Tobias Pieter Tanis

Autonomous Orbit Determination in Cislunar Space

Delft University of Technology

TU Delft

Autonomous Orbit Determination in Cislunar Space

by

Tobias Pieter Tanis

In partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

Delft University of Technology

Student number:	4327276	
Project duration:	November 15, 2021 – August 15, 2022	
Thesis committee:	Prof. dr. E.K.A. Gill,	Chair thesis commitee, TU Delft
	Dr. ir. S. Speretta,	Daily supervisor, TU Delft
	ir. E. Turan,	Daily supervisor, TU Delft
	Dr. ir. D. Dirkx,	External member, TU Delft
Cover image	by Robber Dekker	

TU Delft

Preface

This report describes the work carried out by Tobias Pieter Tanis during the Master's thesis, which is the final part in order to complete the Master's Aerospace Engineering with as specialization Space Engineering. The main objective of this research was to determine the required radio system parameters in order to perform autonomous orbit determination in Cislunar space. This is done by combining link budget analyses via an estimated ranging error to a developed model that executes autonomous orbit determination simulations. This report could therefore be interesting for people with an affinity with autonomous space navigation for small satellite missions.

Large parts of the model were built using the TU Delft Astrodynamics Toolbox (Tudat), which is a powerful set of libraries in C++ and Python that support research in orbital and astrodynamics. I would like to thank Dominic Dirkx for providing Tudat as an open-source toolkit and for the incredible effortless help that is provided in the thesis-tudat-assistance in Slack.

Also, a special thanks goes to Jasper Schol, Charlotte Tanis and Noor Jaarsma. Jasper Schol for his tips and tricks in the use of Python, Charlotte Tanis for the feedback on draft versions and Noor Jaarsma for her occasional motivational speeches during the thesis.

Most of all, I would like to thank Stefano Speretta and Erdem Turan. I really enjoyed the collaboration and supervision during this project and I am sincerely grateful for the help and knowledge I have obtained.

Tobias Pieter Tanis
Delft, 15 August 2022

Abstract

The amount of space missions is growing rapidly. The Unite Nations Office for Outer Space Affairs (UNOOSA) keeps track of object launched into outer space and registered in 2021 42 % more objects launched into outer space than in 2020 and in 2020 a growth of 117 % was measured. Growing interest from commercial, educational and science parties led to an increase in amount of missions, which are possible due to ride-share programs and piggy bag launches of CubeSats. Besides the growing interest in space missions, there is also a growing interest in new ways of orbit determination, preferably autonomous. Autonomous orbit determination could potentially reduce the operational costs due to a reduction of ground support. This research is focused on developing an autonomous orbit determination method that uses inter-satellite measurements to determine the satellite's states. This method is also known as Linked, Autonomous, Interplanetary Satellite Orbit Navigation (LiAISON) and requires multiple (at least two) satellites in locally different gravity fields with one satellite orbiting in a strong asymmetric gravity field, e.g., near an Earth-Moon Lagrange point. In this research two satellites are considered: an Elliptic Lunar Orbiter (ELO) and an Earth-Moon L2 Orbiter (EML2O). Both satellites are based on existing planned missions, which are the Lunar Pathfinder (ELO) and LUMIO (EML2O). The goal of this research is to determine the required radio parameters to perform LiAISON based orbit determination system while complying with the mission requirements of LUMIO. The position and velocity accuracy must be within a 3-D 3σ uncertainty of 1000 m and 1 cm/s.

The autonomous orbit determination model depends on the performance of the on-board radio system and therefore a simplified communication system is taken into account in an effort to demonstrate the feasibility of telemetry ranging. As observables, only range-observations are considered that are implemented in the orbit determination model as normal distributed measurement noise. The performance of the model is highly dependent on the magnitude of the ranging error. Multiple ranging errors, based on an arbitrary bit rate, are calculated and implemented in the autonomous orbit determination model. This iterative process is repeated until the mission requirements of LUMIO are met. As basis radio parameters, Lunar Pathfinder's and on the considered parameters for LUMIO are used. Thereafter the antenna gain of EML2O is increased as variable parameter.

The link budgets are performed for the worst case scenario. This means for the largest inter-satellite distance and highest possible frequencies. The largest inter-satellite distance in this simulation span is 87 237.09 km and the highest possible frequencies based on the available frequencies of the Lunar Pathfinder and the considered antenna for LUMIO are f_{down} = 2290 MHz and f_{up} = 2110 MHz, where subscript $_{down}$ is from EML2O to ELO and subscript $_{up}$ is from ELO to EML2O. The variable link budget parameter is EML2O's antenna gain on which the possible the up- and downlink bit rate depends. As starting point, a nominal antenna gain of 6.5 dBi at boresight is defined for EML2O. From that point, extra antennas are combined in an antenna array to form a higher antenna gain. Considering there are no scaling losses, the link budget is performed for an array of 1, 2, 4, 8, and 16 antennas that are accompanied with an antenna gain of 6.5, 9.5, 12.5, 15.5, and 18.5 dBi. From the link budget analysis a downlink bit rate of 850 bps and an uplink bit rate of 1000 bps could be obtained considering one antenna with 6.5 dBi and a margin of 3 dB. It was found that each doubling in antennas led to a doubling in bit rate and a reduction of factor 4 for the ranging error, as shown in Table 1. From the link budgets, the estimated two-way ranging errors are calculated and are: 385.6 m, 96.6 m, 24.2 m, 6.07 m, and 1.52 m, from 1 to an array of 16 antennas.

The autonomous orbit determination model consists of three sub-models: the dynamic, measurement and estimation model. The dynamic model consists of three input settings: the environment, propagation and integration settings. As environment, all Solar System planets, the Sun and Moon are taken into account and implemented in the acceleration function. Positions and velocities and Solar System body properties are obtained from ephemerides and states are provided in an Earth-centered J2000 reference frame. Additionally, the acceleration exerted from Earth and the Moon are modelled as spherical harmonic gravity of an maximum order and degree of 200 and solar radiation pressure exerted by the Sun is modelled as a cannonball. The simulation is executed from the 21^{th} of March 2024 until the 4^{th} of April 2024 and using a Runge-Kutta 4

numerical integrator, the nominal states are obtained.

The measurement model uses the nominal states and the ranging error of the different antenna configurations to provide the nominal measurements, which exists of the absolute inter-satellite distance plus the ranging error modelled as normal distributed Gaussian noise. Inter-satellite visibility is taken into account to determine in epochs in which no line-of-sight is available and therefore no ranging operations. It was found that only in the first day and the last 2 days, there are short intervals when there is no visibility. The estimation model consists of an Extended Kalman Filter that performs a time update and measurement update each iteration. The measurement time is fixed on 300 s, given visibility and the time update is performed 10 times every 300 s, which is to mitigate the maximum model error, 53.3 m and 67.7 m for EML2O and ELO, between the dynamic and estimation model. In case of no visibility, the measurement update is not executed As output, the estimation model provides the estimated states and the variance. By subtracting the estimated states from the nominal states, the estimated state element errors are found. By taking the square root of the variance and multiply it by three, the 3σ uncertainty per state element is found. The state element error estimation is performed for each antenna configuration accompanied by a different ranging error.

As results, the 3-D 3σ uncertainties were determined in order to visualize whether the given radio parameters meet the mission requirements. It was found that 1 antenna was not able to meet the mission requirements in the given simulation time of 14 days. The 2, 4, 8, and 16 antenna array configurations were able to meet the position and velocity requirements in: 10.9, 8.1, 6.3, and 5.1 days. Additionally, the average 3-D 3σ uncertainties were determined from 6 days of simulation until simulation end time. From 6 days was defined based on the degree of convergence of the 3σ position uncertainties of EML2O before and after 6 days of simulation. The results of each antenna configuration accompanied by the antenna gain, bit rate and ranging error are displayed in Table 1

Table 1: The 3-D 3σ position and velocity uncertainty of Earth-Moon L2 Orbiter (EML2O) based in different radio parameters.

	1 Antenna	2 Antennas	4 Antennas	8 Antennas	16 Antennas
Gain [dBi]	6.5	9.5	12.5	15.5	18.5
bitrate$_{down}$ [bps]	850	1700	3400	6800	13600
bitrate$_{up}$ [bps]	1000	2000	4000	8000	16000
Ranging error [m]	385.6	96.6	24.2	6.07	1.52
Req. time to meet <1000 m 3-D σ uncertainty [days]	N/A	10.8	8.1	6.3	5.1
Req. time to meet <1 m/s 3-D σ uncertainty [days]	N/A	4.1	3.4	2.8	2.3
Average 3-D position estimation accuracy [m]	2755	1431	781.5	315.9	122.3
Average 3-D velocity estimation accuracy [m/s]	1.387e-2	6.473e-3	3.583e-3	1.630e-3	6.538e-4

Table 1 shows the autonomous orbit determination performance for each configuration. An antenna gain up to 9.5 dBi would not be sufficient to fulfill the mission requirements. Equipping an antenna array of 4 would be sufficient if one decides to measure for a longer duration than 8.1 days without interference, e.g., stationkeeping. However, the halo orbit period is approximately 14 days and it is expected that stationkeeping is performed every 7 to 14 days. Therefore, the radio parameters accompanied with 4 antennas would in some cases not be sufficient. An array of 8 antennas can facilitate Linked, Autonomous, Interplanetary Satellite Orbit Navigation (LiAISON) based orbit determination, even if stationkeeping would be performed.

Contents

List of Figures

List of Tables

Acronyms

AOD Autonomous Orbit Determination. 9, 10, 13–15, 17, 23–26, 50, 53, 72

BPSK Binary Phase Shift Keying. 47, 72

CAPSTONE Cislunar Autonomous Positioning System Technology, Operations, and Navigation Experiment. 16, 18, 27, 47

CRTBP Circular Restricted Three-Body Problem. 11, 29, 95, 96

DSN Deep Space Network. 9

E_b/N_0 Energy per bit to noise power spectral density ratio. 43–45, 47, 48, 72

E_s/N_0 Symbol to noise ratio. 47, 72

EIRP Effective Isotropic Radiated Power. 43, 45

EKF Extended Kalman Filter. 15, 25, 26, 53–56, 73

ELFO Elliptic Lunar Frozen Orbit. 20

ELO Elliptic Lunar Orbiter. 1, 3, 21, 26, 27, 29–31, 33–38, 41–45, 51, 53, 54, 57–60, 63–66, 68, 69, 71, 73–75

EML2 Earth-Moon Lagrange 2. 38

EML2O Earth-Moon L2 Orbiter. vi, 1, 3, 10, 26, 27, 29–31, 33–39, 41–45, 51, 54, 57–64, 66–69, 71–75, 93

EPS Electric Power System. 13, 14

ESA European Space Agency. 9, 20, 21, 26

ET Ephemeris Time. 30, 57

G/T Gain-to-noise-temperature. 45

GMSK Gaussian Minimum Shift Keying. 47, 72

ICRF International Celestial Reference System and Frame. 29, 96

IDE Integrated Development Environment. 24

KF Kalman Filter. 53

LEO Low Earth Orbit. 9, 19, 32

LiAISON Linked, Autonomous, Interplanetary Satellite Orbit Navigation. vi, 9–11, 15–20, 23, 64, 66, 68, 71, 72, 75

LME Lunar Meteoroid Environment. 10

LNA Low-Noise Amplifier. 42

LOS Line-of-Sight. 15, 25, 41

LRO Lunar Reconnaissance Orbiter. 16, 27

LUMIO LUnar Meteoroid Impact Observer. 1, 3, 9, 10, 13–19, 23, 24, 26, 27, 30, 33–36, 39, 43, 44, 47, 61, 66, 71, 74, 75

M-ARGO Miniaturised Asteroid Remote Geophysical Observer. 15, 17

MJD Modified Julian Time. 30

NAIF Navigation and Ancillary Information Facility. 25

NASA National Aeronautics and Space Administration. 9, 16, 25, 27

NRHO Near-Rectilinear Halo Orbit. 16, 27

OBC On-Board Computer. 3, 42, 58, 59

RK4 Runge-Kutta 4. 35, 57, 59

SKM Stationkeeping Manoeuvre. 30

SNR Signal-to-Noise ratio. 42, 43, 66

SRP Solar Radiation Pressure. 7, 11, 16, 21, 25, 31–33, 35, 37, 39, 57, 59, 73, 74

SSB Solar System Barycenter. 30

SXM Solar X-ray Monitor. 16

TBD To Be Determined. 44

TT&C Telemetry, Tracking & Command. 24

UHF Ultra-High Frequency. 21

UKF Unscented Kalman Filter. 53

UNOOSA United Nation Office for Outer Space Affairs. 9

XIS X-ray imaging spectrometer. 16

Nomenclature

Physical parameters

d	Distance [m]
m	Mass [kg]
P	Total emitted radiation power [Watt]
S_{ref}	Reference area [m^2]
T	Temperature [K]

Dimensional parameters

r	Vector pointing from satellite to the Sun
a_{SRP}	Acceleration due to Solar Radiation Pressure (SRP) [m/s^2]

Estimation parameters

$\tilde{\mathbf{H}}_k$	Observation-state matrix
K	Kalman gain
P	Covariance matrix
Q	State noise compensation matrix
Φ	State transition matrix
R	Measurement noise covariance
Y	Observation
y	Observation residual

Orbital parameters

$\dot{x}$	velocity in x-direction [m/s]
$\dot{y}$	velocity in y-direction [m/s]
$\dot{z}$	velocity in z-direction [m/s]
Ω	Longitude of ascending node [°]
ω	Argument of periapsis [°]
θ	True anomaly [°]
a	Semi-major axis [m]
e	Eccentricity [-]
i	Inclination [°]
x	position in x-direction [m]
y	position in y-direction [m]

z	position in z-direction [m]	

Physics constants

c	Speed of light in vacuum	$2.99792458 \times 10^{8}\,\mathrm{m\,s^{-1}}$
G	Gravitational constant	$6.67430 \times 10^{-11}\,\mathrm{m^{3}\,kg^{-1}\,s^{-2}}$
k_b	Boltzmann constant	$1.380649 \times 10^{-23}\,\mathrm{J\,K^{-1}}$

Radio parameters

λ	Wavelength [m]
σ_ρ	Ranging error [1σ m]
G	Antenna gain [dBi]
L	Losses [dB]
L_{cable}	Cablelosses [dB]
L_{FSL}	Free space losses [dB]
L_{pol}	Polarization losses [dB]
M	Number of alternative modulation symbols [-]
R_x	Receiver power [dBW]
T_i	Integration time [s]
T_{noise}	Noise Temperature [dB/K]
T_{sd}	Symbol duration [s/bit]
T_x	Transmission power [dBW]

Coefficients

α	Weight factor
C_R	Radiation pressure coefficient [-]

1

Introduction

In the past few years, space flight and space exploration have grown in popularity. In contrast to the 60s and 80s, commercial, educational and science parties have now joined the race to space. Small satellites launched in piggy bags and ride-share programs make it financially possible to launch spacecrafts with a modest budget. United Nation Office for Outer Space Affairs (UNOOSA) keeps track of objects launched into outer space and documented a growth of 41% of launched objects in 2021 with respect to 2020, which in turn experienced a staggering growth of 117% compared to 2019 UNOOSA (2022). Due to recent developments, one can send a 20 kilogram CubeSat into Low Earth Orbit (LEO) for only 600,000 dollar Gelain (2021). Outside LEO, the number of space missions grows as well. Both National Aeronautics and Space Administration (NASA) and European Space Agency (ESA) are involved in developing communication systems, payload systems and constellations of satellites for further exploration of the Moon.

While acquisition costs of satellites are reducing, their operational costs remain significant. Satellites commonly use ground stations for navigation support, scientific data communication and satellite status updates. Deep Space Network (DSN)-supported spacecrafts can expect an hourly rate of 850 to over 10,000 US dollars when requesting ground support, depending on the intensity of support needed NASA (2014). More importantly, the load on ground support stations is nearing its maximum capacity and is expected to be fully exhausted in the near future Towfic et al. (2018). Expanding the current network of ground stations to meet the increasing demand is in progress, but due to the exponential growth of space missions it could be insufficient O'Neill (2021). Another solution could be to reduce the load on ground stations by finding alternatives for the tasks for which ground stations are required. For scientific data handling and spacecraft status updates, this might proof to be a very difficult task as they both require communication to Earth. However, if the need for navigation support could decrease then both operational costs and the pressure on ground stations could also decrease.

A key aspect of navigation support consists of determining a satellite's orbit. Orbit determination requires knowledge of the states of a satellite. One state vector describes the position and velocity of a satellite at one epoch. Consecutive states form the orbit, or trajectory, of a satellite. The position and velocity are defined relative to a chosen reference frame. For successful orbit determination, one needs to determine the absolute states which contain the state elements with respect to a quasi-inertial reference frame such as the center of the Earth. The goal of this research is to investigate the possibilities of autonomous orbit determination of a satellite rather than relying on ground stations.

Literature research showed that Autonomous Orbit Determination (AOD) based on LiAISON could be promising for LUMIO Tanis (2022). This method determines the states of at least two satellites simultaneously using only inter-satellite measurements obtained from ranging operations. The absolute states can be determined if one satellite orbits in a local asymmetric gravity field, because the satellite will have a 'unique' orbit in terms of size, shape and orientation Hill (2007). The states estimated by LiAISON depend on the performance of the radio systems, which are described in terms of link budget parameters. LiAISON based on telemetry range observations provide more accurate estimated states than range-rate observations derived from Doppler shifts Turan et al. (2022).

As use-case satellites, two hypothetical satellites based on the properties of the Lunar Pathfinder and LUMIO are considered. The Lunar Pathfinder is a satellite due to launch in 2023 and one of its main purposes is to provide communication support to future missions in Cislunar space. LUMIO is a 12U CubeSat that is planned to launch in 2024 and will provide information about the Lunar Meteoroid Environment (LME) by observing flashes of meteoroid impacts on the far side of the Moon for the duration of one year Cervone et al. (2022). In this research project, the objective is to determine the position and velocity accuracy of LUMIO, while the Lunar Pathfinder is only used to support the autonomous navigation operations. The feasibility of autonomous orbit determination is judged based on the 3σ position and velocity accuracy requirements, i.e., 1000 m and 1 cm/s respectively, for the LUMIO mission Cervone et al. (2022).

1.1. Research goal and questions

The main research goal of this project is:

> 'To determine radio system parameters for a LiAISON-based autonomous orbit determination model that meet the position and velocity accuracy requirements of LUMIO.'

The performance of a LiAISON-based autonomous orbit determination system is largely determined by the magnitude of the ranging error, which is defined by the link budget parameters. To determine these radio system parameters, a simplified communication system is considered to investigate the feasibility of telemetry ranging. In this system, most radio parameters are fixed, but the number of antennas used can vary. The AOD system is implemented in Python and consists of three sub-models: (i) a dynamic simulation model; (ii) a measurement model; and (iii) an estimation model. The dynamic model contains a high fidelity environment to simulate the motion of LUMIO and the Lunar Pathfinder to obtain reliable nominal trajectories of both satellites. The measurement model estimates the absolute distance between the two satellites while taking the visibility into account. Measurement noise is added to the distances, and the amount of noise is based on the estimated telemetry ranging. The estimation model uses output from the other two sub-models to estimate the position and velocity errors, and their variance for both satellites. The average outer-boundaries of the 3-D position and velocity errors can be determined from the variance. In an iterative process, the position and velocity estimates resulting from different radio parameters were judged based on the requirements of LUMIO.

1.1.1. Research questions

In order to achieve the research goal, the main research question is:

> 'Which radio parameters are required for LUMIO to perform LiAISON-based autonomous orbit determination while complying with its mission requirements?'

To provide an answer to this research question, the following sub-question are stated.

1. Based on the number of antennas, what are the radio performance parameters and the estimated ranging error using telemetry ranging?

2. Which components does a LiAISON based autonomous orbit determination model entail if orbit determination is solely done using inter-satellite ranging?

3. What are the state integrator settings in the estimation filter to provide a 3-D position model error of less than 100 m

4. What are the average 3-D position and velocity estimation accuracy's of EML2O per considered antenna configuration?

5. At which simulation time a 3-D position and velocity estimation accuracy is achieved that is conform with LUMIOs requirements?

1.2. Scientific relevance

LiAISON based orbit determination is a concept developed in the early 2000s by Keric Hill Hill (2007). An extensive literature on the method exists, but most research focuses on theoretical satellites in theoretical (simplified) dynamic models, e.g., the Circular Restricted Three-Body Problem (CRTBP) and occasionally include SRP. Such models often are quite optimistic in performance, and radio systems providing the inter-satellite measurements are often undiscussed. Due to these simplifications, a gap exists between the theoretical models and actual performance of LiAISON based orbit determination. This research aims to narrow this gap by considering less simplified models. Additional factors such as multiple perturbations, visibility and estimated radio performance will also be taken into account. By using a more realistic simulation environment, this research aims to provide top level radio parameters that provide accurate position and velocity estimates to enable autonomous orbit determination for small satellites in practice.

1.3. Structure

First, a literature overview (chapter 2) is provided that discusses the background of the relevant subjects used in this thesis. Second, the method (chapter 3 is provided that explains the steps to take in order to achieve the research goal. Third, the autonomous orbit determination model is discussed in detail in chapter 4, chapter 5 and chapter 6. Forth, the simulations and results obtained from the autonomous orbit determination model are provided in chapter 7. Last, conclusions are drawn based on the results and the research questions are answered in the chapter 8, which also provides a discussion and suggests possible future work.

2

Literature overview

In the literature study preceding this thesis, it was investigated which AOD method could be promising for small satellites. This chapter provides an overview of the requirements for LUMIO, a summary of the different AOD methods, and describes a trade-off between these methods based on the different requirements for LUMIO.

2.1. LUnar Meteoroid Impact Observer

LUMIO is a 12U CubeSat with a wet mass of 22.8 kg orbiting a halo orbit on the farside of the Moon. LUMIO has a mission duration of 1 year (i.e., the operational phase) and is planned to be operational from the 21^{th} of March 2024. Its main mission purpose is to observe meteoroid impacts on the far side of the Moon by detecting flashes that occur during impact. Besides observation, LUMIO keeps track of the amount of flashes and these data is subsequently used to characterize the meteoroids. Ultimately, the LUMIO mission aims to provide a better understanding of the characteristics of the Solar System, e.g., compositions of Solar System bodies (Cervone et al., 2022).

For this research, LUMIO is used as reference mission to determine which AOD method is most suitable. The mission requirements of LUMIO regarding the overall mass and volume and the communication/radio subsystem are provided in Table 2.1. The OVRSYS requirements are obtained from Topputo et al. (2018) and describes the restrictions to its total mass and volume. The communications system (COMMS) requirements are obtained from Speretta et al. (2021) and provide requirements that the radio system should meet. The overall required power in operational phase for the Electric Power System (EPS) is obtained from Cervone et al. (2022).

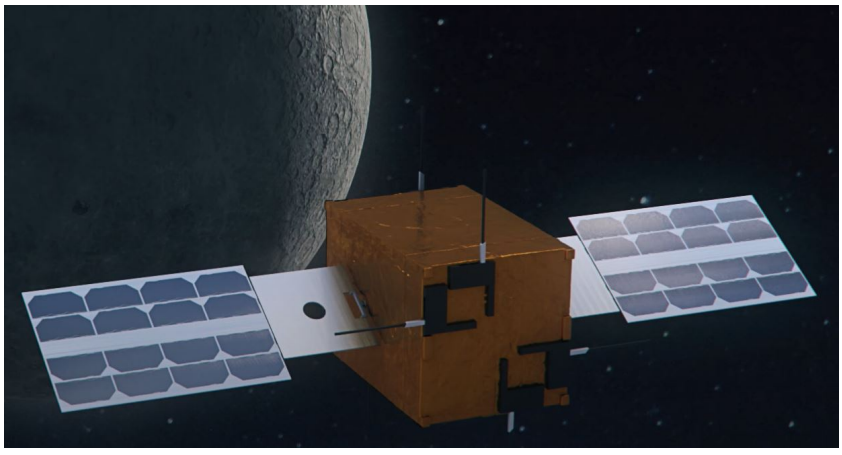


Figure 2.1: Impression of LUMIO in Cislunar space, *Illustration by* Topputo (2017).

Table 2.1: LUMIO mission requirements regarding the overall volume and mass and the communication subsystem.

ID	Requirement	Reference
OVRSYS-001	The initial mass of the S/C (including propellant) shall be no more than 24 kg	(Topputo et al., 2018)
OVRSYS-002	The spacecraft shall be a 12U CubeSat, with a volume of 3x2x2 CubeSat units	(Topputo et al., 2018)
COMMS-010	The spacecraft shall have a telecommand receiver operating according to Recommendation SFCG 32-2R1	(Speretta et al., 2021)
COMMS-020	The spacecraft shall have a telemetry transmitter operating according to Recommendation SFCG 32-2R1	(Speretta et al., 2021)
COMMS-030	The spacecraft shall have a payload data transmitter operating according to Recommendation SFCG 32-2R1	(Speretta et al., 2021)
COMMS-040	Data links should implement a 20% margin on throughput	(Speretta et al., 2021)
COMMS-070 COMMS-070*	The communication system shall provide radio navigation support with a position accuracy of 1 km 3σ The communication system shall provide radio navigation support with a velocity accuracy of 1 cm/s 3σ	(Speretta et al., 2021)
COMMS-080	The satellite receiver shall be operational in all mission phases	(Speretta et al., 2021)
COMMS-090	The payload transmitter shall operate continuously for a continuous ranging session	(Speretta et al., 2021)
COMMS-100	The telecommand receiver shall support a minimum data rate to request at least 10% of the total satellite telemetry	(Speretta et al., 2021)
COMMS-110	The telemetry transmitter shall support transmitting 3 MB of telemetry in one day	(Speretta et al., 2021)
COMMS-120	The payload data transmitter shall be able to transmit 73.6 MB during one Nav&Eng cycle	(Speretta et al., 2021)
COMMS-130	The spacecraft shall be commandable for more than 95% of all spacecraft orientations in all operational scenarios	(Speretta et al., 2021)
COMMS-140	The spacecraft telemetry shall be receivable for more than 95% of all spacecraft orientations in all operational scenarios	(Speretta et al., 2021)
COMMS-150	The communication system shall have a maximum volume of 2U (CubeSat form factor)	(Speretta et al., 2021)
COMMS-160	The communication system shall have a wet mass of no more than 2 kg	(Speretta et al., 2021)
COMMS-170	The communication system shall require a power of no more than 25 W during transmission, and no more than 2 W during reception	(Speretta et al., 2021)
EPS-010	An available power of 29 W is provided by the EPS during operational phase	(Cervone et al., 2022)

2.2. Autonomous orbit determination methods

AOD is determining the states of one or more spacecraft(s) without ground-based interaction. The states must be determined with respect to a known reference frame, e.g, center of the Earth or the Solar System barycenter. AOD methods can be divided into three main concepts: optical, crosslink radiometric, and X-ray pulsar orbit determination.

2.2.1. Optical orbit determination

Optical orbit determination is performed by measuring target bodies using optical sensors to provide the position and velocity relative to the target bodies. This method can be categorized into celestial based and terrain based orbit determination. The chosen category depends on the scale of the environment of the satellite. Celestial optical orbit determination is commonly used for deep space missions in which celestial bodies are far away from the spacecraft or during the transfer phase of the mission. Terrain based optical orbit de-

termination could be executed when a satellite has enough coverage of the terrain of a celestial body. Both methods require optical sensors that are suited for the type of optical orbit determination.

