunar orbit.

This paper also provided visualisations that could help understand the working principle of using GNSS signals by providing 3 different point of reference views as well as showing a working model of the main and side lobe signal connections with the Lunar receivers.

In conclusion, it was found that throughout a whole lunar orbit simulation, Shackleton had a 1.54% more side lobe signals detected than Gateway, nevertheless, Shackleton had a 3.98% more main lobe signals detected, which is of more importance as main lobes have up to 6 times more power than side lobes. This is something that NASA will have to take into account when testing the GPS receivers in Gateway and will provide an approximation of the GPS signal availability to expect for future missions on the Lunar South pole.

Future work has been proposed to understand the specific power levels received on the Moon by taking into account the possible effect of Earth’s atmosphere on the signals as well as understanding the GDOP in more detail.

Acknowledgements

I would like to thank my supervisor Biagio Forte for his guidance and empathy, but most importantly for encouraging me to try new ways or methods when I was uncertain.

I would also like to thank the support from my parents, Ross, Mia, Memphis, Chivi, Massa and JJ.

Table of contents

List of figures

Acronyms

NASA - National Aeronautics and Space Administration

ESA – European Space Agency

JAXA - Japanese

PNT - Positioning, Navigation, and Timing

LuGRE - Lunar GNSS Receiver Experiment

NRHO – Near Rectilinear Halo Orbit

RLP – Rotating Libration Points

FOV – Field of View

LEO – Low Earth Orbit

ML – Main lobe

SL – Side Lobe

LSS – Lunar Surface Station

1 Introduction

1.1 Project Background and Motivation

NASA´s Artemis program highlights the importance of the Moon´s exploration as a building block for future manned missions to Mars and beyond[1]. This test-ground to develop new technologies such as reusable human landers and moon bases [2] has also attracted interest from the China National Space Agency, which plans to build a lunar base in the 2030´s, serving as a geopolitical motivation for NASA. Furthermore, evidence of surface water ice on the Moon[3] paired with exponentially lower launch costs (from $26,800 per kilogram in 1995 to $1,500 per kilogram in 2021 [4]) has transformed the possible economic and scientific exploitation of Earth´s satellite, sparking interest from commercial companies around the globe.

More specifically, NASA plans to land astronauts on the Moon by 2028 in the Artemis III mission. Here, NASA and ESA will approach a challenge that hasn´t been faced since the Apollo program more than 50 years ago: Lunar Surface Navigation.

Lunar PNT is complex. As recently as April 2023, the Hakuto-R M1 mission from the Japanese company iSpace was planning to accomplish the first commercial lunar landing in history [5]. Unfortunately, right before surface contact, the connection was lost and the velocity telemetry readings indicated a higher than expected approximation speed [6], resulting in a failed landing.

Up until now, spacecraft going to the moon (orbiting and landing), such as the Apollo missions, China´s Change 1-5 [7], Artemis 1 [8], India´s Chandrayaan-1 [9] and more than 100 more missions since 1959 to 2023 have been mostly using Telemetry, Tracking and Command (TT&C) subsystems. More specifically, the Tracking segment is responsible for locating and following the spacecraft, which, through a process known as carrier tracking [10], a single ground station is capable of locking onto a satellite and determine its position.

Unfortunately, tracking has a limited communication window, specially for Deep space missions, which rely on 3 ground stations located in Madrid (Spain), Goldstone (USA) and Canberra (Australia)[11]. Moreover, these surface ground stations can be vulnerable to signal interference from space weather and terrestrial radio emissions as well as jamming, spoofing and hacking that could compromise lunar missions. In terms of cost, the increased complexity of having dedicated hardware for both on ground and in spacecraft results in high maintenance costs [12].

This has incentivized NASA and ESA to actively pursue a diverse range of innovative navigation solutions to support the ambitious Artemis moon missions, with options using radiometric, laser altimetry, optical navigation, autonomous navigation, and weak-signal GPS under investigation.

Using signals from GPS in particular constitutes an appealing option, as it leverages infrastructure that is already operational. GPS signals emitted from the 31 satellites that constitute the GPS formation point towards the Earth´s surface, but the signal pattern from the satellite antennas can cover an extended area in free space as shown in Figure 1.

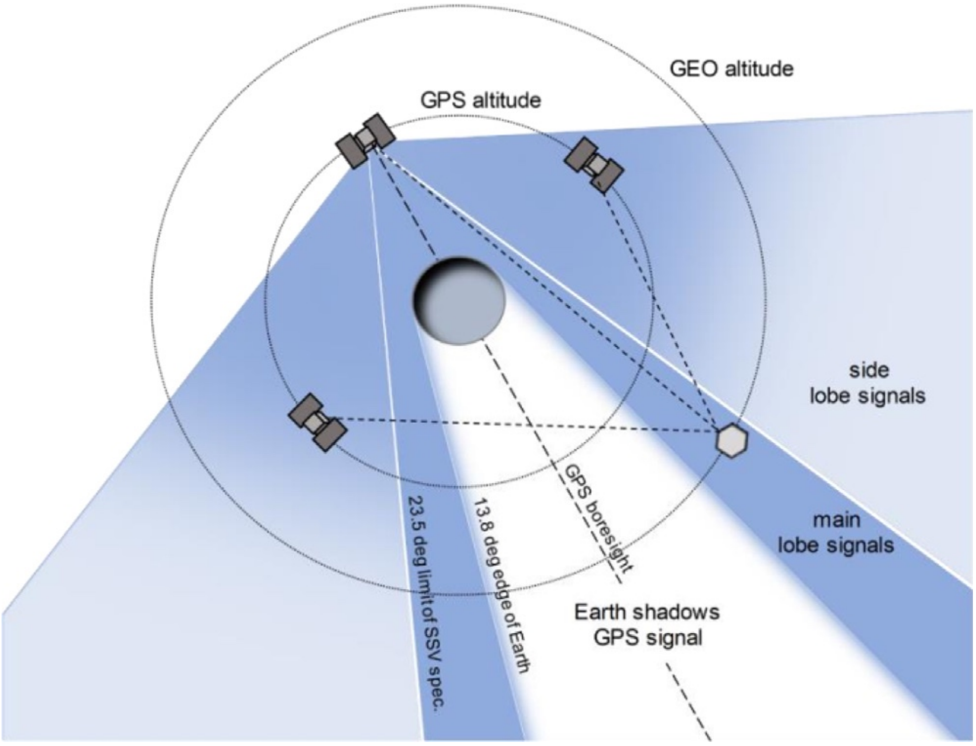


Figure 1 Main lobe and side lobe distribution[13]

The parts of the antenna's radiation pattern outside of the main lobe, which is the major area where the antenna transmits signals with the highest power, are referred to as the side lobes of GPS satellite antennas. Side lobe signals offer a lower decibel power signal, but encompass a wider boresight angle than main lobes, which can be useful when thinking about lunar mission orbits.

1.2 Background theory

1.2.1 GPS working principle

GPS satellites can help determine the receiver's position by a process known as trilateration as shown in Figure 2 below. Each GPS satellite is continuously broadcasting signals that contain information about their current time and the location. Once the signal is received by the receiver, it uses the time difference to calculates a three-dimensional radius (known as pseudorange) at which this receiver could be. Then, with the signal from 2 more satellites, the intersection from these 3d signals determines the receiver´s position. Errors from the receiver´s clock and signal propagation delay through the atmosphere can cause inaccurate measurements known as pseudorange error. Nevertheless, a fourth GPS satellite is used to compensate for the receiver's clock offset, ensuring accurate positioning by taking into account any timing discrepancies between the receiver's clock and the satellite clocks, as even a small error in the receiver´s clock time can result in a large error in its position (due to the high speed of light).

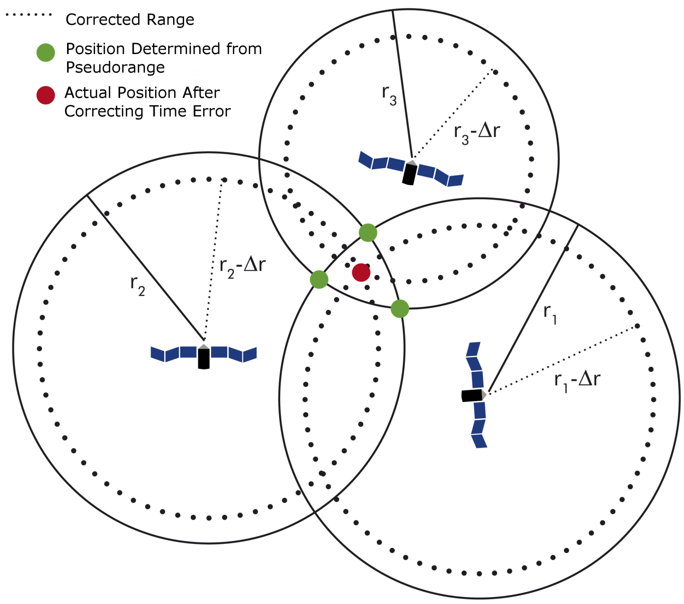


Figure 2 Two-dimensional representation of GPS trilateration[14]

1.2.2 GPS antenna pattern (side and main lobes)

The radiation pattern of an antenna tells us how strongly it radiates in any given direction. The diagram in Figure 3 illustrates this by showing a 3D radiation pattern in which you can see the main and side lobes.

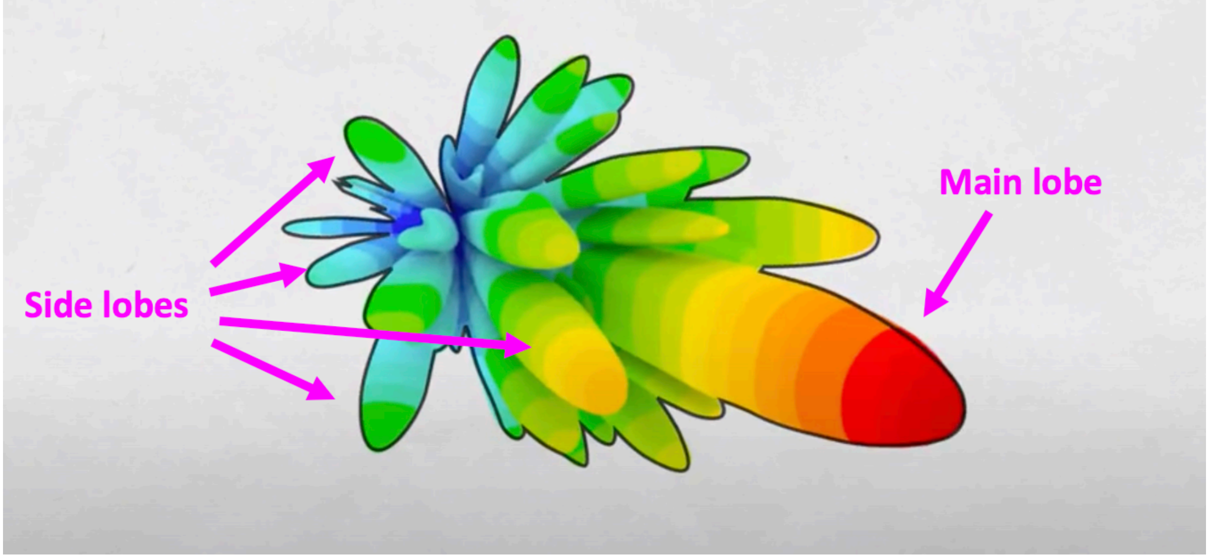


Figure 3 Example of an antenna radiation pattern showing main and side lobes

The main lobe is shown in red and is the largest, showing that it has the highest signal power, also referred to as "Gain". It is usually directed towards the target receiver to maximise the range of the signal and minimise loss.

The rest of the lobes are considered the "side lobes". These can transmit and receive signals to "unwanted" directions causing noise in other receivers and are therefore seen as inefficient in terms of power.

Nevertheless, they offer a greater field of view than the main lobe which can be useful for spacecraft in Lunar missions.

1.3 Software selection

After identifying the main orbital software simulators, the 3 most suitable candidate software's were:

- Systems Tool Kit: Commercial aerospace simulator developed by the company AGI
- GMAT (General Mission Analysis Tool): Orbital simulator developed by NASA to design space missions
- Freeflyer: Commercial aerospace simulator developed by AI Solutions for satellite mission planning

In terms of the User Interface, all of the 3 softwares provide a combination of a user interface and scripting capabilities that allows the creation of objects such as satellites, ground stations etc, in which their parameters can be customized. Nevertheless, GMAT doesn´t have options to simulate signals, which can be a limiting factor in the completion of this paper´s aims.

In terms of the scripting language, GMAT is the most limited option as it doesn´t have a debugger, so, when an error occurs in the script (either logical or syntax), GMAT doesn´t specify the nature of this error, only its location, which can be a large disadvantage when developing complex programs.

The most important factor however, is the software´s availability and price. GMAT is an open source software that can be downloaded by anyone (academic or not) from the SourceForge GMAT website. STK is a commercial software that can be accessed by some institutions with paid licenses (of $2500 a month), of which the University of Bath doesn´t own any. Freeflyer is also a commercial software, but it sometimes offers individual licenses (of a few months) to academics that can prove their academic status.

Freeflyer was therefore selected as it offered better capabilities than GMAT and similar to STK, but unlike STK, a free license can be obtained when the academic status was proved.

1.4 Project Aim

This project aims to create simulation scenarios in an orbital simulation software to illustrate the concept of using GPS main and side lobe signals for educational purposes as well as analysing the feasibility of using GPS signals in these scenarios.

In order to complete the aim of the project, the following objectives were created:

- Evaluate the simulation software options available
- Select which reference mission to simulate
- Build the simulation scenario
- Create visualisations that will help illustrate the mission
- Develop an algorithm that will allow signal detection from the main and side lobes
- Write conclusions, learnt lessons and future work

2 Literature Review

2.1 Introduction and selection process

There are several aspects to be considered from past literature before starting the simulation process. Firstly, the feasibility of using GPS signals above the GPS orbits as well as the use of side lobes was reviewed. Then, more focused on the simulation, the reference mission being implemented, the software being used, the visualizations created and the parameters being measured were analysed. Scopus and Google scholar were used to find relevant literature by using keywords such as "Lunar GNSS", "Lunar Navigation", "Moon GPS", "Lunar PNT", "GPS side lobes" etc. The papers found consisted of a combination of using GNSS signals for positioning in Lunar missions (the ideal paper), investigations on signal strengths received from GPS satellites above LEO and simulations of GPS-like satellite constellations deployed on the Moon for PNT. The two latter were considered as they could still provide valuable information on the software being used, tested reference missions that could be used in the simulation as well as showing different visualization approaches. Most of the research papers were direct or indirect collaborations with NASA and ESA, but there were also independent university and commercial investigations.

2.2 Using GNSS signals for other purposes

Even though GPS signals are designed to be used for positioning on Earth's surface, it has been demonstrated by We et al [15] that Low Earth orbits are completely covered by these signals. In fact, GNSS signals have been used before for missions in LOE (Swarm, Sentinel, MetOp)[16], so it's not the first time their signals are not used for Earth PNT.

But LEOs serve of little use for lunar missions. That is why Benjamin et al [17] analysed the results of the MMS and GOES-16 missions and demonstrated that by taking advantage of both the main and sidelobes (as seen in Figure 5), for Highly elliptical Earth orbits that reached an apogee of 150,000km (40% the distance to the Moon), position could be calculated. Moreover, a recent experiment demonstrated for the first time the capability of receiving GPS signals further away than GEO, with a spacecraft that detected signals 187,176km from Earth, which is half the distance between the earth and the moon (Baird, 2019)[18].

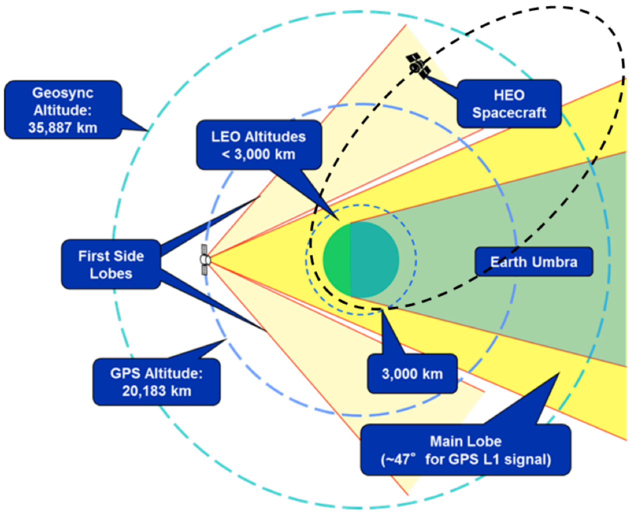


Figure 4 Geometric view of GPS signal use in space.

Further testing of these "beyond Earth" capabilities of weak-signal GPS are on its way as, for example, the NASA and the Italian Space Agency are planning to test this with the Lunar GNSS Receiver Experiment (LuGRE), set to demonstrate GNSS-based positioning, navigation and timing on the moon as soon as 2023 [19].

2.3 Reference mission selection

Within the context of lunar missions, several reference missions have been simulated to test the feasibility of using GNSS signals for positioning. Pietro et al [20] simulated the Deep Space Gateway (DSP) orbit as it argued that it will be used by Artemis´s lunar Gateway, which has been already had its spending approved by the United States congress and is set to send the first module in December 2024, therefore having the practicality of testing the simulation with a real receiver in the near future. Furthermore, the Lunar South Pole´s Shackleton crater is the possible landing site for these Artemis missions as the Moon´s axis crosses right through the crater, giving it almost continuous connection with Gateways highly elliptical orbit.

There were other investigations that have tried simulating the received GNSS signal for future missions. Hugo et al [21], for example, simulated ESA´s Lunar Lander Mission, but it was different to the previous article as this simulation took into account launch, transfer, moon orbit, descent and landing, and surface operations. A trend was identified amongst most simulations in which the reference missions were mostly based on LTO (Lunar Transfer Orbits) as the one shown in Figure 6 from [22], rather than stable orbits already on the Moon or Lunar surface.

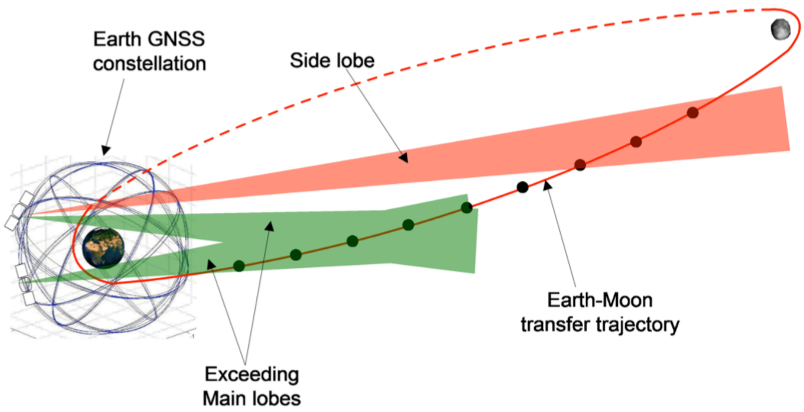
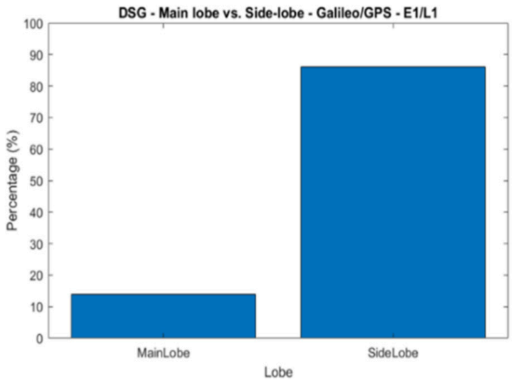


Figure 5 Geometrical visibility of main and side radiation lobes [22]

There were variations within the transfer orbits selected. Jia Tian et al [23] argued that even though there were 2 more novel cost effective (but slower) transfer types, they would simulate a direct Lunar transfer as it has been used in actual lunar missions repeatedly in the 1960´s and 1980´s. A simulation by Alberto et al [24] even included positioning for rover operations up to 500m from the landing site on the Lunar surface, which was also the only article which simulated two different missions by having a second Lunar Transfer Orbit but this time approximating the Moon at an inclination of 30 degrees and landing at a latitude of 25 degrees north.

2.4 Parameters being measured (side/main lobe):

In terms of the metrics being measured in the results, there was a general trend of determining the signal availability from the GPS in almost all lunar simulations investigated. Simulations that focused on orbits already on the moon, like Pietro et al [20] with the a Near rectilinear Halo orbit on the moon, determined signal availability, signal to noise ratio and calculated geometric dilution of precision too. They also made distinction between main and side lobes (one of the few papers that did), however, in the results this was only limited to percentage occurrence of main and side lobes as shown in Figure 7, as the visibility tracking made no distinction between main and side lobes (only between GPS+Galileo and all GNSS constellations as shown in Figure 8).



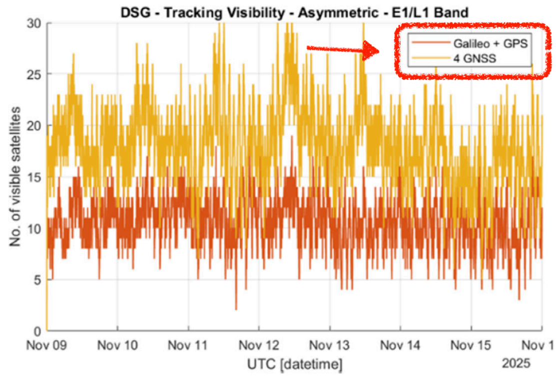


Figure 6 analysis of the GNSS pattern off-boresight angles *Figure 7 Tracking Visibility analysis for Galileo and GPS [20]*

A common trend that was identified was comparing the parameters such as GDOP, signal availability and signal power for different GNSS constellations. Apart from Pietro et al´s article mentioned earlier, Vincenzo et al [23] compared its results between using the GPS and Galileo constellations separately and together. Orbital stability [25], receiver module architecture [22] and descent velocity [20] were other parameters that have also been measured but lack a significant contribution towards understanding GNSS signal detection for lunar missions.

2.5 Antenna Pattern model

GPS satellites have different antenna pattern characteristics as different models have been sent in blocks since the 1990´s, with the most recent block sending its first satellite in 2018. Figure 9 shows the different blocks and their respective operational quantity in orbit.

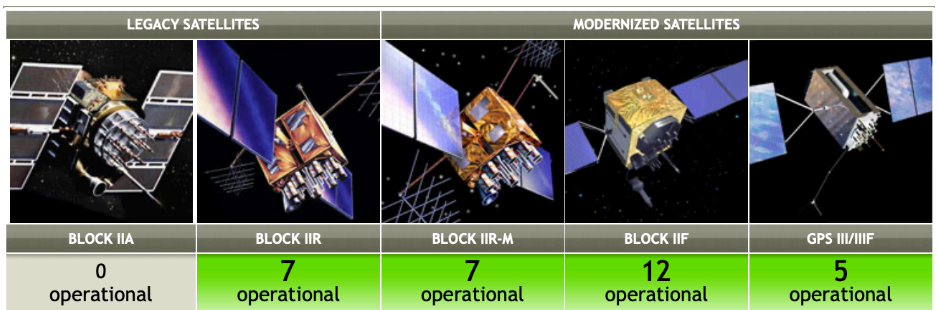


Figure 8 GPS satellite blocks and their respective quantity in orbit [25]

Each different satellite block has a slightly different antenna radiation pattern, therefore, an important aspect to consider is what radiation pattern to use as this can have an effect on the outcome of our simulation.

Most of the articles used a simplified model of the radiation pattern such as Witternigg et al [26], which used a main lobe aperture off-boresight angle of 27 degrees and a sidelobe aperture angle of 57 degrees. Vincenzo et al [23] used the antenna pattern of block IIR, which was developed by Lockeed Martin and the antenna details provided by Marquis and Reigh, 2015 [27]

Pietro et al developed the most realistic model of the examined literature as they used three different types of radiation patterns for three block types. The block IIR model used by Marquis and Reigh, 2015 [27] was also used. For the Block IIF built by Boeing, the radiations patterns were obtained from NASA´s GPS Antenna Characterization Experiment (ACE). Galileo radiation patterns were obtained through a confidential internal study at ESA and were not displayed in their article.