Celestial optical orbit determination
Celestial optical orbit determination is executed by measuring the relative location of the celestial bodies. Combining the distance to multiple celestial bodies of which the location is known, one can estimate a satellite's position using the ephemerides of those celestial bodies (Adams and Peck, 2016). In order to get insight in the applicability on LUMIO, a comparable satellite regarding size and volume is Miniaturised Asteroid Remote Geophysical Observer (M-ARGO), which has a size of 12U and a wet mass of 22.6 kg. This satellite will travel through deep space to observer asteroids. M-ARGO will do its own orbit determination in cruise phase using Line-of-Sight (LOS)-vectors to visible planets and the Sun as shown in Figure 2.2. Regarding AOD, the goal is to achieve a accuracy of 1000 km (3σ) and a velocity accuracy of 1 m/s (Franzese et al., 2021).

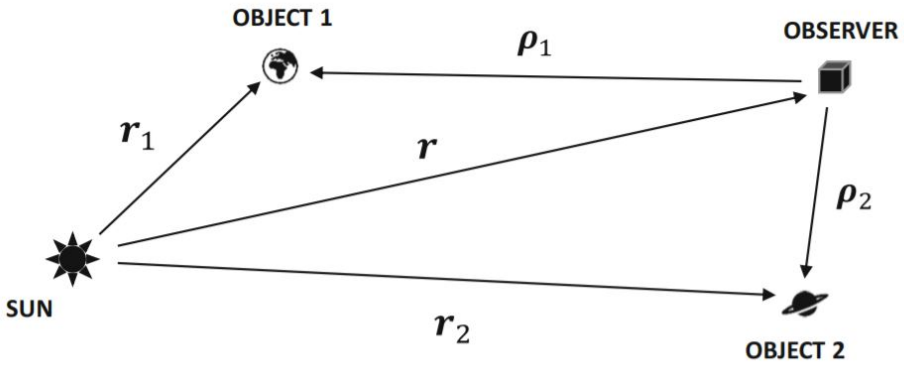


Figure 2.2: LOS vectors from S/C to measurement planets and the Sun (Franzese et al., 2021).

Terrain based optical orbit determination
Terrain based optical orbit determination is performed by measuring known landmarks or comparing terrain images in a body's full disk size with the actual body (Markley, 1984). The accuracy depends on the relative distance to the terrain and the quality of image processing. A smaller relative distance to the terrain and higher resolution images provide a more accurate state estimation (Franzese et al., 2021). For LUMIO, a study was carried out to investigate the possibilities of using LUMIOs payload 'the LUMIO-CAM' for terrain based orbit determination. Preliminary results showed a position error within 10 km (Franzese et al., 2018). However, the estimated measurement noise was 100 km. The authors stated that this noise could be brought back to 10 km, but how exactly this reduction would be possible was unclear.

2.2.2. Crosslink radiometric orbit determination
Crosslink radiometric orbit determination is an orbit determination concept that is solely based on inter-satellite measurements. LiAISON is a crosslink radiometric orbit determination concept in which the absolute states are obtained solely based on inter-satellite measurements, which is possible when one satellite has a unique orbit in terms of size, shape and orientation (Hill, 2007). A unique orbit exists in areas where asymmetry in the gravity field is dominant. Such areas exist locally in three-body problems such as the Earth-Moon system. If the trajectory is unique, the satellite can only follow one particular trajectory. If the trajectory is known, then the absolute states are also known at each time. However, the real trajectory of a satellite will most likely deviate slowly over time from the simulated trajectory due to model errors and unpredicted perturbations. When using a second satellite, however, inter-satellite measurements of the relative position and velocity can be performed. These measurements will differ slightly from the measurements computed using the simulated states. Using both measurements in an estimation filter, e.g., an Extended Kalman Filter (EKF), the absolute states of both satellites can be estimated along with their variance. The accuracy of state estimation depends on the performance of the radio systems of the participating satellites and the visibility between these satellites.

LiAISON based orbit determination has not yet been performed in real cases. Several theoretical cases, e.g., Gao et al. (2016), Hill (2007), show a position and velocity error of less than 1000 m. One satellite called Cislunar Autonomous Positioning System Technology, Operations, and Navigation Experiment (CAPSTONE) is currently on its way to the Moon and will perform LiAISON based orbit determination. CAPSTONE is a 12U CubeSat that will explore the requirements to operate in a Near-Rectilinear Halo Orbit (NRHO) as preparation for NASAs Lunar Gateway and will demonstrate LiAISON based orbit determination with NASAs Lunar Reconnaissance Orbiter (LRO). Simulations of CAPSTONE provided a 3-D position and velocity accuracy of 156 m 3σ and 3.32 cm/s 3σ after three times passing its perilune (Thompson et al., 2022).

A simulation study based on LiAISON based orbit determination of LUMIO and the Lunar Pathfinder is performed in Turan et al. (2022). This study took the radio systems of both satellites into account, and the dynamical model was based on the ephemerides containing the Earth, Moon and Sun as point masses, additionally SRP was applied. Results showed estimated 3-D position and velocity errors lower than 500 m 1σ and 2 mm/s 1σ (Turan et al., 2022).

2.2.3. X-ray pulsar orbit determination

Pulsar navigation is a concept where X-ray pulsars are received and used to determine the satellite's position. The signals originate from fast spinning stable neutron stars of which the positions are known. These signals are pseudo random, so the patterns can be recognized. The recognized patterns are compared with predicted patterns on a certain time and position. The position can be estimated from the difference between actual measurements and predictions.

X-ray pulsar orbit determination can be performed in the entire solar system and most likely also outside the solar system. The ephemerides of the pulsars are known, so spacecraft position relative to pulsars can be computed into an absolute position. X-ray pulsar position determination can be done in a similar fashion as optical navigation, but measurements are extracted from X-ray pulsars instead of optical observations. CubeX will demonstrate whether autonomous orbit determination is possible with X-ray pulsar navigation using the X-ray telescope X-ray imaging spectrometer (XIS) and a Solar X-ray Monitor (SXM) (Kashyap et al., 2020, Stupl et al., 2018). CubeX will orbit the Moon and use four targets as observations to determine its position as shown in Figure 2.3. The telescope is 10x10x60 cm (1x1x6U) with a mass of 5.5 kg. The required power is 8.6 W on average, with peak demands of 12.4 W Stupl et al. (2018). The goal of CubeX is to achieve at least 20 km position accuracy.

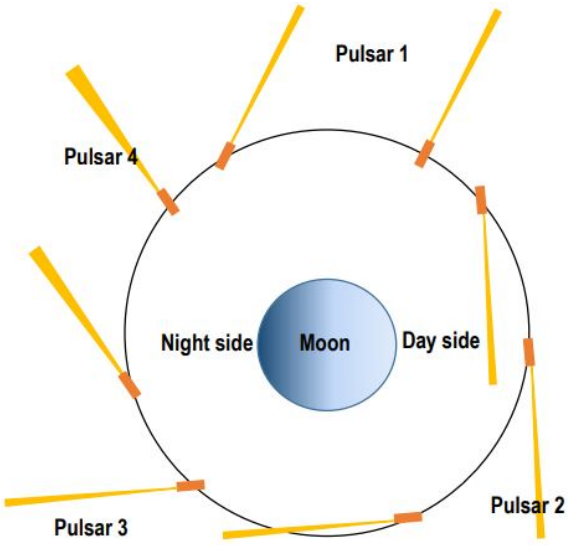


Figure 2.3: Simplistic view of CubeX orbiting the Moon including where pulsar measurements are taken Stupl and Harvard (2018).

2.3. Trade-off

Based on the findings of the four autonomous orbit determination concepts, a trade-off was executed to determine which autonomous orbit determination method would suit a LUMIO mission best. The trade-off is based on four criteria: size, power, position accuracy and velocity accuracy. Since LUMIO has a maximum mass of 24 kg (OVRSYS-001 and 002), the weight of the equipment for AOD has a strict maximum. CubeSats also have limited power due to their restricted size. Low power systems are required to fit inside the satellite's power budget. For the position and velocity accuracy requirements, COMMS-070 and COMMS-070* are used. In what follows, all four AOD methods are evaluated on these four criteria, and a trade-off matrix is provided.

- **Size**
  - *Celestial optical orbit determination*:
    M-ARGO is a 12U CubeSat performing celestial optical orbit determination, so celestial optical navigation instruments should complies for this criteria.
  - *Terrain based optical orbit determination*:
    Terrain based optical orbit determination is often done using the payload of the satellite. For example, it could be possible that LUMIO uses its existing payload for imaging and orbit determination (Cervone et al., 2020, Franzese et al., 2018). Full-disk optical orbit determination complies with this requirement.
  - *LiAISON based orbit determination*:
    LiAISON based orbit determination uses the existing communication system, so complies with size requirements. However, the position and velocity accuracy depends on the performance of inter-satellite measurements, so link budget analyses are required to determine what radio configurations are required.
  - *X-ray pulsar orbit determination*:
    The smallest instrument available at the moment is the CubeX X-ray telescope. This is a 1x1x6U telescope which exceeds the maximum size. So X-ray pulsar orbit determination does not meet the size requirement.
- **Required power**
  - *Celestial optical orbit determination*:
    M-ARGO is a 12U CubeSat of 22.6 kg (Franzese et al., 2021) that will perform autonomous celestial optical orbit determination. Required input power for the instrument is unknown. However, if M-ARGO has enough power for celestial optical orbit determination, then it is likely that LUMIO will have as well. Therefore, celestial optical orbit determination is probably compliant with this criteria.
  - *Terrain based optical orbit determination*:
    This method is considered when the payload can be used for optical navigation. In that case, no additional power budget is necessary for the autonomous orbit determination system, so this method complies with the power requirement.
  - *LiAISON based orbit determination*:
    Required power cannot exceed the required power of the communication system, which is 25 W during transmission and 2 W during receiving. The exact link budget needs to be determined for this AOD concept, but it is expected to comply.
  - *X-ray pulsar orbit determination*:
    According to Stupl and Harvard (2018) the system requires 8.6 W to operate, which is compliant with EPS-010.
- **Position accuracy**
  - *Celestial optical orbit determination*:
    Celestial optical navigation has a position accuracy of several 100s to 1000s of km. So this method is not accurate enough to comply with COMMS-070.

  - *Terrain based optical orbit determination*:
    According to Franzese et al. (2018) a position accuracy of an error within 10 km could be obtained. This is a large improvement compared to celestial optical orbit determination, but does still not comply with COMMS-070.
  - *LiAISON based orbit determination*:
    Simulated results of CAPSTONE show a position and velocity accuracy meeting COMMs-070. Turan et al. (2022), a study focusing on LUMIO show simulated position error of 500 m 1σ, meaning a 3σ position error of 1500 m was obtained, which is slightly exceeding COMMS-070.
  - *X-ray pulsar orbit determination*:
    The goal of CubeX is to achieve less than 20 km position accuracy (Stupl and Harvard, 2018). The real performance is unknown of the CubeX X-ray telescope, but the goal is not compliant with the COMMS-070 mission requirement.

- **Velocity accuracy**
  - *celestial optical orbit determination*:
    Celestial optical navigation could have a velocity accuracy within 1 m/s, but previous missions showed that that is optimistic to accomplish (Franzese et al., 2021). Therefore, celestial optical orbit determination does not comply with COMMS-070* mission requirement.
  - *Terrain based optical orbit determination*:
    According to Franzese et al. (2018), velocity accuracy is within 50 cm/s using an EKF. This exceeds the COMMS-070* mission requirement.
  - *LiAISON based determination*:
    Both simulated results of CAPSTONE and LUMIO show a velocity error that meets COMMS-070* mission requirement.
  - *X-ray pulsar orbit determination*:
    Unknown.

The trade-off matrix based on the findings as stated above is given in Table 2.2.

Table 2.2: Trade-off matrix of the considered autonomous orbit determination methods.

	Size	**Required Power**	**Position accuracy**	**Velocity accuracy**
celestial optical orbit determination	Celestial optical instrument fits in M-ARGO a 12U CubeSat	For exact numbers, more research is necessary regarding power budget	Position accuracy >100 km	Velocity accuracy 1 m/s if optimistic
Full-disk optical orbit determination	If payload is used	Depending on payload power consumption	Position accuracy >1 km	Velocity accuracy <50 cm/s
LiAISON based orbit determination	If uses existing communication system, but is link budget dependent	Depends on communication link budget. Link budget analysis required	Simulated position accuracy <1000 m	Simulated velocity accuracy <2 mm/s
X-ray pulsar orbit determination	Smallest instrument is 6U	Requires 8.4 W	CubeX goal <20 km	Unknown

As shown in Table 2.2, LiAISON based orbit determination seems to be the most promising method for LUMIO. All requirements are met if the required radio parameters stay within the mission requirements as stated in Table 2.1. Celestial optical orbit determination could be a good solution for navigation for interplanetary transfers, but is less suitable for missions that require higher position and velocity accuracy. Terrain based optical orbit determination is a good solution when observing an area and in missions where position accuracy may exceed kilometers. X-ray pulsar needs more research and is not yet applicable for for 12U CubeSats due to size restrictions. Based on this trade-off, this research will explore the possibilities of LiAISON based orbit determination for the LUMIO mission.

2.4. LiAISON based orbit determination

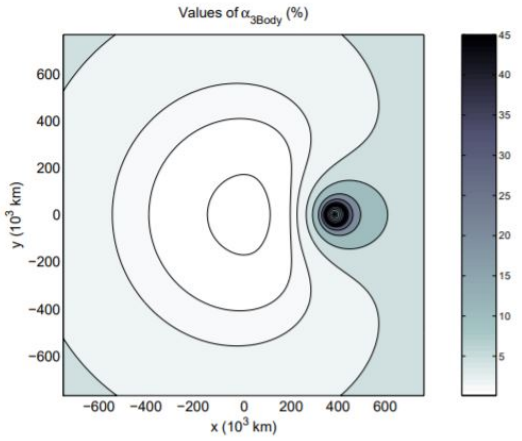
This section provides a more in-depth literature review of LiAISON based orbit determination. Two major aspects of the method are the need for (local) asymmetry in gravity field and the type of inter-satellite ranging. Although LiAISON based orbit determination can be performed with an infinite number of satellites, this section focuses on autonomous orbit determination using only two satellites.

The level of asymmetry in gravity field must be significant enough to provide a unique trajectory. This trajectory exists when the inequality shown in Equation 2.1 holds, where θ is a constant 3 x 3 rotation matrix, Δr is a 3 x 1 constant translation and $\mathbf{X}_n$ is described as in Equation 2.2 Hill (2007). If this inequality holds, it means that there are no possible rotations or translations from an arbitrary state vector to the unique state vector.

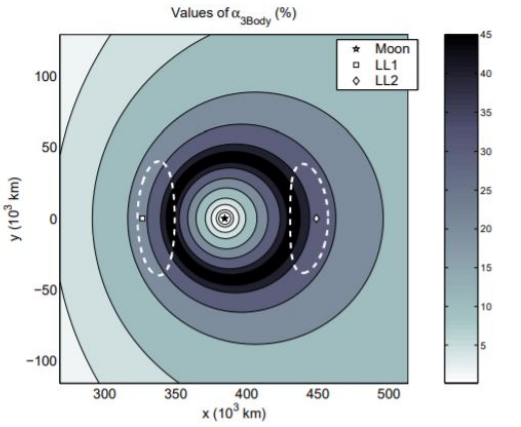
$$\mathbf{X}_2(t) \neq \begin{bmatrix} \theta & \emptyset_{3\times3} \\ \emptyset_{3\times3} & \theta \end{bmatrix} \mathbf{X}_1(t) + \begin{bmatrix} \Delta r \\ [\emptyset]_{3\times1} \end{bmatrix} \tag{2.1}$$

$$\mathbf{X}_n(t) = \begin{bmatrix} x_n(t) & y_n(t) & z_n(t) & \dot{x}_n(t) & \dot{y}_n(t) & \dot{z}_n(t) \end{bmatrix}^T \tag{2.2}$$

During the literature study, is was found that third body perturbations are required to obtain enough asymmetry in the gravity field. Hill (2007) and Qin et al. (2019) discovered that not all state elements could be determined if LiAISON based orbit determination is applied in more symmetric gravity fields. One can determine up to 10 of the 12 state elements when including J_2 perturbation for two LEO orbiters (Hill, 2007) and up to 9 state elements without J_2 (Qin et al., 2019). The Earth-Moon system shows great asymmetry in the gravity field, especially near the Earth-Moon Lagrange 1 and 2 areas (Hill, 2007). The level of asymmetry in gravity field of the Earth-Moon system is provided in Figure 2.4a, which shows that there are certain areas near the Moon that contain high asymmetry. A detailed overview of the level of asymmetry in gravity around the Moon is provided in Figure 2.4b. A sphere around the Moon at the level of the Lagrange points L_1 and L_2 is highly influenced by the third-body perturbation due to Earth.



(a) α distribution of the Earth-Moon three-body problem in the xy-plane Hill (2007).



(b) α distribution near the Moon including the L_1 and L_2 Lagrange points along with a halo orbit in the xy-plane Hill (2007).

Such areas near Earth-Moon L_1 and Earth-Moon L_2 could be interesting areas to perform LiAISON based orbit determination. Since LUMIO orbits a halo orbit near Earth-Moon L_2, the orbital characteristics of LUMIO are used in this research.

2.5. Telemetry ranging

The goal of ranging is to determine the distance between the transmitting and receiving satellites by measuring the time an electromagnetic signal requires to travel between the two satellites (Andrews et al., 2010). To perform the inter-satellite range observations, telemetry based ranging is considered in this research. Telemetry ranging is an alternative to conventional two-way ranging in which the downlink signal is replaced with an asynchronous telemetry signal. It is not dependent on a dedicated ranging signal and enables multiplexed data in one telemetry stream. Using telemetry ranging instead of conventional ranging no separate downlink ranging signal is required, which leads to a reduction in complexity of the radio system, reduction

of required transmission power and an improvement of the link efficiency Andrews et al. (2010).

When using ranging operations for orbit determination in addition to scientific data transfer, the signal power will reduce if both signals are send at the same time (Turan et al., 2022). Additionally, the radio system increase in complexity. In case ranging operations for both goals are operated at different epochs, less time is available for both operations. Therefore, it would be efficient if the same downlink ranging operations could be used for both transmitting data and timing information for determining the inter-satellite range.

Including clock data or a codeword in the telemetry stream, the round-trip light time can be determined from a two-way ranging operation, which is the time required for a signal to travel from the transmitter to the receiver and back to the transmitter. From the round-trip light time and the speed of light, the inter-satellite distance (inter-satellite range) can be computed. In general, a processing delay occur on the receivers satellite, which is the delay between receiving the downlink signal and transmitting transmitting the uplink signal. The delay must be subtracted from the round-trip light time in order to calculate the inter-satellite distance. Furthermore, clock uncertainties and mismatches between the satellite could occur as well, but are not considered in this research.

Telemetry ranging is possible if a time tag or codeword is added to the transmitted scientific data. The receiving satellite receives the scientific data and the associated time tag, and records the time the satellite receives the data. This receiving satellite can then transmit the receiving time tag and the start of transmission time tag back to the initial transmitter. The initial transmitter knows the initial time tag, receives the other two time tags, and records the time at which these tags were received. From these four times, the round-trip light time, and therefore the inter-satellite range, can be computed. A schematic overview is provided in Figure 2.5.

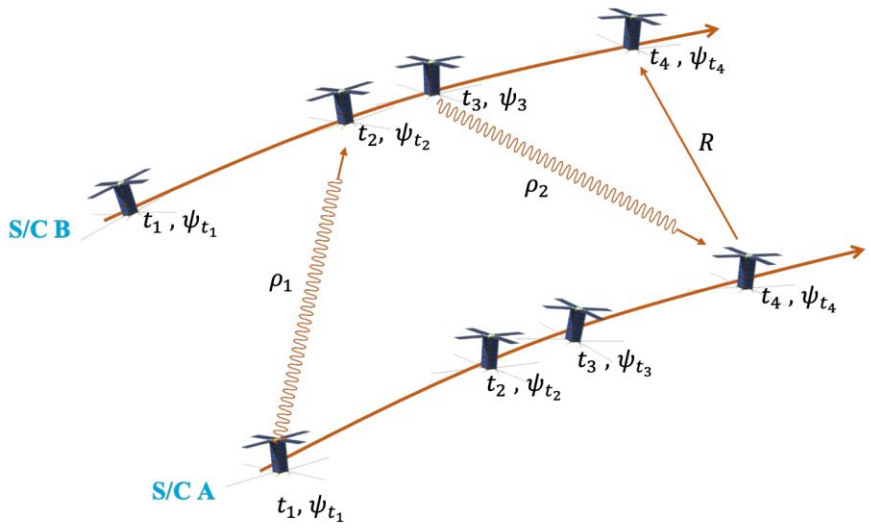


Figure 2.5: Obtaining the time stamps using telemetry ranging (Turan et al., 2022).

2.6. Lunar Pathfinder

LiAISON based orbit determination requires at least two satellites in order to perform inter-satellite ranging. The second satellite that is considered for this research is ESAs Lunar Pathfinder. This satellite is due to launch in 2023 and will operate for 8 years (Harlé and Friend, 2020). The Lunar Pathfinder will orbit in an Elliptic Lunar Frozen Orbit (ELFO) in such a way that the satellite provides full coverage of the Lunar surface and especially long coverage of the South Polar regions. Additionally, the orbit is designed in order to provide long visibility in Earth direction for ground-based communication. One of the Pathfinders main mission purposes is providing communication and navigation services to future Cislunar space missions (Harlé and Friend, 2020) and is therefore equipped with a strong communication system.

Figure 2.6: Impression of the Lunar Pathfinder in Cislunar space, *Illustration by* ESA (2021).

ESAs Lunar Pathfinder is equipped with two simultaneously usable radio channels for Cislunar communication. One channel in S-band, forward 2025 - 2110 MHz and backwards 2200 - 2290 MHz, and one channel in Ultra-High Frequency (UHF), forward 390 - 405 MHz and backwards 435 - 450 MHz Harlé and Friend (2020). The satellite has a wet mass of 280 kg eoPortal (2022) and a 3 m^2 reference area Scotti et al. (2022). SRP is modelled as a cannonball and the reflectivity of the satellite is 1.8.

The Keplerian state elements and the Cartesian state elements of the initial states of the Lunar Pathfinder are provided in Table 2.3. The state elements are with respect to the Moon Fixed reference frame (Scotti et al., 2022). In order to provide the Earth-centered J2000 states of the Lunar Pathfinder, one adds the Earth-centered J2000 states of the Moon to the Moon Fixed Cartesian states. The Earth-centered J2000 Cartesian states of the Moon depend on time and is therefore not given in Table 2.3.

Table 2.3: Keplerian and Cartesian state elements of ELO with respect to the Moon based on (Harlé and Friend, 2020, Scotti et al., 2022)

Keplerian element	**Symbol**	**Value**	**Cartesian element**	**Value**
Semi-major axis	a	$5.737 \cdot 10^6$ [m]	x	$-7.87011 \cdot 10^5$ [m]
Eccentricity	e	0.61 [-]	y	$-1.3970 \cdot 10^6$ [m]
Inclination	i	57.82 [°]	z	$1.72782 \cdot 10^6$ [m]
Argument of periapsis	ω	90 [°]	$\dot{x}$	$6.98341 \cdot 10^2$ [m/s]
Longitude of ascending node	Ω	61.552 [°]	$\dot{y}$	$-1.60431 \cdot 10^3$ [m/s]
True anomaly	θ	30 [°]	$\dot{z}$	$-4.93689 \cdot 10^2$ [m/s]

3

Method

This chapter describes the methods used to determine the position velocity accuracy resulting from LiAISON-based AOD with different radio system parameters. The performance of the autonomous orbit determination model is dependent on the magnitude of the estimated two-way telemetry ranging error. To calculate this error, a link budget analysis is performed in which the bit rate and antenna gain are free parameters. The estimated ranging error can then be used as input in the AOD model, which consists of dynamic, measurement and estimation sub-models. First, the existing radio parameters of the Lunar Pathfinder and LUMIO were used to obtain a baseline for the position and velocity accuracy of LiAISON-based autonomous orbit determination. Next, the radio parameters (i.e., bit rate and antenna gain) were varied in an iterative process to find which parameters values would result in a position and velocity accuracy according LUMIOs requirements. An overview of the simulation model is provided in Figure 3.1.

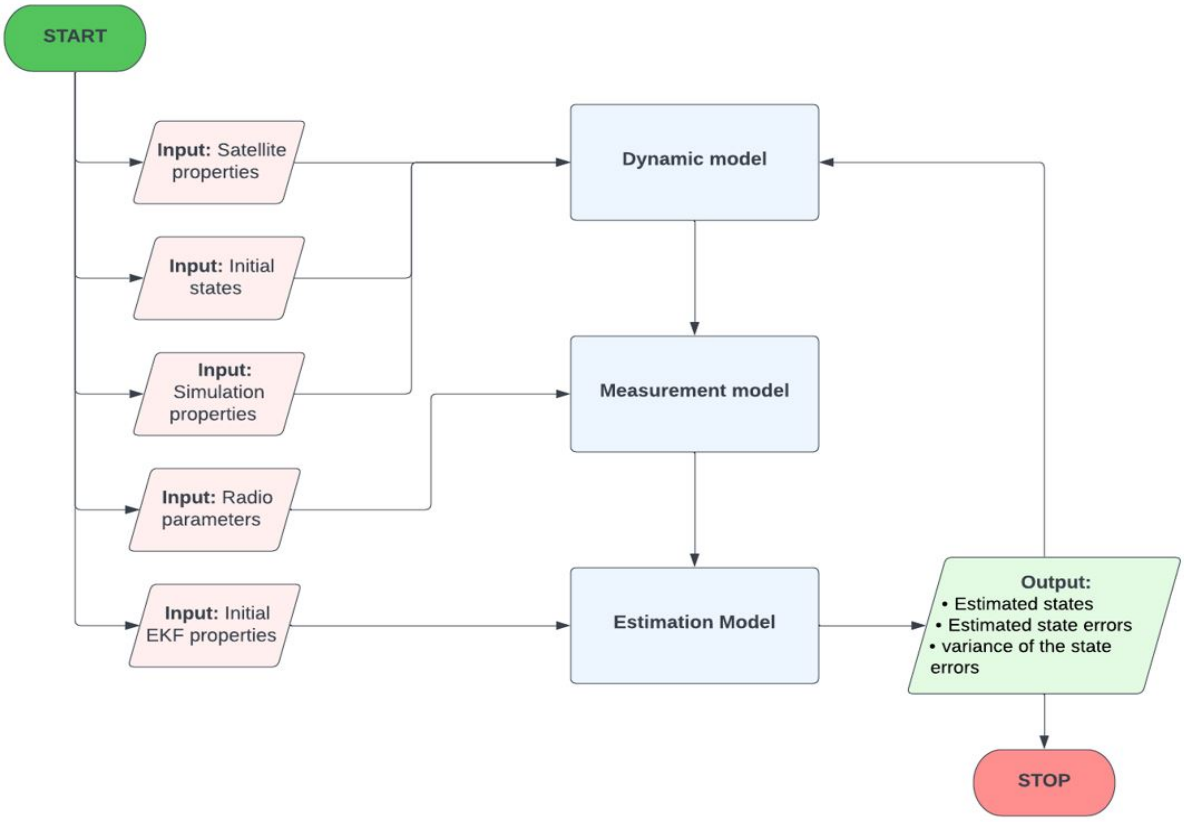


Figure 3.1: Schematic overview of the full autonomous orbit determination model.

3.1. Software and data resources

The AOD model is written in the Python language, which is an open-source programming language. Python is an object orientated programming language, where objects can be defined that contain attributes and the routines to manipulate them. The code is written in Pycharm (community version), an Integrated Development Environment (IDE) specific for Python language. Package and environment handling is done with (mini)Conda, which is a virtual environment manager and package manager. The applied virtual environment is tudatpy. Tudat is a platform that provides a powerful toolkit regarding astrodynamics in both C++ (tudat) and Python (tudatpy). A README of the project and the required packages list can be found in Appendix A and Appendix B.

In addition, a dataset that contains the Earth-centered J2000 states of LUMIO was used. This dataset contains the planned trajectory of LUMIO and also accounts for the stationkeeping manoeuvres. From this dataset, only the initial state of LUMIO was used. The initial state in this research is defined as the first state on the 21^{st} of March 2024, which marks the start of the operational phase. The Earth-centered J2000 initial states are shown in Table 3.1.

Table 3.1: Initial state elements of LUMIO at 00:00:000 the 21^{th} of March 2024 in an Earth-centered J2000 reference frame.

Cartesian state element	Value
x	-3.10537998e+08 m
y	2.49423157e+08 m
z	1.74937757e+08 m
$\dot{x}$	-9.93171842e+02 m/s
$\dot{y}$	-7.66408514e+02 m/s
$\dot{z}$	-5.25173280e+02 m/s

3.2. Link budget and ranging performance

A link budget analysis is performed to get insight in the required radio parameters for a space mission. The link budget analysis takes the gains and losses of the inter-satellite link into account. For space missions, link budgets are calculated for all mission phases and satellite's velocities, attitudes and orientations in order to gain as much information as possible about which radio system meets the mission requirements Speretta (2022). In this research, the link budget will only be analyzed for the orbital scenario in which the inter-satellite distance is largest, because that is the point where highest signal losses in free space occur Board (2004).

To compute a baseline, the initial radio parameters for the Lunar Pathfinder are extracted from Harlé and Friend (2020), and the initial parameters for LUMIO from Turan et al. (2022) and the currently considered Anywaves Telemetry, Tracking & Command (TT&C) antenna Anywaves (2020). To start, the link budget analysis is performed with one antenna on LUMIO which provides a nominal (bore-sight) antenna gain of 6.5 dB Anywaves (2020). This link budget analysis focuses on calculating the required bit rates while keeping the mission requirements in mind. Next, LUMIOs antenna gain is varied in the link budget analysis by considering different numbers of antennas that form an antenna array. Increasing the amount of antennas on LUMIO by turning a single antenna into antenna arrays, will increase the downlink gain. Assuming there are no scaling losses, the antenna gain increases with 3 dB when the amount of antennas is doubled into an array. A higher gain leads to higher possible bit rates and therefore a lower two-way telemetry ranging error. Performing a link budged analyses for different antenna configurations on LUMIO will reveal the relation between antenna gain an bit rate.

The output of a link budget analysis are the possible bit rate and the signal-to-noise ratio. The signal-to-noise ratio must be greater than the required receiving power for the receiving satellite to obtain the signal. In addition, a 3 dB margin in receiving signal is taken into account to ensure link closure. From the bit rate and the signal-to-noise ratio, one can estimate the two-way telemetry ranging error. The round-trip light time can be determined from four time tags that represents the time of receiving and transmitting of a two-way ranging operation as described in section 2.5. Using the time-derived ranging error equation in telemetry

symbols (Turan et al., 2022), a one way ranging error is calculated using

$$\sigma_{\rho_{TD}} = \frac{4cT_{sd}^2}{\pi T_l 10^{\frac{E_S/N_0 - margin}{10}}} \tag{3.1}$$

in meter. In Equation 3.1, $\sigma_{\rho_{TD}}$ is the one-way time-derived ranging error, c is the speed of light, T_{sd} is the symbol duration, T_i is the integration time, and E_S/N_0 is the energy per symbol to noise power spectral density or symbol to noise ratio. The estimated error for each antenna configuration is used as input for the AOD model.

3.3. Autonomous orbit determination model

The autonomous orbit determination model consists of three sub-models that together provide the estimated position and velocity error over the simulated time. The dynamic model provides the 'true' states of the trajectories of the satellites. The measurement model provides the inter-satellite measurements based on the 'true' states and the two-way telemetry ranging error that is determined from link budget parameters. The estimation model uses the initial states and nominal measurements to estimate the position and velocity of the satellites using an estimation filter. By subtracting the estimated states from the true states, the states errors are obtained. Note that the model ignores light-time corrections, which are the displacements of a body's apparent position with respect to its true position. These displacements are negligible since the propagation time is rather small with respect to the relative velocity. Below, a short description of the sub-models is provided. A more elaborate description of these sub-models is given in the following chapters.

The satellites are propagated in the dynamic model, where it is assumed that the accelerations acting on the satellites are solely dependent on its time and position and no stationkeeping is performed. The accelerations acting on the satellites are exerted from the masses of celestial bodies and SRP of the Sun. Since both satellites do not orbit in an atmosphere, there is no acceleration due to drag. Celestial properties are provided by NASAs Navigation and Ancillary Information Facility (NAIF), which developed a toolkit called 'SPICE kernels'. This toolkit is used to obtain the predefined ephemerides of the celestial bodies. Masses of solar system bodies are obtained using DE431 and positions and velocities in time are provided by DE440 ephemerides (Tudat, 2022b). The dynamic model includes all Solar System planets, the Sun, including SRP, and spherical harmonic gravity from both the Earth and the Moon as perturbations from the mentioned ephemerides.

The measurement model takes the nominal states from the dynamic model and the estimated two-way telemetry ranging error as input to provide the nominal measurements as output. The nominal measurements are required as input for the estimation model. Ranging measurements are the absolute distance between the satellites at that epoch plus a certain ranging error that is added to the absolute inter-satellite distance as normal distributed noise. Besides the nominal measurements, a visibility analysis is performed using the nominal states to determine in what epochs the Moon occults the LOS-vector between both satellites. The output of the measurement model are the nominal measurements and whether there is a direct line of sight between the two satellites at the time steps.

Additionally, an estimated measurement bias can be added to the ranging measurements. Measurement bias grows over time and is often not constant. When simulating long periods, e.g., months or years, the effect of measurement bias should be examined. In this research, bias is not considered. It was found that up to 10 m measurement bias, no significant difference occur in the simulations. Above a measurement bias of 10 m, the 3σ uncertainties of the state elements starts to enlarge. At approximately a measurement bias of 20 m, the state element errors start to grow as well. Measurement bias should be implemented when the simulation time is longer and/or the estimated bias exceeds 10 m. Appendix D shows the simulation taking into account measurement bias and without measurement bias.

The estimation model requires the initial states and the nominal measurements as input for the estimation filter. The filter is an EKF with a variable state noise compensation matrix. Theoretically, an EKF performs a time update and a measurement update each iteration (Montenbruck and Gill, 2001). However, in reality, this is not always possible, e.g., due to visibility issues. Therefore the execution of the measurement update

is dependent on the inter-satellite visibility. This EKF is designed in such a way that the time update is executed in a fixed time step and that the measurement update is executed based on the visibility. Therefore, the time update step and measurement update step can differ in time interval. The estimation filter's output are the estimated states and the variance per state element. The variance per state element is acquired from the diagonal of the covariance matrix and the standard deviation is calculated as the square root of the variance. The estimated states are subtracted from the nominal states, which provides the estimated state errors.

3.4. Results

The ranging error for 1, 2, 4, 8 and 16 antennas is determined based on the possible bit rate and antenna gain. After the AOD model was built, simulations ran for the five different antenna configurations in exactly the same environment, time interval and time step given the ranging error as only variable. First, five simulations were carried out using the ranging errors resulting from the chosen number of antennas. Figures of the estimated position and velocity errors along with their 3σ probability are provided to visualize the work done by the AOD model. Second, the epoch is determined from which time the state elements converges. The average 3-D position and velocity errors are determined from that epoch to the simulation end time based on the 3σ state element errors. These errors are computed by taking the norm of $+3\sigma_x, +3\sigma_y$ and $+3\sigma_z$ and the norm of $+3\sigma_{\dot{x}}, +3\sigma_{\dot{y}}$ and $+3\sigma_{\dot{z}}$ on each epoch per satellite. The average estimated 3-D position and velocity errors per satellite are determined by summing those lists and dividing it by the amount of elements. Finally, the 3-D position and velocity errors per antenna configuration are compared with the position and velocity requirements of LUMIO to answer the main research question.

Note that throughout the report the satellite representing LUMIO is called EML2O and the satellite representing the Lunar Pathfinder is called ELO.

3.5. Verification and validation

The work executed in this thesis is validated against known references. Using tudatpy, the trajectories of the satellites are obtained. Tudatpy functions are validated and verified via unit testing at compile time Tudat (2022a), but these function are to be defined in a right way. Therefore, the trajectory of EML2O is verified against the trajectory of LUMIO over the simulation span of 14 days from March the 21^{st} to April the $4th$ of 2024. The trajectory of LUMIO is provided by Politecnico di Milano and verified by ESA. The difference in absolute position error is provided in Figure 3.2. Especially, in the last days of simulation, the position error starts to grow exponentially to a 3-D position error of 90.17 km. It is known that during the this time interval, LUMIO is in its science cycle and therefore no stationkeeping is performed until the $4th$ of April. The difference in states is due to the use of a different dynamic model, because the dynamic model settings are unknown for the trajectory of LUMIO. Therefore, it cannot be said whether the states from Politecnico di Milano or the states of EML2O obtained in this research provide more accuracy. However, a model error of 90.17 km compared to the distance between the Moon and EML2O, which varies between approximately 35 000 km to 85 000 km is negligible. Therefore, it is validated that the orbit of EML2O is propagated in a correct way.

For ELO, such a validation trajectory was not available. Therefore, an acceleration breakdown was executed in subsection 4.3.1 and compared to the acceleration breakdown for the Lunar Pathfinder performed in Scotti et al. (2022) where similar magnitudes of the acceleration distribution was found. Additionally, the maximum and minimum inter-satellite distance during the simulation complies with the inter-satellite distances of LUMIO and the Lunar Pathfinder in Speretta et al. (2021) and Moon-centered initial conditions of ELO complies with the Lunar Pathfinder initial conditions in Harlé and Friend (2020), Scotti et al. (2022), Turan et al. (2022). ELO is propagated in the same settings as EML2O and its trajectory is validated on the basis of the points mentioned above.

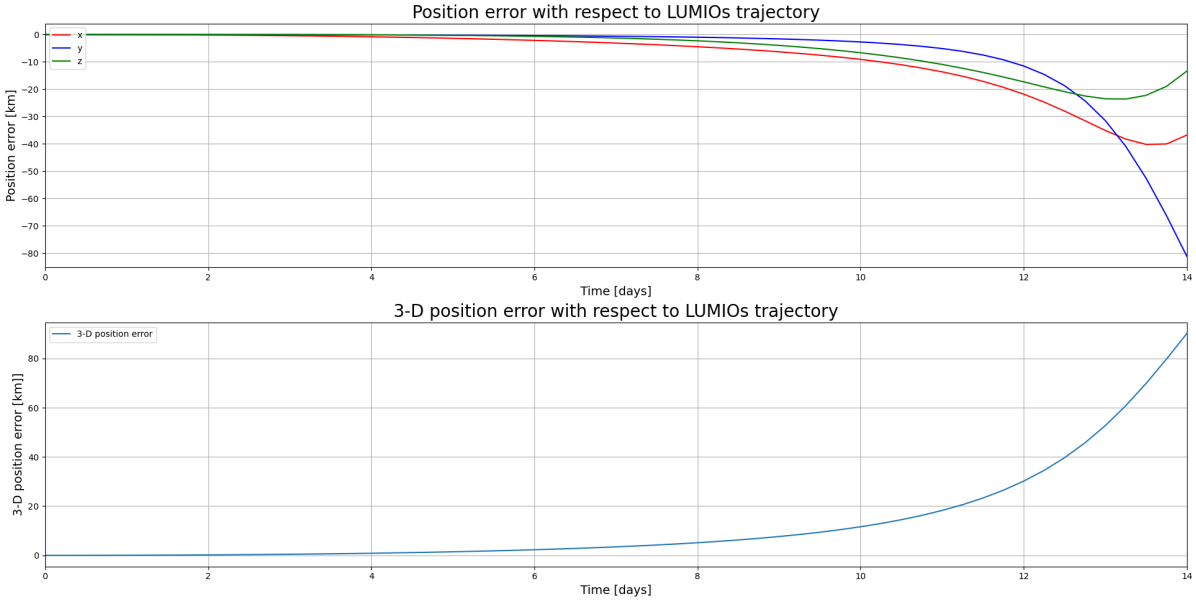


Figure 3.2: Position error of the dynamic model with respect to LUMIOs planned trajectory.

For data handling, e.g., saving and loading, and for self defined functions, unit tests were written. The script of unit tests is provided as Appendix C and its output is shown in Figure 3.3.

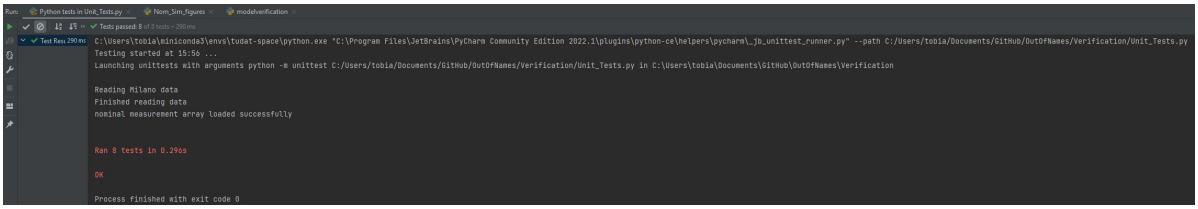


Figure 3.3: Output of unit tests.

Initial radio parameters were based on existing and considered parameters for both LUMIO and the Lunar Pathfinder (Harlé and Friend, 2020, Speretta et al., 2021, Turan et al., 2022). Known link budget equations are used to determine the unknown parameters and values remained within range between the possible outer-boundaries.

The performance of the autonomous orbit determination is validated with CAPSTONEs simulated performance. Although CAPSTONE orbits in a NRHO and communicates with NASAs LRO, it is one of the few references available where a high fidelity environment is taken into account in Cislunar space. CAPSTONE has an antenna gain of 20.5 dBi, which is 2 dBi higher than the maximum gain considered in this research. Furthermore, transmission power is equal, operates at similar frequencies and has an orbit of 6.5 days. Table 4 in Thompson et al. (2022) shows that a 3-D 3σ position uncertainty of 156 m can be obtained after it passes its perilune for the third time. This result is quite similar in value to the results obtained for EML2O with an antenna gain of 18.5 dBi. Results may differ due to different orbit orientations. For example, the distribution and magnitude of the uncertainty of EML2O differs from ELO. Additionally, the results of the autonomous orbit determination model are validated against Turan et al. (2022), in which similar distributions of the estimated errors and uncertainties are shown. Turan et al. (2022) performed comparable simulations with LUMIO and the Lunar Pathfinder, but initial conditions and dynamic model differ.

4

Dynamic model

The dynamic model describes the position and velocity in time in a specified environment. Both the nominal trajectories of EML2O and ELO are the output of the dynamic model. As input, the model requires the initial states in Cartesian form of both EML2O and ELO and a simulation span in real time. This chapter describes the settings of the dynamic model. The simulator that outputs the nominal trajectories consists of three input settings: environment, propagation, and integration settings.

4.1. Circular restricted three-body problem

The CRTBP is part of the three-body problem family. The three-body problem states: 'What are the motions of three given bodies (point masses) moving under the influence of their mutual gravitational attraction?' (Wakker, 2015). The circular restricted three-body problem is a special form of the three-body problem and uses the following assumptions (Wakker, 2015):

- The mass of two bodies (P1 and P2) are much larger than the mass of the third body (P3), which causes the third body to move due to the gravity field of the bodies of P1 and P2. The attraction exerted from P3 to P1 and P2 is negligible.
- The two larger bodies, P1 and P2, move in circular orbits about their mutual barycenter.
- The masses of the two larger bodies are known, making the CRTBP a problem to determine the motion of the third body.

During this research, CRTBP is investigated along with the transformation of the normalized, rotating barycentric coordinates to Earth-centered, non-rotating coordinates (International Celestial Reference System and Frame (ICRF)). The CRTBP is not used in further calculations, since the trajectories propagated in the CRTBP and transformed to ICRF only consists of an environment with two large point masses (in this case the Earth and Moon) and the satellites. Such an environment is not accurate enough to provide realistic states of the satellites, because perturbations, e.g., spherical harmonic gravity of the Moon and Earth and other Solar System bodies, are not taken into account in the acceleration function. Since the acceleration function acting on the satellite differ, the trajectory will deviate over time. However, the CRTBP does provide first estimations and insight in the motion of a satellite (Turan et al., 2022). A brief description of the transformation from CRTBP to ICRF reference frame is provided in Appendix E.

4.2. High fidelity model

The high fidelity model based on ephemerides is a dynamic model in which more perturbations are taken into account to give a more accurate simulation of the trajectories of the satellites. In order to obtain high fidelity nominal trajectories, one wants to take into account all Solar System planets, the Sun, and the Moon and include spherical harmonic gravity to all bodies. However, due to the environment, this research considers only spherical harmonics for both the Earth and the Moon. For Earth's harmonic gravity field, the GOCO05c gravity model is provided that goes up to an order and degree of 720 (Tudat, 2022b). For the Moon's harmonic gravity field, the gggrx1200 model is provided which goes up to an order and degree of 1199 (Tudat, 2022b).

A dynamic model is required that includes Solar System body characteristics, positions and velocities. Besides the position and velocity, their physical characteristics are obtained using ephemerides and loaded in by tudatpy (Tudat, 2022b). Furthermore, for the masses of the Sun, Moon and Solar System planets DE431 ephemeris is applied and for the Solar System body motions (position and velocity) DE440 ephemerides is applied (Tudat, 2022b).

The environment, propagation, and integration inputs are fed to the dynamic simulator that provides an output consisting of the trajectory over the simulated time and the chosen output, e.g., Keplerian state elements, over the simulated time. These aspects of the dynamic model are discussed after the simulation time setup, in which the real time span is defined for the simulations.

4.2.1. Simulation setup

The simulation setup describes the initial, final time, and fixed time step required for the numerical integration. The initial simulation time is based on the planned time that LUMIO starts its operational phase in orbit: 00:00:000 the 21^{th} of March 2024. The simulation duration is fixed to 14 days for two reasons. First, the period of the halo orbit is approximately 14 days, so approximately one full halo orbit period for EML2O is simulated. Second, it is assumed that Stationkeeping Manoeuvre (SKM) are performed every 7 to 14 days for EML2O in order to repeat the nominal orbit over the full operational phase duration of 1 year. This research does not take into account station-keeping and therefore, the trajectory of EML2O deviates slowly from the planned trajectory, computed by Politecnico di Milano. The simulation end time is the simulation start time plus the simulation time of 14 days, meaning the simulation end time is on April 4^{th} 2024.

In real-time astrodynamics, dates are impractical and Modified Julian Time (MJD) and Ephemeris Time (ET) are used to indicate time. MJD is given in days that goes back to 00:00:000 17^{th} of November 1858. This transformation leads to a simulation start time of 60390.00 MJD and a simulation end time of 60404.00 MJD. Ephemerides require another form of time called ET, which is the number of seconds since 00:00:000 01-01-2000. The simulation start time in ET is 764251269.18 seconds and the simulation end time 14 days later is 765460869.18 seconds. The values of the simulation start- and end time are provided in Table 4.1 as time and date, MJD and ET. Furthermore, the fixed time step, i.e., numerical integration interval, in the dynamic model is 10 seconds, because the time interval in a numerical integrator affects the outcome, so a small time step is desired. However, a smaller time step, e.g., 1 second was unfortunately not possible due to computational limitations of the available hardware

Table 4.1: Simulation time setup for the dynamic simulator.

Date	MJD	ET
21 - 03 - 2024	60390.00	764251269.18 s
04 - 04 - 2024	60404.00	765460869.18 s

4.2.2. Environment setup

The environment is created by defining a system of bodies consisting of relevant celestial bodies and artificial bodies. Relevant celestial bodies are bodies that exert an acceleration on the propagated bodies and/or are an area of interest in the research. The bodies are defined with respect to a certain reference frame orientation and reference frame origin. Possible frame orientations are J2000 reference frame and ECLIPJ2000 reference frame. The frame origin can be celestial body centered or centered in the Solar System Barycenter (SSB). Celestial body centered reference frames are right-handed inertial frames with the x-axis towards the vernal equinox. Additionally, for J2000 the z-axis is aligned with the Earth's rotation axis and EPCLIPJ2000 has its z-axis perpendicular to the ecliptic. For both reference frames, the y-axis finishes the right-hand rule and the origin is the center of mass of the celestial body.

Artificial bodies are self-defined bodies and are added to the system. One creates a body by providing it a name and then one can add properties to this body, e.g., mass, aerodynamic settings and/or solar radiation pressure settings. In this research, all Solar System planets, the Sun and the Moon are included in the environment and the reference fame was chosen to be Earth-centered J2000. Furthermore, EML2O and ELO are added to the environment as artificial bodies. Table 4.2 provides the environment settings for the dynamic model.

Table 4.2: Environment setup for the dynamic simulator.

System of bodies	Sun, Mercury, Venus, Earth, Moon, Mars, Jupiter, Saturn, Uranus, Neptune
Artificial bodies	EML2O, ELO
Frame origin	Earth
Frame orientation	J2000

4.2.3. Propagation setup

The propagation setup describes the settings for the bodies to be propagated. Ingredients for the propagation setup are: bodies to propagate, central bodies, initial states, acceleration conditions, termination conditions and desired output values. Every aspect of the propagation setup is individually discussed and thereafter all propagation settings are provided in Table 4.4.

Bodies to propagate

The bodies to propagate are the bodies from which the trajectories are determined by the dynamic simulator. This dynamic model is made to obtain the nominal trajectories of EML2O and ELO, meaning they are the bodies to propagate.

Central bodies

In tudat, a central body must be defined for each body to be propagated. The central bodies can differ from the global frame origin, because the central body depends on the provided initial states. For example, if one has the initial state of satellite X with respect to the Moon, the central body of that satellite is the Moon. Tudatpy provides the transformation of the Moon based states to the Earth based states.

Initial states

An initial state is required for each body to propagate. Initial states must be provided in Cartesian state elements. The initial states must comply with the central bodies. In this research, both initial states are provided with respect to an Earth-centered J2000 reference and are given in Table 4.4. The initial states of EML2O are obtained from the database provided by politecnico di milano at the simulation start time. The initial state of ELO are obtained by adding the Moon based state to the state of the Moon with respect to Earth, from ephemerides, at simulation start time.

Accelerations

The accelerations define which bodies exert a certain type of acceleration on the bodies to propagate. These bodies must be defined in the system of bodies as described in subsection 4.2.2, because the ephemerides are only available for the Solar System bodies that are defined in the environment setup. Besides the addition of a body, there are several acceleration types. Common types of accelerations are drag due to atmosphere, solar radiation pressure and gravity from celestial bodies. The types of acceleration are discussed below. Note that bold letters are vectors.

Solar radiation pressure

SRP is an acceleration that is exerted by the radiation of the Sun. Massless photons travelling at the speed of light colliding with a satellite exchange their momentum to the satellite that causes an acceleration on the satellite. The magnitude of the solar radiation pressure depends on the distance from the Sun, the extent in which a celestial body is blocking the sunlight (occulting body), the area of the satellite facing the Sun and material properties of the area facing the emitted light Out (2017). Solar radiation pressure can be modelled in two different ways, namely as cannonball radiation pressure and as panelled radiation pressure. The equation for acceleration due to cannonball solar radiation pressure is given in Equation 4.1 Tudat (2022c).

$$\mathbf{a}_{SRP} = \left(\frac{P}{4\pi c}\right)\left(\frac{C_r S_{ref}}{m}\right)\frac{\hat{\mathbf{r}}}{r^2} \tag{4.1}$$

Where the parameters stand for:

- $\mathbf{a}_{SRP}$: Acceleration due to solar radiation pressure [m/s^2]
- C_R: Radiation pressure coefficient [-]

- S_{ref}: Reference area [m^2]
- m: mass of the body experiencing A_{SRP}
- **r**: Vector pointing from the satellite to the Sun
- P: Total emitted radiation power for the body exerting the acceleration
- c: Speed of light [m/s]

As shown in Equation 4.1, the equation leaves the occulting body out. However, tudatpy takes into account situations where a celestial body is (partially) occulting the Sun.

Panelled radiation pressure is a more complex way to describe the acceleration due to solar radiation pressure. However, for this to work properly, one needs to include the orientation of the satellite with respect to the Sun which is not included in this research. Therefore, SRP modelled as cannonball radiation pressure is used.

Point mass gravity
An acceleration acting on a satellite is described in terms of a gravity potential as

$$\ddot{\mathbf{r}} = \nabla U, \tag{4.2}$$

where

$$\nabla = \frac{\partial}{\partial x}\mathbf{u}_x + \frac{\partial}{\partial y}\mathbf{u}_y + \frac{\partial}{\partial z}\mathbf{u}_z, \tag{4.3}$$

and U is the gravity potential. In case of a point mass, the gravity potential U is written as

$$U = GM\frac{1}{\mathbf{r}}. \tag{4.4}$$

The acceleration due to point mass gravity can be described using Newton's Law of Gravity, as shown in Equation 4.5 Montenbruck and Gill (2001), Tapley et al. (2004). Where $\ddot{\mathbf{r}}$ is the acceleration of the satellite due to the point mass in m/s^2, G is the universal gravitational constant, which equals 6.67259·10^{-11} $m^3kg^{-1}s^{-2}$, M is the mass of the point mass in kg, where it is assumed that the mass of the satellite is negligible with respect to the mass of the point mass, and **r** is the vector from the point mass to the satellite.

$$\ddot{\mathbf{r}} = GM \cdot \frac{\mathbf{r}}{r^3} \tag{4.5}$$

Spherical harmonic gravity
In the propagation of satellites it is often assumed that the total mass of a body is located in the center of the body, hence the name point mass gravity. Bodies far away can indeed be defined as point mass perturbations, since the mass distribution is negligible. However, for celestial bodies nearby satellites, e.g., a LEO orbiter, a more accurate gravity model could be taken into account, to increase the accuracy of the acceleration exerted. An approximation of gravity, based on spherical harmonic components can be used to model the shape and mass distribution of the celestial body. The form of the acceleration function remains the same as in Equation 4.2, but the gravity potential U is redefined by including the shape and mass distribution as shown in Equation 4.6. In this equation, ρ is the density within the body at point **s**, dx, dy, dz represents the volume of dm, **r** is the position vector from the frame origin of the body to the satellite and **s** is the position vector from the frame origin of the body to the mass element of body M, so r - s is the position vector from mass element of body M to the satellite.

$$U = G\int\int\int \frac{\rho(\mathbf{s})dxdydz}{r-s} = G\int \frac{\rho(\mathbf{s})dm}{r-s} \tag{4.6}$$

In order to get more insight in Equation 4.6, the inverse of the distance between the mass particle and the satellite (1/(r - s)) is expanded in a series of Legendre polynomials

$$\frac{1}{r-s|} = \frac{1}{r}\sum_{n=0}^{\infty}\left(\frac{s}{r}\right)^n P_n(cos(\gamma)), \text{ with } \quad cos(\gamma) = \frac{\mathbf{r}\cdot\mathbf{s}}{rs}, \tag{4.7}$$

where

$$P_n(u) = \frac{1}{2^n n!}\frac{d^n}{du^n}(u^2-1)^n. \tag{4.8}$$

The Legendre polynomial P_n(u) is of degree n and γ is the angle between vector **r** and vector **s**. Using longitude λ and latitude ϕ, the position elements of **r** and **s** can be described as

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos(\phi)\cos(\lambda) \\ r\cos(\phi)\sin(\lambda) \\ r\sin(\phi) \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r\cos(\phi')\cos(\lambda') \\ r\cos(\phi')\sin(\lambda') \\ r\sin(\phi') \end{bmatrix}. \tag{4.9}$$

Then, the addition theorem of Legendre polynomials can be applied stating that

$$P_n(\cos(\gamma)) = \sum_{m=0}^{n}(2-\delta_{0m})\frac{(n-m)!}{(n+m)!}P_{nm}(\sin(\phi))P_{nm}(\sin(\phi'))\cos(m(\lambda-\lambda')), \tag{4.10}$$

where P_{nm} is the associated Legendre polynomial of degree n and order m which is defined as

$$P_{nm}(u) = (1-u^2)^{m/2}\frac{d^m}{du^m}P_n(u). \tag{4.11}$$

Using several substitutions, the gravity potential of mass M can be written as

$$U = \frac{GM}{r}\sum_{n=0}^{\infty}\sum_{m=0}^{n}\frac{R^n}{r^n}P_{nm}(\sin(\phi)(C_{nm}\cos(m\lambda)) + S_{nm}\sin(m\lambda)), \tag{4.12}$$

where C_{nm} and S_{nm} are the geopotential coefficients and are given as

$$\begin{aligned} C_{nm} &= \frac{2-\delta_{0m}}{M}\frac{(n-m)!}{(n+m)!}\int\frac{s^n}{R^n}P_{nm}\sin(\phi')\cos(m\lambda')\rho(\mathbf{s})dx'dy'dz'\mathbf{s} \\ S_{nm} &= \frac{2-\delta_{0m}}{M}\frac{(n-m)!}{(n+m)!}\int\frac{s^n}{R^n}P_{nm}\sin(\phi')\sin(m\lambda')\rho(\mathbf{s})dx'dy'dz'\mathbf{s}. \end{aligned} \tag{4.13}$$

So, the acceleration function acting on a satellite due to spherical harmonic gravity is

$$\ddot{\mathbf{r}} = \nabla\cdot\frac{GM}{r}\sum_{n=0}^{\infty}\sum_{m=0}^{n}\frac{R^n}{r^n}P_{nm}(\sin(\phi)(C_{nm}\cos(m\lambda)) + S_{nm}\sin(m\lambda)). \tag{4.14}$$

Both satellites orbit close to the Moon and relatively close to Earth. Therefore, in this research, a spherical harmonic gravity model is applied for the Earth and the Moon. Both are defined with a relative high maximum order of 200 and degree of 200 to obtain accurate trajectories. To determine what the effect of spherical harmonics is on the trajectories of the satellites, multiple simulations were executed using all Solar System planets, the Sun (incl SRP) and the Moon, but with a different order and degree of spherical harmonic gravity. The maximum 3-D position difference with respect to the model using Earth and Moon spherical harmonics with an order and degree of 200 and 200 is given in Table 4.3. The absolute distance between LUMIO from the dataset and the propagated EML2O after 14 days is approximately 90 172 m.