Some investigations also performed the simulations using the same antenna pattern (shown in Figure 10) for all the satellites, even using GPS antenna radiation patterns for Hugo et al [21] Galileo satellites due to the lack of data available for these.

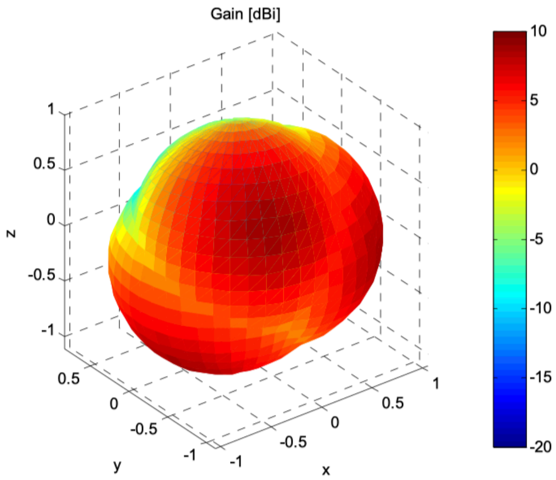


Figure 9 Composite radiation pattern of the 3 high gain antenna layout [21]

2.6 Visualizations created

Most of the visualizations presented by the literature reviewed had the following issues:

- Lack of different perspectives. As most of the papers only presented one point of view as shown in Figures 11 and 12 by Pietro et al [20] and Vincenzo et al [23] respectively, it was difficult to understand how the GPS signals were being received by the Gateway or Shackleton receivers

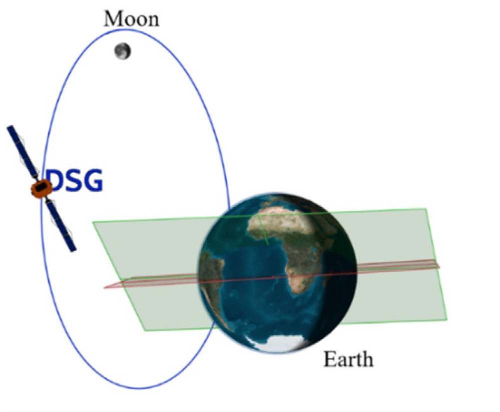


Figure 10 The NRHO of the DSG depicted in STK with the Earth and Moon equatorial planes in green and red (left) [20]

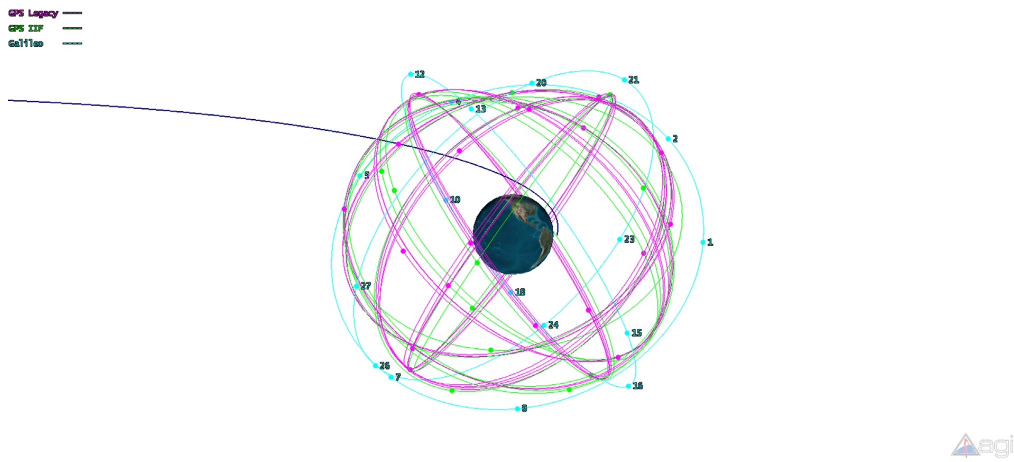


Figure 11 Super-GPS orbit-like orbit (red: GPS orbits; blue: ESMO orbit) [28]

- No closeups: All the visualisations from the literature were from distances in which the
- No visibility vectors: It wasn´t possible to visualise how many GPS signals were receivable from GPS or Gateway at any given point.
- No progression: There were no timelapses in the orbits which helped give a sense of time. Only Giovanni et al [28] provided some sort of timelapse, even though it was about the Lunar Transfer orbit and the celestial objects and spacecraft weren´t labelled.
- No main and side lobe visibility: There are no papers which provide a visualization of the main and side lobes, making it difficult to understand how the system works.

2.7 Literature review conclusions and link to Project aims

Most of the literature is simulating Lunar Transfer Orbits that don´t resemble any future missions. Therefore, as the Gateway mission which will orbit the moon in a highly elliptical orbit (NRHO) can be simulated as the first module with signal receivers will launch as early as December 2024 so that the simulation results can be compared with the real measurements. Moreover, as part of NASA´s Artemis III mission will include landing on the lunar south pole (most likely Shackleton crater), a simulation of the GPS signals received by a ground station here could help investigate the difference in signal availability between these two scenarios. Furthermore, as there is no literature that shows the main and side lobe signal availability throughout the spacecraft receiver´s orbit, this would be one of the papers objectives. An effort to include more perspectives in the visualisations as well as show how many satellites are connected at any given point making a distinction between main and side lobes.

3 Method

3.1 Simulation scenario

The simulation will include the following objects:

- Earth, Moon and Sun (taking into account their gravity)
- 31 GPS satellites (the number of current operational GPS satellites)
- Gateway spacecraft following its NRHO (Near Rectilinear Halo Orbit)
- Lunar Surface Station (located at Shackleton crater)
- Side and main lobe signal emitter models (referred to as sensors in Freeflyer) for each GPS satellite
- Vectors connecting GPS main and side lobe signals with the Gateway/Shackleton receivers

3.2 Simulation intro

To build the simulation, first the following Freeform scripts were created by following the steps in Figure 13.

- **Set up Objects**: Where all the objects such as GPS satellites, Gateway, visibility segments, vectors etc will be created and their parameters specified
- **Set up Outputs**: Where the different visualisation views and graphs will be defined
- **Propagate**: Where all the spacecrafts will be iterated to the next time step and the visibility conditions will be analysed.

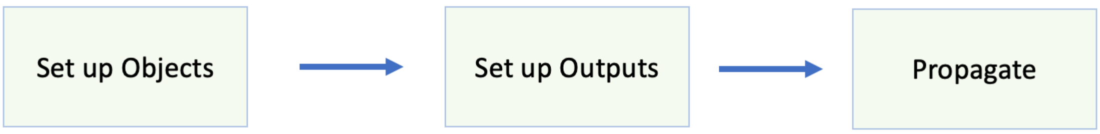


Figure 12 Freeflyer script order

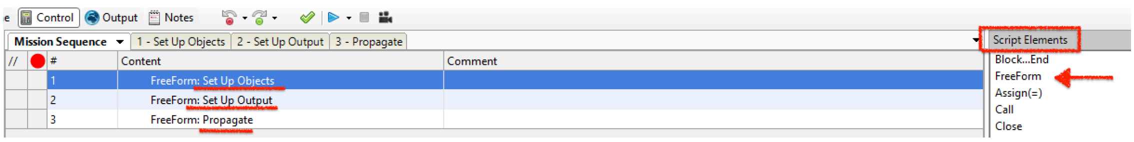


Figure 13 Freeflyer freeform script creation

3.3 Set up objects:

Objects in Freeflyer include anything from spacecraft, celestial bodies, vectors, plots etc. They can represent physical entities, such as spacecraft or ground stations, as well as processes like finite burns or orbit determination. Every FreeFlyer mission plan contains objects, each uniquely named. These objects can be created and edited through the Object Editors (GUI as show in Fig.x) and FreeForm Script Editors.

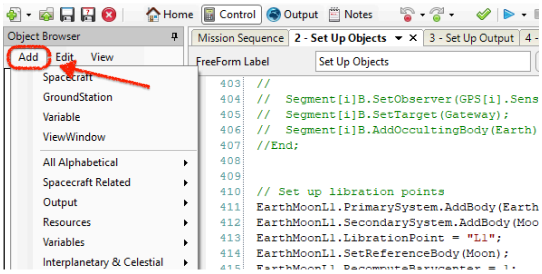


Figure 14 Adding object through Freeflyer GUI

3.3.1 GPS

There are currently 31 operational GPS satellites orbiting Earth [25]. They are distributed in 6 orbital planes, with 5 satellites in each plane except 1 plane which has 6. The orbits are specified through a set of parameters known as the Keplerian orbital elements.

Keplerian orbital elements are a set of six parameters that define an orbit in a two-body problem, such as a satellite orbiting a planet. They describe the size, shape, orientation, and position of the orbit in a given coordinate system.

The orbit size is represented by the semi-major axis (a), which is half the distance between the orbit's closest and farthest points. Eccentricity (e) indicates the form of the orbit, with 0 representing a circular orbit and values ranging from 0 to 1 indicating an elliptical orbit. Inclination (i) is the angle formed by the orbit plane and a reference plane that defines the tilt of the orbit. The right angle of the ascending node defines the orbit's orientation in space, whereas the argument of periapsis determines the orbit ellipse's orientation within the plane. The orbiting body's position along its orbit is described by true anomaly (v) measuring the angle between periapsis and the body's current position.

These six orbital elements uniquely define an orbit and are used to determine the position and velocity of an orbiting body at any given time. These elements are represented visually in Figure 17.

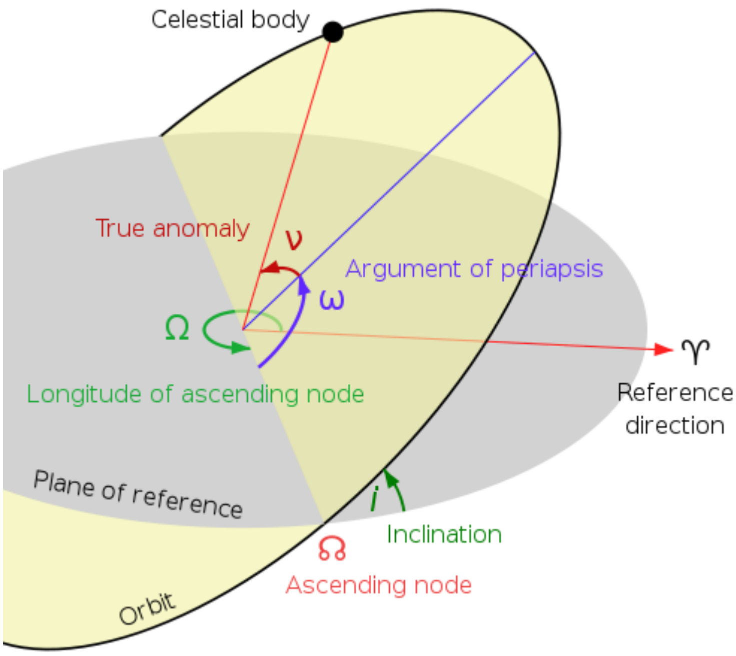


Figure 15 Keplerian orbital parameters [29]

In the specific GPS constellation, every satellite has the same inclination of 55 degrees. Each plane has the same RAAN with a 60 degree difference of RAANS between planes. Within each plane, the satellites are separated by their true anomaly, in 60 degree intervals. The precise parameters obtained were from Celestark [36], an organisation who works with National Coordination Office for Space-Based Positioning, Navigation, and Timing to keep an Almanac of orbital parameters as shown in figure x below

```
******** Week 195 almanac for PRN-01 ********
ID:                          01
Health:                      000
Eccentricity:                0.1216316223E-001
Time of Applicability(s):   61440.0000
Orbital Inclination(rad):    0.9890425019
Rate of Right Ascen(r/s):   -0.7703178011E-008
SQRT(A)  (m 1/2):            5153.612793
Right Ascen at Week(rad):   -0.1384280966E+001
Argument of Perigee(rad):    0.937892182
Mean Anom(rad):              0.2504369694E+001
Af0(s):                      0.2298355103E-003
Af1(s/s):                   -0.3637978807E-011
week:                         195
```

Figure 16 GPS 1 Orbital parameters example from Celestark [36]

The initial orbital parameters used were from UTC 1st of January 2023 to 28th January 2023. All these values were added through the GUI by selecting the desired satellite in the object browser column to the on the left side of the GUI, then selecting the "Orbit" option, changing the element type to Keplerian as shown in Figure 19. and then entering the specific orbital parameters as mentioned previously.

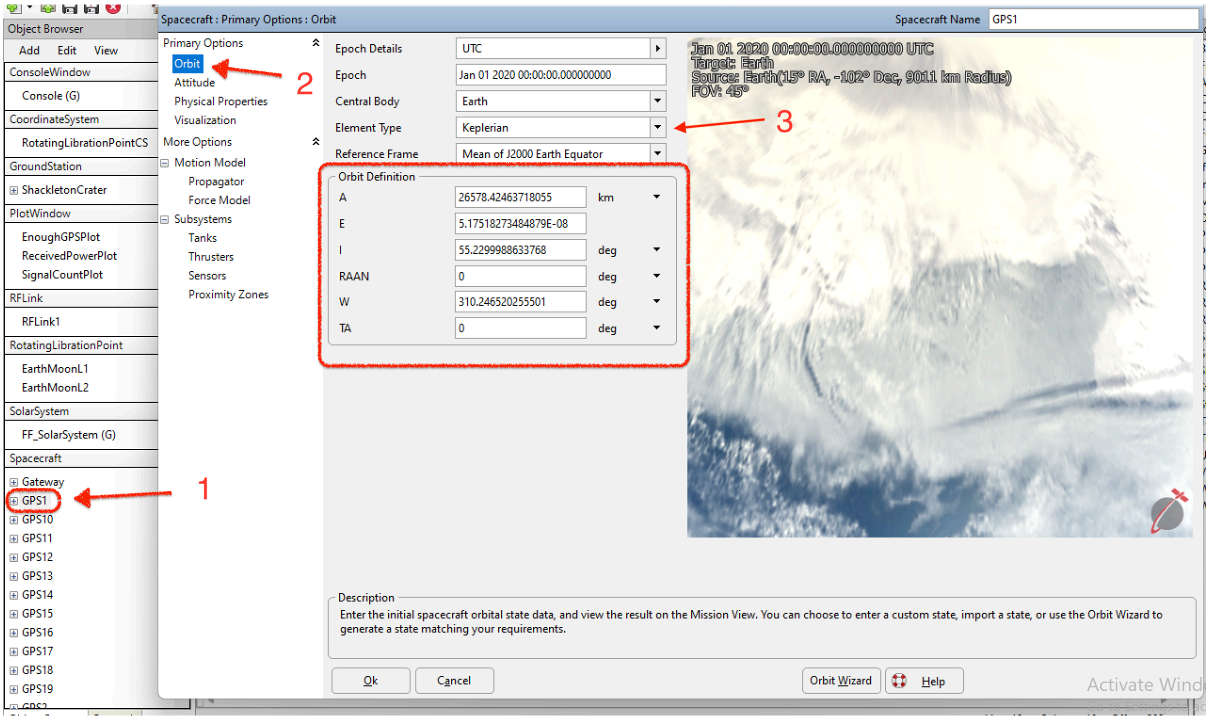


Figure 17 Freeflyer Orbit Determination Keplerian Parameters

3.3.2 Sensors (Signal)

The radio signal emitted by each GPS can be modelled by creating a sensor that replicates the geometry of the antenna radiation pattern through a simplified cone model with the same aperture angles as shown in Figure 20. The antenna gain pattern used was the IFF block from because as shown in the literature review, recent studies are using the antenna gain pattern of the IFF block, which constitutes the most numerous satellite block of the current GPS constellation with 12 satellites [25].

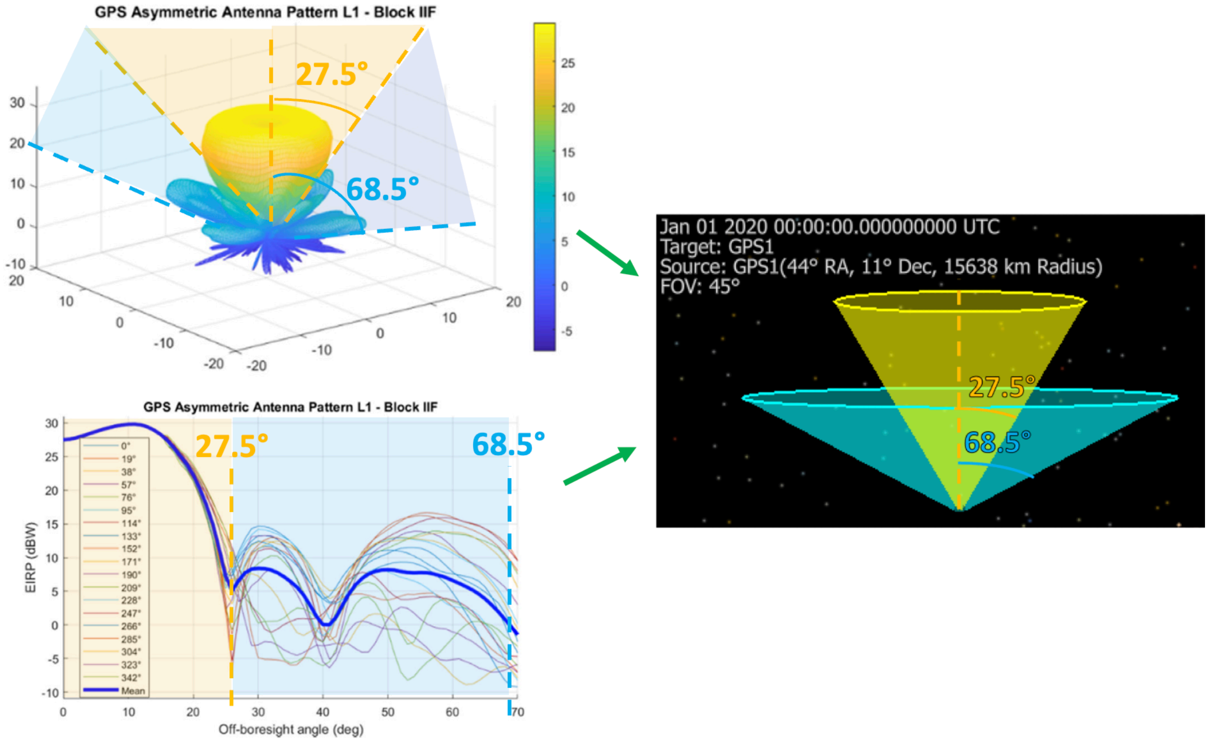


Figure 18 Model creation of the GPS IIF block antenna pattern into Freeflyer

The main and side lobe sensors shown in the figure above in Freeflyer can be added as a subsystem of each GPS as shown in the following Figure 21 and the half angle edited accordingly to match the antenna pattern aperture angles.

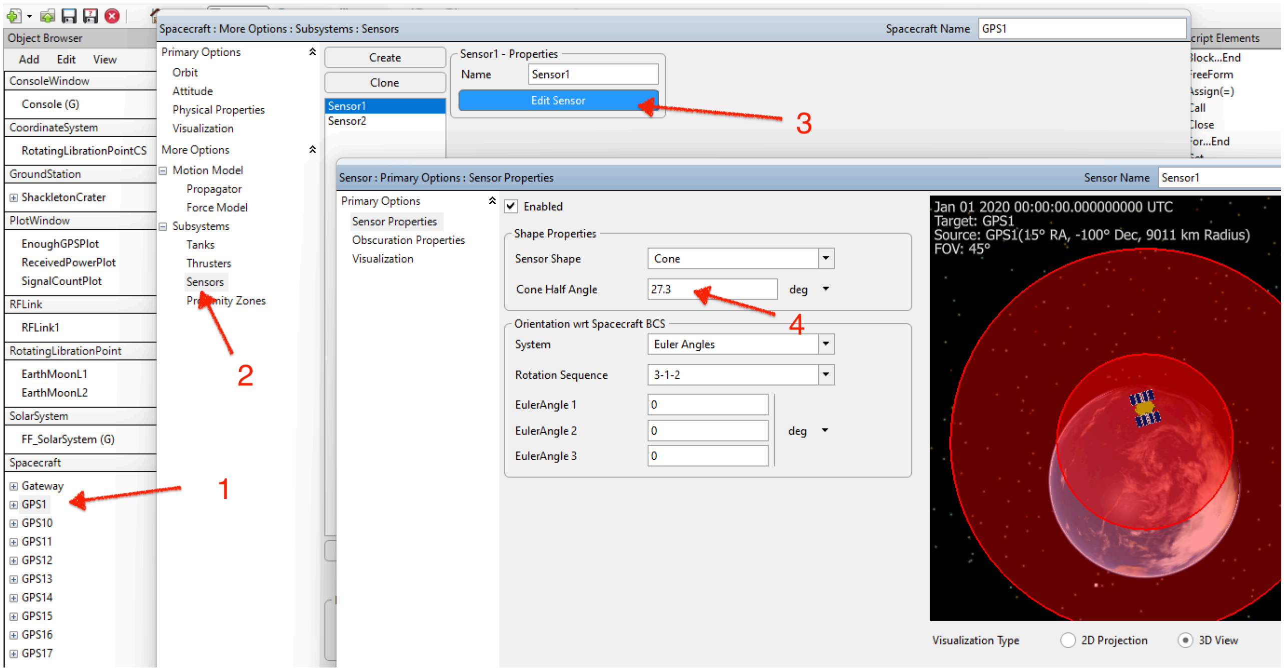


Figure 19 Main and side lobe model creation through Freeflyer object subsystem

3.3.3 Gateway

The Gateway station will follow the NRHO of the moon mentioned in the literature review. This orbit is complex as it orbits the moon but its fixed to the two Lagrange points of the Earth and the Moon. Orbits can be determined through orbital parameter inputs (such as Keplerian as used for the GPS), but also by using Ephemeris files.

An ephemeris file in FreeFlyer is a data file that contains details about the position, velocity, and other pertinent characteristics of a celestial object at particular time intervals, such as a spacecraft or planetary body. For mission planning, analysis, and visualisation, the ephemeris data can be utilised in FreeFlyer to initialise and propagate object orbits as well as to simulate object motion.

Several ephemeris file formats are supported by FreeFlyer, including:
SPICE kernels: Can store ephemeris data as well as other ancillary information including orientation and reference frame data. They were created by the Navigation and Ancillary Information Facility (NAIF) at NASA's Jet Propulsion Laboratory (JPL).

TLE: composed of two lines of text (frequently used for Earth-orbiting satellites), which carry the Keplerian Orbital Parameters

Freeflyer provides the Ephemeris file of the NRHO, which can be accessed in the “Support_Files” folder, located within the “Sample_Missions_Plan” folder. This Ephemeris file was retrieved from NASA's Navigation and Ancillary Information Facility (NAIF) [30] at JPL by Freeflyer.

To upload it into the Gateway spacecraft the following steps shown in Figure 20 were followed.

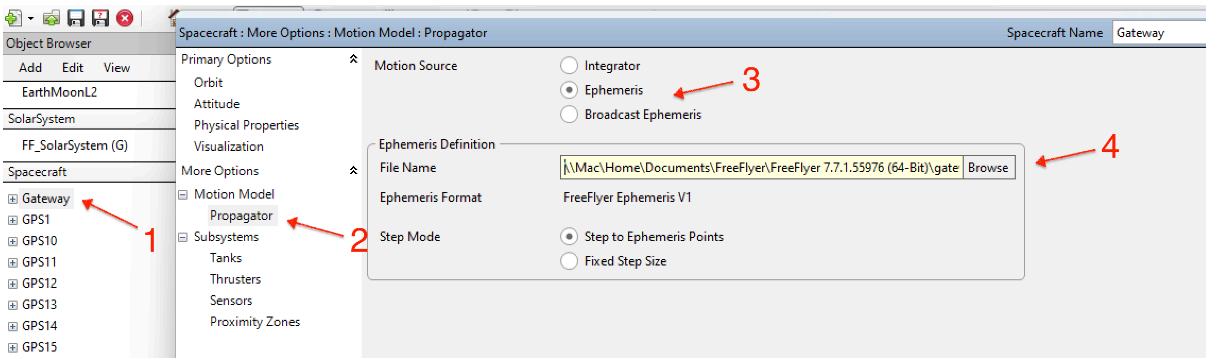


Figure 20 Steps to upload Ephemeris file as propagator model for Gateway

3.3.4 Shackleton

The Lunar ground station located at shackleton crater can be added through the object browser by selecting the “GroundStation” object and its location can be edited by following the steps in Figure 23 below. The coordinates of the crater are 89.9 degrees South Latitude and 0 degrees East longitude and the planetodetic height was set to 17 metres (which is the same as NASA´s Deep Space Network Antennas on Earth [32].