Table 4.3: 3-D position difference due to the dynamic model difference in spherical harmonics in the Earth and the Moon (reference case is max. order and degree of Earth and the Moon 200 and 200).

Earth order & degree	**Moon order & degree**	**EML2O 3-D position difference**	**ELO 3-D position difference**
100, 100	100, 100	$<1\times10^{-16}$ m	3.087×10^{-3} m
50, 50	50, 50	$<1\times10^{-16}$ m	1.523×10^{-2} m
40, 40	40, 40	$<1\times10^{-16}$ m	1.275 m
30, 30	30, 30	$<1\times10^{-16}$ m	2.791×10^{1} m
20, 20	20, 20	$<1\times10^{-16}$ m	2.438×10^{2} m
10, 10	100, 100	$<1\times10^{-16}$ m	3.087×10^{-3} m
2, 2	20, 20	1.449×10^{-1} m	2.438×10^{2} m
Point mass	20, 20	1.995×10^{3} m	2.000×10^{2} m
Point mass	10, 10	1.995×10^{3} m	1.567×10^{4} m
Point mass	Point mass	2.045×10^{3} m	1.573×10^{6} m

An maximum order and degree of 2 has a significant effect on the trajectory of EML2O. Above two, the significance of increasing the maximum order and degree of the spherical harmonics of Earth decreases. A factor of 10^4 in 3-D position difference occurs in modelling Earth as a point mass or spherical harmonics with an order and degree of 2, 2 is a significant difference to consider. In general, the Moon's spherical harmonics has less influence than Earth on EML2O.

ELO is especially affected by the Moon's spherical harmonics. Even in a relative high order and degree for the Moon's spherical harmonics, there is still a 3-D position difference due to the different gravity models. However, the effect above a maximum order and degree of 50 starts to become negligible. The Earth's spherical harmonics do not have a significant effect on the trajectory of ELO. Although a improvement in maximum 3-D position error occur in when comparing modelling Earth as a point mass or with spherical harmonics with an order and degree of 2.

It can be observed in Table 4.3 that spherical harmonic gravity has a significant influence on the states of the satellites up to a certain order and degree. Dependent on a trade-off in computation time/power and desired dynamic model accuracy one can define the maximum order and degree for spherical harmonic gravity. To implement a high fidelity model, it is desired to obtain the nominal states by taking into account spherical harmonics of all bodies to improve the fidelity of the model. However, due to impracticality's, this is not possible. To improve the fidelity of the dynamic model, the the Earth and Moon are modelled as spherical harmonics with an order and degree of 200.

Termination conditions
Propagation ends when the termination conditions are met. This can be either time dependent or when a parameter reaches a certain value. The simulation end time is used as termination condition in this research.
Output values
The output values are the values one wants to save after the simulation is complete. Besides the state history over the trajectory one may want to save other elements. In this case, the total acceleration acting on a body and the acceleration contributions per perturbation were also saved.
Overview of the propagation setup The propagation setup for the nominal trajectory is provided in Table 4.4. The initial state of EML2O is provided by a dataset containing the states of LUMIO over a time span. The initial state of ELO are obtained by adding state elements from Table 2.3 to the states of the Moon at the epoch of initial time.

Table 4.4: Propagation setup for the dynamic model in an Earth-centered J2000 reference frame.

Propagation setup	
Bodies to propagate	EML2O, ELO
Central bodies	Earth, Earth
Initial states EML2O	
x	$-3.105\,379\,98 \times 10^8$ m
y	$2.494\,231\,57 \times 10^8$ m
z	$1.749\,377\,57 \times 10^8$ m
$\dot{x}$	$-9.931\,718\,42 \times 10^2$ m/s
$\dot{y}$	$-7.664\,085\,14 \times 10^2$ m/s
$\dot{z}$	$-5.251\,732\,80 \times 10^2$ m/s
Initial states ELO	
x	$-2.802\,868\,25 \times 10^8$ m
y	$2.524\,427\,58 \times 10^8$ m
z	$1.469\,229\,56 \times 10^8$ m
$\dot{x}$	$-2.878\,971\,46 \times 10^2$ m/s
$\dot{y}$	$-2.411\,604\,03 \times 10^3$ m/s
$\dot{z}$	$-2.972\,268\,74 \times 10^2$ m/s
Acceleration settings (both EML2O & ELO)	
Earth	Spherical harmonic gravity (200, 200)
Moon	Spherical harmonic gravity (200, 200)
Sun	Point mass gravity & SRP
Mercury	Point mass gravity
Venus	Point mass gravity
Mars	Point mass gravity
Jupiter	Point mass gravity
Saturn	Point mass gravity
Uranus	Point mass gravity
Neptune	Point mass gravity
Termination condition	Simulation end time (765460869.18 [s])
Desired output values	States history and acceleration contributions of each perturbation

4.2.4. Integration setup

The Runge-Kutta 4 (RK4) is used as numerical integrator. The RK4 method provides a decent balance between computation time and accuracy LUMIOs states were obtained using a RK4. Therefore, the same numerical integrator is considered. RK4 is a renowned numerical integrator and proven to be more accurate than the forward Euler integrator. The integrator uses a fixed time step, which is preferred in the determination of the nominal states. In general, the RK4 works as follows:

1. One has an initial value problem in the form of:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \mathbf{x}(t_0) = x_0 \tag{4.15}$$

   The goal is to obtain an unknown vector **x** (or scalar x) at time t, which is approximated using the RK4 numerical approach.

2. The states update is provided in Equation 4.16, where x_i is the previous state, x_{i+1} is the state to be obtained and dt is the fixed time step.

$$\mathbf{x}_{i+1} = x_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)dt \tag{4.16}$$

Factors k_n are given as:

$$\begin{aligned} k_1 &= f(t_i, \mathbf{x}_i) \\ k_2 &= f(t_i + \frac{dt}{2}, \mathbf{x}_i + dt\frac{k_1}{2}) \\ k_3 &= f(t_i + \frac{h}{2}, \mathbf{x}_i + dt\frac{k_2}{2}) \\ k_4 &= f(t_i + dt, \mathbf{x}_i + dt \cdot k_3 \end{aligned} \tag{4.17}$$

3. These calculations are done for every time step until final time is reached.

4.2.5. Dynamic simulation

After defining the environment, propagation and integration settings, the simulator can be created. The simulation can be executed by defining and solving the state derivative equation:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t; \mathbf{p}), \text{ where } \quad \mathbf{x}(t_0) = \mathbf{x}_0. \tag{4.18}$$

In Equation 4.18, $\mathbf{x}$ defines the state vector that is to be propagated and $\mathbf{x}_0$ represents the initial state at t_0. Furthermore, vector $\mathbf{p}$ is taken into account, which represents the dependence on various environmental and system parameters defined in the environment setup. Next, the state derivative function $\mathbf{f}$ is defined and the differential equations are solved using the chosen integrator. The simulator stops when meeting the termination condition. An overview of the dynamic model is presented in Figure 4.1.

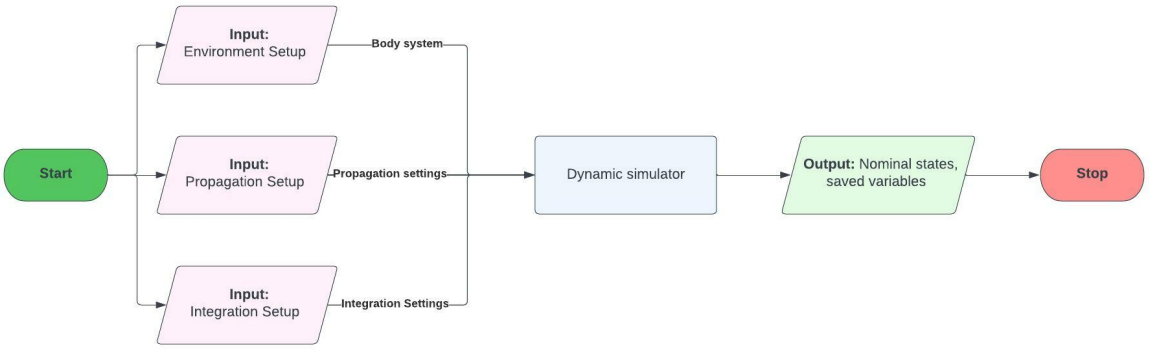


Figure 4.1: Overview of the dynamic model.

4.3. Nominal simulation

This section presents the simulations performed using the input parameters defined as in section 4.2. Figure 4.2 shows the trajectories of EML2O, ELO and the Moon in an Earth-centered J2000 frame. Figure 4.3 shows the same trajectories in a Moon-centered frame. In the figures, the dot represents the initial position of the satellite. The halo trajectory of EML2O is not fully closed, which is due to absence of stationkeeping. The minimum inter-satellite distance is 36 434 km and the maximum inter-satellite distance in this simulation is 87 237 km. The maximum and minimum inter-satellite are consistent with Speretta et al. (2021) in which a inter-satellite distance of 31 772 km and 89 870 km is considered between the Lunar Pathfinder and LUMIO.

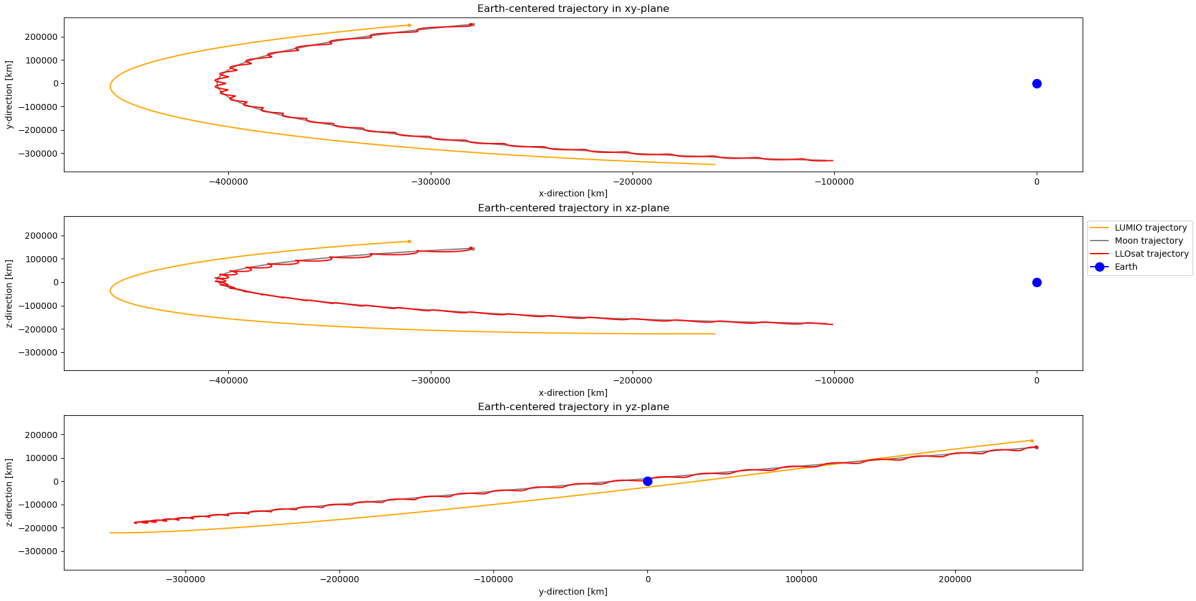


Figure 4.2: Overview of the trajectories of EML2O and ELO in an Earth-centered J2000 frame.

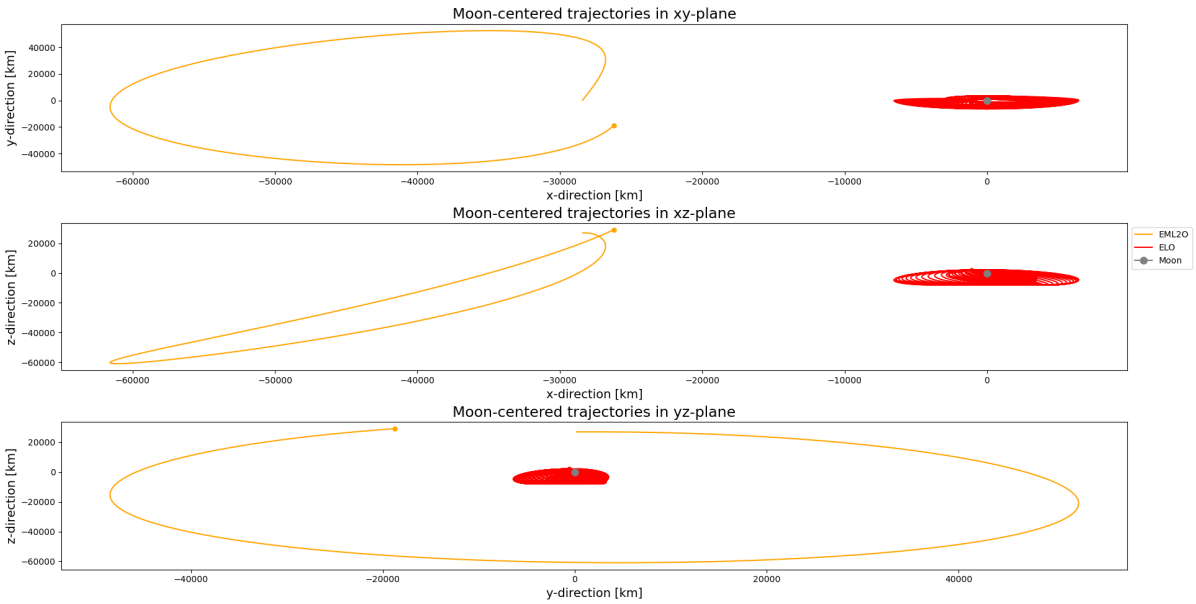


Figure 4.3: Overview of the trajectories of EML2O and ELO in a Moon-centered reference frame.

4.3.1. Accelerations

The trajectories of both satellites are defined by the accelerations acting on the satellites. For both satellites a figure is provided containing the magnitude of the considered acceleration contributions.

Elliptic Lunar Orbiter

The main contributor to the acceleration of ELO is the Moon. The satellite orbits the Moon in an elliptic orbit, which can be observed from Figure 4.4 due to the periodicity of the acceleration curve due to the Moon. During the peaks of the acceleration curve of the Moon, the satellite is in its periapsis and in its troughs ELO is in its apoapsis. In addition, ELO experiences a significant acceleration due to the Earth's gravity, the Sun's point mass gravity and SRP. The magnitude of SRP is in the order 1×10^{-7} m/s^2 and forms a straight line with one gap during day four of the simulation when the Earth occults the Sun, which is verified simulating separately the Earth and the Moon in the occultation model. The other celestial bodies exert a significantly

lower acceleration to ELO, due to their larger distance. Jupiter exerts acceleration at an order of approximately $1 \times 10^{-12}\,\mathrm{m/s^2}$ and the other planets contribute a magnitude in the order of less than $1 \times 10^{-13}\,\mathrm{m/s^2}$. Due to the low contributions in acceleration, some of these celestial bodies could be ignored without causing a notable difference in the trajectory. However, in a full acceleration breakdown analysis, the direction of the acceleration contributions must be taken into account as well.

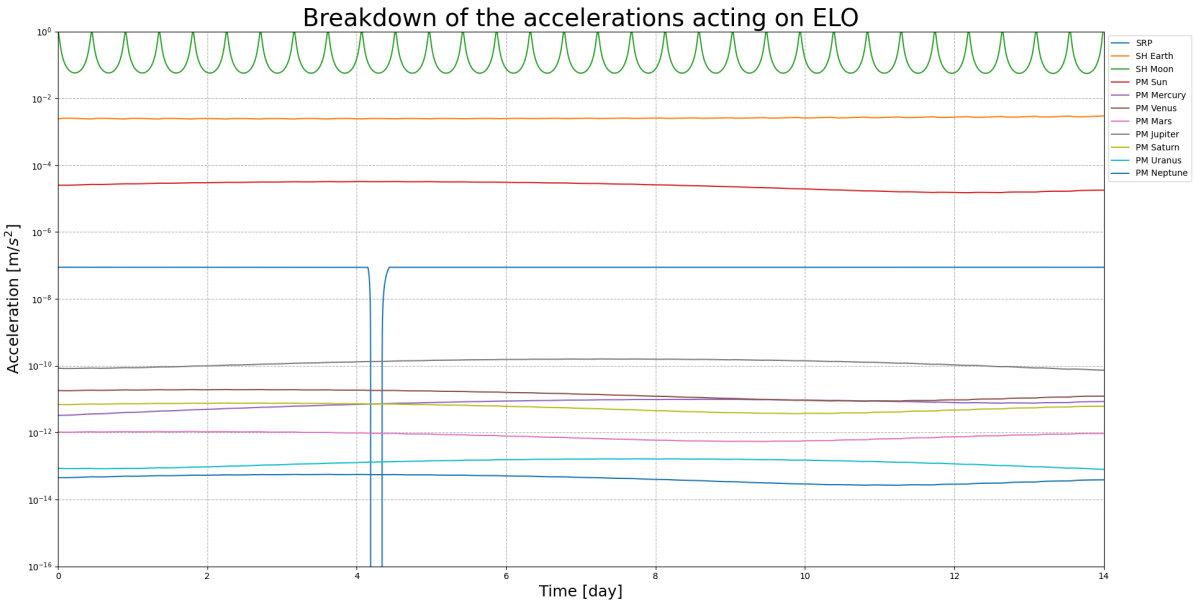


Figure 4.4: Acceleration breakdown acting on ELO.

Earth-Moon L2-Orbiter

EML2O experiences less total acceleration due to its environment than ELO, because EML2O orbits near a libration point and has a significantly larger distance to the Moon than ELO. Because EML2O orbits near the Earth-Moon Lagrange 2 (EML2) point, the accelerations exerted by Earth and the Moon are quite similar. In this simulation, EML2Os initial state is near the Moon, which is visible in Figure 4.3 where the orange dot represents the initial position. It can also be observed in Figure 4.5, where the acceleration due to the Moon is slightly higher than the acceleration due to Earth until day 1 of simulation time. Between day 1 and 13, the acceleration due to Earth becomes slightly higher, but during the last day of simulation (near the Moon), the acceleration due to the Moon is higher again. Contributions due to the other Solar System planets and the Sun are quite similar compared to the acceleration breakdown of ELO, since the relative distances from ELO and EML2O to the Solar System bodies are similar.

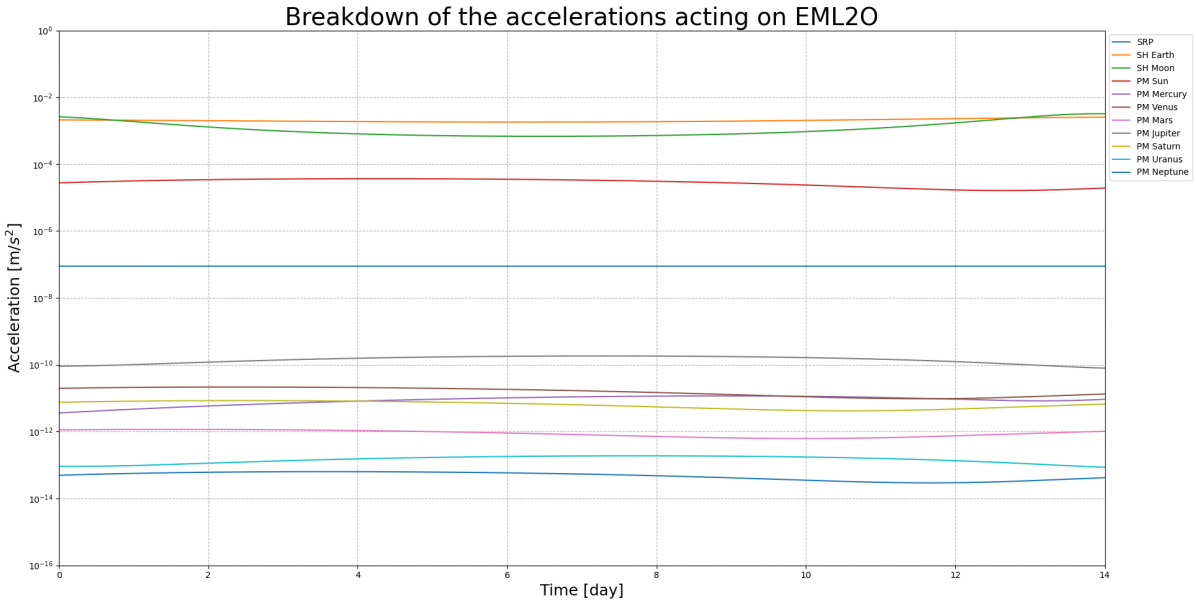


Figure 4.5: Acceleration breakdown acting on EML2O.

The acceleration breakdown can be (partially) verified using Scotti et al. (2022), where an acceleration breakdown for 1 day is applied on the Lunar Pathfinder including the Earth, Moon, Sun (incl. SRP), and Jupiter. The magnitude and progression of the periodicity of the Moons gravity complies with the simulated acceleration breakdown, as well as the magnitude of the acceleration due to the Earth, Sun, SRP, and Jupiter. For LUMIO, such an acceleration breakdown is not available, but since the acceleration model is identical for both simulated satellites, it is assumed that the progression of both satellites can be verified on the basis of Scotti et al. (2022).

5

Measurement model

The measurement model provides measurements based on the nominal states and a two-way ranging error. Inter-satellite measurements can be obtained when there is a line-of-sight between both satellites and when both satellites equip a radio system to form a link. This chapter first discusses both the visibility and link budget analysis. Next, the link budget analysis to compute the ranging error is done using the baseline radio parameters. Thereafter, the possible bit rates for the different antenna configurations are determined from which the estimated ranging errors are obtained. Last, the method to determine nominal measurements is discussed.

5.1. Visibility analysis

Range measurements can only be obtained when there is a LOS vector between the transmitting and receiving satellites. The inter-satellite link is formed by EML2O and ELO, so the Moon could block the line-of-sight at some epochs. The visibility analysis is performed using the nominal trajectories. The nominal positions can differ from the actual positions, so taking into account arbitrary position errors, a margin in size of the Moon is added. The radius of the Moon is approximately 1737.4 km. By including a margin, the radius of the Moon plus margin is set on 1800 km, which is rounded up for simplification and to be on the conservative side in the visibility analysis.

The visibility is based on trigonometry by determining whether the distance between the center of mass of the Moon perpendicular to the inter-satellite distance vector is larger than the radius of the moon plus margin. A schematic overview is provided in Figure 5.1. First, the angle between the inter-satellite distance and the vector from EML2O to the Moon is determined, which is done using Equation 5.1 where **a** is the inter-satellite vector and **b** is the vector from EML2O to the center of mass of the Moon. Second, the length of **h** can be determined using Equation 5.2, where h and b are both scalars representing the absolute distances. Last, visibility is determined by checking whether h is larger than 1800 km.

$$cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|} \tag{5.1}$$

$$h = b \cdot sin(\theta) \tag{5.2}$$

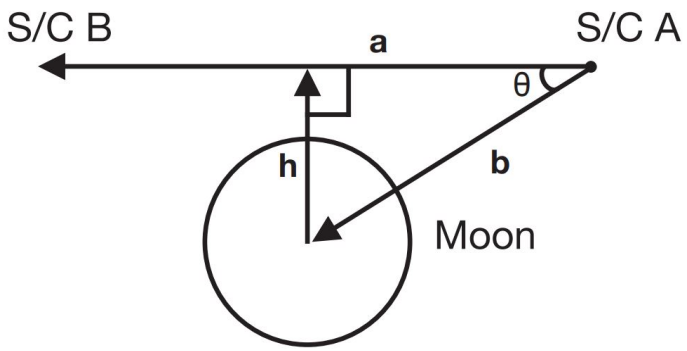


Figure 5.1: Schematic overview of the trigonometry of the visibility between ELO and EML2O.

5.2. Link budget

The link budget analysis is only preformed for one orbital scenario, which is the worst case scenario during the trajectories in the given simulation time. Considering a constant wavelength, the worst case scenario exists when the distance between the two satellites is largest, since the distance between the two antennas is the largest. The free space losses between two antennas are calculated in dB (Board, 2004). As shown in Equation 5.3, where λ is the wavelength in meter and d is the distance in meter, the free space losses are larger for higher distance and/or higher frequencies.

$$L_{FSL} = 20 \cdot log_{10}\left(\frac{4\pi d}{\lambda}\right) \tag{5.3}$$

For the link budget analysis, a top level communication link is used that consists of a transmitting satellite, a free space in-between and a receiving satellite, as shown in Figure 5.2. A communication system is composed by the following blocks (Speretta, 2022):

- **Transmitting satellite**
  1. Input data is acquired from the OBC to transmit
  2. **Packet engine**: The data is packed into blocks
  3. **Code engine**: The blocks/packets are encoded to improve their robustness to noise and allow eventual error correction
  4. **Modulator**: Data is converted to analog signals at the right frequency
  5. **Power amplifier**: The signal needs to be amplified to increase the output power to guarantee reception
  6. **Antenna**: The electrical signal is turned into radio waves and sends it in the direction of the receiver
- **Free space**
  1. Free space losses occur due to the decrease of the power flux density with the square of the distance (Board, 2004)
- **Receiving satellite**
  1. **Antenna**: Transforms the radio waves into an electric signal
  2. **Low-Noise Amplifier (LNA)**: Amplifies the low power signal without decreasing (drastically) the Signal-to-Noise ratio (SNR)
  3. **Demodulation**: Converts the analog signal to digital bits
  4. **Code engine**: Decodes the digital signal
  5. **Packet engine**: Packet reconstruction
  6. Data output to the receiving satellite OBC

As shown in Figure 5.2, the power level varies along the full transmission / receiving chain. The transmitter is fed with an initial transmission power. Losses occur on the cable within the transmission system and then the signal strength increases due to the transmitting antenna gain. The resulting power leaving the transmitting satellite is the Effective Isotropic Radiated Power (EIRP) Equation 5.4, which is often described in a relative unit dBm instead of Watt. Transforming units from Watt to dBm is done using Equation 5.5. After transmitting, the signal power does not actually decreases in power, as is shown in Figure 5.2, but the power flux density decreases since the signal power spreads over a larger area. This phenomena is called 'free space losses' and causes the reduction in signal power at the receiving antenna. The free space losses are a function of wavelength and distance and is shown in Equation 5.3. When the signal arrives at the receiver, the signal strength increases due to the receiver's antenna gain. Subtracting the cable losses at the receivers side provides the receiving power R_x. R_x must be greater than the receivers sensitivity in order to be discriminated from the receivers internal noise. The receivers sensitivity is the minimum signal strength needed for the receiver to detect. It is strongly recommended to have a 3 dB margin for a fixed speed link, which is therefore taken into account in the link budget.

$$EIRP = T_x - L_{cable} + G_{transmit} \tag{5.4}$$

$$P(dBm) = 10\log_{10}\left(\frac{1000P(W)}{1W}\right) \tag{5.5}$$

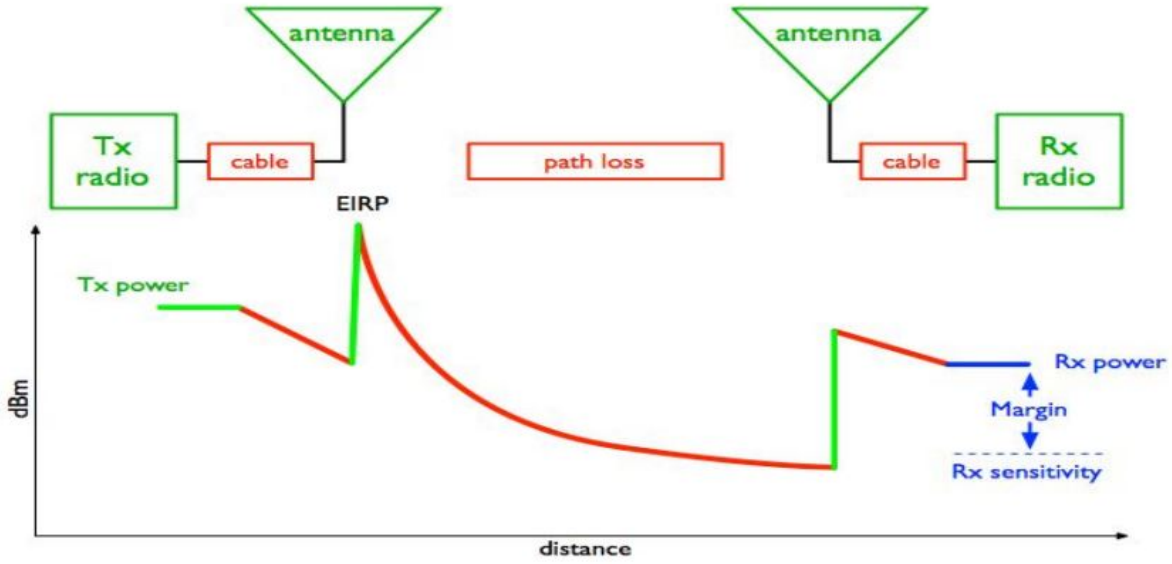


Figure 5.2: Schematic overview of a top level communication link (Speretta, 2022)

5.2.1. Initial radio parameters

For the link budget calculations, the radio parameters from the Lunar Pathfinder Harlé and Friend (2020) and the LUMIO mission Anywaves (2020), Turan et al. (2022) are used. The transmission power on the Lunar Pathfinder is 3 dBW, meaning the system has 2 W of output power. The available frequencies of the Lunar Pathfinder are in S-band 2025 - 2110 MHz and UHF 390 - 405 MHz (Harlé and Friend, 2020). In this study, the frequency is fixed at 2110 MHz, because the link budget is carried out for the worst case scenario. The highest frequency give the shortest wavelength, see Equation 5.6 and a shorter wavelength leads to a higher free space losses (Equation 5.3). Cable losses, antenna gain, polarization losses and required Energy per bit to noise power spectral density ratio (E_b/N_0) are obtained from Turan et al. (2022) in which similar simulations are performed. Polarization losses are caused due to mismatch of the antennas and E_b/N_0 is the normalized SNR. The available bit rate of the Lunar Pathfinder ranges from 0.5 to 128 kbps. The bit rate is one of the variables in the link budget, so it must be within the supported range. The margin is universal for both ELO and EML2O and is based on the heuristic of 3 dB as in Cheung (2015), Speretta et al. (2021), Turan et al. (2022). The noise temperature is set to 26.9 dB/K and is obtained from Speretta et al. (2021). The radio parameters

for ELO are provided in Table 5.1.

EML2O has a transmission power of 3 dBW (Speretta et al., 2021). The frequency is set to 2290 MHz, which is the highest S-band frequency the Lunar Pathfinder can receive and the highest possible frequency the considered antenna for LUMIO can transmit (Anywaves, 2020). Cable losses, polarization loss and required E_b/N_0 are based on Speretta et al. (2021), Turan et al. (2022) and one antenna has an antenna gain of 6.5 dBi. The bit rate is determined in the link budget analysis. The radio parameters of EML2O are provided in Table 5.2.

The link budget is executed from the perspective of EML2O. This means that ranging operations from EML2O to ELO are downlink and from ELO to EML2O are uplink ranging operations. Therefore, the calculations done in the link budget analysis contain a subscript $_{up}$ for ELO and a subscript $_{down}$ for EML2O.

Table 5.1: Uplink parameters of ELO.

Link parameter	**Physical quantity**	**Value**
Transmission power	T_x	3 dBW
Frequency [1]	f	2110 MHz
Cablelosses	L_{cable}	1 dB
Antenna gain	G	23.6 dBi
Polarization loss	L_{pol}	0.5 dB
Required E_b/N_0	$(E_b/N_0)_{req}$	2.5 dB
Bit rate [2]	bitrate	0.5 - 128 kbps
Margin	margin	3 dB
Noise temperature	T_{noise}	26.9 dB/K

Table 5.2: Downlink parameters of EML2O.

Link parameter	**Physical quantity**	**Value**
Transmission power	T_x	3 dBW
Frequency	f	2290 MHz
Cablelosses	L_{cable}	1 dB
Antenna gain [3]	G	6.5 dBi
Polarization loss	L_{pol}	0.5 dB
Required E_b/N_0	$(E_b/N_0)_{req}$	2.5 dB
Bit rate	bitrate	To Be Determined (TBD)
Margin	margin	3 dB
Noise temperature	T_{noise}	26.9 dB/K

5.2.2. Link budget analysis

The wavelength, which is required to calculate free space losses, is determined using Equation 5.6

$$\lambda = \frac{c}{f}, \tag{5.6}$$

and gives $\lambda_{down} = \approx 0.1309$ m and $\lambda_{up} \approx 0.1421$ m. Given these wavelengths and the fact that $d_{max} \approx 87.237$ km, the free space losses are calculated using Equation 5.3 and have the following value: $L_{FSL_{down}} \approx 198$ dB and $L_{FSL_{up}} \approx 197$ dB. Both up- and downlink wavelengths and the free space losses are constant in this link budget analysis in order to perform the link budget analysis for worst case. The link budget is elaborated once for the baseline conditions as provided in Table 5.1 and Table 5.2. Equations used in the link budget are obtained

[1] Possible transmit frequencies are S-band 2025-2110 MHZ and UHF 390-405 MHz (Harlé and Friend, 2020). Chosen for highest frequency, since L_{FSL} are highest then and link budget analysis is computed in worst scenario

[2] This states the possible bit rate. The bit rate used will vary, but within these boundaries.

[3] The antenna gain of 6.5 dBi is based on the antenna considered for LUMIO, which has >6.5 dBi gain at boresight (Anywaves, 2020).

from Speretta (2022).

The EIRP is determined for both up and down link using Equation 5.4 and gives $EIRP_{down}$ = 8.5 dB and $EIRP_{up}$ = 25.6 dB. The received signal power and the Gain-to-noise-temperature (G/T) are calculated using Equation 5.7 and Equation 5.8. This gives for receiving signal Rx_{down} (from EML2O → ELO) ≈ −190.5 dB and Rx_{up} (from ELO → EML2O) ≈ −172.7 dB. The downlink G/T is −20.4 dB and the uplink G/T is −3.3 dB.

$$Rx = EIRP - L_{FSP} - L_{pol} \tag{5.7}$$

$$G/T = G - T_{noise} \tag{5.8}$$

The bit rate can be obtained using Equation 5.9. However, the actual E_b/N_0 is not determined yet. The actual E_b/N_0 has a minimum of 5.5 dB, since it is equal to the required E_b/N_0 plus the margin. This leaves only the bit rate as unknown. Setting the E_b/N_0 at 5.5 dB, the maximum bit rate can be obtained by solving Equation 5.9.

$$\begin{aligned}(E_b/N_0)_{down} &= Rx_{down} + G/T_{up} - 10log_{10}(k_b \cdot bitrate_{down}) \\ (E_b/N_0)_{up} &= Rx_{up} + G/T_{down} - 10log_{10}(k_b \cdot bitrate_{up})\end{aligned} \tag{5.9}$$

The maximum up-and downlink bit rate are 1011 bps and 859 bps. The bit rates are rounded down to whole numbers, since they are binary values. For simplification, the bit rates are rounded down slightly further (dividable by 50). Therefore, the downlink bit rate is 850 bps and the uplink bit rate is 1000 bps. Accompanied with these bit rates, the downlink and uplink margin are slightly higher than 3 dB, respectively 3.046 dB and 3.052 dB. Making the actual downlink E_b/N_0 (2.5 + 3.046) 5.546 dB and the actual uplink E_b/N_0 (2.5 + 3.052) 5.552 dB. Table 5.3 provides the calculated performance parameters based on the initials as described in Table 5.1 and Table 5.2.

Table 5.3: Calculated radio parameters based on Table 5.1 and Table 5.2.

Link parameter	**Downlink**	**Uplink**
Wavelength	0.1309 m	0.1421 m
Free space losses	198.5 dB	197.8 dB
EIRP	8.5 dB	25.6 dB
G/T	−20.4 dB	−3.3 dB
E_b/N_0	5.546 dB	5.552 dB
Bit rate	850 bps	1000 bps
Margin	>3 dB	>3 dB

5.3. Two-way ranging performance

As described in section 2.5, telemetry ranging is considered in order to obtain the inter-satellite distance. If one can calculate the round-trip light time using four specific timings, the inter-satellite distance can be determined. The four required epochs are the transmission time of EML2O, the receiving time of ELO, the transmitting time of ELO and the receiving time of EML2O, see Figure 5.3.

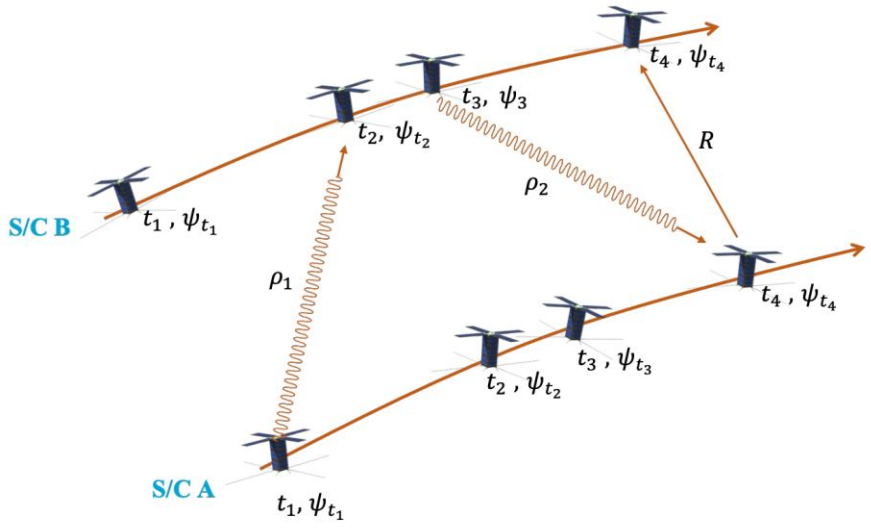


Figure 5.3: Time-derived inter-satellite ranging in order to obtain the four timestamps (Turan et al., 2022).

Assuming that each bit is a statistical independent measurement and knowing that data travels with the speed of light, the ranging error per bit can be determined by dividing the speed of light by the bit rate, as shown in Equation 5.11. This equation gives a downlink ranging error per bit of 352.70 km and an uplink ranging error per bit of 299.80 km/bit. To obtain the two-way ranging error per bit, Pythagorean theorem is applied. Meaning the two-way ranging error per bit is the square root of the squares of the one-way ranging errors, resulting in a total ranging error per bit of 462.90 km/bit. These are very high with respect to the required ranging errors, but can be drastically reduced by increasing the number of measurements (N). Since the ranging errors calculated in Equation 5.11 are per bit, increasing the bits leads to a reduction in ranging error. In case the number of measurements is equal the amount of bits (downlink) transmitted in one second the ranging error reduces to 15 877.13 m, as shown in Equation 5.10.

$$\sigma_{\rho_{conv}} = \frac{\frac{\sigma_{range}}{bit}}{\sqrt{N}} = \frac{462.90e3}{\sqrt{850}} = 15\,877.13\,\text{m} \tag{5.10}$$

$$\frac{\sigma_{\text{range}}}{\text{bit}} = \frac{c}{\text{bitrate}} \tag{5.11}$$

If one sets the integration time to 1 s, the number of measurements would be numerical equal to the number of bits transmitted in one second. To reduce the ranging error even more, one can increase the number of measurements by increasing integration time Equation 5.12, which is the time needed to transmit the desired amount of data. For example, one can set the number of measurements to 51 000 bits, which is 60 times the downlink bit rate. The downlink integration time then becomes 51000/850 = 60 s and the uplink integration time 51000/1000 = 51 s. The uplink integration time is lower due to the higher uplink bit rate. Such an increase in the number of measurements has positive effect on the ranging error which will be reduced to $462.90e3/\sqrt{51000}$ = 2049.73 m. However, the equation does not consider the motion of the satellites during the ranging operation. The higher the ranging time, the larger the satellites deviate from their initial positions. This effect should be taken into account for. Motions during ranging operations can be compensated for, but still a certain drift applies. This research fixes the downlink integration time to 1 second.

$$T_i = \frac{N}{\text{bitrate}} \tag{5.12}$$

The above calculations for the ranging error are done in the assumption that a whole bit-time can be considered as the basic distance measurement unit. Both one-way ranging errors can be reduced using a time-derived ranging error equation in telemetry symbols (Turan et al., 2022).

$$\sigma_{\rho_{TD}} = \frac{4cT_{sd}^2}{\pi T_l 10^{\frac{E_S/N_0 - margin}{10}}}. \tag{5.13}$$

In Equation 5.13, $\sigma_{\rho_{TD}}$ is the one-way time-derived ranging error in meter, c is the speed of light, T_{sd} is the symbol duration, T_i is the integration time and Symbol to noise ratio (E_s/N_0) is the energy per symbol to noise power spectral density or symbol to noise ratio. To determine the ranging error in meter, E_s/N_0 must be defined. E_s/N_0 is related to E_b/N_0 as described in Equation 5.14, where M represents the number of alternative modulation symbols. It is chosen to consider a modulation with M = 2, so that E_s/N_0 equals E_b/N_0. Possible modulations where M = 2 are Gaussian Minimum Shift Keying (GMSK) or Binary Phase Shift Keying (BPSK) modulation.

$$E_s/N_0 = E_b/N_0 log_2(M) \tag{5.14}$$

The one-way ranging errors can be determined for both down- and uplink. The term in the denominator including E_s/N_0 and the margin is transforming both E_s/N_0 and the margin out of the dB-scale. Taking the norm of both gives the ranging error in meter:

$$\sigma_{down} = \frac{4c(1/850)^2}{\pi \cdot 1 \cdot 10^{\frac{5.546-3}{10}}} = 293.93\,\text{m}$$
$$\sigma_{up} = \frac{4c(1/1000)^2}{\pi \cdot 0.85 \cdot 10^{\frac{5.552-3}{10}}} = 249.54\,\text{m}$$

$$\sigma_\rho = \sqrt{293.93^2 + 249.54^2} = 385.57\,\text{m} \tag{5.15}$$

Using the initial parameters as described as in Table 5.1, Table 5.2, Table 5.3 and the fixed modulation index and integration time, a two-way ranging error of 385.57 m is obtained. Increasing the bit rate has a positive effect on the two-way ranging error, but is limited based on the antenna gain. Therefore, multiple link budgets and ranging error calculations are executed based on different antenna gains, which are based on different antenna configurations.

5.4. Antenna arrays

The following assumption is made:
↑ Antenna gain → ↑ bit rate → ↓ two-way ranging error → ↓ 3-D state errors and ↓ 3σ uncertainties
However, exact relation and the degree of influence of each parameter to the other is unknown. In order to obtain insight in the above relations, the state element errors and 3σ uncertainties are calculated on the basis of different downlink antenna gains. To do so, different antenna arrays are considered that are based on the number of Anywaves antennas.

Calculations and simulation are executed for five configurations with an array of (1), 2, 4, 8, and 16 antennas respectively. It is assumed that there are no scaling losses, meaning that doubling the amount of downlink antennas lead to an increase in downlink antenna gain of 3 dB, as shown in Table 5.4. The considered downlink antenna is based on the antenna that will most likely be equipped by LUMIO. One single antenna has the following dimensions: L 79.8 x W 79.8 x H 12.1 mm^3 (Anywaves, 2020). The required surface area is also provided in Table 5.4 and it can be observed that the required surface area for 8 and 16 antennas is too large to fit on a 12U CubeSat and will exceed LUMIOs mission requirements. However, to overcome this issue, one can apply an antenna panel on top of the satellite as is done with CAPSTONE to provide enough room for the antennas. CAPSTONE uses an antenna array built by patch antennas, but details of the system are considered a commercial secret. CAPSTONE is able to obtain a downlink antenna gain of 20.5 dBi (Thompson et al., 2022). Assuming one patch antennas have a slightly higher antenna gain than Anywaves antennas, CAPSTONE will have approximately 12 to 24 antennas. This research considers an array up to 16 Anywaves antennas, which gives a maximum antenna gain of 18.5 dBi. Figure 5.4 shows the area where the antennas of CAPSTONE are placed.

Table 5.4: Downlink antenna gain based on number of antennas.

No. of antennas	Downlink antenna gain	Surface area (L x B) mm^2
1	6.5 dBi	79.8 x 79.8
2	9.5 dBi	159.6 x 79.8
4	12.5 dBii	159.6 x 159.6
8	15.5 dBi	319.2 x 159.6
16	18.5 dBi	319.2 x 319.2

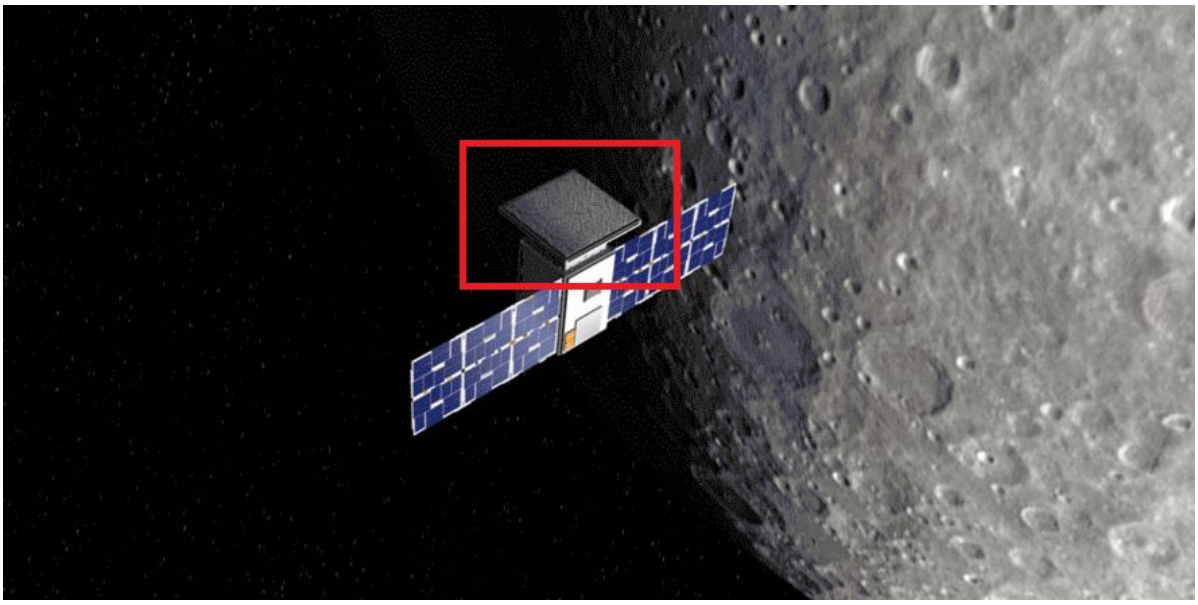


Figure 5.4: Image of CAPSTONE's antenna panel, *Illustration by NASA/Daniel Rutter* NASA (2022).

The link budget analysis plus ranging error calculations are performed using the different antenna gains from Table 5.4, where the ranging error is written as function of downlink antenna gain and down- and uplink bit rates. The downlink antenna gain is set from 6.5 to 18.5 dB, so from one to sixteen antennas. Then using the fact that the actual E_b/N_0 is 5.5 dB, the bit rate can be computed from Equation 5.9. The progression of the ranging error with respect to the downlink antenna gain is provided in Figure 5.5. The red dots represent the different antenna configurations that are considered in this research. Note that the y-axis is logarithmically scaled.

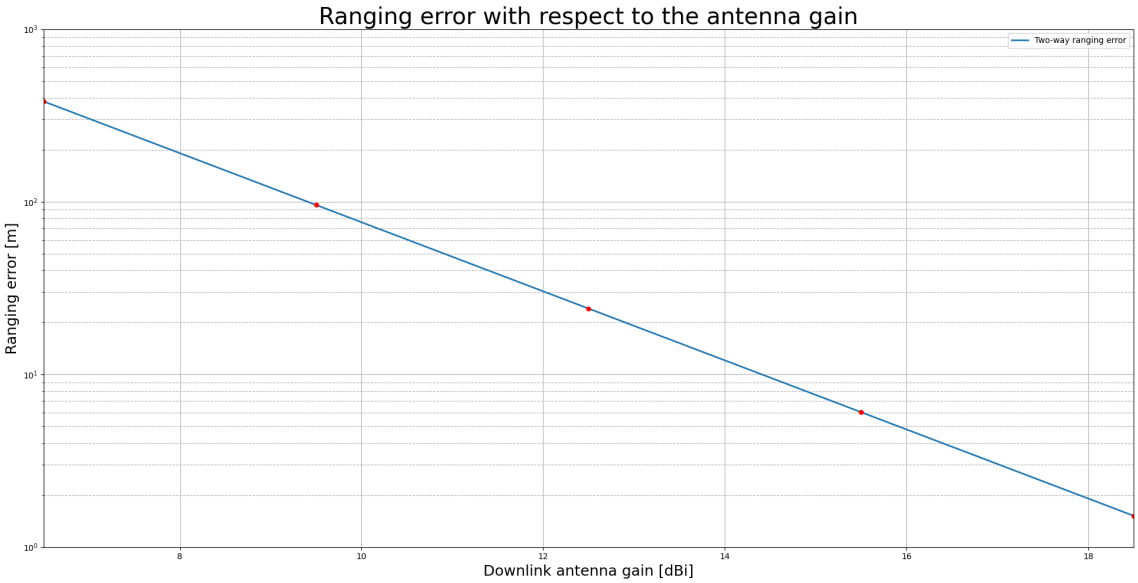


Figure 5.5: Two-way ranging error with respect to the downlink antenna gain.

Figure 5.6 provides a plot to observe the relation between the uplink and downlink bit rate and the overall ranging error. It is shown that the ranging error drastically reduces by increasing the bit rate to 2000 bps, but thereafter, increasing the bit rate has less effect on the reduction of the ranging error. Doubling the amount of antennas, the gain increases each time with 3 dB and the bit rate can be doubled leading to a reduction in ranging error by factor 4. The two-way ranging error with one antenna is 385.57 m. By doubling the antennas, the ranging error reduces by factor 4, therefore eventually the significance of this effect decreases and has smaller impact on the absolute values.

Figure 5.6: Up- and downlink bit rate with respect to the ranging error.

Figure 5.7 provides the relation between downlink gain and bit rate. This could be useful when a certain bit rate is required. One can determine what minimum downlink gain would be necessary to obtain the required bit rate and vice-versa.

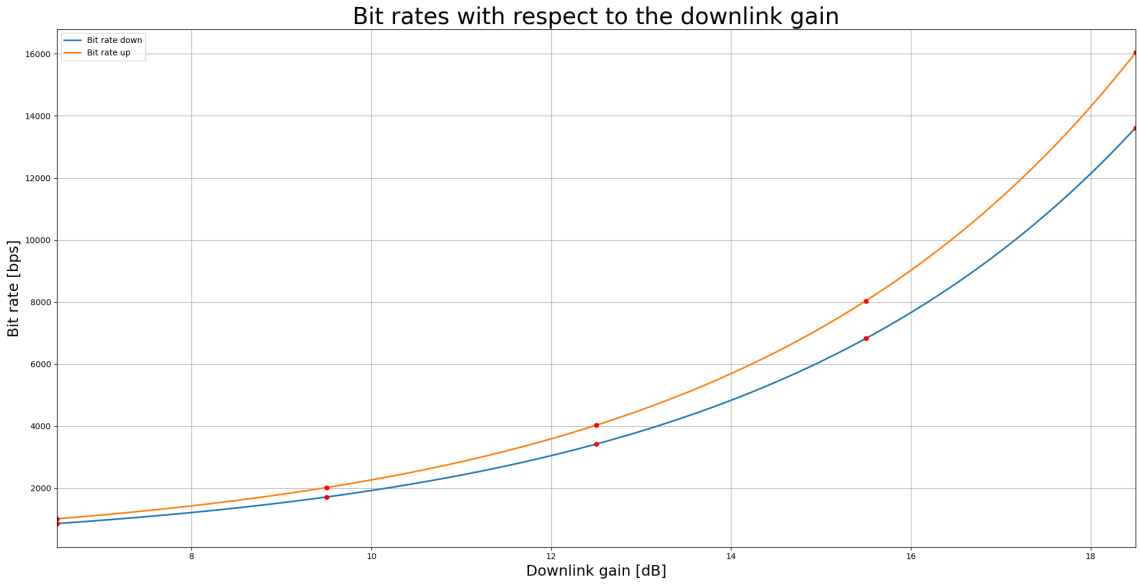


Figure 5.7: Up- and downlink bit rate with respect to the downlink gain.