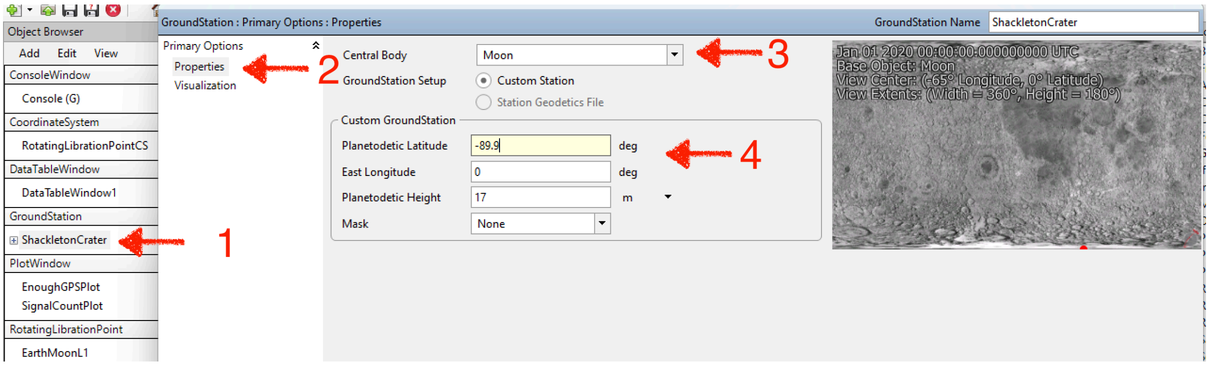


Figure 21 Steps to customize lunar surface station at Shackleton crater

3.3.5 RotatingLibrationPoints & Coordinate Systems

The two Earth-Moon lagrange points as well as the rotating coordinate system that will be used in the output window to visualize the Earth-Moon fixed reference frame, but firstly, the objects must be created through the object browser by adding two “RotatingLibrationPoint” objects and a “Coordinate System” object shown in Figure 24 and Figure 25 respectively.

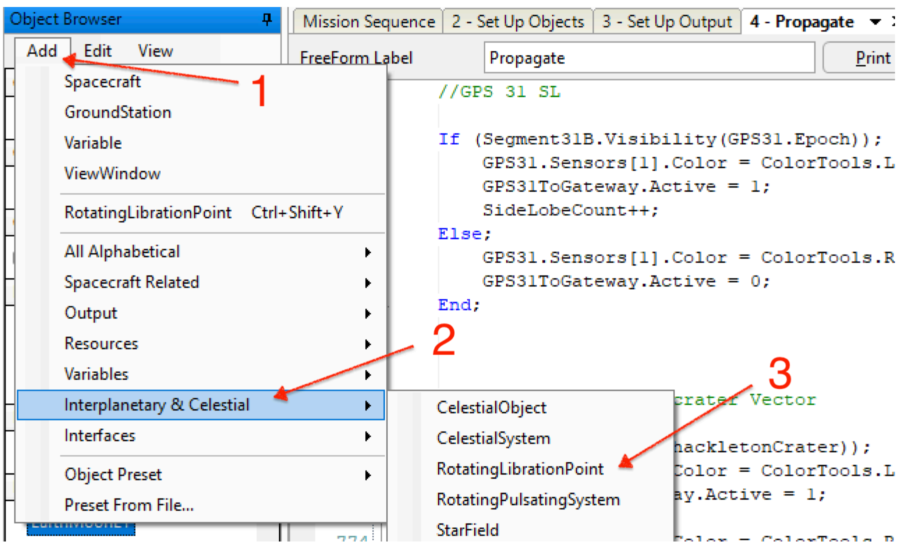


Figure 22 Steps to add a "RotatingLibrationPoint" in Freeflyer

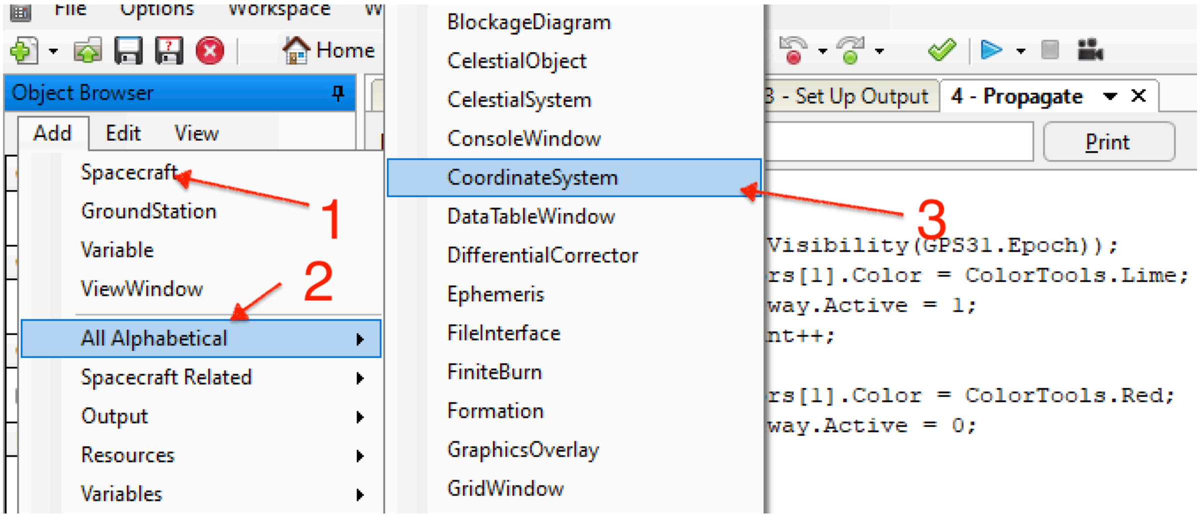


Figure 23 Steps to add a "CoordinateSystem" in Freeflyer

The "RotatingLibrationPoint" objects for Earth-Moon L1 and L2 must then be edited in the Freeflyer Script to specify the following parameters:

- "PrimarySystem" and "SecondarySystem": Primary and secondary systems of the liberation point (in this case Earth is the primary and the Moon is the secondary)
- "LibrationPoint": L1 and L2
- "SetReferenceBody": The Moon in this case as Gateway will be orbiting the Moon.
- "RecomputeBarycenter": The Barycentre (which is the centre of mass between two objects orbiting each other) must be set to "= 1" as this means that all the bodies in the system will be used to compute the system barycentre instead of setting it to "= 0" which would mean that the primary system (Earth in this case) would be used as the barycentre.

The figure below shows how the script was implemented inside the "Set Up Objects" window:

```
// Set up libration points
EarthMoonL1.PrimarySystem.AddBody(Earth);
EarthMoonL1.SecondarySystem.AddBody(Moon);
EarthMoonL1.LibrationPoint = "L1";
EarthMoonL1.SetReferenceBody(Moon);
EarthMoonL1.RecomputeBarycenter = 1;

EarthMoonL2.PrimarySystem.AddBody(Earth);
EarthMoonL2.SecondarySystem.AddBody(Moon);
EarthMoonL2.LibrationPoint = "L2";
EarthMoonL2.SetReferenceBody(Moon);
EarthMoonL2.RecomputeBarycenter = 1;
```

Figure 24 Freeflyer script parameter specification for the L1 and L2 Earth-Moon points

3.2.6 Vectors

In Freeflyer, vectors are objects that represents a straight line between 2 objects. This could be from a ground station to a spacecraft, a celestial object to a spacecraft or between spacecrafts and are mainly used for visualization purposes.

For this simulation, vectors were created between Gateway and every one of the GPS satellites. A second group of vectors was similarly created between the Shackelton Crater ground station and the 31 GPS satellites.

Firstly, a "Vector" object was created for every existing Gateway-GPS connection and Shackelton-GPS connection as shown in Fig.27 (in different mission simulations). Then within the Freeflyer script, the vector targets (Gateway and GPSx) and type were specified by typing the vector name, then using the ".BuildVector" method and adding the "Object-to-Object" vector which is represented by value "9" (Figure 29).

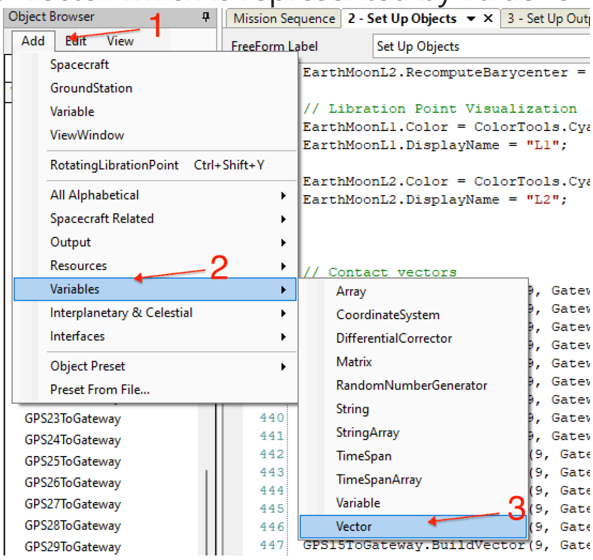


Figure 25 Creating a "Vector" object through Freeflyer object browser

```
// Contact vectors
GPS1ToGateway.BuildVector(9, Gateway, GPS1);
GPS2ToGateway.BuildVector(9, Gateway, GPS2);
GPS3ToGateway.BuildVector(9, Gateway, GPS3);
GPS4ToGateway.BuildVector(9, Gateway, GPS4);
GPS5ToGateway.BuildVector(9, Gateway, GPS5);
GPS6ToGateway.BuildVector(9, Gateway, GPS6);
GPS7ToGateway.BuildVector(9, Gateway, GPS7);
GPS8ToGateway.BuildVector(9, Gateway, GPS8);
GPS9ToGateway.BuildVector(9, Gateway, GPS9);
GPS10ToGateway.BuildVector(9, Gateway, GPS10);
GPS11ToGateway.BuildVector(9, Gateway, GPS11);
GPS12ToGateway.BuildVector(9, Gateway, GPS12);
```

Figure 27 Freeflyer script to specify the parameters of each vector

Value	Label
0	Cartesian
1	Spherical
2	Cross Product
3	Normalized Vector
4	Body Axis
5	Body Angular Momentum
6	Body Position
7	Body Velocity Vector
8	Existing X-axis
9	Object-to-Object
10	Anti-Vector
11	Star
12	Object-to-Star

Figure 26 Vector types in Freeflyer [33]

3.2.7 Visibility Segments

Visibility Segments allow the program to determine if there is line of sight between Freeflyer objects (such as spacecraft and ground stations) taking into account occulting bodies, celestial object shape models and refraction [34].

In this mission, a Visibility Segment was created between each GPS and Gateway/Shackleton Lunar station, adding the Earth as an occulting body by using the ".AddOccultingBody" method (Figure 30). Within each GPS satellite, two Visibility Segments were created: one for the main lobe and one for the side lobe, with "Segment1A" referring to main lobe of GPS1, "Segment1B" referring to side lobe of GPS1, "Segment2A" referring to side lobe of GPS2 and so forth.

```
//Visibility segment of GPS1 Main Lobe
VisibilitySegment Segment1A;

Segment1A.SetObserver(GPS1.Sensors[0]);
Segment1A.SetTarget(Gateway);
Segment1A.AddOccultingBody(Earth);

//Visibility segment of GPS1 Side Lobe
VisibilitySegment Segment1B;

Segment1B.SetObserver(GPS1.Sensors[1]);
Segment1B.SetTarget(Gateway);
Segment1B.AddOccultingBody(Earth);

```

Figure 28 Visibility Segment creation for GPS-Gateway

3.2.8 Variables

In the propagation section of the method, the Visibilty Segments will be used to count how many satellites main/side lobes are visible at each time interval. To keep track of this count, the following variables are created by adding the "Variable" object from the Object browser:

- MainLobeCount: How many GPS **main lobe** signals are detected from the Gateway/Shackleton Lunar Station at every time interval
- SideLobeCount: How many GPS **side lobe** signals are detected from the Gateway/Shackleton Lunar Station at every time interval
- TotalLobeCount: How many GPS **total lobes** (ie. main and side lobes) signals are detected from Gateway/Shackleton Lunar Station at every time interval

3.3 Set Up Output (Visualizations and Graphs):

3.3.1 Visualizations:

As concluded in earlier sections, we want 3 different views to illustrate the simulation:

- Earth Inertial View
- Moon Inertial View
- Rotating Libration Point View

The Earth and Moon Inertial views was created by adding a ViewWindow object, selecting the Viewpoint option within the ViewWindow object, setting the source as Earth or Moon, selecting the Inertial reference frame and setting the Target and Tail Reference accordingly as shown in Figure 30.

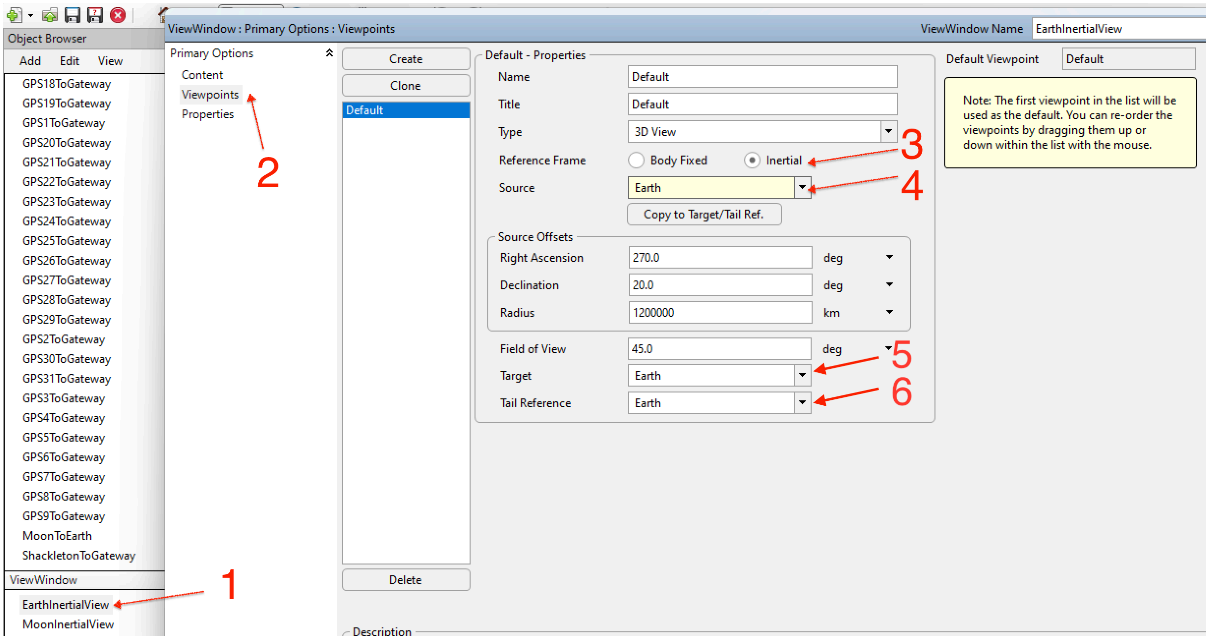


Figure 29 Creation in Freeflyer of Earth Inertial View

The Rotating Libration point view is more complex as the view is fixed to the rotating L1 and L2 points of the Earth-Moon system. The following steps were carried out to build this view:

1. Create a ViewWindow object and set its Viewpoint source as “Fixed” to the “RotatingLibrationPointCS” created in section 3.3.5 as shown in Figure 32.

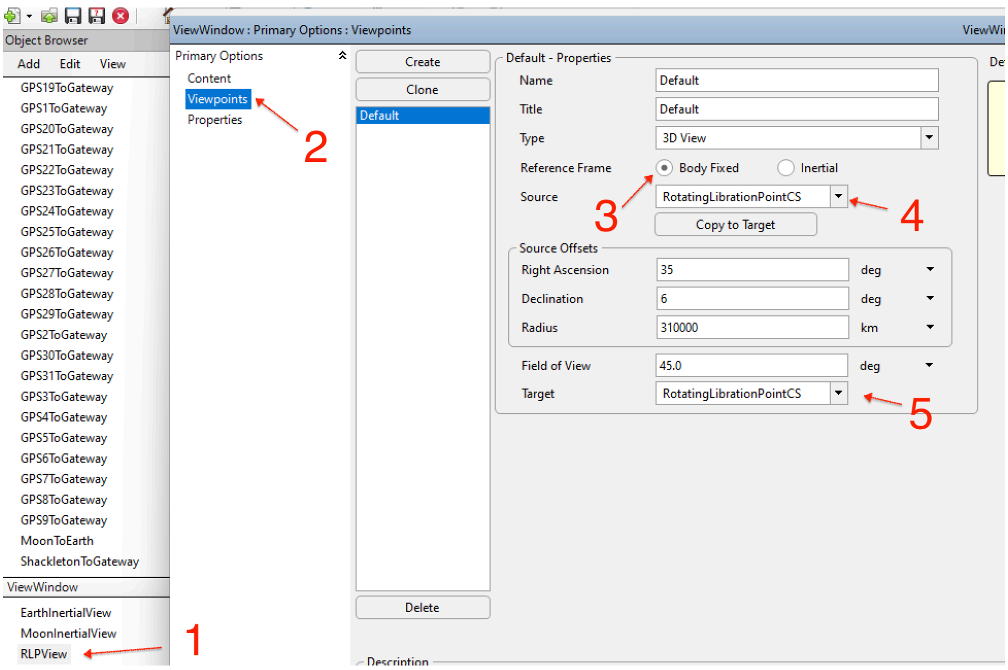


Figure 30 Setting up the RLP View object in Freeflyer

2. Add a FreeForm script element and name it as “Set up Output”.

3. A procedure from one of the freeflyer demo missions was used to update the Lagrange points of the RLPView to follow the fixed frame of the L1 and L2 points as shown in the Figure 33.

```
Define Procedure DefineRLPCS(TimeSpan epoch, RotatingLibrationPoint rlp, CoordinateSystem cs);

    Matrix dcm = rlp.GetDCMAtEpoch(epoch).Transpose();

    Vector x;
    Vector y;
    Vector pos;
    Vector z;

    x.Epoch = epoch;
    x.Element = (dcm * [1; 0; 0]).ToArrayColumnMajor();

    y.Epoch = epoch;
    y.Element = (dcm * [0; 1; 0]).ToArrayColumnMajor();

    z.Epoch = epoch;
    z.Element = (dcm * [0; 0; 1]).ToArrayColumnMajor();

    pos.Epoch = epoch;
    pos.Element = rlp.SecondarySystem.GetPositionAtEpoch( epoch );

    cs.Epoch = epoch;
    cs.BuildCoordinateSystem(1, x, 2, y, pos);

EndProcedure;

// Set up RLP view
RotatingLibrationPointCS.CentralBody = Moon.DisplayName;
RLPView.CurrentViewpoint.ThreeDView.TailReference = RotatingLibrationPointCS.ObjectId;
Call DefineRLPCS(Gateway.Epoch, EarthMoonL2, RotatingLibrationPointCS);
```

Figure 31 RLP View update procedure

This procedure establishes a coordinate system based on the location and orientation of the Earth-Moon L2 Lagrange point. The coordinate system is constructed using the derived vectors as axes after calculating the transformation matrix required to convert the positions of the system's objects. In order to understand the spacecraft's behaviour close to the L2 point, it is helpful to be able to represent the spacecraft's motion in this unique coordinate system.

3.3.2 Graphs:

Graph that shows signal availability from main and side lobes at each given time step were created by adding “PlotWindow” objects and using the variables created in section 3.2.8. As shown in Figure 34, firstly, the X-axis was set as the elapsed time for the Gateway and the Y axis would show the number of main, side and total lobe count at each given time.

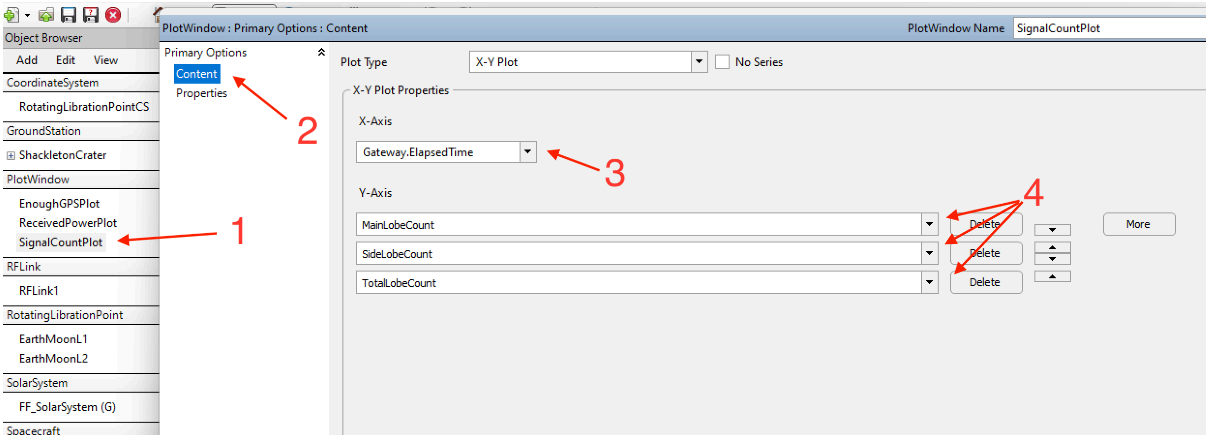


Figure 32 Creating a "PlotWindow" for the lobe counts

3.4 Propagate:

Now that the static "scene" has been created, every spacecraft must be moved to its next position in space, this is known as propagation. Propagation uses mathematical models of forces like gravity, solar radiation pressure etc. in combination with specific integrators (a type of algorithm) to simulate the motion of an object in space. And for each "Step" in time, the program must check if Gateway is within the side lobe or main love FOV. The procedure of the algorithm is shown in the block diagram in Figure 35.

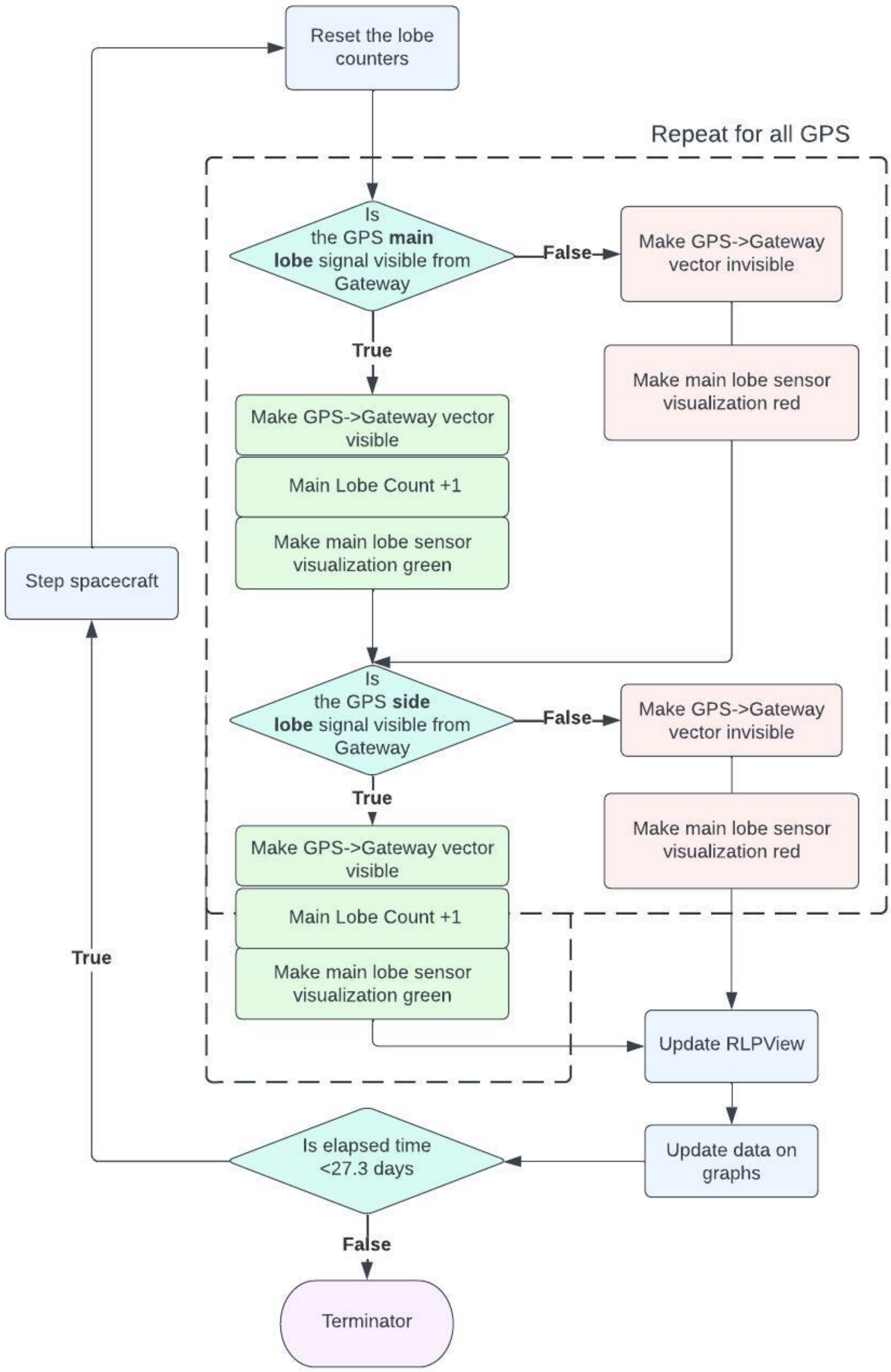


Figure 33 Propagation algorithm flow chart

The most common integrator used in orbital mechanics is the Runge-Kutta, a method to estimate the future positions and velocities of objects in space, given their initial conditions and the forces acting on them. The Runge Kutta 8 or eighth-order Runge Kutta which involves 8 different approximations of the derivatives of the next time step and then determines the average. This is the highest order Runge Kutta integrator and provides the highest precision. The integrator can be edited within each spacecraft Object as shown in the Figure 36, but the default integrator is Runge Kutta.

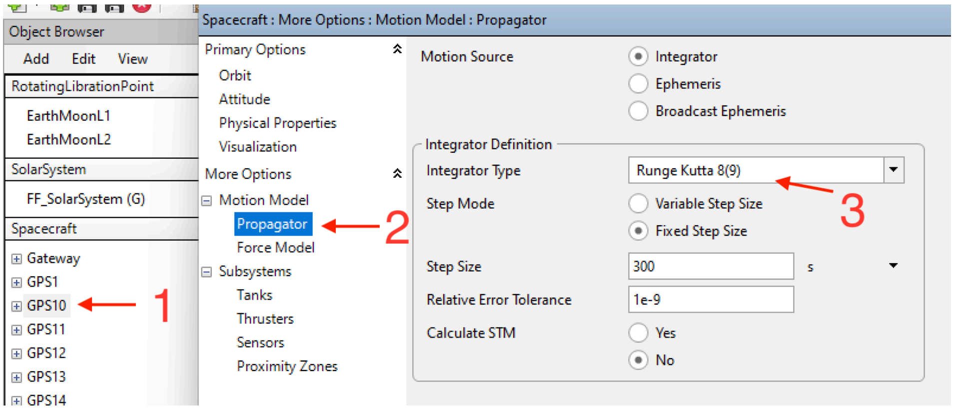


Figure 34 Changing propagator for spacecraft

To accomplish this, a While loop was created with the condition of the elapsed time of the Gateway spacecraft being less than 27.3 days (the duration of Moon orbit of the Earth). So as long as the condition is met, the program will "Step" each GPS and Gateway to its next position in space. This part of the code is shown in Figure 37.

```
While (Gateway.ElapsedTime < TIMESPAN(27.3 days));
    Step Gateway;
    Step GPS1 to (GPS1.Epoch == Gateway.Epoch);
    Step GPS2 to (GPS2.Epoch == Gateway.Epoch);
    Step GPS3 to (GPS3.Epoch == Gateway.Epoch);
    Step GPS4 to (GPS4.Epoch == Gateway.Epoch);
    Step GPS5 to (GPS5.Epoch == Gateway.Epoch);
    Step GPS6 to (GPS6.Epoch == Gateway.Epoch);
    Step GPS7 to (GPS7.Epoch == Gateway.Epoch);
    Step GPS8 to (GPS8.Epoch == Gateway.Epoch);
    Step GPS9 to (GPS9.Epoch == Gateway.Epoch);
    Step GPS10 to (GPS10.Epoch == Gateway.Epoch);
    Step GPS11 to (GPS11.Epoch == Gateway.Epoch);
    Step GPS12 to (GPS12.Epoch == Gateway.Epoch);
```
Figure 35 Freeflyer script to step the spacecraft to the next timestep

For every step in time, the program will restart set the "SideLobeCount", "MainLobeCount" and "TotalLobeCount" to zero as shown in Figure 38.

```
// Reset the counters
MainLobeCount = 0;
SideLobeCount = 0;
TotalLobeCount = 0;
```

Figure 36 Freeflyer script to reset the lobe counts

It will then consider the Visibility segments between Gateway/Shackleton Crater and each GPS satellite main and side lobes. If the main lobe visibility segment for a specific GPS is active, the respective GPS-Gateway vector is activated, the “MainLobeCount” variable adds 1 to its value and the main lobe sensor turns green in the visualisation. If the visibility segment for the main lobe of a GPS is inactive (ie not in the field of view of the main lobe) the respective GPS-Gateway vector is inactive (not present in the visualisation) and the main lobe sensor turns red. The same steps are repeated for the side lobes as shown in Figure 39.

```

//GPS 1 Main lobe
If (Segment1A.Visibility(GPS1.Epoch)); //and Gateway.Contact(ShackletonCrater));
    GPS1.Sensors[0].Color = ColorTools.Lime;
    GPS1ToGateway.Active = 1;
    MainLobeCount++;
    SideLobeCount--;
Else;
    GPS1.Sensors[0].Color = ColorTools.Red;
    GPS1ToGateway.Active = 0;
End;

// GPS 1 Side lobe

If (Segment1B.Visibility(GPS1.Epoch));
    GPS1.Sensors[1].Color = ColorTools.Lime;
    GPS1ToGateway.Active = 1;
    SideLobeCount++;
Else;
    GPS1.Sensors[1].Color = ColorTools.Red;
    GPS1ToGateway.Active = 0;
End;

```

Figure 37 Freeflyer script to determine if Gateway is within FOV of the main/side lobes of GPS

With the following script in Figure 40, The RLP View is then updated so that its fixed to the Lagrange points (ie. the Moon and Earth remain stationary in the view). The rest of the “EarthInertialView” and “MoonInetrtialView” as well as the “SignalPlotCount” are updated with the new data for this time step.

```

    MoonToEarth.Epoch = Gateway.Epoch;
    Call DefineRLPCS(Gateway.Epoch, EarthMoonL2, RotatingLibrationPointCS);

    Update EarthInertialView;
    Update MoonInertialView;
    Update RLPView;
    Update SignalCountPlot;
    Update EnoughGPSPlot;

End;
```

Figure 38 Freeflyer script to update output views and graphs

If the elapsed time is less than 27.3 days, all the Spacecraft will be updated to the next time step, if its greater than 27.3 days, the program will stop.

4. Results and Analysis

4.1 Side and Main lobe counts

The variables “MainLobeCount”, “SideLobeCount” and “TotalLobeCount” were used to determine the number of signals detected from each lobe of the GPS satellites by Gateway and Shackleton receivers. These were plotted in the graphs shown below every time step (20 minutes) for a total period of 27.3 days (one whole Moon orbit of Earth) using the “PlotWindow” object in Freeflyer as shown in section 3.3.

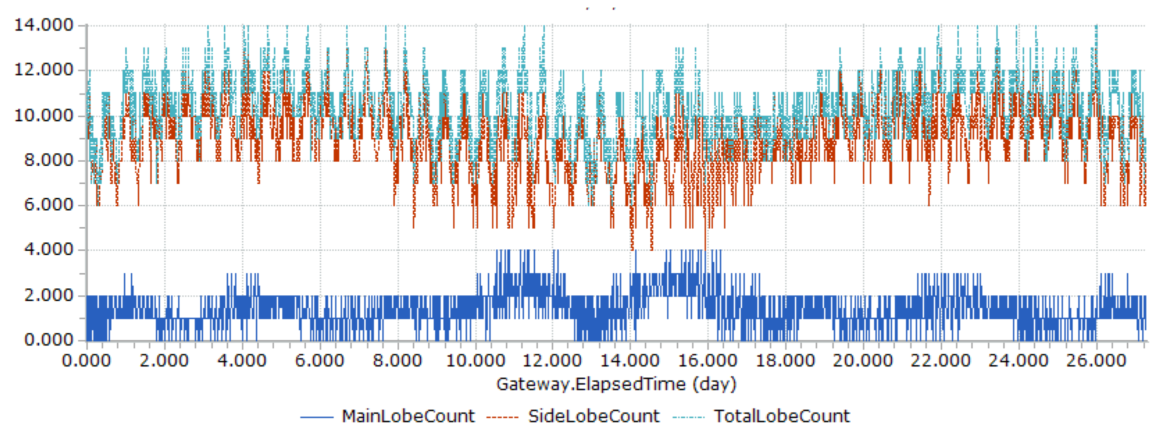


Figure 39 Gateway Main and Side Lobe Count for a whole Moon orbit of Earth

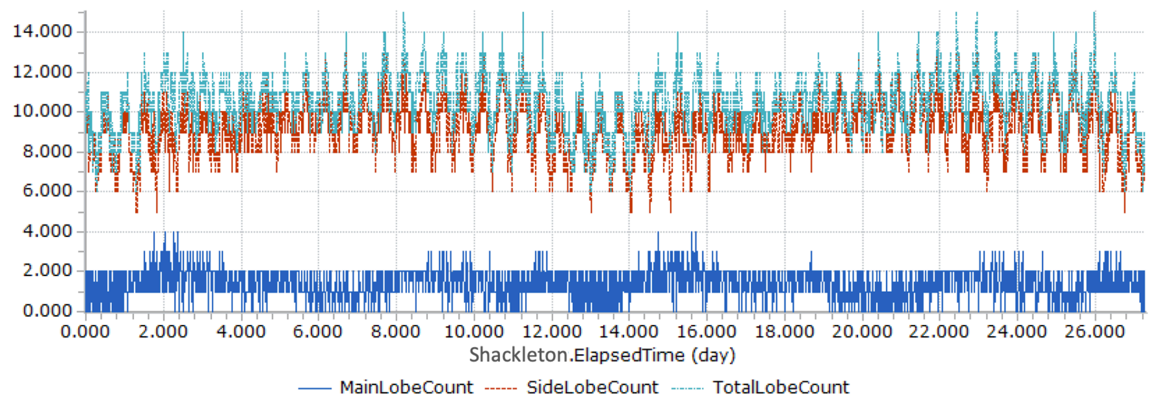


Figure 40 Shackleton Main and Side Lobe Count for a whole Moon orbit of Earth

The graphs in Figures 41 and 42 show the number of main lobe (dark blue), side lobe (red) and total lobe (cyan) signals are visible from the Lunar Gateway (first graph) and the Lunar Surface Station (second graph) for the duration of a complete orbit of the moon around the Earth (27.3 days).

It can be observed that the **ML signals ranged between 0 and 4** at any given time step for both Gateway and the LSS. The **SL signals ranged between 4 and 13 for Gateway and 5 and 13 for the LSS**, showing a slightly higher minimum.
The minimum number of required signals to calculate position is 4 pseudoranges, of which there were 4 ML signals at 24 time steps in the Gateway simulation compared to only at only 7 time steps in the Lunar Station.

The average ML count for the whole Lunar orbit simulation (27.3 days) can be calculated by dividing the sum of all the detected ML by the number of time steps in the simulation as shown in equation.

$$Average\ lobe\ count = \frac{Sum\ of\ lobe\ counts}{Number\ of\ time\ steps}$$

The total number of time steps for the whole simulation was 1965 time steps and this was used to calculate the following average lobe counts.

	Average Lobe Count	
	Main Lobe	Side Lobe
Gateway	1.41	8.98
Shackleton	1.35	8.98

Table 1 Average number of ML and SL signals visible in the whole simulation

The average SL count was surprisingly close in Gateway and Shackleton, with 8.98 on average. Within the ML count however, Shackleton had an average of 1.35 compared to 1.41, which is a 4.44% increase from Shackleton to Gateway.

The Side lobes occurrence have a sinusoidal nature to them as shown in Figure 43 below. They follow a variation of up to 5 visible satellites in a period of half an Earth day approximately. This could be of great significance as it means that the changing satellite SL connections occurs at a high rate in the short timeframe, which could result in more inaccurate position measurements.

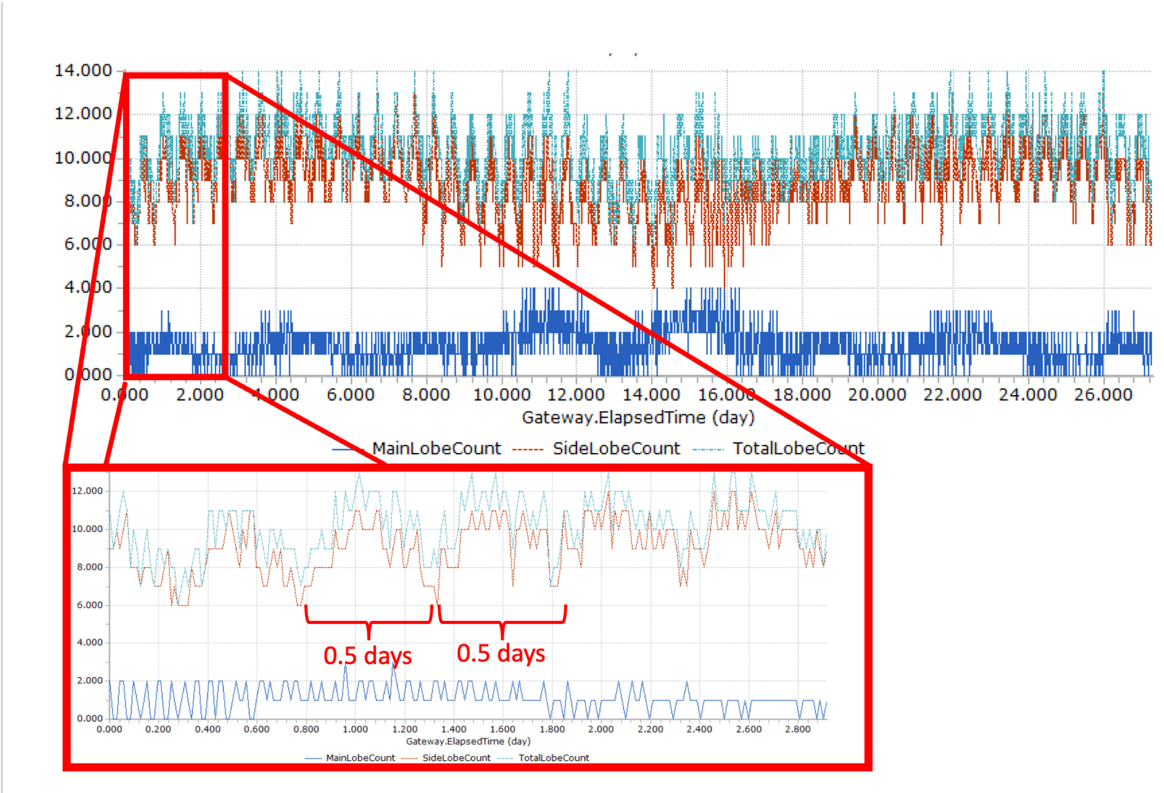


Figure 41 Zoom-in into the Gateway timeframe

The overall lobe counts (which represent the sum of the lobe counts for the duration of the whole simulation), were added and presented in tabular form by Freeflyer as shown in the figures below.

OverallSideLobeCount	OverallMainLobeCount	OverallTotalLobeCount
17652.0000000	2766.0000000	20418.0000000

Table 2 Overall Side, Main and Total lobe count for Gateway simulation

OverallSideLobeCount	OverallMainLobeCount	OverallTotalLobeCount
17924.0000000	2660.0000000	20584.0000000

Table 3 Overall Side, Main and Total lobe count for Shackleton simulation

We can observe that the **overall total lobe count and side lobe count are slightly higher for Shackleton** (0.8% and 1.54% increase from Shackleton to Gateway respectively). However, the **overall main lobe count for Gateway is 3.98% higher than Shackleton**.

Another way to analyse the side and main lobe occurrence of each simulation scenario is by comparing the occurrence percentage of each lobe type. This was calculated and presented using Matlab in the chart in Figure 44.

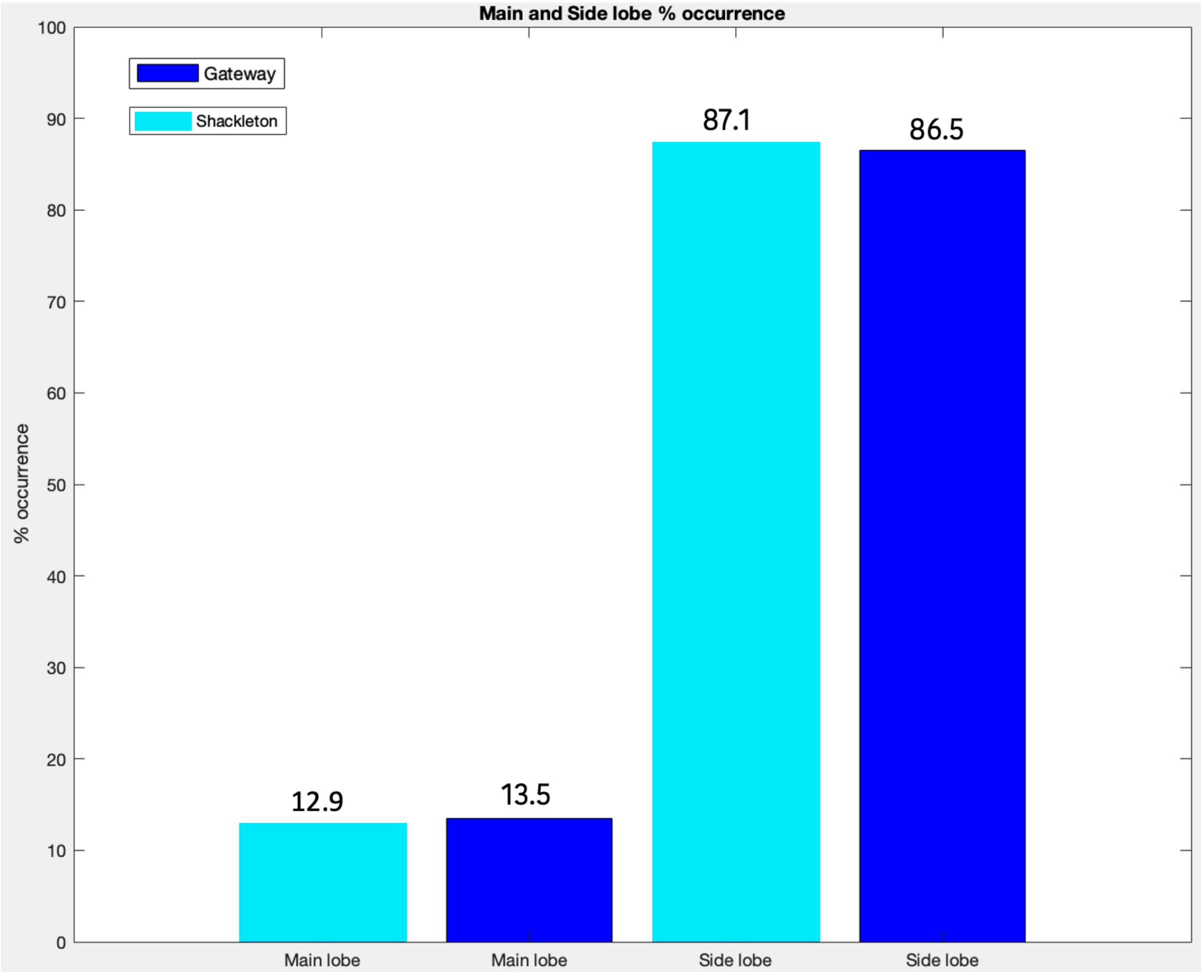


Figure 42 Percentage occurrence of ML and SL for Gateway and Shackleton

This bar chart shows the greater dominance of occurrence in SL signals for both Shackleton and Gateway, but with a 0.6% more dominance in Shackleton. However, in the ML signals, Gateway had 0.6% higher occurrence than Shackleton.

4.2 Visualisations

The main purpose of the visualisations was to illustrate a comprehensive model that would show the working principle of using GNSS signals for Lunar navigation by having responsive GPS sensors and signal vectors that would activate and deactivate if the receiver (Gateway/Shackleton) was within the FOV of the main or side lobe distinctively.