The bit rate used are slightly lower than the maximum bit rate possible to ensure the bit rate numbers can be divided by 50 bps. Therefore, the margin and ranging error is slightly higher as well. These values are provided in Table 5.5 and their corresponding ranging errors are used in the AOD model and their output is compared in terms of position and velocity accuracy.

Table 5.5: Two-way ranging error based on the antenna gain and calculated bit rate.

$N_{antennas}$	**Gain** [dBi]	$bitrate_d$ [bps]	$bitrate_{up}$ [bps]	$Margin_d$ [dB]	$Margin_{up}$	σ_ρ [m]
1	6.5	850	1000	3.046	3.052	**385.6**
2	9.5	1700	2000	3.036	3.041	**96.6**
4	12.5	3400	4000	3.026	3.031	**24.2**
8	15.5	6800	8000	3.016	3.021	**6.07**
16	18.5	13600	16000	3.005	3.011	**1.52**

5.5. Measurements

The measurement input consists of the absolute inter-satellite distance plus the estimated two-way ranging error. The absolute inter-satellite distance can be calculated by i subtracting the delay between the uplink and downlink signal; ii dividing this time by two; and (iii) calculating the absolute distance by multiplying the time left with the speed of light. It is assumed that the propagation time through the transponder and therefore bias is compensated statically, as such that it therefore is not considered in this work. Additionally, bias up to 10 m has no significant effect on the performance of the autonomous orbit determination model as shown in Appendix D.

To simulate this process, the absolute inter-satellite distance is determined using the state elements provided from the dynamic model. An estimated ranging error, distributed as white Gaussian noise with a 1σ standard deviation equal to the ranging error, is added to the absolute inter-satellite distance to simulate the actual two-way telemetry ranging error. The ranging errors considered are the ranging errors shown in Table 5.5.

$$\rho = G(\mathbf{X}, t) + \sigma_{noise} = \sqrt{(x_{eml2o} - x_{elo})^2 + (y_{eml2o} - y_{elo})^2 + (z_{eml2o} - z_{elo})^2} + \sigma_{noise} \tag{5.16}$$

To include the visibility, the predefined measurements are tagged with an ID tag containing a 0 (no visibility) or 1 (visibility), which are used in the estimation filter. In case of no visibility, the estimation filter cannot perform the measurement update. If there is visibility, both the time and measurement update are performed. Figure 5.8 shows that time intervals where there is no visibility only occurs at the beginning and

end of the simulation. The visibility issues at the start and end of the simulation are due to the position of EML2O as shown in Figure 4.3. In those cases, EML2O is:

- Closest to the Moon
- Slightly above the z-axis with respect to the Moon
- Especially at simulation end time, around the y-axis with respect to the Moon

When ELO orbits at the near side of the Moon, these three factors could cause visibility issues.

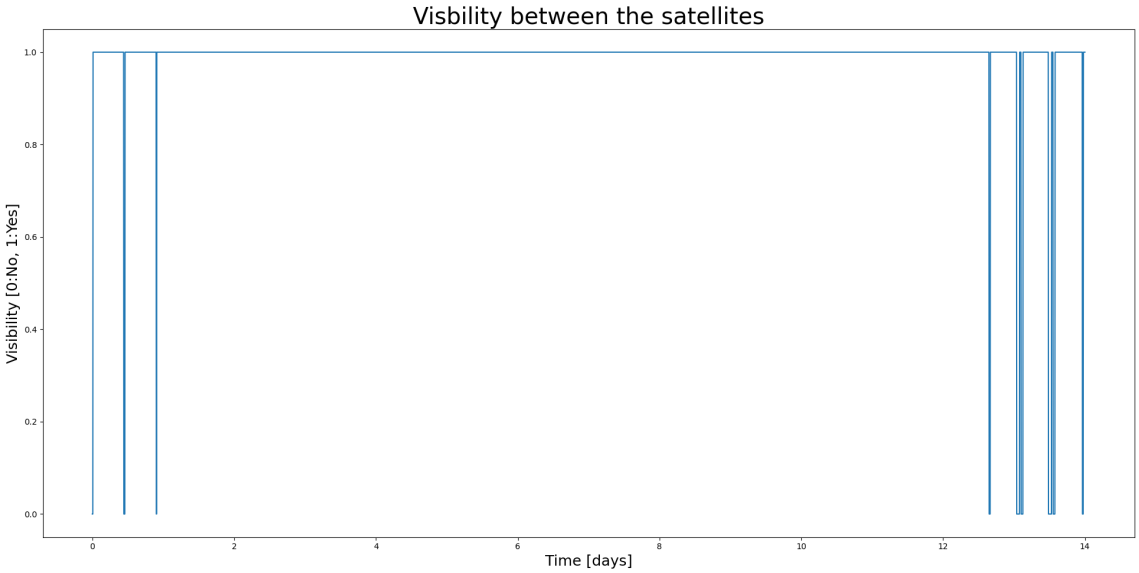


Figure 5.8: Visibility between EML2O and ELO.

6

Estimation Model

The estimation model is the final sub-model of the AOD model. This chapter discusses the required input, estimation method and all its necessary equations that are obtained from Montenbruck and Gill (2001), Tapley et al. (2004). Except for the measurement update function of the Q-matrix, which is an addition to the EKF and obtained from Akhlaghi et al. (2017).

6.1. Extended Kalman Filter

There are two types of estimation methods: sequential and batch estimations. In this research, a sequential filter is considered, which is often preferred over batch processors when on-board orbit determination is applied (Montenbruck and Gill, 2001). Sequential filters contain two advantages compared to batch processors: 1) they require less computational power and time than batch processors, 2) they can serve as real-time applications and therefore stationkeeping can be performed in an instance based on autonomous orbit determination.

As a sequential filter, the EKF is considered, which is broadly used in space applications (Gao et al., 2017). It is an extension of the regular Kalman Filter (KF) that estimates states based on linear dynamic equations. However, orbital dynamics and therefore the dynamic system in this research is non-linear. Therefore, an extension of the regular Kalman filter must be used. Possible Kalman filters are the EKF and Unscented Kalman Filter (UKF). The EKF linearizes non-linear dynamic equations of motion in order to estimate the states.

An EKF iteration consists of a time update and a measurement update. During the time update, the states and covariance are updated to the next epoch using a state integrator. The measurement update adjusts the time updated states and covariance based on inter-satellite measurements in an attempt to converge to the nominal trajectory, as shown in Figure 6.1. In chapter 5, it was determined that in some time intervals, there is no visibility. Therefore, inter-satellite measurements cannot be obtained and the measurement update cannot be performed in those epochs. When measurements are not possible due to the lack of visibility, only the time update is executed in order to update the states and covariance. This is necessary since, the state integrator is a numerical integrator and therefore dependent on the values of the previous iteration. A too long time step would lead to numerical integration errors, especially in the state elements of ELO. Due to the relative short orbital period, ELO is subject to large velocity changes in magnitude and direction, making the orbit sensitive to calculation errors using a larger time step for states integration.

In orbit, it is impracticable to measure every 10 seconds as is done to obtain the nominal states. In reality, the observations will be obtained in a variable time interval exceeding up to minutes or maybe hours, so another time step must be defined to use in the estimation filter. A fixed time step is considered, instead of a variable time step, but will be larger than the fixed time step as in the dynamic model. The time update in the EKF does not require external measurements, so can be updated in a fixed time interval. The measurement update can only be executed when an inter-satellite measurement is obtained. When operational, it is estimated that range observations can be obtained every 5 to 20 minutes. Therefore, the measurement fixed time step is set on 5 minutes[1] to be on the liberal side.

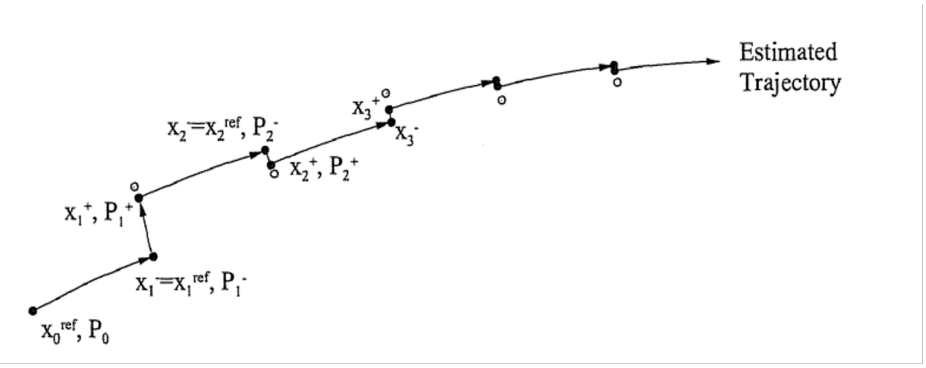


Figure 6.1: Visualization of the Extended Kalman Filter Montenbruck and Gill (2001).

6.2. Estimation setup

The estimation filter requires several input variables to run properly. This section discusses the estimation setup used in the estimation model.

Initial states

Initial states of both satellites are required for the estimation filter. The initial states, $\mathbf{X_0}$, contain state elements of both satellites

$$\mathbf{X_0} = \begin{bmatrix} x_1 & y_1 & z_1 & \dot{x}_1 & \dot{y}_1 & \dot{z}_1 & x_2 & y_2 & z_2 & \dot{x}_2 & \dot{y}_2 & \dot{z}_2 \end{bmatrix}^T \tag{6.1}$$

where $_1$ indicates EML2O and $_2$ indicates ELO. The initial state vector differs from the initial states vector of the nominal dynamic model, because an initial error is considered above the initial conditions, as shown in Equation 6.3. The initial state error is applied since the position and velocity state elements of both satellites are not exactly unknown. A limitation of the EKF with respect to a batch processor is that the initial conditions must be well-defined in order to provide convergence. Important initial condition is the initial state error, which has limitations in magnitude (Montenbruck and Gill, 2001). In Cislunar space, the estimated initial position errors are often defined between 100 m and 1000 m and the velocity errors between 1×10^{-4} m/s and 1×10^{-2} m/s (Hill and Born, 2008, Hill, 2007, Turan et al., 2022). In this research, the initial states error vector in meter and meter per second is defined as:

$$\mathbf{x_{0_{error}}} = [500 \quad 500 \quad 500 \quad 1e-3 \quad 1e-3 \quad 1e-3 \quad 500 \quad 500 \quad 500 \quad 1e-3 \quad 1e-3 \quad 1e-3]^T. \tag{6.2}$$

$$\mathbf{X_0} = \mathbf{X_{0_{true}}} + \mathbf{x_{0_{error}}} \tag{6.3}$$

Nominal measurements

The nominal measurements are obtained in the measurement model as described in chapter 5. They consist of the nominal ranging measurement and an ID-tag containing information regarding the visibility.

Initial covariance matrix

The covariance matrix is a matrix with the length and height equal to the number of initial states entering the estimation filter Equation 6.1. In this case, it is a 12x12 matrix. The initial covariance matrix is defined as a diagonal, that represents the variance of the state elements. Matrix elements outside the diagonal represents the covariance between state elements. The initial variance must be greater than the initial state errors. The root of the variance is the standard deviation (1σ) and 3σ represents a 99.7% confidence interval. If the initial errors are initially defined outside the 3σ uncertainty, then the initial covariance matrix is ill-defined.

[1] Most research (Gao et al., 2016, 2017, Hill and Born, 2008, Hill, 2007, Wang et al., 2019) do not define the measurement interval or consider a measurement interval of 60 to 100 s, which is to short to be applicable in space.

Initially, the variance of the position elements were set as the square of the initial position errors and the variance of the velocity errors to a few cm/s. In an iterative process, the values were reduced, and eventually, the initial values are

$$\mathbf{P}_0 = [10e+4\ 10e+4\ 10e+4\ 1e-2\ 1e-2\ 1e-2\ 10e+4\ 10e+4\ 10e+4\ 1e-2\ 1e-2\ 1e-2] \times \mathbf{I_{12x12}}. \tag{6.4}$$

in m^2 for the position elements and m^2/s^2 for the velocity elements.

As results from the simulations, the 3σ position- and velocity element uncertainties are plotted along the estimated state element errors. If the state errors are outside the 3σ state element uncertainties, the initial covariance matrix could be ill-defined or the initial state noise compensation matrix needs tuning.

State noise compensation matrix
The state noise compensation matrix, covariance matrix of process noise, or Q-matrix, is a matrix added to the filter to prevent filter saturation. The Q-matrix is one of the few input matrices of which its values are free to choose. However, the EKF is highly affected by the Q-matrix and poor values for elements in the Q-matrix lead to a significant downgrade in the performance of the EKF (Akhlaghi et al., 2017). The initial Q-matrix ($\mathbf{Q}_{ini}$ is defined similar as in Wang et al. (2019), which is

$$\mathbf{Q}_{ini} = \Gamma \mathbf{Q} \Gamma^T, \tag{6.5}$$

where Γ is

$$\Gamma = \begin{bmatrix} \frac{\Delta t^2}{2} \times \mathbf{I_{3\times 3}} & \mathbf{0_{3\times 3}} \\ \Delta t \times \mathbf{I_{3\times 3}} & \mathbf{0_{3\times 3}} \\ \mathbf{0_{3\times 3}} & \frac{\Delta t^2}{2} \times \mathbf{I_{3\times 3}} \\ \mathbf{0_{3\times 3}} & \Delta t \times \mathbf{I_{3\times 3}} \end{bmatrix} \tag{6.6}$$

and $\mathbf{Q}$ is defined as in Equation 6.7. By examining the effect of the 10^n outside the matrix, the magnitude of the order is determined in an iterative process. Thereafter, the factor was adjusted in an attempt to fine tune the Q-matrix.

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \cdot 5 * 10^{-10} \tag{6.7}$$

To further improve the Q-matrix, the Q-matrix is updated based on the previous measurement update. The measurement update of the Q-matrix is given in Equation 6.8, where α is the weight factor. A weight factor of 1 means a constant $\mathbf{Q}$ over time and the lower α gets, the more adaptive $\mathbf{Q}$ becomes Akhlaghi et al. (2017). In this research, α is set on 0.3 as is done in Akhlaghi et al. (2017).

$$\mathbf{Q}_{k+1} = \alpha \cdot \mathbf{Q}_k + (1-\alpha)\mathbf{K}_k \mathrm{y}_k \mathrm{y}_k \mathbf{K}_k^T \tag{6.8}$$

Measurement noise covariance
The measurement noise or weight observation is the last required input for the EKF. The measurement noise equals the square of the ranging error, which makes it dependent of the number of antennas. The measurement noise is dependent on the type of measurement. In case a range-rate measurement is obtained, the measurement noise is a scalar which equals the square of the estimated range-rate error. However, only range observations are considered, so the measurement noise is fixed to R = σ^2_{noise}, where σ_{noise} is defined as in Table 5.5. In case of two measurement types, the measurement noise is still handled as a scalar, since it is assumed that it is unlikely to happen that two measurement types, e.g., range and range-rate are obtained at exactly the same time.

6.3. Estimation filter

In this section, the the estimation filter is discussed. First, the time update in the EKF is described, which provides the time updated states, state transition matrix and time updated covariance matrix. Second, the measurement update is described, in which the observations are used to correct the time updated states. The

state integrator used in the EKF differs from the nominal dynamic model. Last, the state integrator used for the time update is described and discussed as well.

For both time update and measurement update, the sequence of operations are described. In the equations, two notations are used to indicate time and measurement updates. In case the subscript is $_{k-1}$, the variable is not time updated yet and when the subscript is $_k$, the variable is time updated. In case the superscript is an $^-$, the variable is not measurement updated and in case of a superscript of $^+$, the variable is measurement updated.

The simulation starts by initializing the variables needed to perform the estimation loop. For the first iteration:

- $\mathbf{X}_{k-1}^- = \mathbf{X}_0$
- $\mathbf{P}_{k-1}^- = \mathbf{P}_0$

6.3.1. Time update

The time update consists of integrating the states, state transition and covariance matrix to the next time step k. The states and state transition matrices are integrated using the state integrator[2] as described as in section 6.4. In Equation 6.9, $\mathbf{X}_k^-$ is the time updated states vector, t_k is the next time step and $\mathbf{X}_{t_{k-1}} = \mathbf{X}_{k-1}$ is the previous states vector.

$$\mathbf{X}_k^- = \mathbf{X}(t_k; \mathbf{X}_{t_{k-1}} = \mathbf{X}_{k-1}) \tag{6.9}$$

The state transition matrix $\Phi_{k,k-1}$ is determined by the dynamic simulator, describes the transformation from x_{k-1} to x_k and is required to time update the covariance matrix Equation 6.10.

$$\mathbf{P}_k^- = \Phi_{k,k-1}\mathbf{P}_{k-1}\Phi_{k,k-1}^T + \mathbf{Q}_k \tag{6.10}$$

The last step of the time update is to read the ID-tag of the nominal observation at time k (Y_k), which initializes at the start of each iteration. If the ID-tag equals 0, there is no line-of-sight visibility between the satellites. The ranging operations are not executed and a new iteration starts with $\mathbf{X}_{k-1} = \mathbf{X}_k^-$, $\mathbf{P}_{k-1} = \mathbf{P}_k^-$. If ID-tag equals 1, the measurement update is executed.

6.3.2. Measurement update

First, the observation-state matrix $\tilde{\mathbf{H}}_k$ is determined using Equation 6.11. The observation state matrix is the partial derivative of the observation function. In case of ranging observations, the observation-state matrix $\tilde{\mathbf{H}}_k$ is the partial derivative of Equation 5.16. Second, the difference between the nominal trajectory and measurements is determined, as shown in Equation 6.12.

$$\tilde{\mathbf{H}}_k = \left[\frac{\partial G}{\partial \mathbf{X}}\right]_k^* = \left[\begin{array}{cccccccccccc} \frac{\partial\rho}{\partial x_1} & \frac{\partial\rho}{\partial y_1} & \frac{\partial\rho}{\partial z_1} & \frac{\partial\rho}{\partial \dot{x}_1} & \frac{\partial\rho}{\partial \dot{y}_1} & \frac{\partial\rho}{\partial \dot{z}_1} & \frac{\partial\rho}{\partial x_2} & \frac{\partial\rho}{\partial y_2} & \frac{\partial\rho}{\partial z_2} & \frac{\partial\rho}{\partial \dot{x}_2} & \frac{\partial\rho}{\partial \dot{y}_2} & \frac{\partial\rho}{\partial \dot{z}_2} \end{array}\right]_k \tag{6.11}$$

$$y_k = \mathrm{Y}_{k_{nom}} - \mathrm{G}(\mathbf{X}_\mathbf{k}^-, t_k) \tag{6.12}$$

Second, the Kalman gain used to weight the observations, is computed. The higher the Kalman gain, the higher the weight of the observation and therefore a larger difference in the time updated states and the measurement updated states.

$$\mathbf{K}_k = \mathbf{P}_k^-\tilde{\mathbf{H}}_k(\tilde{\mathbf{H}}_k\mathbf{P}_k^-\tilde{\mathbf{H}}_k^T + R_k)^{-1} \tag{6.13}$$

Last, the measurement update is performed by updating the states, covariance matrix and the Q-matrix using Equation 6.14 for the states, Equation 6.15 for the covariance matrix and Equation 6.8 for the Q-matrix.

$$\mathbf{X}_k^+ = \mathbf{X}_k^- + \mathbf{K}_k \cdot y_k \tag{6.14}$$

$$\mathbf{P}_k^+ = (\mathbf{I}_{\mathbf{12\times 12}} - \mathbf{K}_k\tilde{\mathbf{H}}_k)\mathbf{P}_k^- \tag{6.15}$$

Last, the updated variables are initialized for the new iterations such that:

- $\mathbf{X}_{k-1}^- = \mathbf{X}_k^+$

[2] The state integrator differs from the state integrator in the dynamic model. Therefore a separate section (section 6.4) is dedicated to the state integrator in the EKF

- $\mathbf{P}_{k-1}^{-} = \mathbf{P}_{k}^{+}$
- $\mathbf{Q}_{k+1} = \mathbf{Q}_{k}$

In the simulation, this loop continues until the end of the simulation time is reached. In reality, this loop can be executed in real-time until a position error with respect to the desired nominal orbit is reached. Then a stationkeeping manoeuvre could be triggered. However, this is not part of this research, but would be interesting in future research.

6.3.3. Output

From the simulation, the estimated states and the standard deviation (1σ) is extracted and saved to visualize the results. The estimated states are saved in an array to subtract from the nominal states to determine the estimated state errors. The standard deviation is obtained from the covariance matrix at each iteration to provide the 3σ uncertainty per state element, which is done by taking the square root of the variance elements of the covariance matrix. The simulation figures in chapter 7 contain both the estimated state errors and their 3σ uncertainties serving as out-boundaries of the state element errors to prove that the estimation filter presents data that are valid.

6.4. State integrator

The state integrator numerically integrates the states to the next time step and obtains the state transition matrix. The state transition matrix is the matrix describing the motion of $\mathbf{X}_{k-1}$ to $\mathbf{X}_{k}$ and is required for the time update of the covariance matrix. The state transition matrix is obtained separately for each satellite and is merged in one total state transition matrix with the form Equation 6.16. The states are updated similar as in the nominal dynamic model. However, the environment created for the estimator differs from the high fidelity environment as described as in subsection 4.2.2, because such high spherical harmonics and certain celestial bodies is not necessarily required. As was also shown in Table 4.3, subsection 4.3.1.

$$\Phi_{k,k_{k-1}} = \begin{bmatrix} \Phi_{eml2o_{k,k-1}} & \mathbf{0_{6x6}} \\ \mathbf{0_{6x6}} & \Phi_{elo_{k,k-1}} \end{bmatrix} \tag{6.16}$$

Based on the acceleration breakdowns for EML2O and ELO in Figure 4.5 and Figure 4.4 several celestial bodies are not considered due to their contribution to the acceleration function. Celestial bodies not considered are: Neptune, Uranus, Saturn and Mars, meaning that the Sun (including SRP), the Moon, Earth, Mercury, Venus and Jupiter are implemented in the integration environment. In the dynamic model a maximum order and degree of 200 for the spherical harmonics of the Earth and the Moon were considered. However, it was shown in Table 4.3 that such high order and degree are not necessary in order to provide accurate states.

The propagation and integration settings are defined within the estimation loop, because these input settings are dependent of the current states and ET and they are updated each iteration. As mentioned in section 6.1, the iteration time step is 300 s (5 minutes). A RK4 integrator is used and as simulator tudatpy' 'SingeArcVariationalSimulator' to determine the states and the state transition matrix. However, six state integrations are required in order for the SingleArcVariationalSimulator to determine the state transition matrix. Otherwise an insufficient number of data points (epochs) occur in order to create a Lagrange interpolator. As base, the integration time step is set on 50 s, so that six state integrations can be executed within one measurement time step.

To define the spherical harmonics for Earth and the Moon in the estimators integrator, the computation time and model errors must be considered. In Table 4.3 the difference in trajectory due to spherical harmonic gravity of the Earth and the Moon was investigated. but a different time step was not taken into account yet. Since a numerical integrator is applied, the different time step can cause model errors. To determine the state integrator settings, first, the computation time is investigated and, second, the model errors are discussed.

6.4.1. Computation time

When applying this method in real-time, a hard requirement is that the computation time must be faster than the applied time step. The computation time per iteration is determined for different state integrators that includes the Sun (incl. SRP), the Moon, Earth, Mercury, Venus and Jupiter in the integration environment.

The difference between the integration settings are the order and degree of the spherical harmonics of Earth and the Moon and the goal is to obtain knowledge about the correlation between computation time and the values of the order and degree of the spherical harmonics. The same values for the order and degree are used as in Table 4.3 except for the point masses, since it was found that the 3-D position errors are too large with respect to the nominal trajectory.

Two days of simulation time are executed for each state integrator to determine the required computation time per iteration of solely the estimation filter, excluding initialization processes. Two days of simulation equals 576 iterations and during the computation an average processor speed of 3.5 GHz was measured. Considering an ISISpace OBC, a processor speed of 400 MHz is available. The computation time is inversely proportional to the processor speed as in Equation 6.17, so by determining the computation time on this computer, the computation time in-space can be calculated.

The calculation time per iteration of this computer is calculated by dividing the amount of iterations by the computation time. Using Equation 6.17, the computation time per iteration of an OBC can be determined, where T_{obc} is the computation time per iteration of the OBC, T_{pc} is the computation time per iteration of this computer, and CS_{obc}, CS_{pc} are the processor speed.

$$\frac{T_{obc}}{T_{pc}} = \frac{CS_{pc}}{CS_{obc}} \quad \rightarrow \quad T_{obc} = \frac{CS_{pc}}{CS_{obc}} \cdot T_{pc} \tag{6.17}$$

Table 6.1: Computation time per iteration of both the pc and OBC for different spherical harmonics for Earth and Moon.

Spherical harmonics Earth	**Spherical harmonics Moon**	**2 day computation time [s]**	**Computation time per iteration (pc) [s/it]**	**Computation time per iteration (OBC) [s/it]**
200, 200	200, 200	641.9	1.114	9.751
100, 100	100, 100	129.9	2.256e-1	1.974
50, 50	50, 50	32.50	5.641e-2	4.936e-1
40, 40	40, 40	23.22	4.031e-2	3.527e-1
30, 30	30, 30	14.02	2.434e-2	2.129e-1
20, 20	20, 20	7.025	1.220e-2	1.067e-1
10, 10	100, 100	84.87	1.473e-1	1.289
2, 2	20, 20	5.468	9.493e-3	8.306e-2

Table 6.1 shows the computation time per iteration. Every model would comply with the required time to perform an iteration, so simplification of the model is not necessarily required. However, the model errors have to be discussed before making choices regarding the state integrator in the estimation model.

6.4.2. Model error

In this section the difference with respect to the nominal states is investigated due to a different integration environment and time step. This is done by propagating the nominal initial states by excluding the initial estimated states error and measurements in the estimator filter. The maximum error between the states represents the model error and is determined for each spherical harmonic gravity configuration as described in Table 6.1. Table 6.2 shows the model error of each configuration using the 300 s time step including 6 integrations in the state integrator.

The model error of EML2O remains constant at 53 m, regardless of the spherical harmonic gavity settings of the Moon and Earth. What is striking is that the model error for ELO enlarges when the maximum order and degree of the spherical harmonic gravity of the Moon increases above 20 and remains constant from an order and degree of 40. One would expect a decrease in model error when the order and degree of spherical harmonic gravity is increased for the Moon. Decreasing the order and degree for the Moon lead to an even higher model error, as shown in Table 6.2.

Table 6.2: Model error of the different spherical harmonic gravity configurations.

Spherical harmonics Earth	**Spherical harmonics Moon**	**3-D position model error EML2O [m]**	**3-D position model error ELO [m]**
200, 200	200, 200	53.1	799.7
100, 100	100, 100	53.1	799.7
50, 50	50, 50	53.1	799.7
40, 40	40, 40	53.1	798.4
30, 30	30, 30	53.1	771.7
20, 20	20, 20	53.1	555.7
10, 10	100, 100	53.1	799.7
2, 2	20, 20	53.3	555.7
2, 2	15, 15	53.3	2835.5
2, 2	10, 10	53.3	16421.6

The model error is explained by the difference in time step, since the numerical integrator is sensitive to the time step. Increasing the maximum order and degree of the spherical harmonic gravity of the Moon is counterproductive using the given integrator time step of 50 s. To answer sub-question 3 of the research questions, the integrators time step need to be adjusted in order to obtain a model error of less than 100 m.

6.4.3. Integrator time step

Based on the values in Table 4.3 and Table 6.2 a maximum order and degree of 2 for Earth's spherical harmonic gravity is sufficient. In Table 4.3 it is observed that from a maximum order and degree of 40 for the spherical harmonics of the Moon, the accuracy difference stalls. The same can be observed in Table 6.2, where the model error remains constant from an order and degree of 40 for the Moon. Therefore, the spherical harmonic gravity of the Earth and Moon are modelled as 2, 2 and 40, 40 respectively. By adjusting the time step within the state integrator, the time step is determined for which a model error of less than 100 m holds.