4.2.1 Earth-centred inertial view:

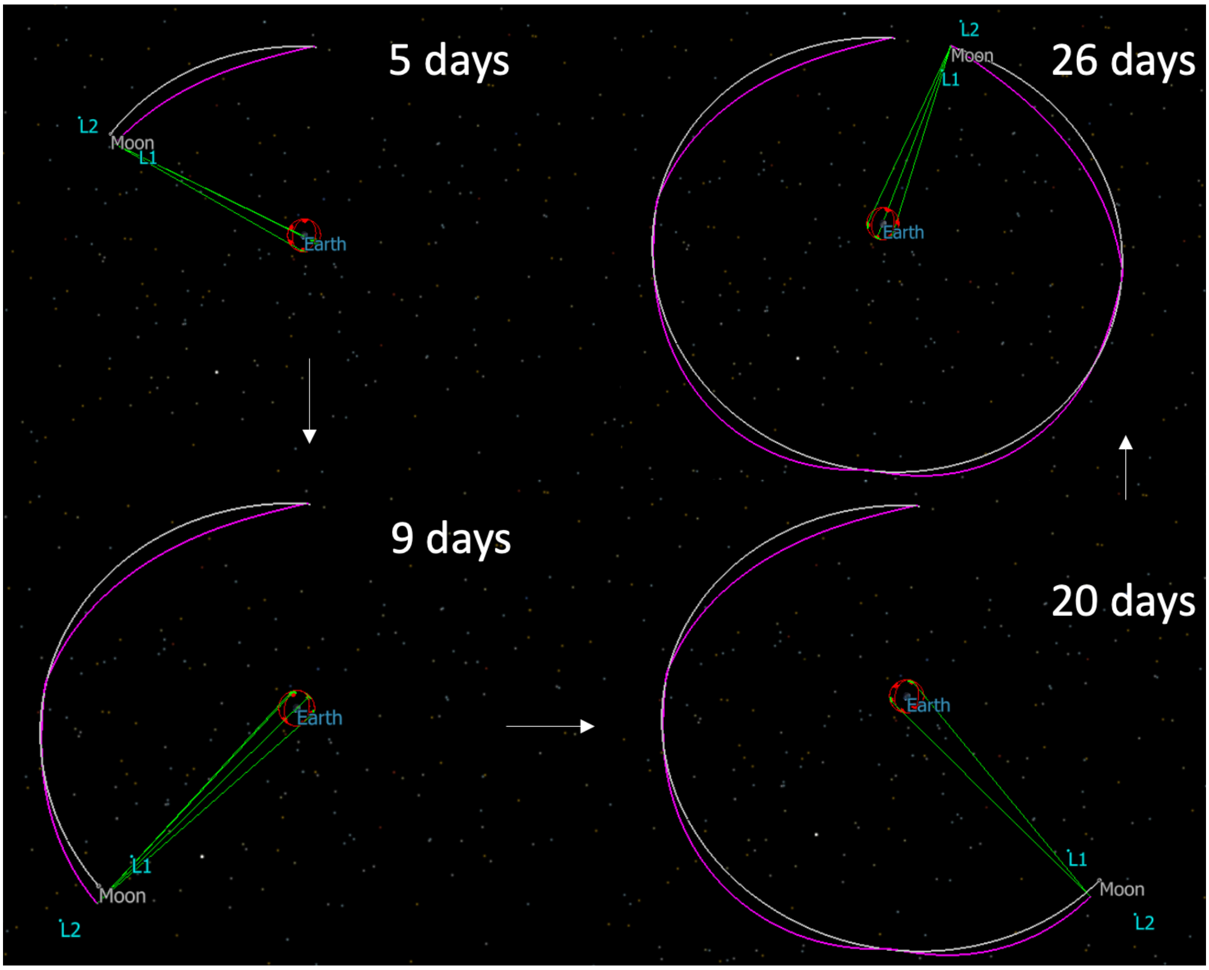


Figure 43 Earth-centred inertial view timelapse

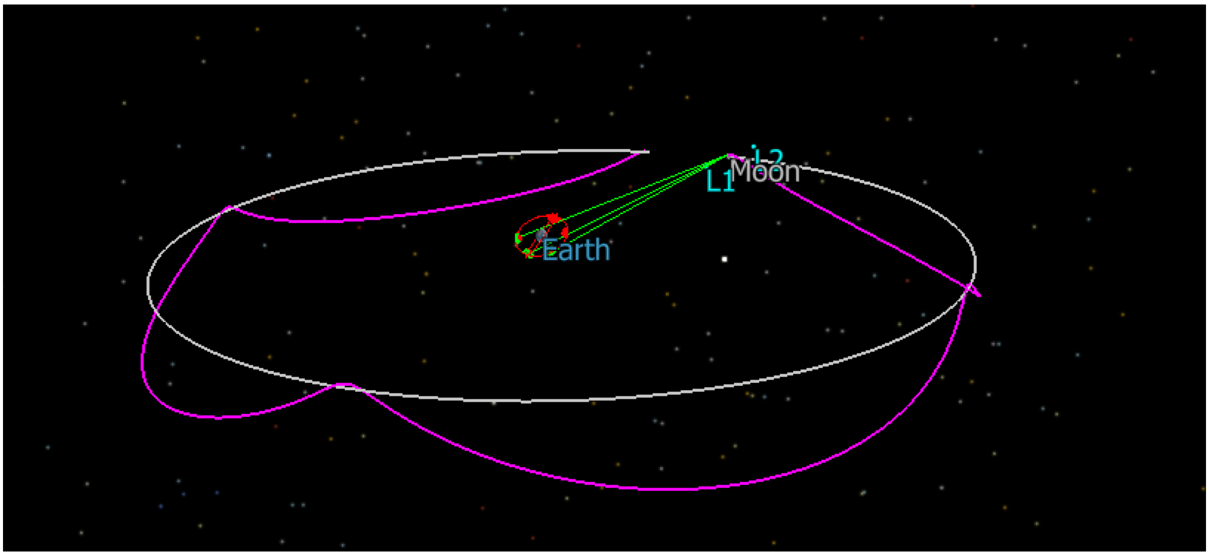


Figure 44 Earth centred inertial view at a lower angle

In the Earth centred inertial view shown in the Figures 43 and 44 above, we can observe the evolution of the Moon´s orbit (white) as well as the Gateway orbit (pink), relative to Earth. The two RLP´s L1 and L2 can be seen in the simulation and, as observed in figure 29, throughout the progression of days, they follow the expected path by adjusting to the correct position between the Earth and the Moon. The NRHO orbit followed by the Gateway station can be seen in Figure 44 to be highly elliptical, only crossing the moon once every 6.75 days (check the actual value).

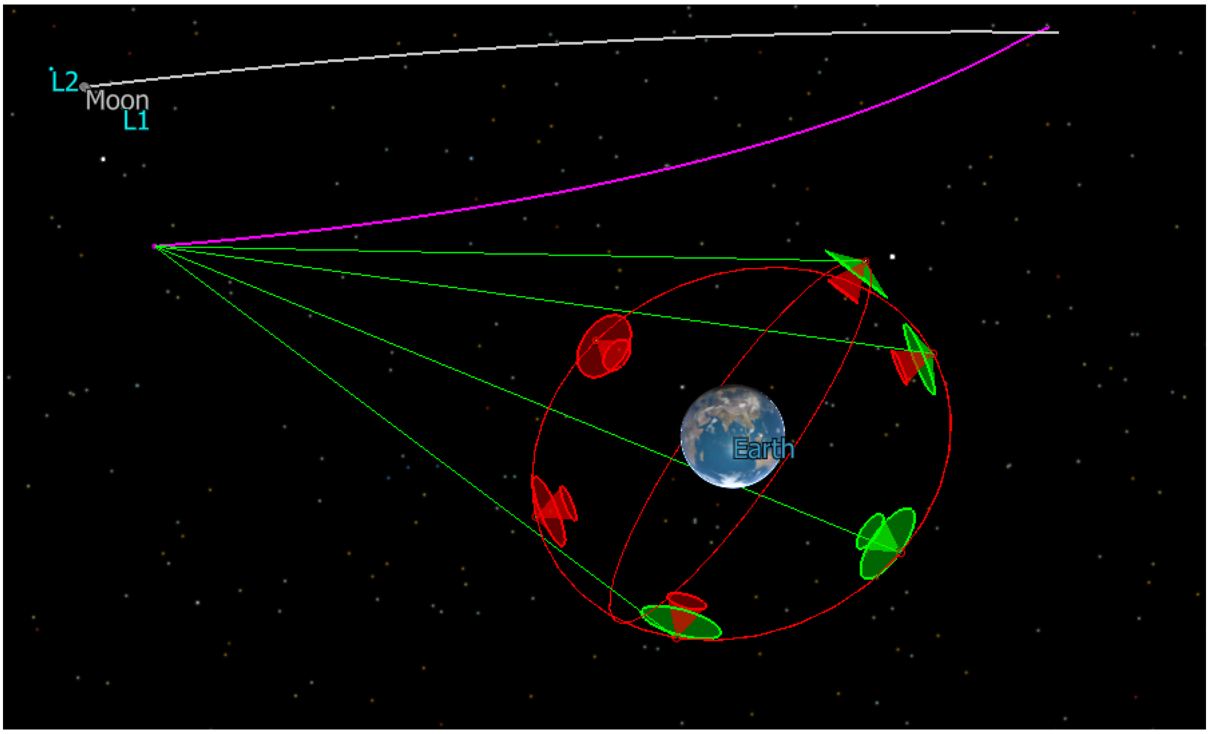


Figure 45 Earth centred inertial view from Earth

Figure 45 shows 4 satellites that were added as a reference to illustrate how the receiving GPS signals from Gateway would operate, showing a green vector from the GPS satellite to Gateway station when Gateway is within the FOV as well as turning the sensor lobe green, proving that the simulation works as desired. Each satellite has 2 cones, which represent the main lobe and side lobe of the antenna radiation pattern.

4.2.2 Moon inertial View:

This Moon Inertial View (Figure 46) was shown to illustrate how the NRHO changes with respect to the fixed frame on the moon. This is also useful because as most of the planned landing sites will be in the lunar south pole, little axial rotation from the Moon is experienced, making it a more realistic view point. This is a snapshot of the simulation, but when animated, the Lagrange points L1 and L2 can be seen orbiting the Moon, being the reason the orbit looks like this.

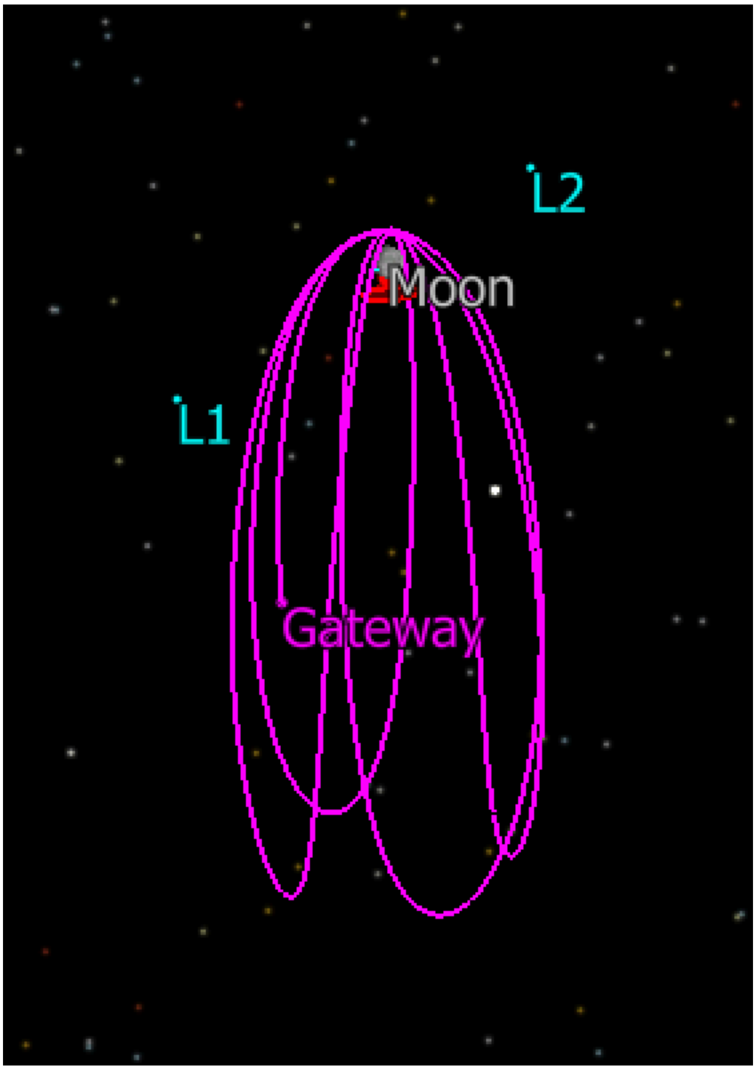


Figure 46 Moon inertial view of Gateway Orbit with Lagrange points

4.2.3 RLP View:

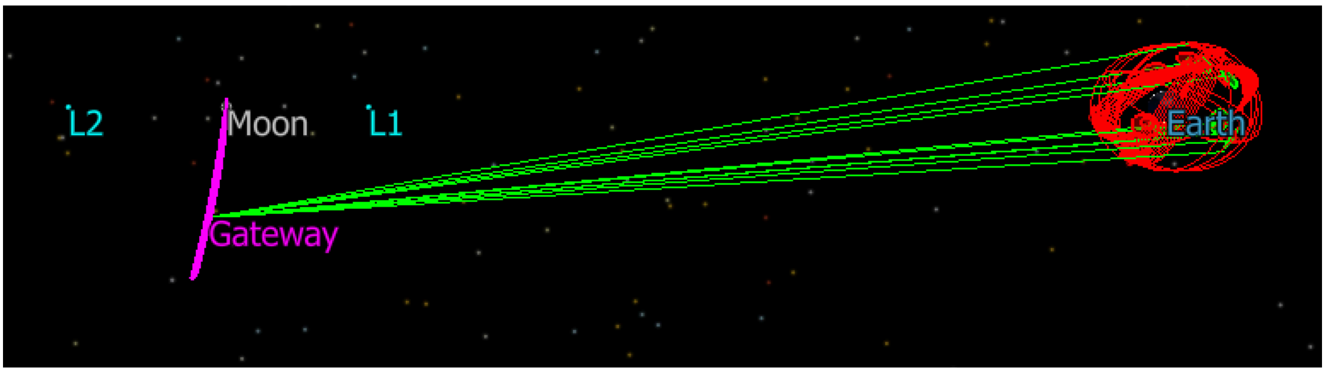


Figure 47 Rotating libration point view perpendicular to Moon-Earth vector for Gateway

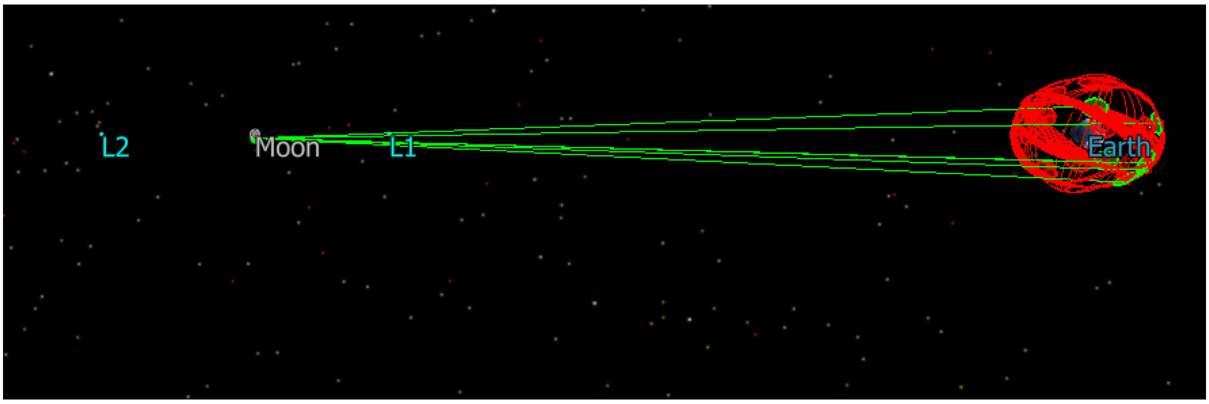


Figure 48 Rotating libration point view perpendicular to Moon-Earth vector for Shackleton

The Rotating Libration point view shown in Figure 48 shows the Moon (grey), the Gateway orbit (pink), the 2 Lagrange points (L1 and L2), the GPS orbits (red), the GPS main and side lobe FOVs (red or green) and the GPS to Gateway vectors (green). In this Field of view, the Libration points L1 and L2, the Moon and the Earth all remain in a static position relative to each other. Therefore, the only moving elements are Gateway and the GPS spacecraft. Figure 48 shows the same view, but for the simulation using a receiver at Shackleton instead, showing the GPS vector signals (green) going to the Moon´s South pole.

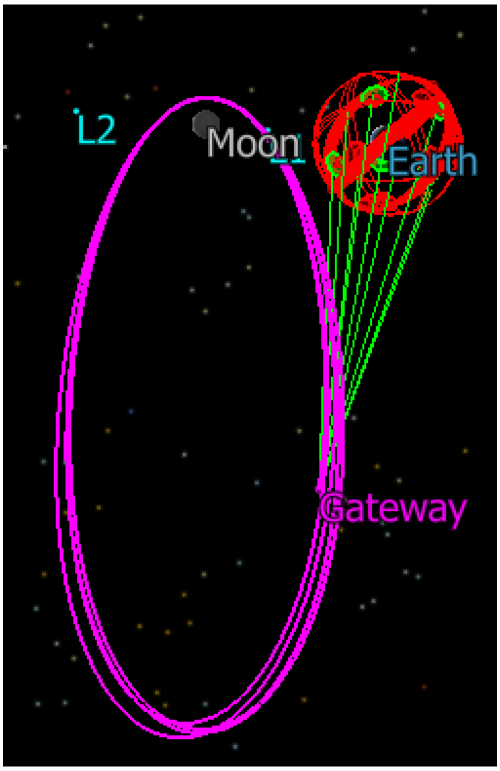


Figure 49 RLP view from the Moon (for Gateway)

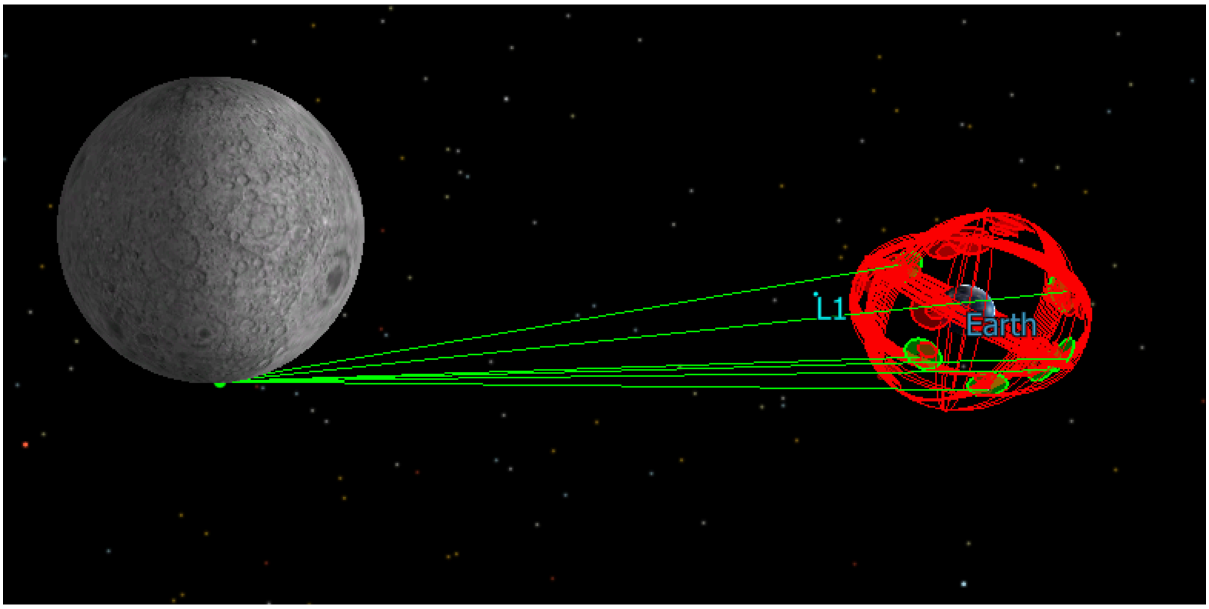


Figure 50 RLP view closeup of the Moon (for Shackleton)

Figures 49 and 50 show a closeup view from the receivers end (Gateway and Shackleton). Here, the orbit of Gateway is clearly shown to have a continuous line of sight with Earth, without the Moon blocking it at any given point.

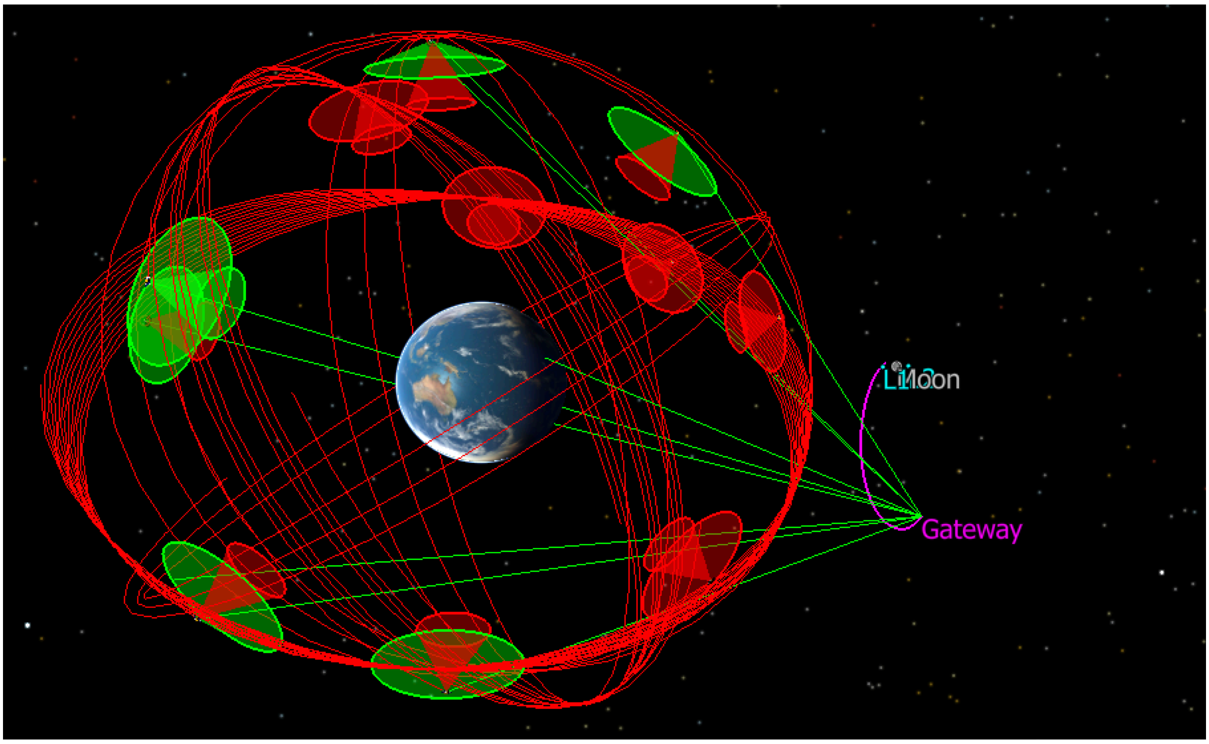


Figure 51 RLP view closeup on Earth (for Gateway)

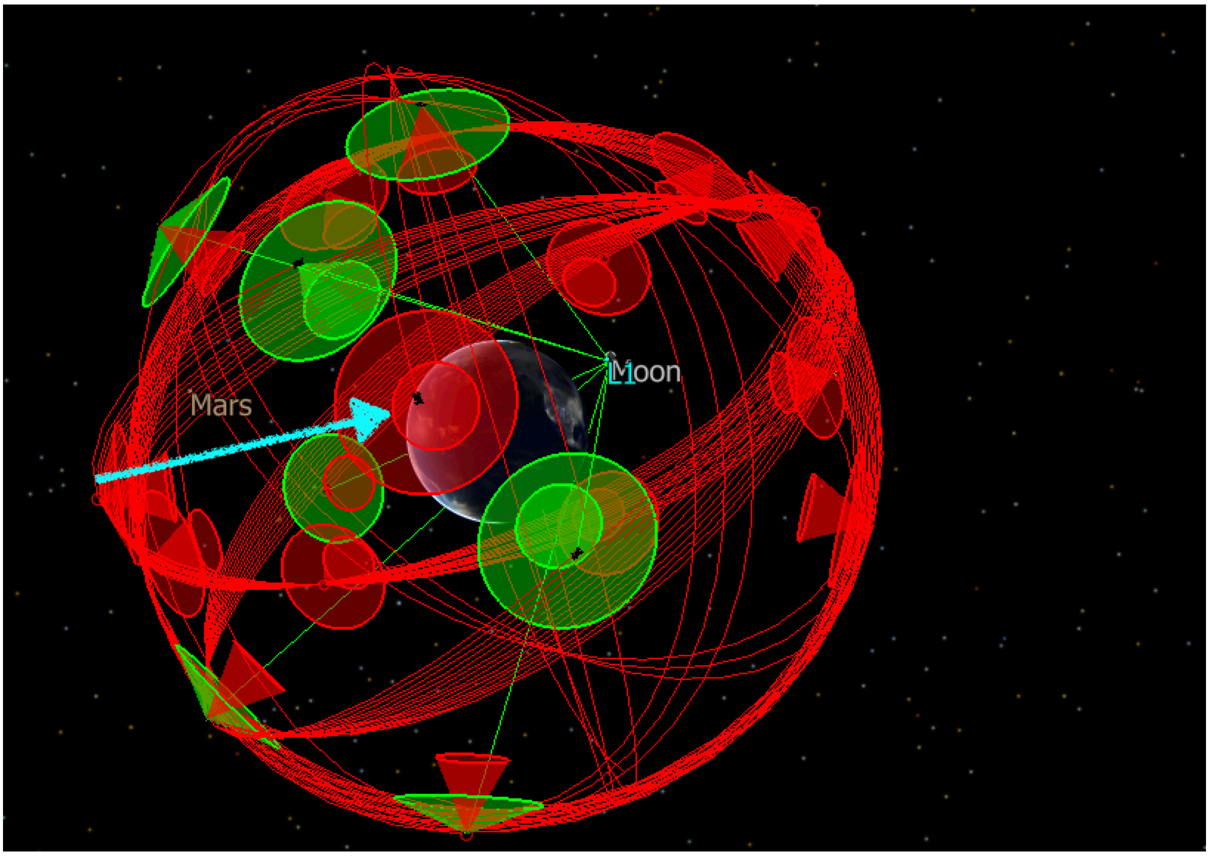


Figure 52 RLP view closeup of Earth (for Shackleton)

Figures 51 and 52 show closeups of the Earth and GPS constellations for both the Gateway and Shackleton simulations. Here we can observe the correct behaviour of the sensors (representing the main and side lobes of the satellites) as when the receiver (Gateway or Shackleton) is within the FOV of the sensor, the main or side lobe model will turn green and the vector will appear. Figure 52 shows an example of a captured snapshot in which even though Shackleton is within the FOV of the GPS satellite marked with the blue arrow, the Earth is obstructing the signal so the vector and lobes don´t activate, proving that the ".AddOccultingBody(Earth)" added method to the visibility segments is working correctly.