It has been discovered that if the state integrator updates every 30 s, the maximum model errors for the EML2O and ELO are 53.3 m and 66.7 m respectively. The total simulation of 14 days, consisting of 4033 iterations, was executed in 160.4 s. This gives a computation time per iteration of 3.977×10^{-2} s for this computer (3.5 GHz) and for an ISISpace OBC (400 MHz) the computation time per iteration is 3.480×10^{-1} s. Therefore, the chosen time and spherical harmonic gravity settings hold the computation requirement.

6.4.4. State integrator settings and its model error

The settings of the state integrator used in the estimation filter are provided in Table 6.3.

Table 6.3: State integrator settings in the estimation filer.

State integrator setup	
Measurement time[3]	300 s
State update time	30 s
Reference frame origin	Earth-centered
Reference frame orientation	J2000
Integrator	RK4 integrator
Dynamic simulator	Single arc variational simulator
Acceleration settings (both EML2O and ELO)	
Earth	Spherical harmonic gravity (2, 2)
Moon	Spherical harmonic gravity (40, 40)
Sun	Point mass gravity & SRP
Mercury	Point mass gravity
Venus	Point mass gravity
Jupiter	Point mass gravity

[3]Given visibility

Using the integrator settings from Table 6.3, Figure 6.2 and Figure 6.3 show the progression of the model error for both the EML2O and ELO. The model error of EML2O slowly exponentially grows over time to 53.3 m. The progression of the model error of ELO grows periodically. The peaks in 3-D position and velocity errors occur when ELO is in its periapsis, which is where ELO experiences its highest velocity in magnitude. Due to the different integration environments, the acceleration functions differ. Therefore, the parts of the orbit with high velocity changes are most vulnerable to states errors. The next chapter describes the simulations and results obtained from the autonomous orbit determination model based on the estimator settings defined in this chapter.

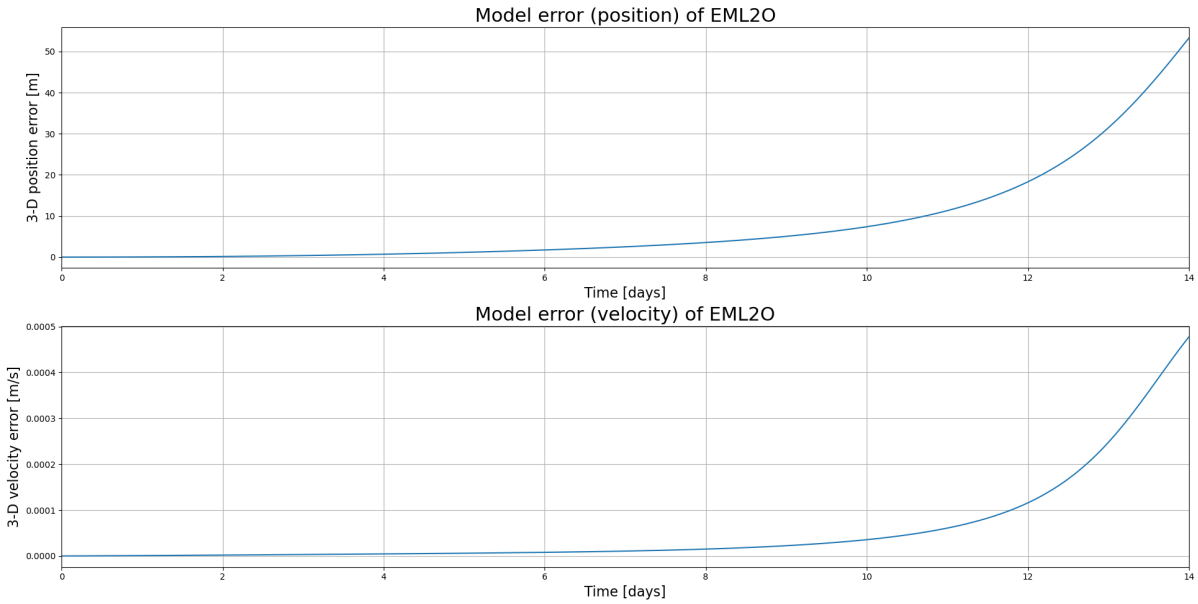


Figure 6.2: 3D-position and velocity errors of EML2O due to a different state integrator with respect to the dynamic model.

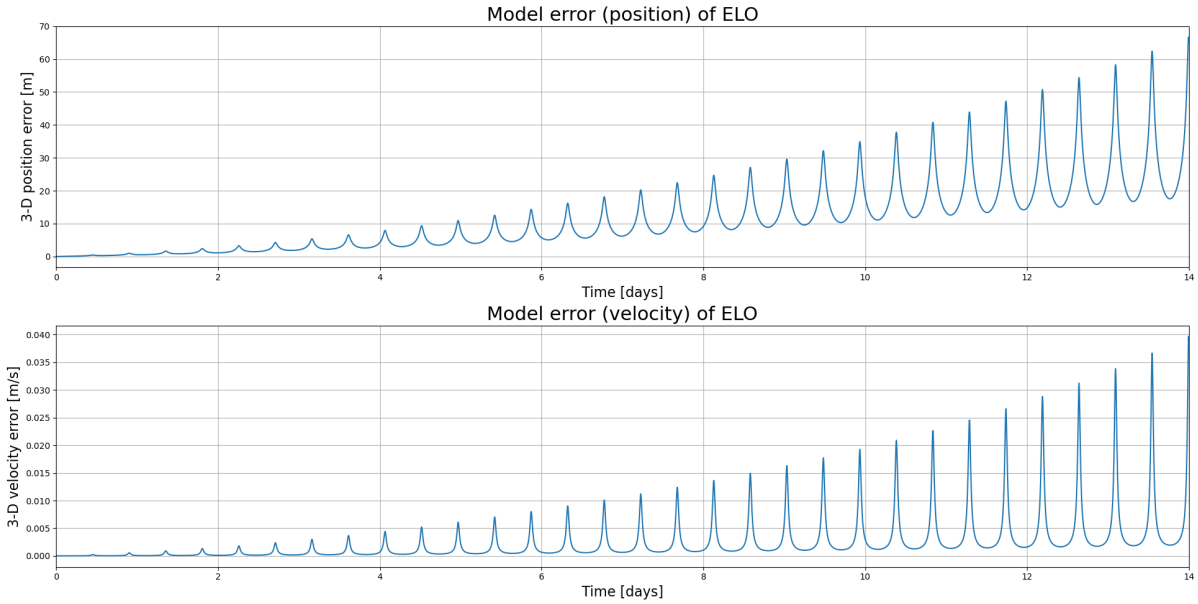


Figure 6.3: 3D-position and velocity errors of ELO due to a different state integrator with respect to the dynamic model.

7

Simulations and results

This chapter displays and discusses the simulation and results obtained from the autonomous orbit determination model. First, the progression of the state elements and their 3σ uncertainties are provided and discussed. Second, the progression of the 3-D state estimation accuracy is displayed and discussed with respect to LUMIOs mission requirements. Last, the average estimated 3-D estimation accuracy is provided and discussed. By 3-D estimation accuracy, the 3-D 3σ uncertainty of both position and velocity is meant. The position and velocity requirements for LUMIO are COMMS-070 and COMMS-070*, stating a required position and velocity accuracy of 1000 m and 1 cm/s respectively.

7.1. Simulation results

The simulation results are shown for each antenna array configuration using the settings of the estimation filter as defined in chapter 6. The figures show the progression of the estimated state element errors along with their 3σ uncertainties. When referring to a state element error, the notations x_n is used and for the 3σ uncertainties, $3\sigma_n$ is used with $n = x, y, z, \dot{x}, \dot{y}, \dot{z}$. Note that the very start of the simulation the estimator is still 'searching' for convergence. Therefore, the y-axis limits are set to provide view on the whole progression leaving some peaks at the start of the simulation out.

7.1.1. Simulation of the EML2O

Figure 7.1 until Figure 7.5 show the progression of the estimated errors and 3σ's of EML2O. In each configuration, the errors have a similar progression over the simulation span in terms of shape. The magnitude decreases with a decreasing ranging error. In 1 antenna configuration, the $3\sigma_y$ and $3\sigma_z$ converge just short of 1000 m, while in a 2 antenna configuration the 3σ's are below 1000 m after 10 to 11 days of simulation time. For 4 antennas, this already happens at day 8, for 8 antennas at day 6 and for 16 antennas on day 4 of simulation time. There seems to be a trend in doubling the amount of antennas to decreasing the amount of days by 2 to provide the 3σ's below 1000 m.

Regarding velocity, the 1 antenna configuration is not able to obtain a 3σ velocity accuracy of below 1 cm/s, but from antenna configuration 2 the velocity requirement COMMS-070* of LUMIO is met shortly after the simulation starts. Besides antenna configuration 8 and 16, the $3\sigma_{\dot{y}}$ forms a 'belly' between day 4 and 10 of simulation time, which flattens each configuration with a lower ranging error.

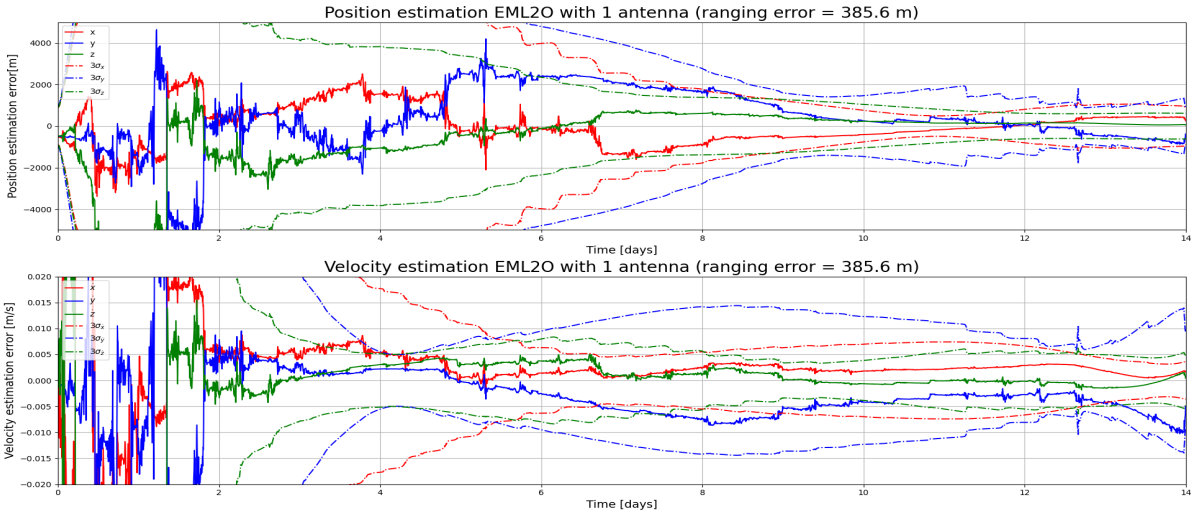


Figure 7.1: State estimation of EML2O using 1 antenna.

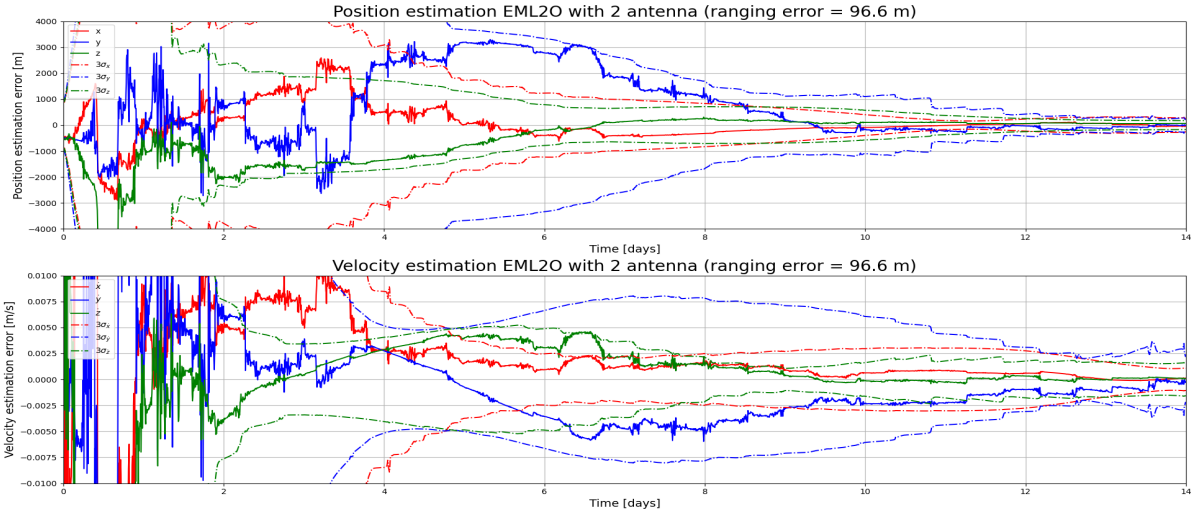


Figure 7.2: State estimation of EML2O using 2 antennas.

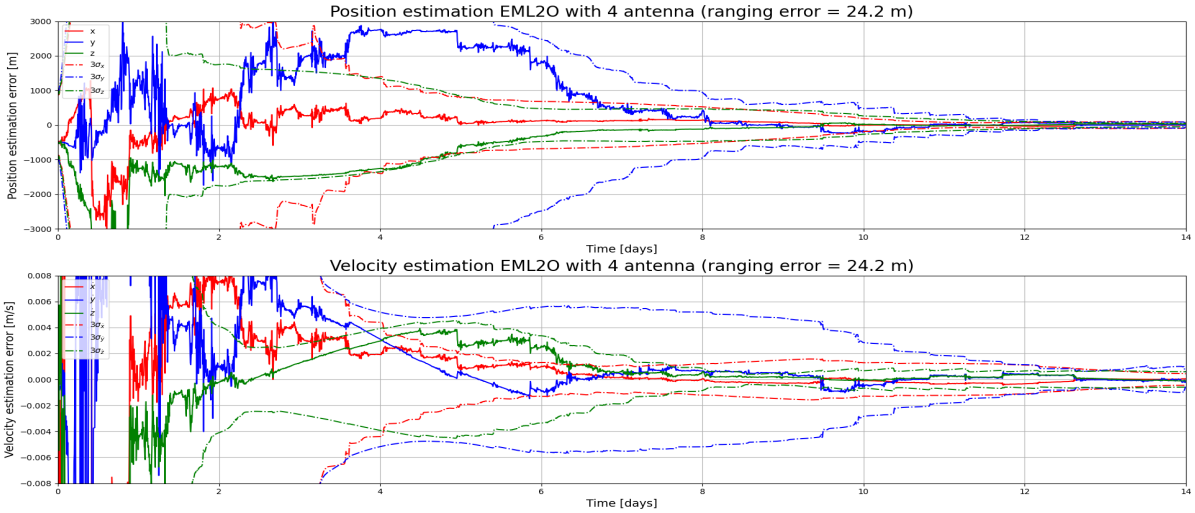


Figure 7.3: State estimation of EML2O using 4 antennas.

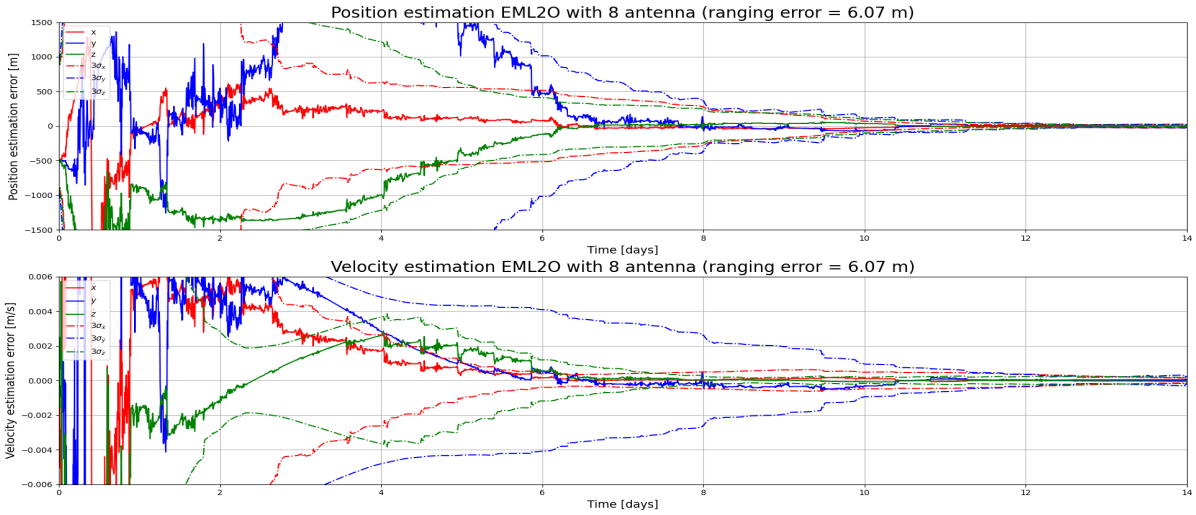


Figure 7.4: State estimation of EML2O using 8 antennas.

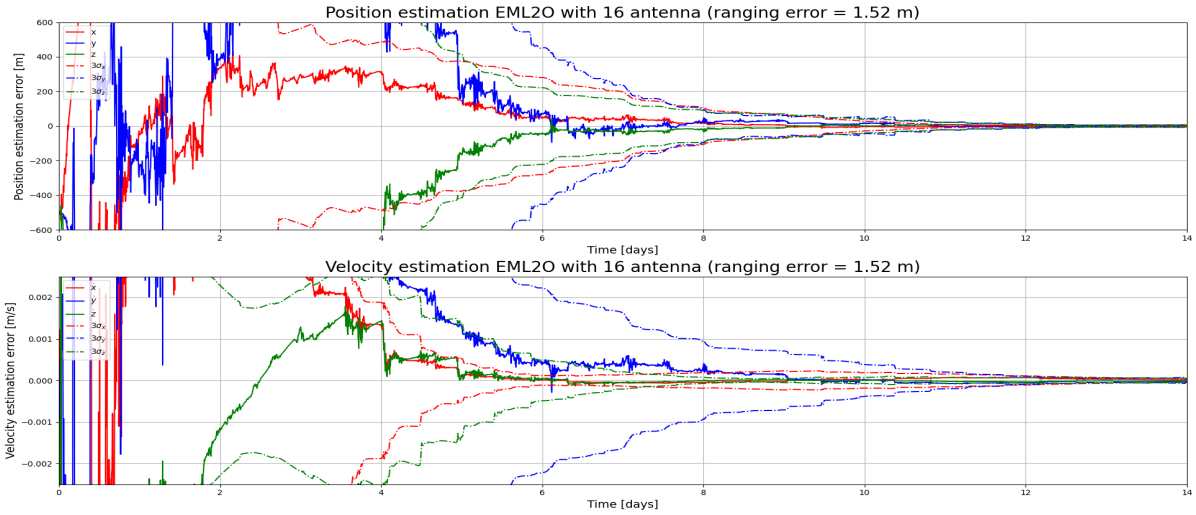


Figure 7.5: State estimation of EML2O using 16 antennas.

7.1.2. Simulation of ELO

Figure 7.6 until Figure 7.10 shows the progression of the estimated errors and uncertainties of ELO, which are on average higher than for EML2O. During the simulation period, EML2O finishes approximately one halo orbit, while ELO finishes a full revolution every 10.5 hours, leading to a different shape of the estimated errors and uncertainties. ELOs progression is periodic and the velocity uncertainties peak in its perilune, while the position estimation improves in its perilune for configuration 8 and 16. This effect is less visible for lower configurations, due to relatively higher ranging errors. As shown in Figure 6.3, the model error in the apoapsis remains below 20 m, while in its perilune the model error peaks as well to a maximum of 66.7 m. Both high model errors and high velocity in the perilune lead to high peaks in estimated errors and 3σ's.

As well as for EML2O, the estimated errors and 3σ's decrease with a decreasing ranging error. The magnitude differs however. Especially the $3\sigma_y$ and its velocity component are relatively high with respect to the other 3σ's. The same is observed in the figures for EML2O where the estimated error in y-direction and $3\sigma_y$ are the dominant error and uncertainty.

In general, the velocity errors for ELO are higher than for EML2O. The velocity errors are highest near its periapsis due to the high magnitude in velocity and the high changes in direction of the satellite. Moving out of the perilune, the magnitude of the errors and uncertainties decreases along with ELOs velocity. For ELO,

no specific position and velocity requirements are defined to comply to, since ELO is used as the ranging subject for EML2O. However, the plots are provided to observe the differences in progression of the errors and uncertainties due to a different orbit.

In all simulation figures of ELO relatively high peaks are shown around 3 and 10 days of simulation time. The tops of the peaks are not plotted since large parts of the progression will become less visible. However, they are shortly discussed. Over the simulation time, the y-element error and its 3σ are dominant. Except for the peaks around 3 and 10 days which are mostly caused by $3\sigma_x$ and $3\sigma_z$. These reoccurring peaks are related to the relative geometry of the two satellites. During these periods, the inter-satellite distance remains almost constant at approximately 72 000 km. Therefore, subsequently inter-satellite measurements are constant as well leading to a reduction in performance of LiAISON based orbit determination, since it requires no similarities in size, shape and orientation and constant measurements lead to an increase in uncertainty.

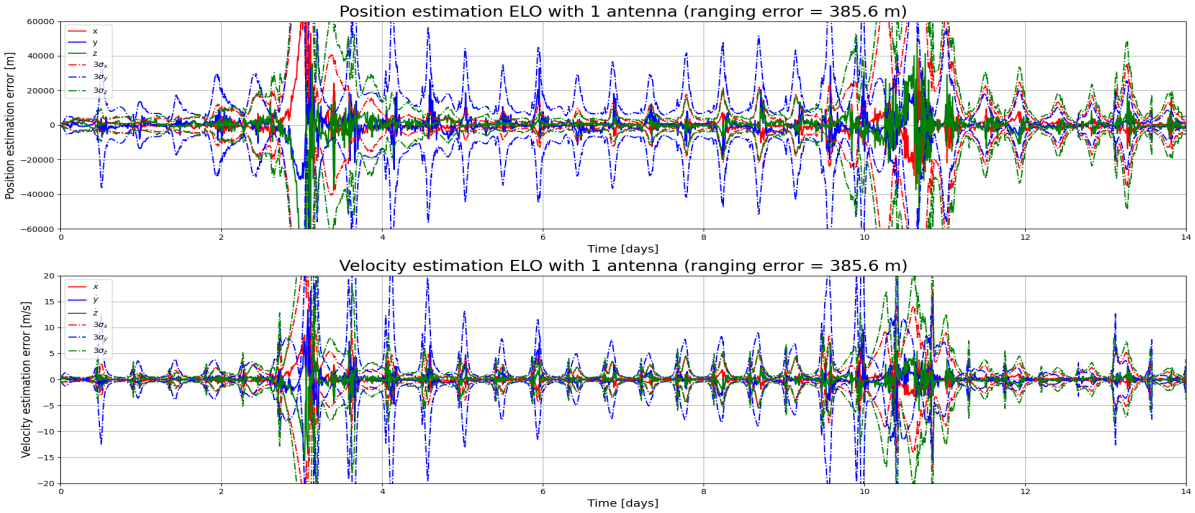


Figure 7.6: State estimation of ELO using 1 antenna.

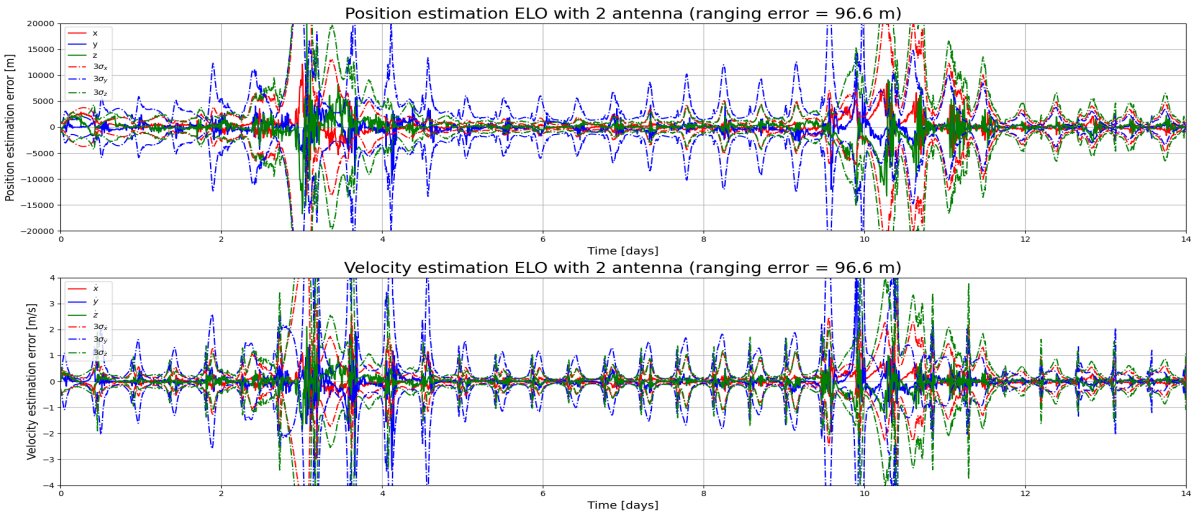


Figure 7.7: State estimation of ELO using 2 antennas.

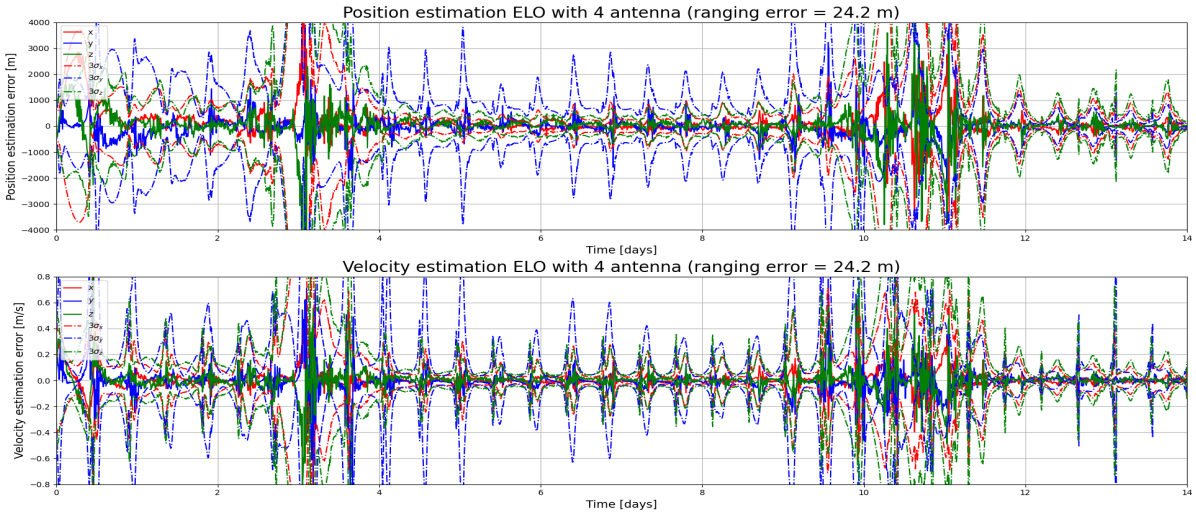


Figure 7.8: State estimation of ELO using 4 antennas.