5 Discussion

5.1 Outcomes of your project in relation to the literature and past work

Pietro et al [20] was the only investigation which simulated the Gateway scenario like the one performed in this project. However, they used the signals from Galileo and GPS instead of only GPS (as in this report), so a question that could be investigated is if the % occurrence of main and side lobes from Earth´s signals would be different if it were GPS or GPS + Galileo. When we compare the side and main lobe occurrence obtained in our simulation for Gateway (Figure 54) and compare it side to side with what Pietro et al [20] (Figure 53) obtained in their simulation we can see that the results are extremely similar, with percentage of main lobe occurrence around 15% and 85% for sidelobes. This is significant as the main lobe signals have more power (up to 23dbW) [20] than the side lobes, resulting in a more reliable signal (and therefore a pseudorage) which is less affected by free space loss, leading to more accurate positioning.

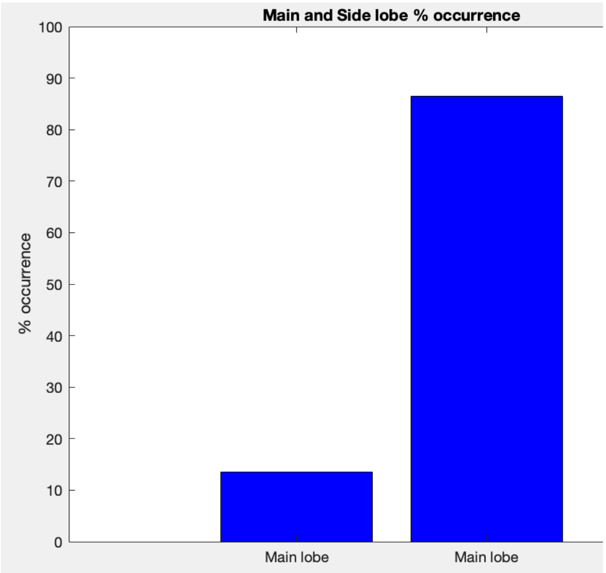


Figure 54 Main and side lobe ocuurence (for Gateway)

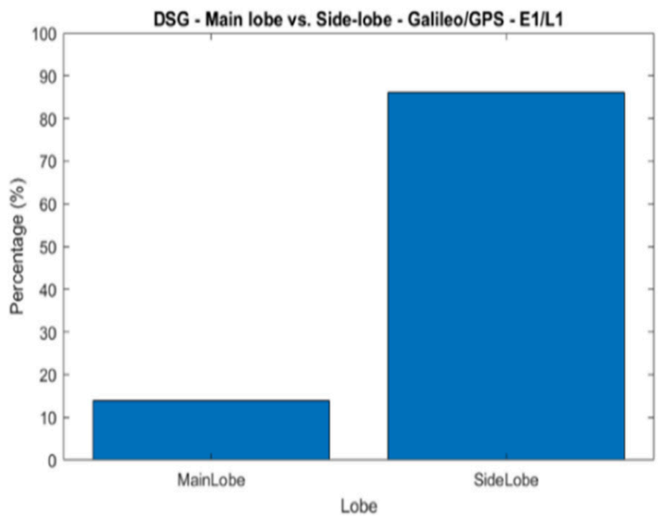


Figure 53 Main and side lobe ocurrence for Gateway (Pietro et al)[20]

But a more interesting measurement to investigate is not only the percentage occurrence but the total occurrence progression throughout the lunar orbit, in other words, how much of a difference does using Gaileo signals on top of GPS affect the amount of satellite signals available at any given time. This is shown in the comparison below.

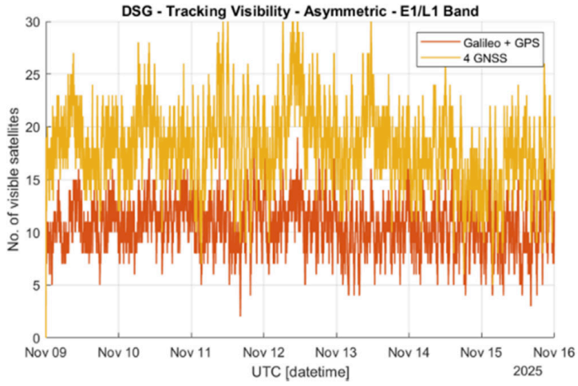


Figure 55 Total number of visible lobes from gateway [20]

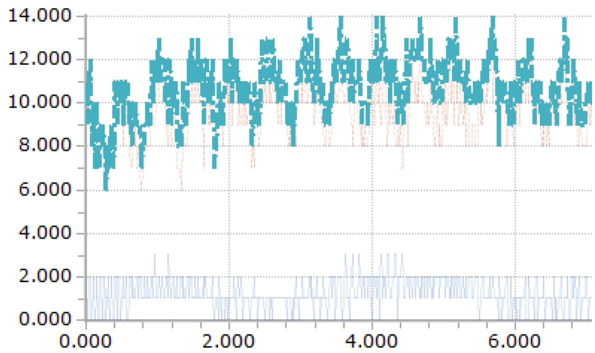


Figure 56 Total number of visible lobes from gateway

Here we can observe that in Pietro et al´s simulation, for the Galileo + GPS (dark orange) the number of satellites visible for the specific time period of 7 days shown in the graph ranges between 4 and 15, but having 9 time intervals that less than 5 satellites are visible

In our simulation (right), however, the total available signals (main and side lobe) ranged from 6 to 14 for a time period of 7 days with only the GPS constellation. How is this possible? Well, at first it could be due to the 7 day period selected, but as you can see in the Figure below, for the whole lunar orbit, there isn´t a point in time where there are less than 5 signals detected from the GPS satellites.

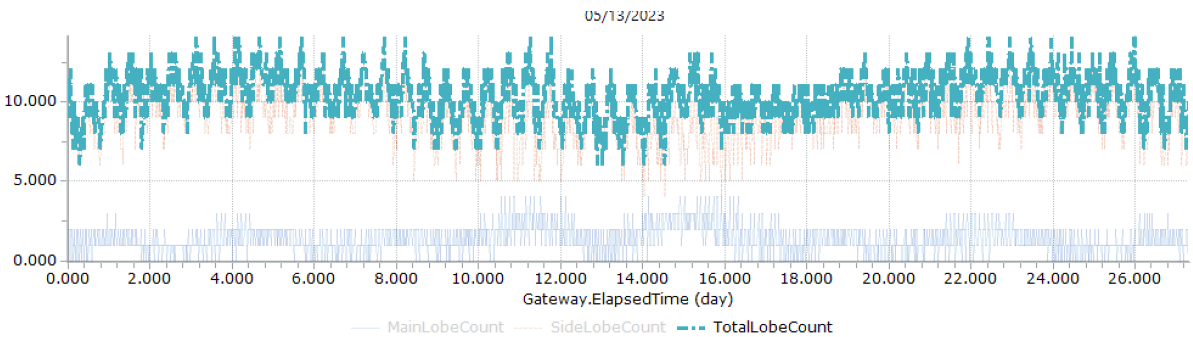


Figure 57 Total number of visible lobes from gateway for a whole lunar orbit of the Earth

If we take a closer look at the constrains used by Pietro et al [20] we can see that only a signal was counted as detected if the signal to noise ratio is "above the tracking threshold of 15 dB-Hz", which would exclude many the side lobe signals being counted in our simulation.

On that note, Pietro et al nor any other article distinguished between side and main lobes in during the time progression of the simulation, something which is important when taking into account the low power of the side lobe signals. As shown in Figure 58 during the days 10 and 16 of the simulation, the side lobe signals visible decreased in both scenarios to around 5 and up to only 4 visible satellites. In Gateway´s case, the main lobes increased accordingly to up to 4 visible, whereas Shackleton remained the same (around 2).

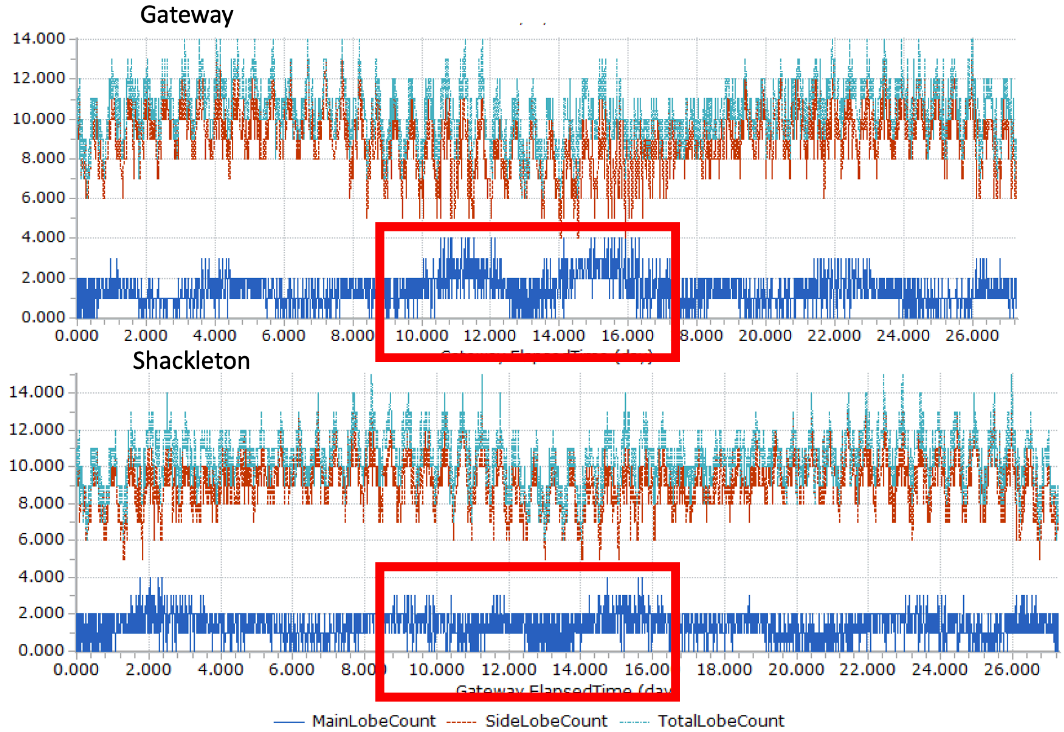


Figure 58 Main and side lobe visibility from Gateway (Top) and Shackleton (Bottom)

In terms of the visualization, this simulation managed to achieve not only 3 different views (giving perspective), but also provide a closeup of the satellites showing the main and side lobe activation as shown in the comparison in Figure 59

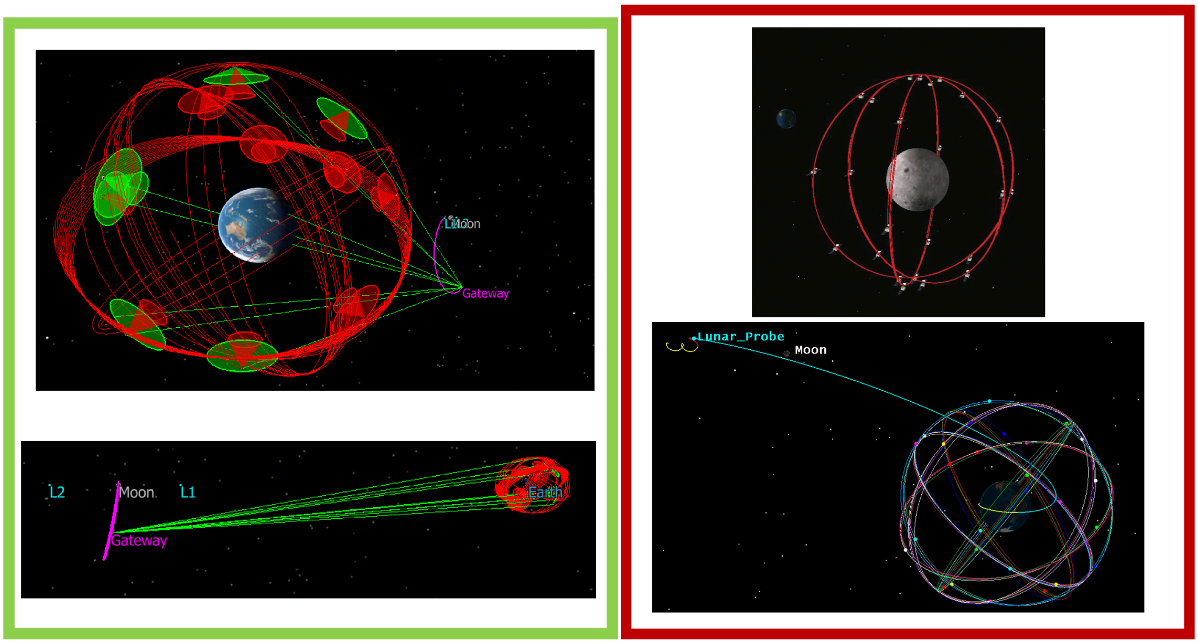


Figure 59 Project visualization (left) vs previous visualisations (right)

Furthermore, showing the Lagrange points (L1 and L2) in the Earth Inertial view allowed to see their orbit during the Moon´s orbital cycle. This served to understand why Gateway´s orbit in the Moon Inertial View was different than in the RLP view.

5.2 Uncertainty and incompleteness of any information used and how they affect the project outcomes

The problem with using the exact radiation pattern in the simulation is that the signal cannot be separated into the main lobe and side lobe as its represented as a single continuous model of varying power. This can affect the visualization as no distinction can be seen between the lobes (it´s all one continuous volume) as well as in the signal detection, which would have to observe the analogue signal power instead of a binary representation of lobe visibility. Nevertheless, the simplification of our model doesn´t take into account possible lower power levels between side lobes as shown in Figure 60, which will lead to inaccurate results.

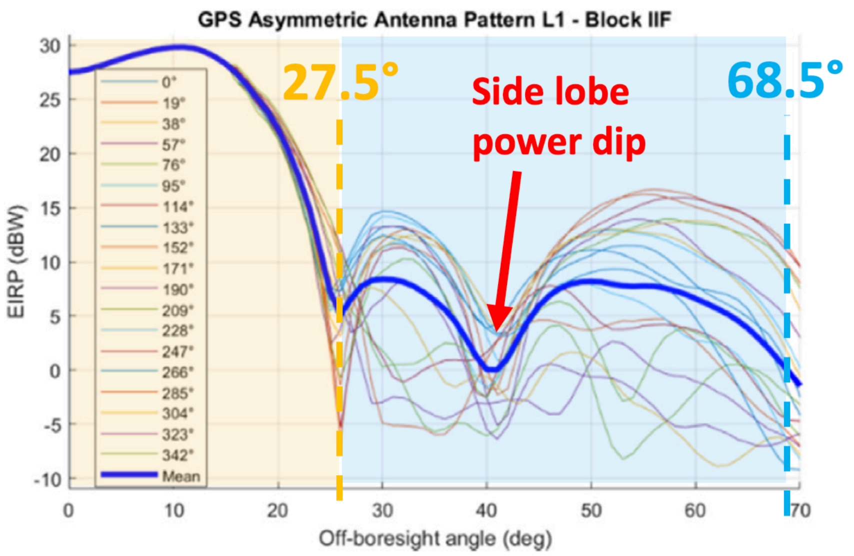


Figure 60 GPS antenna radiation pattern

Another assumption is that all the GPS satellites had the same radiation pattern. This is not the case as there are different blocks (talk about different blocks and how they have different radiation patterns (including aperture angles) but the IIF pattern was used as this is the most numerous one, the newest one and the one that is going to be used for replacement of the old satellites, so taking into account that the first section of the Gateway (the propulsion and habitation module) are planned to launch no earlier than late 2024, this should be fine. On top of that, we would have to know exactly which satellite was of which block (and its specific radiation pattern) or if not it would result in a more inaccurate model (and this information is not available).

Another assumption is that the GPS antennas are always facing to earth's centre (with no changes on the attitude), when it can have an effect as shown in Vincenzo et al´s paper "GPS based" [34].

5.3 Future work

As the signal being received by Gateway or Shackleton from the GPS is sometimes obstructed by the Earth, one might ask what the effect of Earth´s atmosphere could have on the receiving signal. The research presented on this topic has mainly assumed only freespace loss of the signal, that is to say a continuous decrement in signal power with distance travelled through space. However, taking into account the refraction and perturbations that the ionosphere and mesopsphere in the signal could be of interest as the effect would take place at the near the emitter of the signal, and as the signal then has to travel 384,000km to the moon, this error could have a great impact on the accuracy of the pseudorange calculated.

Some suggestions to implementing this could include determining the perpendicular distance between the GPS to Gateway signal and the surface of the Earth. Which could then be used to calculate through which part of the atmosphere the signal travels as well as for how far.

The antenna power wasn´t taken into account in this paper´s simulation. The reason being to understand the occurrence of the main and side lobes as a binary representation as mentioned previously. Nevertheless, if errors such as free-space loss and the effect of the atmosphere want to be measured, an antenna gain pattern would have to be uploaded as an ephemeris file in the "sensor" section of each GPS object.

5.4 Review of project

Limitations in the initial software selected (GMAT) made the project reach a roadblock that couldn´t be changed. This was expected to happen, but instead of quickly changing to some other software, as there weren´t any readily options available, and this roadblock occurred well into the simulations development, I persisted for several weeks trying to find workarounds which ultimately led nowhere.

Then, whilst asking for help in an online forum about aerospace engineering to see if anyone had tried to implement the cone visualisations, the responsible of academic licenses from ai solutions contacted me suggesting Freeflyer was capable of doing what I was trying to do. However, after seeing that licenses were required, I suggested to the director of academic licenses that even though my university hadn´t licensed this software, to allow me to use a license freely in order to prove to the academics in the university that this software could provide useful tools that weren´t previously available in GMAT.

Therefore, to avoid these types of setbacks, making sure that every required functionality of the simulation is confirmed possible to execute in the software before starting to develop the program is of crucial importance. However, as this might not always be possible, developing the simulation in 2 different softwares simultaneously or at least having the basic initial building blocks developed could prevent this type of behaviour occurring in the future as in case of a roadblock, the student could swiftly transition into another software without having to start the simulation from scratch.

In terms of the Freeflyer algorithm, a lot of time could be saved if for loops would have been implemented in the GPS creation. However, this would have been done by creating an list or formation object of the GPS constellations, which didn´t have a method to add the sensors, a crucial part of the simulation.

Also, towards the end of the project the Galileo satellite constellation (Europe´s GPS equivalent) was added to see if any further conclusions could be extracted from this data. However, the simulation crashed and wasn´t capable of executing properly. It was suspected that this could be from having too many objects simulated simultaneously, so the propagator for every spacecraft was changed from Rungekutta 8 to Rungekutta 4 crashing could be avoided by doing this. This would make the model less precise but reduce computational power, but the simulation still crashed. Nevertheless higher order RungeKutta integrators should be used in long duration simulations as small inaccuracies in the propagation can have a snowball effect and accumulate over time.

5.5 Conclusion

This project has investigated the signal detection from main and side lobes of GPS satellites by Gateway and Shackleton receivers. Throughout the 27.3-day lunar orbit simulation, the number of main lobe signals fluctuated between 0 and 4 at any given time for both receivers, whereas side lobe signals ranged between 4 and 13 for Gateway and 5 and 13 for Shackleton. The minimum number of signals required to calculate position (4 pseudoranges) were found more frequently in the Gateway simulation.

When calculating the average lobe counts, Shackleton demonstrated a slight increase in the side lobe count, but Gateway displayed a higher count in the main lobe. Side lobe occurrences showed a sinusoidal pattern, fluctuating with a period of approximately 12 hours, which implies a potential impact on the accuracy of position measurements due to rapid changes in satellite side lobe connections.

The overall total lobe count was slightly higher for Shackleton, with the main lobe count being higher for Gateway. When comparing the occurrence percentage of each lobe type, side lobes were more dominant in both receivers, with a slightly higher dominance in Shackleton. However, Gateway had a slightly higher occurrence in main lobes.

The project's visualisations successfully illustrated the principles of using GNSS signals for Lunar navigation with responsive GPS sensors and signal vectors. The Earth-centered inertial view showed the evolution of the moon's orbit and the Gateway orbit relative to Earth. A closeup view from the receivers end clearly demonstrated that the Gateway orbit maintains a continuous line of sight with Earth, without the Moon obstructing it.

The Moon Inertial View and the Rotating Libration point view were useful in showing how the Near Rectilinear Halo Orbit (NRHO) changes with respect to the fixed frame on the moon and how different elements move and interact in this dynamic environment. Moreover, the visualisations confirmed the correct behaviour of the sensors, representing the main and side lobes of the satellites, with the receivers within the Field of View (FOV) of the sensor.

Once the Gateway spacecraft is sent to its future NRHO orbit with a GPS receiver, real signal measurements will be taken and compared with the results with this project, which, if similar, could also validate the results obtained for the Shackleton crater.

7 References

[1] Creech, S., Guidi, J. and Elburn, D. (2022). Artemis: An Overview of NASA's Activities to Return Humans to the Moon. *2022 IEEE Aerospace Conference (AERO)*. doi:https://doi.org/10.1109/aero53065.2022.9843277.

[2] Sherwood, B. (2019). Principles for a practical Moon base. *Acta Astronautica*, 160, pp.116–124. doi:https://doi.org/10.1016/j.actaastro.2019.04.018.

[3] Spudis, P.D., Bussey, D.B.J., Baloga, S.M., Cahill, J.T.S., Glaze, L.S., Patterson, G.W., Raney, R.K., Thompson, T.W., Thomson, B.J. and Ustinov, E.A. (2013). Evidence for water ice on the Moon: Results for anomalous polar craters from the LRO Mini-RF imaging radar. *Journal of Geophysical Research: Planets*, [online] 118(10), pp.2016–2029. doi:https://doi.org/10.1002/jgre.20156.

[4] Brukardt, R. (2022). *How will the space economy change the world? | McKinsey*. [online] www.mckinsey.com. Available at: https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/how-will-the-space-economy-change-the-world.

[5] ispace-inc.com. (n.d.). *ispace*. [online] Available at: https://ispace-inc.com/m1.

[6] Martin, D. (2023). *La sonda japonesa HAKUTO-R no logra alunizar con éxito*. [online] Eureka. Available at: https://danielmarin.naukas.com/2023/04/26/la-sonda-japonesa-hakuto-r-no-logra-alunizar-con-exito/ .

[7] Wanhong, H., Zan, L., Guangliang, D., Hong, W. and Haitao, L. (2016). *Software carrier tracking of Chinese Chang'E-3 lunar explorer in the power descent phase*. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/abstract/document/7802828 .

[8] McIntosh, D., Baker, J. and Matus, J. (2020). The NASA Cubesat Missions Flying on Artemis-1. *Small Satellite Conference*. [online] Available at: https://digitalcommons.usu.edu/smallsat/2020/all2020/44/.

[9] Tsvetan Dachev, B.T. Tomov, Yu.N. Matviichuk, Dimitrov, P., Vadawale, S.V., Goswami, J.N., G. de Angelis and V. Girish (2011). An overview of RADOM results for earth and moon radiation environment on Chandrayaan-1 satellite. *Advances in Space Research*, 48(5), pp.779–791. doi:https://doi.org/10.1016/j.asr.2011.05.009.

[10] Guest, A.N. (2013). Telemetry, Tracking, and Command (TT&C). *Handbook of Satellite Applications*, [online] pp.1067–1078. doi:https://doi.org/10.1007/978-1-4419-7671-0_69.

[11] Deep Space Network Now. (2019). *Deep Space Network Now*. [online] Available at: https://eyes.nasa.gov/dsn/dsn.html

[12] Squadrito, P., Livreri, P., Di Donato, L., Squadrito, C. and Sorbello, G. (2019). A Telemetry, Tracking, and Command Antennas System for Small-Satellite Applications. *Electronics*, 8(6), p.689. doi:https://doi.org/10.3390/electronics8060689.

[13] Donaldson, J.E., Parker, J.J.K., Moreau, M.C., Highsmith, D.E. and Martzen, P.D. (2020). Characterization of on-orbit GPS transmit antenna patterns for space users. *NAVIGATION*, 67(2), pp.411–438. doi:https://doi.org/10.1002/navi.361.

[14] insights.globalspec.com. (n.d.). *How Does GPS Work? | Engineering360*. [online] Available at: https://insights.globalspec.com/article/10315/how-does-gps-work.

[15] Wu, S., Bertiger, W., Kuang, D., Lichten, S., Nandi, S., Romans, L. and Srinivasan, J. (1997). *MicroGPS for Low-Cost Orbit Determination*. [online] Available at: https://ipnpr.jpl.nasa.gov/progress_report/42-131/131E.pdf .

[16] Lopes, H.D., Silva, J.S., Silva, P.F., Musumeci, L., Dovis, F., Serant, D., Calmettes, T., Challamel, R., Ospina, J.A., Pessina, I. and Perello, J.V. (2014a). GNSS-Based Navigation for Lunar Missions. *Ion.org*, [online] pp.1536–1553. Available at: https://www.ion.org/publications/abstract.cfm?articleID=12471.

[17] Ashman, B., Bauer, F.H., Parker, J. and Donaldson, J. (2018). GPS Operations in High Earth Orbit: Recent Experiences and Future Opportunities. *2018 SpaceOps Conference*. doi:https://doi.org/10.2514/6.2018-2568.

[18] Garner, R. (2019). *Record-Breaking NASA Mission Advances High-Altitude GPS*. [online] NASA. Available at: https://www.nasa.gov/feature/goddard/2019/record-breaking-satellite-advances-nasa-s-exploration-of-high-altitude-gps/ .

[19] Parker, J.S., Fabio Dovis, Anderson, B.O., Ansalone, L., Ashman, B.W., Bauer, F.H., D'Amore, G., Facchinetti, C., Samuele Fantinato, G. Impresario, McKim, S., Egidio Luis Miotti, Miller, J.H., Musmeci, M., Pozzobon, O., Schlenker, L., Tuozzi, A. and Valencia, L.M. (2022). The Lunar GNSS Receiver Experiment (LuGRE). [online] doi:https://doi.org/10.33012/2022.18199.

[20] Delépaut, A., Giordano, P., Ventura-Traveset, J., Blonski, D., Schönfeldt, M., Schoonejans, P., Aziz, S. and Walker, R. (2020). Use of GNSS for lunar missions and plans

for lunar in-orbit development. *Advances in Space Research*, 66(12), pp.2739–2756. doi:https://doi.org/10.1016/j.asr.2020.05.018.

[21] Lopes, H.D., Silva, J.S., Silva, P.F., Musumeci, L., Dovis, F., Serant, D., Calmettes, T., Challamel, R., Ospina, J.A., Pessina, I. and Perello, J.V. (2014b). GNSS-Based Navigation for Lunar Missions. *Ion.org*, [online] pp.1536–1553. Available at: https://www.ion.org/publications/abstract.cfm?articleID=12471.

[22] Manzano-Jurado, M., Alegre-Rubio, J., Pellacani, A., Seco-Granados, G., López-Salcedo, J.A., Guerrero, E. and García-Rodríguez, A. (2014a). *Use of weak GNSS signals in a mission to the moon*. [online] IEEE Xplore. doi:https://doi.org/10.1109/NAVITEC.2014.7045151.

[23] Capuano, V., Botteron, C., Leclère, J., Tian, J., Wang, Y. and Farine, P. - A. (2015). Feasibility study of GNSS as navigation system to reach the Moon. *Acta Astronautica*, [online] 116, pp.186–201. doi:https://doi.org/10.1016/j.actaastro.2015.06.007.

[24] Manzano-Jurado, M., Alegre-Rubio, J., Pellacani, A., Seco-Granados, G., López-Salcedo, J.A., Guerrero, E. and García-Rodríguez, A. (2014b). *Use of weak GNSS signals in a mission to the moon*. [online] IEEE Xplore. doi:https://doi.org/10.1109/NAVITEC.2014.7045151.

[25] GPS.gov (2019). *GPS.gov: Space Segment*. [online] Gps.gov. Available at: https://www.gps.gov/systems/gps/space/.

[26] Mbh Leonhardstraße, F., Witternigg, N., Obertaxer, G., Schönhuber, M., Palmerini, G., Rodriguez -Diaee, F., Sapienza, L., Capponi, I., Mozzetti, C., Buccinnà, B., Borghi - Spacesys, N., Soualle, I., Fernandez-Prim, D., Floch, J-J., Defence, A., Space, G. and Graz (2015). *Weak GNSS Signal Navigation on the Moon Executive Summary Report Contents*. [online] Available at: https://nebula.esa.int/sites/default/files/neb_study/1161/C4000107097ExS.pdf .

[27] Marquis, W.A. and Reigh, D.L. (2015). The GPS Block IIR and IIR-M Broadcast L-band Antenna Panel: Its Pattern and Performance. *Navigation*, 62(4), pp.329–347. doi:https://doi.org/10.1002/navi.123.

[28] Palmerini, G.B., Sabatini, M. and Perrotta, G. (2009). En route to the Moon using GNSS signals. *Acta Astronautica*, [online] 64(4), pp.467–483. doi:https://doi.org/10.1016/j.actaastro.2008.07.022.

[29] rebound.readthedocs.io. (n.d.). *Orbital elements - REBOUND*. [online] Available at: https://rebound.readthedocs.io/en/latest/orbitalelements/ .

[30] naif.jpl.nasa.gov. (n.d.). *The Navigation and Ancillary Information Facility (NAIF)*. [online] Available at: https://naif.jpl.nasa.gov/naif/naif/ .

[31] geohack.toolforge.org. (n.d.). *GeoHack - Shackleton (crater)*. [online] Available at: https://geohack.toolforge.org/geohack.php?pagename=Shackleton_(crater)¶ms=89.9_S_0.0_E_globe:moon_type:landmark .

[32] Anon, (n.d.). *Antennas – Madrid Deep Space Communications Complex*. [online] Available at: https://www.mdscc.nasa.gov/index.php/en/antennas/ .

[33] ai-solutions.com. (n.d.). *Vector.BuildVector(Variable) Method*. [online] Available at: https://ai-solutions.com/_help_Files/vector_buildvector_0_nanosecond.htm .

[34] ai-solutions.com. (n.d.). *VisibilitySegment Object*. [online] Available at: https://ai-solutions.com/_help_Files/visibilitysegment_nanosecond.htm

[35] Shehaj, E., Capuano, V., Botteron, C., Blunt, P. and Farine, P.-A. (2017). GPS Based Navigation Performance Analysis within and beyond the Space Service Volume for Different Transmitters' Antenna Patterns. *Aerospace*, 4(3), p.44. doi:https://doi.org/10.3390/aerospace4030044.

[36] celestrak.org. (n.d.). *CelesTrak: GPS Yuma Almanacs*. [online] Available at: https://celestrak.org/GPS/almanac/Yuma/ [Accessed 16 May 2023].

8 Appendices

Appendix A: Freefyler script for objects Set up

```
// Set Up Objects


// Set step size
Gateway.Propagator.StepSize = TimeSpan.FromMinutes(20);



//Visibility segment of GPS1 Main Lobe
VisibilitySegment Segment1A;

Segment1A.SetObserver(GPS1.Sensors[0]);
Segment1A.SetTarget(Gateway);
Segment1A.AddOccultingBody(Earth);

//Visibility segment of GPS1 Side Lobe
VisibilitySegment Segment1B;

Segment1B.SetObserver(GPS1.Sensors[1]);
Segment1B.SetTarget(Gateway);
Segment1B.AddOccultingBody(Earth);

VisibilitySegment Segment2A;

Segment2A.SetObserver(GPS2.Sensors[0]);
Segment2A.SetTarget(Gateway);
Segment2A.AddOccultingBody(Earth);

VisibilitySegment Segment2B;

Segment2B.SetObserver(GPS2.Sensors[1]);
Segment2B.SetTarget(Gateway);
Segment2B.AddOccultingBody(Earth);


VisibilitySegment Segment3A;

Segment3A.SetObserver(GPS3.Sensors[0]);
Segment3A.SetTarget(Gateway);
Segment3A.AddOccultingBody(Earth);

VisibilitySegment Segment3B;

Segment3B.SetObserver(GPS3.Sensors[1]);
Segment3B.SetTarget(Gateway);
Segment3B.AddOccultingBody(Earth);
```

```
VisibilitySegment Segment4A;

Segment4A.SetObserver(GPS4.Sensors[0]);
Segment4A.SetTarget(Gateway);
Segment4A.AddOccultingBody(Earth);

VisibilitySegment Segment4B;

Segment4B.SetObserver(GPS4.Sensors[1]);
Segment4B.SetTarget(Gateway);
Segment4B.AddOccultingBody(Earth);

VisibilitySegment Segment5A;

Segment5A.SetObserver(GPS5.Sensors[0]);
Segment5A.SetTarget(Gateway);
Segment5A.AddOccultingBody(Earth);

VisibilitySegment Segment5B;

Segment5B.SetObserver(GPS5.Sensors[1]);
Segment5B.SetTarget(Gateway);
Segment5B.AddOccultingBody(Earth);

VisibilitySegment Segment6A;

Segment6A.SetObserver(GPS6.Sensors[0]);
Segment6A.SetTarget(Gateway);
Segment6A.AddOccultingBody(Earth);

VisibilitySegment Segment6B;

Segment6B.SetObserver(GPS6.Sensors[1]);
Segment6B.SetTarget(Gateway);
Segment6B.AddOccultingBody(Earth);

VisibilitySegment Segment7A;

Segment7A.SetObserver(GPS7.Sensors[0]);
Segment7A.SetTarget(Gateway);
Segment7A.AddOccultingBody(Earth);

VisibilitySegment Segment7B;

Segment7B.SetObserver(GPS7.Sensors[1]);
Segment7B.SetTarget(Gateway);
Segment7B.AddOccultingBody(Earth);

VisibilitySegment Segment8A;

Segment8A.SetObserver(GPS8.Sensors[0]);
Segment8A.SetTarget(Gateway);
Segment8A.AddOccultingBody(Earth);

VisibilitySegment Segment8B;

Segment8B.SetObserver(GPS8.Sensors[1]);
Segment8B.SetTarget(Gateway);
Segment8B.AddOccultingBody(Earth);

VisibilitySegment Segment9A;
```

```
Segment9A.SetObserver(GPS9.Sensors[0]);
Segment9A.SetTarget(Gateway);
Segment9A.AddOccultingBody(Earth);

VisibilitySegment Segment9B;

Segment9B.SetObserver(GPS9.Sensors[1]);
Segment9B.SetTarget(Gateway);
Segment9B.AddOccultingBody(Earth);

VisibilitySegment Segment10A;

Segment10A.SetObserver(GPS10.Sensors[0]);
Segment10A.SetTarget(Gateway);
Segment10A.AddOccultingBody(Earth);

VisibilitySegment Segment10B;

Segment10B.SetObserver(GPS9.Sensors[1]);
Segment10B.SetTarget(Gateway);
Segment10B.AddOccultingBody(Earth);

VisibilitySegment Segment11A;

Segment11A.SetObserver(GPS11.Sensors[0]);
Segment11A.SetTarget(Gateway);
Segment11A.AddOccultingBody(Earth);

VisibilitySegment Segment11B;

Segment11B.SetObserver(GPS11.Sensors[1]);
Segment11B.SetTarget(Gateway);
Segment11B.AddOccultingBody(Earth);

VisibilitySegment Segment12A;

Segment12A.SetObserver(GPS12.Sensors[0]);
Segment12A.SetTarget(Gateway);
Segment12A.AddOccultingBody(Earth);

VisibilitySegment Segment12B;

Segment12B.SetObserver(GPS12.Sensors[1]);
Segment12B.SetTarget(Gateway);
Segment12B.AddOccultingBody(Earth);

VisibilitySegment Segment13A;

Segment13A.SetObserver(GPS13.Sensors[0]);
Segment13A.SetTarget(Gateway);
Segment13A.AddOccultingBody(Earth);

VisibilitySegment Segment13B;

Segment13B.SetObserver(GPS13.Sensors[1]);
Segment13B.SetTarget(Gateway);
Segment13B.AddOccultingBody(Earth);

VisibilitySegment Segment14A;
```

```
Segment14A.SetObserver(GPS14.Sensors[0]);
Segment14A.SetTarget(Gateway);
Segment14A.AddOccultingBody(Earth);

VisibilitySegment Segment14B;

Segment14B.SetObserver(GPS14.Sensors[1]);
Segment14B.SetTarget(Gateway);
Segment14B.AddOccultingBody(Earth);

VisibilitySegment Segment15A;

Segment15A.SetObserver(GPS15.Sensors[0]);
Segment15A.SetTarget(Gateway);
Segment15A.AddOccultingBody(Earth);

VisibilitySegment Segment15B;

Segment15B.SetObserver(GPS15.Sensors[1]);
Segment15B.SetTarget(Gateway);
Segment15B.AddOccultingBody(Earth);

VisibilitySegment Segment16A;

Segment16A.SetObserver(GPS16.Sensors[0]);
Segment16A.SetTarget(Gateway);
Segment16A.AddOccultingBody(Earth);

VisibilitySegment Segment16B;

Segment16B.SetObserver(GPS16.Sensors[1]);
Segment16B.SetTarget(Gateway);
Segment16B.AddOccultingBody(Earth);

VisibilitySegment Segment17A;

Segment17A.SetObserver(GPS17.Sensors[0]);
Segment17A.SetTarget(Gateway);
Segment17A.AddOccultingBody(Earth);

VisibilitySegment Segment17B;

Segment17B.SetObserver(GPS17.Sensors[1]);
Segment17B.SetTarget(Gateway);
Segment17B.AddOccultingBody(Earth);

VisibilitySegment Segment18A;

Segment18A.SetObserver(GPS18.Sensors[0]);
Segment18A.SetTarget(Gateway);
Segment18A.AddOccultingBody(Earth);

VisibilitySegment Segment18B;

Segment18B.SetObserver(GPS18.Sensors[1]);
Segment18B.SetTarget(Gateway);
Segment18B.AddOccultingBody(Earth);

VisibilitySegment Segment19A;

Segment19A.SetObserver(GPS19.Sensors[0]);
```

```
Segment19A.SetTarget(Gateway);
Segment19A.AddOccultingBody(Earth);

VisibilitySegment Segment19B;

Segment19B.SetObserver(GPS19.Sensors[1]);
Segment19B.SetTarget(Gateway);
Segment19B.AddOccultingBody(Earth);

VisibilitySegment Segment20A;

Segment20A.SetObserver(GPS20.Sensors[0]);
Segment20A.SetTarget(Gateway);
Segment20A.AddOccultingBody(Earth);

VisibilitySegment Segment20B;

Segment20B.SetObserver(GPS20.Sensors[1]);
Segment20B.SetTarget(Gateway);
Segment20B.AddOccultingBody(Earth);

VisibilitySegment Segment21A;

Segment21A.SetObserver(GPS21.Sensors[0]);
Segment21A.SetTarget(Gateway);
Segment21A.AddOccultingBody(Earth);

VisibilitySegment Segment21B;

Segment21B.SetObserver(GPS21.Sensors[1]);
Segment21B.SetTarget(Gateway);
Segment21B.AddOccultingBody(Earth);

VisibilitySegment Segment22A;

Segment22A.SetObserver(GPS22.Sensors[0]);
Segment22A.SetTarget(Gateway);
Segment22A.AddOccultingBody(Earth);

VisibilitySegment Segment22B;

Segment22B.SetObserver(GPS22.Sensors[1]);
Segment22B.SetTarget(Gateway);
Segment22B.AddOccultingBody(Earth);

VisibilitySegment Segment23A;

Segment23A.SetObserver(GPS23.Sensors[0]);
Segment23A.SetTarget(Gateway);
Segment23A.AddOccultingBody(Earth);

VisibilitySegment Segment23B;

Segment23B.SetObserver(GPS23.Sensors[1]);
Segment23B.SetTarget(Gateway);
Segment23B.AddOccultingBody(Earth);

VisibilitySegment Segment24A;

Segment24A.SetObserver(GPS24.Sensors[0]);
Segment24A.SetTarget(Gateway);
```

```
Segment24A.AddOccultingBody(Earth);

VisibilitySegment Segment24B;

Segment24B.SetObserver(GPS24.Sensors[1]);
Segment24B.SetTarget(Gateway);
Segment24B.AddOccultingBody(Earth);

VisibilitySegment Segment25A;

Segment25A.SetObserver(GPS25.Sensors[0]);
Segment25A.SetTarget(Gateway);
Segment25A.AddOccultingBody(Earth);

VisibilitySegment Segment25B;

Segment25B.SetObserver(GPS24.Sensors[1]);
Segment25B.SetTarget(Gateway);
Segment25B.AddOccultingBody(Earth);

VisibilitySegment Segment26A;

Segment26A.SetObserver(GPS26.Sensors[0]);
Segment26A.SetTarget(Gateway);
Segment26A.AddOccultingBody(Earth);

VisibilitySegment Segment26B;

Segment26B.SetObserver(GPS26.Sensors[1]);
Segment26B.SetTarget(Gateway);
Segment26B.AddOccultingBody(Earth);

VisibilitySegment Segment27A;

Segment27A.SetObserver(GPS27.Sensors[0]);
Segment27A.SetTarget(Gateway);
Segment27A.AddOccultingBody(Earth);

VisibilitySegment Segment27B;

Segment27B.SetObserver(GPS27.Sensors[1]);
Segment27B.SetTarget(Gateway);
Segment27B.AddOccultingBody(Earth);

VisibilitySegment Segment28A;

Segment28A.SetObserver(GPS28.Sensors[0]);
Segment28A.SetTarget(Gateway);
Segment28A.AddOccultingBody(Earth);

VisibilitySegment Segment28B;

Segment28B.SetObserver(GPS28.Sensors[1]);
Segment28B.SetTarget(Gateway);
Segment28B.AddOccultingBody(Earth);

VisibilitySegment Segment29A;

Segment29A.SetObserver(GPS29.Sensors[0]);
Segment29A.SetTarget(Gateway);
Segment29A.AddOccultingBody(Earth);
```

```
VisibilitySegment Segment29B;

Segment29B.SetObserver(GPS29.Sensors[1]);
Segment29B.SetTarget(Gateway);
Segment29B.AddOccultingBody(Earth);

VisibilitySegment Segment30A;

Segment30A.SetObserver(GPS30.Sensors[0]);
Segment30A.SetTarget(Gateway);
Segment30A.AddOccultingBody(Earth);

VisibilitySegment Segment30B;

Segment30B.SetObserver(GPS30.Sensors[1]);
Segment30B.SetTarget(Gateway);
Segment30B.AddOccultingBody(Earth);

VisibilitySegment Segment31A;

Segment31A.SetObserver(GPS31.Sensors[0]);
Segment31A.SetTarget(Gateway);
Segment31A.AddOccultingBody(Earth);

VisibilitySegment Segment31B;

Segment31B.SetObserver(GPS31.Sensors[1]);
Segment31B.SetTarget(Gateway);
Segment31B.AddOccultingBody(Earth);




// Set up libration points
EarthMoonL1.PrimarySystem.AddBody(Earth);
EarthMoonL1.SecondarySystem.AddBody(Moon);
EarthMoonL1.LibrationPoint = "L1";
EarthMoonL1.SetReferenceBody(Moon);
EarthMoonL1.RecomputeBarycenter = 1;

EarthMoonL2.PrimarySystem.AddBody(Earth);
EarthMoonL2.SecondarySystem.AddBody(Moon);
EarthMoonL2.LibrationPoint = "L2";
EarthMoonL2.SetReferenceBody(Moon);
EarthMoonL2.RecomputeBarycenter = 1;

// Libration Point Visualization
EarthMoonL1.Color = ColorTools.Cyan;
EarthMoonL1.DisplayName = "L1";

EarthMoonL2.Color = ColorTools.Cyan;
EarthMoonL2.DisplayName = "L2";



// Contact vectors
GPS1ToGateway.BuildVector(9, Gateway, GPS1);
GPS2ToGateway.BuildVector(9, Gateway, GPS2);
GPS3ToGateway.BuildVector(9, Gateway, GPS3);
```

```
GPS4ToGateway.BuildVector(9, Gateway, GPS4);
GPS5ToGateway.BuildVector(9, Gateway, GPS5);
GPS6ToGateway.BuildVector(9, Gateway, GPS6);
GPS7ToGateway.BuildVector(9, Gateway, GPS7);
GPS8ToGateway.BuildVector(9, Gateway, GPS8);
GPS9ToGateway.BuildVector(9, Gateway, GPS9);
GPS10ToGateway.BuildVector(9, Gateway, GPS10);
GPS11ToGateway.BuildVector(9, Gateway, GPS11);
GPS12ToGateway.BuildVector(9, Gateway, GPS12);
GPS13ToGateway.BuildVector(9, Gateway, GPS13);
GPS14ToGateway.BuildVector(9, Gateway, GPS14);
GPS15ToGateway.BuildVector(9, Gateway, GPS15);
GPS16ToGateway.BuildVector(9, Gateway, GPS16);
GPS17ToGateway.BuildVector(9, Gateway, GPS17);
GPS18ToGateway.BuildVector(9, Gateway, GPS18);
GPS19ToGateway.BuildVector(9, Gateway, GPS19);
GPS20ToGateway.BuildVector(9, Gateway, GPS20);
GPS21ToGateway.BuildVector(9, Gateway, GPS21);
GPS22ToGateway.BuildVector(9, Gateway, GPS22);
GPS23ToGateway.BuildVector(9, Gateway, GPS23);
GPS24ToGateway.BuildVector(9, Gateway, GPS24);
GPS25ToGateway.BuildVector(9, Gateway, GPS25);
GPS26ToGateway.BuildVector(9, Gateway, GPS26);
GPS27ToGateway.BuildVector(9, Gateway, GPS27);
GPS28ToGateway.BuildVector(9, Gateway, GPS28);
GPS29ToGateway.BuildVector(9, Gateway, GPS29);
GPS30ToGateway.BuildVector(9, Gateway, GPS30);
GPS31ToGateway.BuildVector(9, Gateway, GPS31);




GPS1ToGateway.Color = ColorTools.Lime;
GPS2ToGateway.Color = ColorTools.Lime;
GPS3ToGateway.Color = ColorTools.Lime;
GPS4ToGateway.Color = ColorTools.Lime;
GPS5ToGateway.Color = ColorTools.Lime;
GPS6ToGateway.Color = ColorTools.Lime;
GPS7ToGateway.Color = ColorTools.Lime;
GPS8ToGateway.Color = ColorTools.Lime;
GPS9ToGateway.Color = ColorTools.Lime;
GPS10ToGateway.Color = ColorTools.Lime;
GPS11ToGateway.Color = ColorTools.Lime;
GPS12ToGateway.Color = ColorTools.Lime;
GPS13ToGateway.Color = ColorTools.Lime;
GPS14ToGateway.Color = ColorTools.Lime;
GPS15ToGateway.Color = ColorTools.Lime;
GPS16ToGateway.Color = ColorTools.Lime;
GPS17ToGateway.Color = ColorTools.Lime;
GPS18ToGateway.Color = ColorTools.Lime;
GPS19ToGateway.Color = ColorTools.Lime;
GPS20ToGateway.Color = ColorTools.Lime;
GPS21ToGateway.Color = ColorTools.Lime;
GPS22ToGateway.Color = ColorTools.Lime;
GPS23ToGateway.Color = ColorTools.Lime;
GPS24ToGateway.Color = ColorTools.Lime;
GPS25ToGateway.Color = ColorTools.Lime;
GPS26ToGateway.Color = ColorTools.Lime;
GPS27ToGateway.Color = ColorTools.Lime;
GPS28ToGateway.Color = ColorTools.Lime;
```

```
GPS29ToGateway.Color = ColorTools.Lime;
GPS30ToGateway.Color = ColorTools.Lime;
GPS31ToGateway.Color = ColorTools.Lime;


//Initialise Vectors
GPS1ToGateway.Active = 0;
GPS2ToGateway.Active = 0;
GPS3ToGateway.Active = 0;
GPS4ToGateway.Active = 0;
GPS5ToGateway.Active = 0;
GPS6ToGateway.Active = 0;
GPS7ToGateway.Active = 0;
GPS8ToGateway.Active = 0;
GPS9ToGateway.Active = 0;
GPS10ToGateway.Active = 0;
GPS11ToGateway.Active = 0;
GPS12ToGateway.Active = 0;
GPS13ToGateway.Active = 0;
GPS14ToGateway.Active = 0;
GPS15ToGateway.Active = 0;
GPS16ToGateway.Active = 0;
GPS17ToGateway.Active = 0;
GPS18ToGateway.Active = 0;
GPS19ToGateway.Active = 0;
GPS20ToGateway.Active = 0;
GPS21ToGateway.Active = 0;
GPS22ToGateway.Active = 0;
GPS23ToGateway.Active = 0;
GPS24ToGateway.Active = 0;
GPS25ToGateway.Active = 0;
GPS26ToGateway.Active = 0;
GPS27ToGateway.Active = 0;
GPS28ToGateway.Active = 0;
GPS29ToGateway.Active = 0;
GPS30ToGateway.Active = 0;
GPS31ToGateway.Active = 0;
```

Appendix B Freeflyer script for Output setup

```
// Set Up Output

Define Procedure DefineRLPCS(TimeSpan epoch, RotatingLibrationPoint rlp,
CoordinateSystem cs);

      Matrix dcm = rlp.GetDCMAtEpoch(epoch).Transpose();

      Vector x;
      Vector y;
      Vector pos;
      Vector z;

      x.Epoch = epoch;
      x.Element = (dcm * [1; 0; 0]).ToArrayColumnMajor();

      y.Epoch = epoch;
      y.Element = (dcm * [0; 1; 0]).ToArrayColumnMajor();

      z.Epoch = epoch;
      z.Element = (dcm * [0; 0; 1]).ToArrayColumnMajor();

      pos.Epoch = epoch;
      pos.Element = rlp.SecondarySystem.GetPositionAtEpoch( epoch );

      cs.Epoch = epoch;
      cs.BuildCoordinateSystem(1, x, 2, y, pos);

EndProcedure;

// Set up RLP view
RotatingLibrationPointCS.CentralBody = Moon.DisplayName;
RLPView.CurrentViewpoint.ThreeDView.TailReference =
RotatingLibrationPointCS.ObjectId;
Call DefineRLPCS(Gateway.Epoch, EarthMoonL2, RotatingLibrationPointCS);



OutputLayout.ApplyUpdates();
```

Appendix C Freeflyer script for propagation

```
OverallSideLobeCount = 0;
OverallMainLobeCount = 0;
OverallTotalLobeCount = 0;


While (Gateway.ElapsedTime < TIMESPAN(27.3 days));
      Step Gateway;
      Step GPS1 to (GPS1.Epoch == Gateway.Epoch);
      Step GPS2 to (GPS2.Epoch == Gateway.Epoch);
      Step GPS3 to (GPS3.Epoch == Gateway.Epoch);
      Step GPS4 to (GPS4.Epoch == Gateway.Epoch);
      Step GPS5 to (GPS5.Epoch == Gateway.Epoch);
      Step GPS6 to (GPS6.Epoch == Gateway.Epoch);
      Step GPS7 to (GPS7.Epoch == Gateway.Epoch);
      Step GPS8 to (GPS8.Epoch == Gateway.Epoch);
      Step GPS9 to (GPS9.Epoch == Gateway.Epoch);
      Step GPS10 to (GPS10.Epoch == Gateway.Epoch);
      Step GPS11 to (GPS11.Epoch == Gateway.Epoch);
      Step GPS12 to (GPS12.Epoch == Gateway.Epoch);
      Step GPS13 to (GPS13.Epoch == Gateway.Epoch);
      Step GPS14 to (GPS14.Epoch == Gateway.Epoch);
      Step GPS15 to (GPS15.Epoch == Gateway.Epoch);
      Step GPS16 to (GPS16.Epoch == Gateway.Epoch);
      Step GPS17 to (GPS17.Epoch == Gateway.Epoch);
      Step GPS18 to (GPS18.Epoch == Gateway.Epoch);
      Step GPS19 to (GPS19.Epoch == Gateway.Epoch);
      Step GPS20 to (GPS20.Epoch == Gateway.Epoch);
      Step GPS21 to (GPS21.Epoch == Gateway.Epoch);
      Step GPS22 to (GPS22.Epoch == Gateway.Epoch);
      Step GPS23 to (GPS23.Epoch == Gateway.Epoch);
      Step GPS24 to (GPS24.Epoch == Gateway.Epoch);
      Step GPS25 to (GPS25.Epoch == Gateway.Epoch);
      Step GPS26 to (GPS26.Epoch == Gateway.Epoch);
      Step GPS27 to (GPS27.Epoch == Gateway.Epoch);
      Step GPS28 to (GPS28.Epoch == Gateway.Epoch);
      Step GPS29 to (GPS29.Epoch == Gateway.Epoch);
      Step GPS30 to (GPS30.Epoch == Gateway.Epoch);
      Step GPS31 to (GPS31.Epoch == Gateway.Epoch);

      // Gateway.Color = ColorTools.InterpolateColorRGB({ColorTools.Blue,
ColorTools.OrangeRed, ColorTools.Maroon}, Gateway.VMag);

      // Reset the counters
      MainLobeCount = 0;
      SideLobeCount = 0;
      TotalLobeCount = 0;
      EnoughGPS = 0;
      NotEnoughGPS = 0;

      //GPS 1 Main lobe
      If (Segment1A.Visibility(GPS1.Epoch)); //and
Gateway.Contact(ShackletonCrater));
            GPS1.Sensors[0].Color = ColorTools.Lime;
            GPS1ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
```

```
  Else;
        GPS1.Sensors[0].Color = ColorTools.Red;
        GPS1ToGateway.Active = 0;
End;

  // GPS 1 Side lobe

  If (Segment1B.Visibility(GPS1.Epoch));
        GPS1.Sensors[1].Color = ColorTools.Lime;
        GPS1ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS1.Sensors[1].Color = ColorTools.Red;
        GPS1ToGateway.Active = 0;
End;

  //GPS 2 ML

  If (Segment2A.Visibility(GPS2.Epoch));
        GPS2.Sensors[0].Color = ColorTools.Lime;
        GPS2ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS2.Sensors[0].Color = ColorTools.Red;
        GPS2ToGateway.Active = 0;
End;

  //GPS 2 SL

  If (Segment2B.Visibility(GPS2.Epoch));
        GPS2.Sensors[1].Color = ColorTools.Lime;
        GPS2ToGateway.Active = 1;
        SideLobeCount++;

  Else;
        GPS2.Sensors[1].Color = ColorTools.Red;
        GPS2ToGateway.Active = 0;

End;

  //GPS 3 ML

  If (Segment3A.Visibility(GPS3.Epoch));
        GPS3.Sensors[0].Color = ColorTools.Lime;
        GPS3ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS3.Sensors[0].Color = ColorTools.Red;
        GPS3ToGateway.Active = 0;
End;

  //GPS 3 SL

  If (Segment3B.Visibility(GPS3.Epoch));
        GPS3.Sensors[1].Color = ColorTools.Lime;
        GPS3ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS3.Sensors[1].Color = ColorTools.Red;
```

```
        GPS3ToGateway.Active = 0;
End;

  //GPS 4 ML

  If (Segment4A.Visibility(GPS4.Epoch));
        GPS4.Sensors[0].Color = ColorTools.Lime;
        GPS4ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS4.Sensors[0].Color = ColorTools.Red;
        GPS4ToGateway.Active = 0;
End;

  //GPS 4 SL

  If (Segment4B.Visibility(GPS4.Epoch));
        GPS4.Sensors[1].Color = ColorTools.Lime;
        GPS4ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS4.Sensors[1].Color = ColorTools.Red;
        GPS4ToGateway.Active = 0;
End;

  //GPS 5 ML

  If (Segment5A.Visibility(GPS5.Epoch));
        GPS5.Sensors[0].Color = ColorTools.Lime;
        GPS5ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS5.Sensors[0].Color = ColorTools.Red;
        GPS5ToGateway.Active = 0;
End;

  //GPS 5 SL

  If (Segment5B.Visibility(GPS5.Epoch));
        GPS5.Sensors[1].Color = ColorTools.Lime;
        GPS5ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS5.Sensors[1].Color = ColorTools.Red;
        GPS5ToGateway.Active = 0;
End;

  //GPS 6 ML

  If (Segment6A.Visibility(GPS6.Epoch));
        GPS6.Sensors[0].Color = ColorTools.Lime;
        GPS6ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS6.Sensors[0].Color = ColorTools.Red;
        GPS6ToGateway.Active = 0;
End;
```

```
    //GPS 6 SL

    If (Segment6B.Visibility(GPS6.Epoch));
          GPS6.Sensors[1].Color = ColorTools.Lime;
          GPS6ToGateway.Active = 1;
          SideLobeCount++;
    Else;
          GPS6.Sensors[1].Color = ColorTools.Red;
          GPS6ToGateway.Active = 0;
  End;

    //GPS 7 ML

    If (Segment7A.Visibility(GPS7.Epoch));
          GPS7.Sensors[0].Color = ColorTools.Lime;
          GPS7ToGateway.Active = 1;
          MainLobeCount++;
          SideLobeCount--;
    Else;
          GPS7.Sensors[0].Color = ColorTools.Red;
          GPS7ToGateway.Active = 0;
  End;

    //GPS 7 SL

    If (Segment7B.Visibility(GPS7.Epoch));
          GPS7.Sensors[1].Color = ColorTools.Lime;
          GPS7ToGateway.Active = 1;
          SideLobeCount++;
    Else;
          GPS7.Sensors[1].Color = ColorTools.Red;
          GPS7ToGateway.Active = 0;
  End;

    //GPS 8 ML

    If (Segment8A.Visibility(GPS8.Epoch));
          GPS8.Sensors[0].Color = ColorTools.Lime;
          GPS8ToGateway.Active = 1;
          MainLobeCount++;
          SideLobeCount--;
    Else;
          GPS8.Sensors[0].Color = ColorTools.Red;
          GPS8ToGateway.Active = 0;
  End;

    //GPS 8 SL

    If (Segment8B.Visibility(GPS8.Epoch));
          GPS8.Sensors[1].Color = ColorTools.Lime;
          GPS8ToGateway.Active = 1;
          SideLobeCount++;
    Else;
          GPS8.Sensors[1].Color = ColorTools.Red;
          GPS8ToGateway.Active = 0;
  End;

    //GPS 9 ML

    If (Segment9A.Visibility(GPS9.Epoch));
          GPS9.Sensors[0].Color = ColorTools.Lime;
```

```
            GPS9ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS9.Sensors[0].Color = ColorTools.Red;
            GPS9ToGateway.Active = 0;
    End;

      //GPS 9 SL

      If (Segment9B.Visibility(GPS9.Epoch));
            GPS9.Sensors[1].Color = ColorTools.Lime;
            GPS9ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS9.Sensors[1].Color = ColorTools.Red;
            GPS9ToGateway.Active = 0;
    End;

      //GPS 10 ML

      If (Segment10A.Visibility(GPS10.Epoch));
            GPS10.Sensors[0].Color = ColorTools.Lime;
            GPS10ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS10.Sensors[0].Color = ColorTools.Red;
            GPS10ToGateway.Active = 0;
    End;

      //GPS 10 SL

      If (Segment10B.Visibility(GPS10.Epoch));
            GPS10.Sensors[1].Color = ColorTools.Lime;
            GPS10ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS10.Sensors[1].Color = ColorTools.Red;
            GPS10ToGateway.Active = 0;
    End;

      //GPS 11 ML

      If (Segment11A.Visibility(GPS6.Epoch));
            GPS11.Sensors[0].Color = ColorTools.Lime;
            GPS11ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS11.Sensors[0].Color = ColorTools.Red;
            GPS11ToGateway.Active = 0;
    End;

      //GPS 11 SL

      If (Segment11B.Visibility(GPS11.Epoch));
            GPS11.Sensors[1].Color = ColorTools.Lime;
            GPS11ToGateway.Active = 1;
            SideLobeCount++;
      Else;
```

```
        GPS11.Sensors[1].Color = ColorTools.Red;
        GPS11ToGateway.Active = 0;
End;

  //GPS 12 ML

  If (Segment12A.Visibility(GPS12.Epoch));
        GPS12.Sensors[0].Color = ColorTools.Lime;
        GPS12ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS12.Sensors[0].Color = ColorTools.Red;
        GPS12ToGateway.Active = 0;
End;

  //GPS 12 SL

  If (Segment12B.Visibility(GPS12.Epoch));
        GPS12.Sensors[1].Color = ColorTools.Lime;
        GPS12ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS12.Sensors[1].Color = ColorTools.Red;
        GPS12ToGateway.Active = 0;
End;

  //GPS 13 ML

  If (Segment13A.Visibility(GPS13.Epoch));
        GPS13.Sensors[0].Color = ColorTools.Lime;
        GPS13ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS13.Sensors[0].Color = ColorTools.Red;
        GPS13ToGateway.Active = 0;
End;

  //GPS 13 SL

  If (Segment13B.Visibility(GPS13.Epoch));
        GPS13.Sensors[1].Color = ColorTools.Lime;
        GPS13ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS13.Sensors[1].Color = ColorTools.Red;
        GPS13ToGateway.Active = 0;
End;

  //GPS 14 ML

  If (Segment14A.Visibility(GPS14.Epoch));
        GPS14.Sensors[0].Color = ColorTools.Lime;
        GPS14ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS14.Sensors[0].Color = ColorTools.Red;
        GPS14ToGateway.Active = 0;
End;
```

```
    //GPS 14 SL

    If (Segment14B.Visibility(GPS14.Epoch));
          GPS14.Sensors[1].Color = ColorTools.Lime;
          GPS14ToGateway.Active = 1;
          SideLobeCount++;
    Else;
          GPS14.Sensors[1].Color = ColorTools.Red;
          GPS14ToGateway.Active = 0;
  End;

    //GPS 15 ML

    If (Segment15A.Visibility(GPS15.Epoch));
          GPS15.Sensors[0].Color = ColorTools.Lime;
          GPS15ToGateway.Active = 1;
          MainLobeCount++;
          SideLobeCount--;
    Else;
          GPS15.Sensors[0].Color = ColorTools.Red;
          GPS15ToGateway.Active = 0;
  End;

    //GPS 15 SL

    If (Segment15B.Visibility(GPS15.Epoch));
          GPS15.Sensors[1].Color = ColorTools.Lime;
          GPS15ToGateway.Active = 1;
          SideLobeCount++;
    Else;
          GPS15.Sensors[1].Color = ColorTools.Red;
          GPS15ToGateway.Active = 0;
  End;

    //GPS 16 ML

    If (Segment16A.Visibility(GPS16.Epoch));
          GPS16.Sensors[0].Color = ColorTools.Lime;
          GPS16ToGateway.Active = 1;
          MainLobeCount++;
          SideLobeCount--;
    Else;
          GPS16.Sensors[0].Color = ColorTools.Red;
          GPS16ToGateway.Active = 0;
  End;

    //GPS 16 SL

    If (Segment16B.Visibility(GPS16.Epoch));
          GPS16.Sensors[1].Color = ColorTools.Lime;
          GPS16ToGateway.Active = 1;
          SideLobeCount++;
    Else;
          GPS16.Sensors[1].Color = ColorTools.Red;
          GPS16ToGateway.Active = 0;
  End;

    //GPS 17 ML

    If (Segment17A.Visibility(GPS17.Epoch));
```

```
        GPS17.Sensors[0].Color = ColorTools.Lime;
        GPS17ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS17.Sensors[0].Color = ColorTools.Red;
        GPS17ToGateway.Active = 0;
End;

  //GPS 17 SL

  If (Segment17B.Visibility(GPS17.Epoch));
        GPS17.Sensors[1].Color = ColorTools.Lime;
        GPS17ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS17.Sensors[1].Color = ColorTools.Red;
        GPS17ToGateway.Active = 0;
End;

  //GPS 18 ML

  If (Segment18A.Visibility(GPS18.Epoch));
        GPS18.Sensors[0].Color = ColorTools.Lime;
        GPS18ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS18.Sensors[0].Color = ColorTools.Red;
        GPS18ToGateway.Active = 0;
End;

  //GPS 18 SL

  If (Segment18B.Visibility(GPS18.Epoch));
        GPS18.Sensors[1].Color = ColorTools.Lime;
        GPS18ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS18.Sensors[1].Color = ColorTools.Red;
        GPS18ToGateway.Active = 0;
End;

  //GPS 19 ML

  If (Segment19A.Visibility(GPS19.Epoch));
        GPS19.Sensors[0].Color = ColorTools.Lime;
        GPS19ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS19.Sensors[0].Color = ColorTools.Red;
        GPS19ToGateway.Active = 0;
End;

  //GPS 19 SL

  If (Segment19B.Visibility(GPS19.Epoch));
        GPS19.Sensors[1].Color = ColorTools.Lime;
        GPS19ToGateway.Active = 1;
        SideLobeCount++;
```

```
  Else;
        GPS19.Sensors[1].Color = ColorTools.Red;
        GPS19ToGateway.Active = 0;
End;

  //GPS 20 ML

  If (Segment20A.Visibility(GPS20.Epoch));
        GPS20.Sensors[0].Color = ColorTools.Lime;
        GPS20ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS20.Sensors[0].Color = ColorTools.Red;
        GPS20ToGateway.Active = 0;
End;

  //GPS 20 SL

  If (Segment20B.Visibility(GPS20.Epoch));
        GPS20.Sensors[1].Color = ColorTools.Lime;
        GPS20ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS20.Sensors[1].Color = ColorTools.Red;
        GPS20ToGateway.Active = 0;
End;

  //GPS 21 ML

  If (Segment21A.Visibility(GPS21.Epoch));
        GPS21.Sensors[0].Color = ColorTools.Lime;
        GPS21ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS21.Sensors[0].Color = ColorTools.Red;
        GPS21ToGateway.Active = 0;
End;

  //GPS 21 SL

  If (Segment21B.Visibility(GPS21.Epoch));
        GPS21.Sensors[1].Color = ColorTools.Lime;
        GPS21ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS21.Sensors[1].Color = ColorTools.Red;
        GPS21ToGateway.Active = 0;
End;

  //GPS 22 ML

  If (Segment22A.Visibility(GPS22.Epoch));
        GPS22.Sensors[0].Color = ColorTools.Lime;
        GPS22ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS22.Sensors[0].Color = ColorTools.Red;
        GPS22ToGateway.Active = 0;
```

```
    End;

      //GPS 22 SL

      If (Segment22B.Visibility(GPS22.Epoch));
            GPS22.Sensors[1].Color = ColorTools.Lime;
            GPS22ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS22.Sensors[1].Color = ColorTools.Red;
            GPS22ToGateway.Active = 0;
    End;

      //GPS 23 ML

      If (Segment23A.Visibility(GPS23.Epoch));
            GPS23.Sensors[0].Color = ColorTools.Lime;
            GPS23ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS23.Sensors[0].Color = ColorTools.Red;
            GPS23ToGateway.Active = 0;
    End;

      //GPS 23 SL

      If (Segment23B.Visibility(GPS23.Epoch));
            GPS23.Sensors[1].Color = ColorTools.Lime;
            GPS23ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS23.Sensors[1].Color = ColorTools.Red;
            GPS23ToGateway.Active = 0;
    End;

      //GPS 24 ML

      If (Segment24A.Visibility(GPS24.Epoch));
            GPS24.Sensors[0].Color = ColorTools.Lime;
            GPS24ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS24.Sensors[0].Color = ColorTools.Red;
            GPS24ToGateway.Active = 0;
    End;

      //GPS 24 SL

      If (Segment24B.Visibility(GPS24.Epoch));
            GPS24.Sensors[1].Color = ColorTools.Lime;
            GPS24ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS24.Sensors[1].Color = ColorTools.Red;
            GPS24ToGateway.Active = 0;
    End;

      //GPS 25 ML
```

```
  If (Segment25A.Visibility(GPS25.Epoch));
        GPS25.Sensors[0].Color = ColorTools.Lime;
        GPS25ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS25.Sensors[0].Color = ColorTools.Red;
        GPS25ToGateway.Active = 0;
End;

  //GPS 25 SL

  If (Segment25B.Visibility(GPS25.Epoch));
        GPS25.Sensors[1].Color = ColorTools.Lime;
        GPS25ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS25.Sensors[1].Color = ColorTools.Red;
        GPS25ToGateway.Active = 0;
End;

  //GPS 26 ML

  If (Segment26A.Visibility(GPS26.Epoch));
        GPS26.Sensors[0].Color = ColorTools.Lime;
        GPS26ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS26.Sensors[0].Color = ColorTools.Red;
        GPS26ToGateway.Active = 0;
End;

  //GPS 26 SL

  If (Segment26B.Visibility(GPS26.Epoch));
        GPS26.Sensors[1].Color = ColorTools.Lime;
        GPS26ToGateway.Active = 1;
        SideLobeCount++;
  Else;
        GPS26.Sensors[1].Color = ColorTools.Red;
        GPS26ToGateway.Active = 0;
End;

  //GPS 27 ML

  If (Segment27A.Visibility(GPS27.Epoch));
        GPS27.Sensors[0].Color = ColorTools.Lime;
        GPS27ToGateway.Active = 1;
        MainLobeCount++;
        SideLobeCount--;
  Else;
        GPS27.Sensors[0].Color = ColorTools.Red;
        GPS27ToGateway.Active = 0;
End;

  //GPS 27 SL

  If (Segment27B.Visibility(GPS27.Epoch));
        GPS27.Sensors[1].Color = ColorTools.Lime;
        GPS27ToGateway.Active = 1;
```

```
            SideLobeCount++;
      Else;
            GPS27.Sensors[1].Color = ColorTools.Red;
            GPS27ToGateway.Active = 0;
    End;

      //GPS 28 ML

      If (Segment28A.Visibility(GPS28.Epoch));
            GPS28.Sensors[0].Color = ColorTools.Lime;
            GPS28ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS28.Sensors[0].Color = ColorTools.Red;
            GPS28ToGateway.Active = 0;
    End;

      //GPS 28 SL

      If (Segment28B.Visibility(GPS28.Epoch));
            GPS28.Sensors[1].Color = ColorTools.Lime;
            GPS28ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS28.Sensors[1].Color = ColorTools.Red;
            GPS28ToGateway.Active = 0;
    End;

      //GPS 29 ML

      If (Segment29A.Visibility(GPS29.Epoch));
            GPS29.Sensors[0].Color = ColorTools.Lime;
            GPS29ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS29.Sensors[0].Color = ColorTools.Red;
            GPS29ToGateway.Active = 0;
    End;

      //GPS 29 SL

      If (Segment29B.Visibility(GPS29.Epoch));
            GPS29.Sensors[1].Color = ColorTools.Lime;
            GPS29ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS29.Sensors[1].Color = ColorTools.Red;
            GPS29ToGateway.Active = 0;
    End;

      //GPS 30 ML

      If (Segment30A.Visibility(GPS30.Epoch));
            GPS30.Sensors[0].Color = ColorTools.Lime;
            GPS30ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS30.Sensors[0].Color = ColorTools.Red;
```

```
            GPS30ToGateway.Active = 0;
    End;

      //GPS 30 SL

      If (Segment30B.Visibility(GPS30.Epoch));
            GPS30.Sensors[1].Color = ColorTools.Lime;
            GPS30ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS30.Sensors[1].Color = ColorTools.Red;
            GPS30ToGateway.Active = 0;
    End;

      //GPS 31 ML

      If (Segment31A.Visibility(GPS31.Epoch));
            GPS31.Sensors[0].Color = ColorTools.Lime;
            GPS31ToGateway.Active = 1;
            MainLobeCount++;
            SideLobeCount--;
      Else;
            GPS31.Sensors[0].Color = ColorTools.Red;
            GPS31ToGateway.Active = 0;
    End;

      //GPS 31 SL

      If (Segment31B.Visibility(GPS31.Epoch));
            GPS31.Sensors[1].Color = ColorTools.Lime;
            GPS31ToGateway.Active = 1;
            SideLobeCount++;
      Else;
            GPS31.Sensors[1].Color = ColorTools.Red;
            GPS31ToGateway.Active = 0;
    End;


      TotalLobeCount = MainLobeCount + SideLobeCount;



      OverallSideLobeCount = OverallSideLobeCount + SideLobeCount;
      OverallMainLobeCount = OverallMainLobeCount + MainLobeCount;
      OverallTotalLobeCount = OverallTotalLobeCount + TotalLobeCount;


      Call DefineRLPCS(Gateway.Epoch, EarthMoonL2,
RotatingLibrationPointCS);

      Update EarthInertialView;
      Update MoonInertialView;
      Update RLPView;
      Update SignalCountPlot;

End;

Update DataTableWindow1;
```