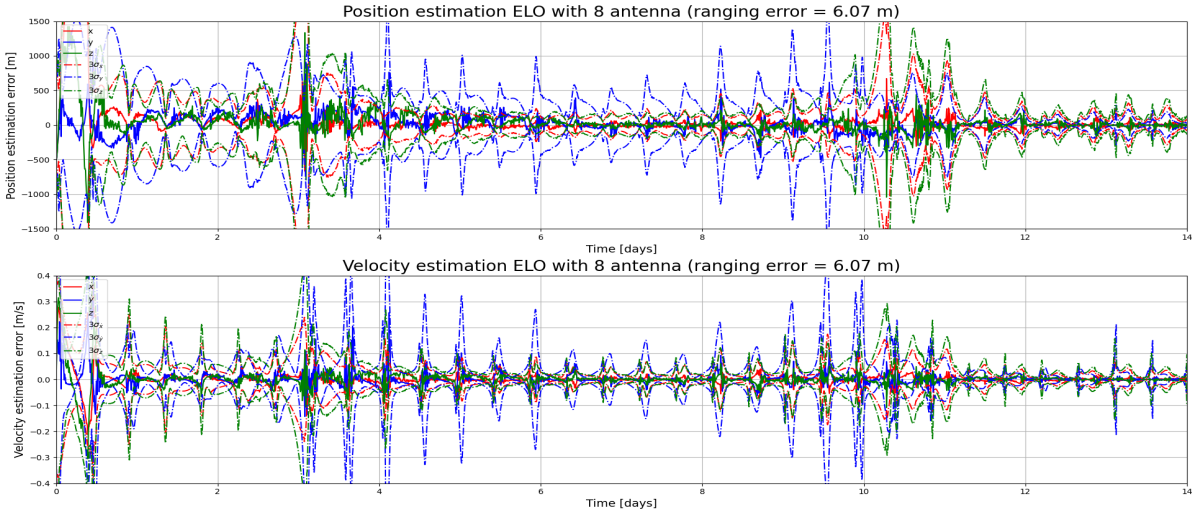


Figure 7.9: State estimation of ELO using 8 antennas.

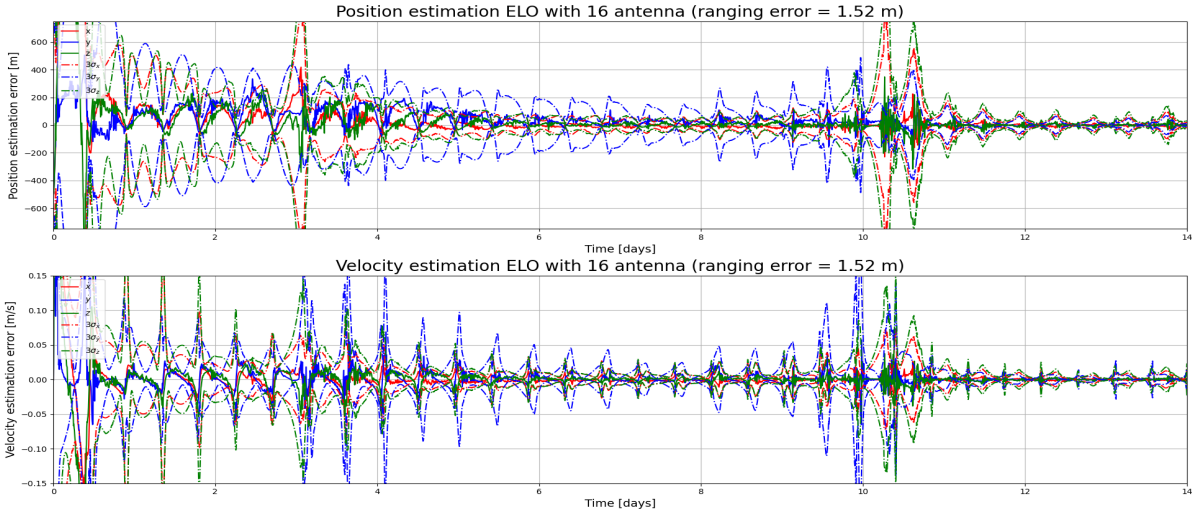


Figure 7.10: State estimation of ELO using 16 antennas.

7.1.3. Discussion of the simulations

Five simulations were carried out with a different ranging error and model error of 53.3 m and 66.7 m for EML2O and ELO. Due to different trajectories of both satellites, the distribution of the estimated error differs in shape and size. The estimated errors and 3σ's decrease with a decreasing ranging error, making it possible to comply with LUMIO's position and velocity requirements COMMS-070 and COMMS-070*, which will be further elaborated in section 7.2.

In the simulations a constant ranging error is considered that was calculated based on worst case scenario, which is at maximum distance. The inter-satellite distance differs over time and therefore the SNR varies, leading to a variable ranging error. However, a variable ranging error is not considered in this research, but could lead to better performances of the estimation filter. Both a variable ranging error and a reduction in model error could improve the performance of the estimations. The model error could be reduced by improving the estimation integrator, but would lead to a larger computation time. This research set the limit a maximum model error of 100 m for both satellites, which is sufficient in achieving a 3-D position and velocity uncertainty of 1000 m and 1 cm/s. However, implementing a reduction in model error by taking into account computation time and a variable ranging error could be interested to look into in future research.

7.2. Results

The estimated state element errors, are possible errors during the simulation. However, due to the normal distributed ranging error, the estimated state errors differ in every simulation and thus offer little guidance in determining the absolute 3-D position and 3-D velocity accuracy. Therefore, the progression of the 3σ's of 3-D position and velocity are defined as the 3-D state estimation accuracy and plotted over the simulation time. By including the position and velocity requirements for LUMIO, it is determined which configuration meets the mission requirements.

Additionally, the average 3-D estimation accuracy for position and velocity from 6 days to simulation end time are provided for both EML2O and ELO. The time epoch of 6 days is chosen for two reasons. First, the position error accuracy of EML2O strongly converges until 6 days. Second, the duration of the orbit of EML2O is approximately 14 days and will perform stationkeeping every 7 to 14 days. Therefore, determining the average 3-D estimation accuracy before 7 days of simulation time seems the most useful epoch.

7.2.1. EML2O: 3-D estimation accuracy

The progression of the 3-D estimation accuracy for the five configuration shows similarities and the distance between the progressions of each configuration grows over the simulation time. The 3-D position estimation accuracy of each configuration deviate until 1 day of simulation time, thereafter the curves start to converge until 13 days of simulation time and during the last day of simulation, the lines flatten or slightly diverges. Given this knowledge, one could consider to reduce measurement time to 13 days, since the accuracy does not improve afterwards.

In Figure 7.11 it is shown that configuration 1 converges to a 3-D position estimation accuracy higher than the position requirement and therefore the basis radio parameters are not sufficient to perform LiAISON based orbit determination. All other four configurations are capable of obtaining a 3-D position estimation accuracy of less than 1000 m. Using an array of 2 antennas, this occurs at day 10.8, with 4 antennas at day 8.1, with 8 antennas at day 6.3 and with 16 antennas at day 5.1.

The 3-D velocity estimation accuracy requirement is met in a shorter amount of time than the position requirement. From an array with 2 to 16 antennas this requirement is met at day 4.1, 3.4, 2.8, and 2.3. The ranging error using 1 antenna is however too large to converge below the velocity requirement. The configurations converge relatively fast to near the velocity requirement, but thereafter the values remain constant for 1, 2, 4 antennas. From day 10, the 3-D velocity estimation accuracy reduces further and from day 13 the values remain constant or even slightly diverge. Table 7.1 provides the simulation time required to meet LUMIOs position and velocity accuracy requirements.

Table 7.1: Simulation time required to comply to COMMS-070 and COMMS-070*.

No. of antennas	Ranging error	T_{req} for COMMS-070	T_{req} for COMMS-070*
1	385.6 m	N/A	N/A
2	96.6 m	10.8 days	4.1 days
4	24.2 m	8.1 days	3.4 days
8	6.07 m	6.3 days	2.8 days
16	1.52 m	5.1 days	2.3 days

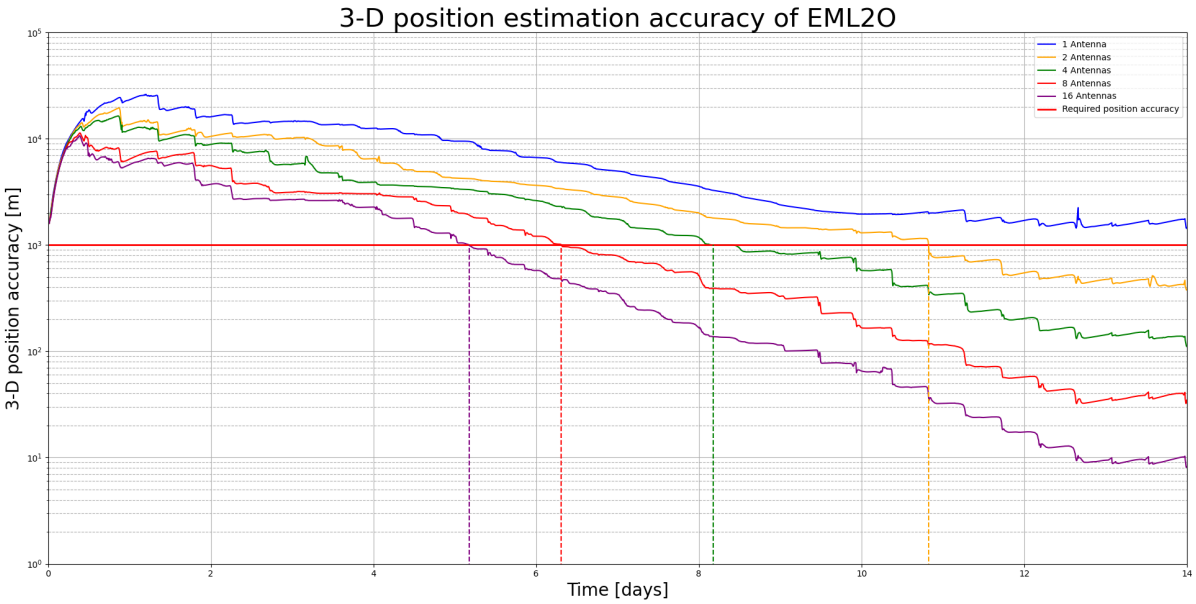


Figure 7.11: The 3-D position estimation accuracy of EML2O of each antenna configuration.

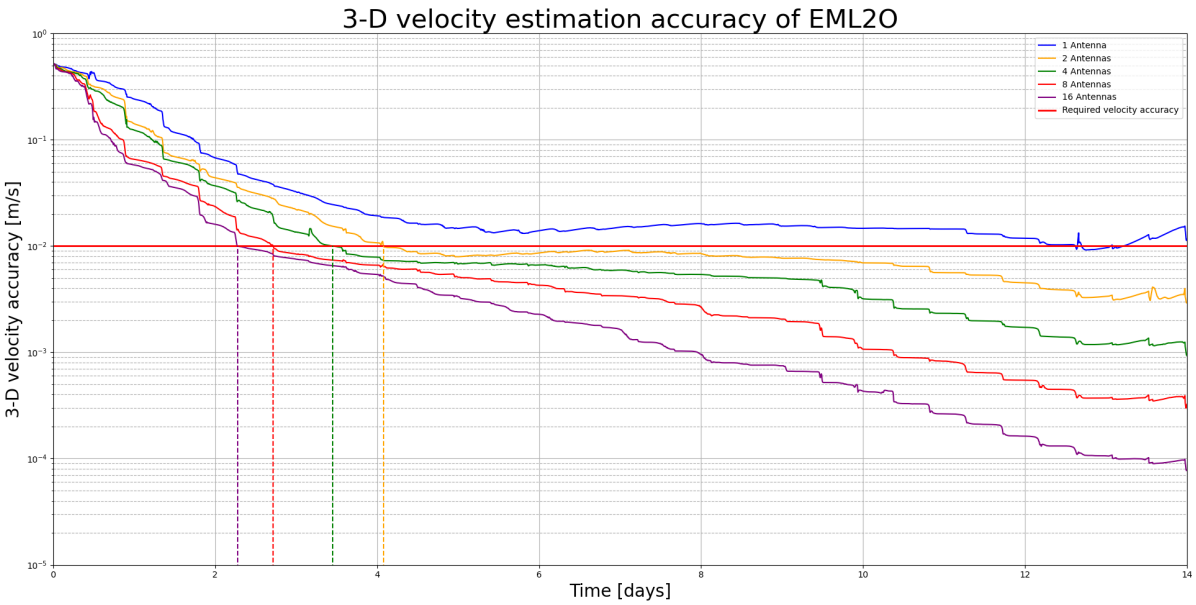


Figure 7.12: The 3-D velocity estimation accuracy of EML2O of each antenna configuration.

7.3. ELO: 3-D estimation accuracy

The progression of the 3-D state estimation accuracy's are provided in Figure 7.13 and Figure 7.14. The shapes of the position estimation accuracy of each antenna configuration is much alike. Both shapes of the 3-D position and velocity estimation accuracy is almost periodic, above its periodicity due to the elliptic orbit. Describing the average of the progression of the 3-D state estimation accuracy's, first, the progression grows or remains constant until day 3, where the high peaks due to inter-satellite geometry occur. Second, the progressions decrease until day 7 and increase again until day 10 to 11. Third, the values reduce in size again until simulation end time.

Zooming in on the periodicity due to the elliptic orbit of ELO. It is, especially, well visible in configuration 8 and 16 that downward peaks in position occur during the perilune, while upward peaks occur in the velocity progression. Highest velocity occur when ELO orbits near its perilune and therefor the uncertainties in velocity grow. For lower configurations, the downward peaks are less visible, which could be explained due to the higher ranging error.

The 3-D position estimation accuracy of configuration 1 keeps above 10 000 m for almost the whole simulation time, while equipping an array of 2 antennas already reduce the 3-D position estimation accuracy between 1000 m and 10 000 m. The 3-D state estimation performance of ELO is much more sensitive for high ranging errors than EML2O. For ELO no 3-D position and velocity accuracy requirements were defined, but by showing and discussing the results of ELO it is observed that LiAISON based orbit determination provides more accurate results for EML2O than for ELO. In an attempt to improve the estimation accuracy for ELO, one could consider, in future work, applying a ranging strategy to avoid measurements near its perilune to mitigate the reoccurring peaks leading to a reduction in errors and uncertainties for ELO.

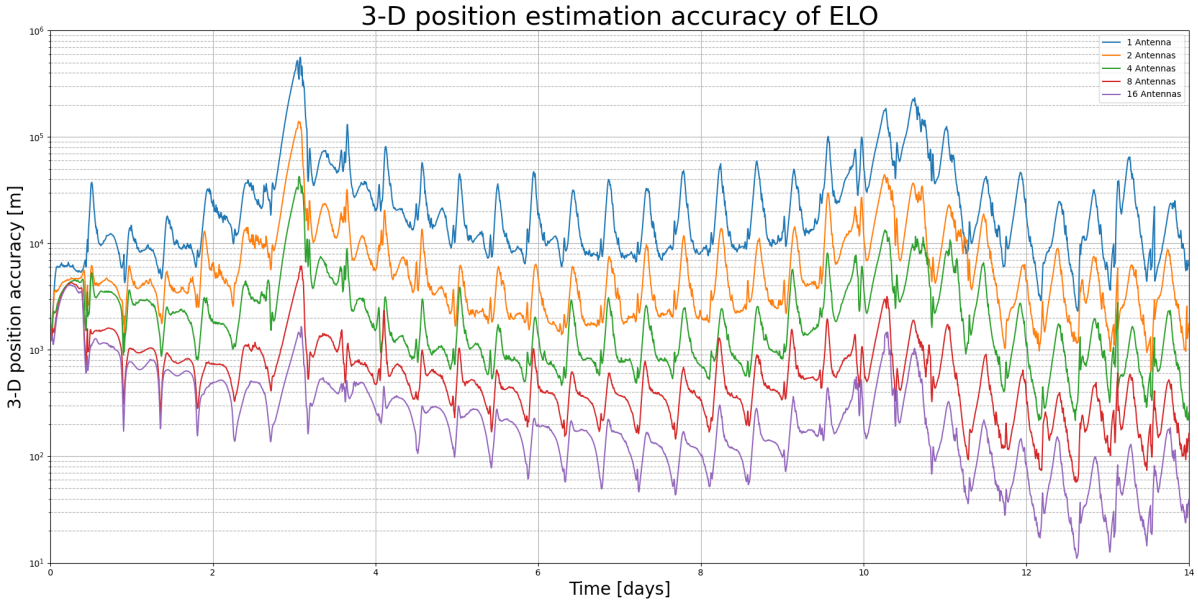


Figure 7.13: The 3-D position estimation accuracy of ELO of each antenna configuration.

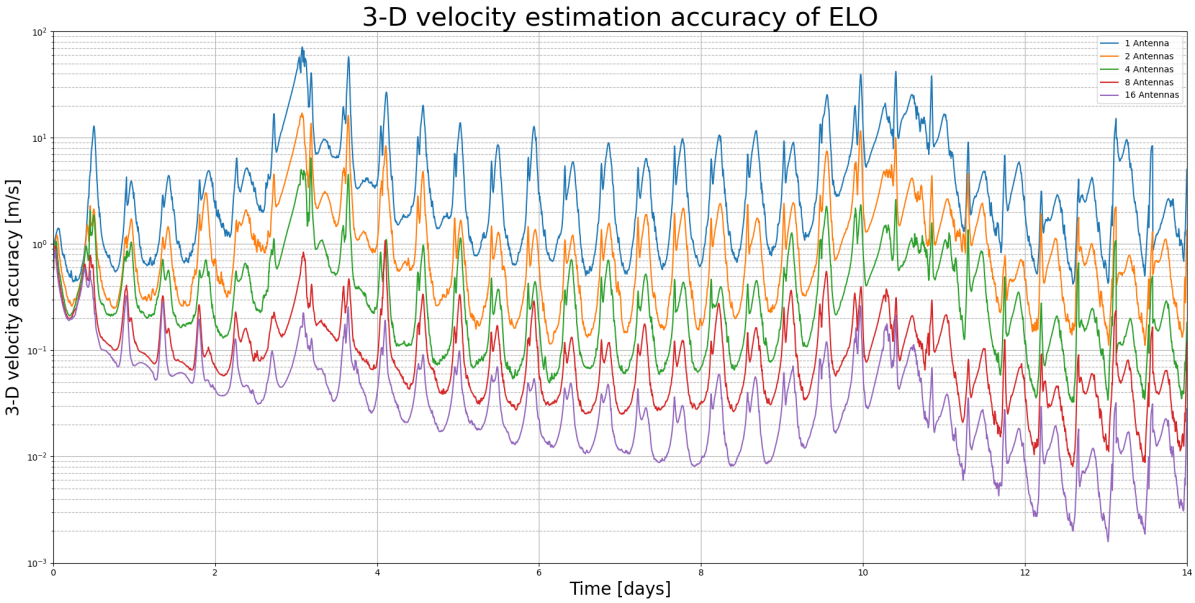


Figure 7.14: The 3-D velocity estimation accuracy of ELO of each antenna configuration.

7.3.1. The average 3-D state estimation accuracy

The average 3-D state estimation accuracy after 6 days simulation is provided in Table 7.2 and also includes the corresponding radio parameters and the ranging error. Table 7.2 shows that if EML2O equips at least 4 Anywaves antennas providing an antenna gain of 12.5 dBi, the average 3-D position uncertainty is below the mission requirement of a 1000 m 3σ position accuracy.

Considering 1 antenna, the average 3-D position estimation accuracy is 10 times lower for EML2O than for ELO. When increasing the antennas to an array of 8 or 16, the average 3-D position error of ELO becomes almost similar to EML2O. This shows that the ranging error has more influence on the estimation accuracy of ELO than for EML2O. The velocity estimation accuracy remains a few orders higher for ELO, which is due to the high uncertainties occurring in during the perilune. For EML2O, the 3-D position estimation accuracy reduces approximately by factor 2 when applying more (+3 dBi) antenna gain. For higher configurations the average 3-D position uncertainty is reduced by factor 2.5 when doubling the antennas. For ELO, the improvement in 3-D position estimation accuracy in doubling the antennas is approximately factor 4.3 for 1 to 2 antennas. The reduction factor is 3.2 for doubling from 2 to 4, 4.3 for 4 to 8 and 3.04 for 8 to 16 antennas.

These results are based on one simulation run per antenna (array) configuration. The ranging error is normal distributed, which lead to slightly different simulations each simulation run. In order to improve the fidelity of the results, one could apply a Monte Carlo simulation of X times and determine the average and cumulative distribution function of the estimated errors and uncertainties. This could be considered in future research, but are not applied in this research due to the time intensity of Monte Carlo simulation.

Table 7.2: Average 3-D state estimation accuracy's per antenna configuration after 6 days simulation.

				EML2O		ELO	
$N_{antenna}$	σ_{rho} [m]	G_{down} [dBi]	bitrates [bps]	$3\sigma_{3d.pos}$ [m]	$3\sigma_{3d.vel}$ [m/s]	$3\sigma_{3d.pos}$ [m]	$3\sigma_{3d.vel}$ [m/s]
1	385.6	6.5	850, 1000	2755.5	1.387e-2	29658	4.963
2	96.6	9.5	1700, 2000	1431.0	6.473e-3	6797.9	1.139
4	24.2	12.5	3400, 4000	781.45	3.583e-3	2128.3	3.544e-1
8	6.07	15.5	6800, 8000	315.85	1.630e-3	495.12	8.092e-2
16	1.52	18.5	13600, 16000	122.27	6.538e-4	163.48	2.609e-2

8

Conclusion

This research was carried out to investigate whether LiAISON based autonomous orbit determination is possible in the Lunar vicinity. As use-case satellites, two hypothetical satellites were considered, EML2O and ELO, which were based on characteristics and properties of LUMIO and the Lunar Pathfinder respectively. The goal of this research was to develop an autonomous orbit determination system in which only inter-satellite ranging is performed and is conform to LUMIOs position and velocity requirements of a 3σ 3-D position and velocity accuracy of 1000 m and 1 cm/s. The developed autonomous orbit determination model was capable of estimating the position and velocity errors accompanied by their uncertainties. The performance of the model is dependent on the initial radio parameters from which an estimated two-way ranging error can be computed. In order to accomplish the research goal the following main research question was formulated:

> '***Which radio parameters are required for LUMIO to perform LiAISON-based autonomous orbit determination while complying with its mission requirements?***'

To answer this research question several link budget analyses were executed from which the estimated two-way telemetry ranging error was determined that was subsequently used as input in the autonomous orbit determination model. The number of antennas was considered a free parameter in the link budget. One antenna has a gain of 6.5 dBi and doubling the amount of antennas was assumed to lead to a +3 dB gain, meaning there are no scaling loses. The link budget analysis was performed for five different antenna array configurations, from which four configurations were capable of obtaining a 3σ 3-D state estimation accuracy within the simulation time. Only three configurations had an average 3-D state estimation accuracy (from six days until simulation end time) that comply with the mission requirements.

Table 8.1: The 3-D state estimation accuracy of EML2O based in different radio parameters.

	1 Antenna	2 Antennas	4 Antennas	8 Antennas	16 Antennas
Gain [dBi]	6.5	9.5	12.5	15.5	18.5
bitrate$_{down}$ [bps]	850	1700	3400	6800	13600
bitrate$_{up}$ [bps]	1000	2000	4000	8000	16000
Ranging error [m]	385.6	96.6	24.2	6.07	1.52
Req. time to meet COMMS-070 [days]	N/A	10.8	8.1	6.3	5.1
Req. time to meet COMMS-070* [days]	N/A	4.1	3.4	2.8	2.3
Average 3-D position estimation accuracy [m] [1]	2755	1431	781.5	315.9	122.3
Average 3-D velocity estimation accuracy [m/s] [1]	1.387e-2	6.473e-3	3.583e-3	1.630e-3	6.538e-4

[1] From day 6 of until simulation end time

Table 8.1 shows the autonomous orbit determination performance for each configuration. An antenna gain up to 9.5 dBi would not be sufficient to fulfill the mission requirements. Equipping an antenna array of 4 would be sufficient if one decides to measure for a longer duration than 8.1 days without interference, e.g., stationkeeping. However, the halo orbit period is approximately 14 days and it is expected that stationkeeping is performed every 7 to 14 days. Therefore, the radio parameters accompanied with 4 antennas would in some cases not be sufficient. An array of 8 antennas can facilitate LiAISON based orbit determination, even if stationkeeping would be performed. The required 3-D position accuracy is obtained after 6.3 days using the following radio parameters for EML2O:

Table 8.2: Radio parameters of EML2O.

Transmission power	T_x	3 dBW
Frequency (down)	f_{down}	2290 MHz
Frequency (up)	f_{up}	2110 MHz
Cablelosses	L_{cable}	1 dB
Polarization loss	L_{pol}	0.5 dB
Required E_b/N_0	$(E_b/N_0)_{req}$	2.5 dB
Margin	margin	3 dB
Noise temperature	T_{noise}	26.9 dB/K
Antenna gain	G	15.5 dBi
Bit rate down	bitrate_{down}	6800 bps
Bit rate up	bitrate_{up}	8000 bps

The answer to the main research question was based on the findings of several sub-questions, which are answered below.

1. Based on the number of antennas, what are the radio performance parameters and the estimated ranging error using telemetry ranging?

Link budget analyses were performed for five different antenna array configurations with an antenna gain of 6.5, 9.5, 12.5, 15.5, and 18.5 dBi. The up- and downlink bit rate were set as variables and the required E_b/N_0 (signal-to-noise ratio) of 2.5 dB and margin of 3 dB were defined. Leaving the other radio parameters constant, the up- and downlink bit rates were determined.

The estimated two-way ranging error was determined based on the bit rate, integration time and E_s/N_0 (symbol-to-noise ratio). The bit rates were determined in the link budgets and the integration time is 1 s. In this research, E_s/N_0 equals E_b/N_0 since a modulation with $M = 2$ is chosen, e.g., BPSK or GMSK. Both the link budgets and the ranging error calculations provide the following parameters:

Table 8.3: Radio performance parameters and two-way ranging error.

$N_{antenna}$	G_{EML2O} [dBi]	bitrate_{down} [bps]	bitrate_{up} [bps]	σ_ρ [m]
1	6.5	850	1000	385.6
2	9.5	1700	2000	96.6
4	12.5	3400	4000	24.2
8	15.5	6800	8000	6.07
16	18.5	13600	16000	1.52

2. Which components does a LiAISON based autonomous orbit determination model entail if orbit determination is solely done using inter-satellite ranging?

An AOD model that uses only inter-satellite ranging as measurements to estimate the absolute states of the satellites requires three sub-models, namely:

- **Dynamic model**
  The dynamic model requires initial states and a simulation time span as input and provides the nominal

states over the simulation time span as output. Nominal states represents the nominal trajectories the satellites would follow over that simulation time. The nominal states would most likely differ from the nominal trajectory in reality, since the states are obtained within a model using a virtual environment. However, the reliability of the nominal states increases if one improves the fidelity of the integration environment. In this research, this is done by defining the accelerations exerted from Solar System bodies based on the ephemerides of those bodies and by modelling the accelerations exerted by Earth and the Moon as spherical harmonic gravity.

- **Measurement model**
  The measurement model uses the nominal states to simulate nominal measurements. The nominal measurements consist of the absolute inter-satellite distance plus a simulated normal distributed ranging error that is determined based on the radio performance parameters.

  - **Link budget analysis and ranging error**
    The link budget parameter model performs several link budget analysis' in order to determine the required radio parameters and to calculate the ranging error as is done in the first research sub-question. The ranging error is input as normal distributed Gaussian white noise in the measurement model, that is added on top of the absolute inter-satellite distance. The nominal inter-satellite distance plus the noise represents the nominal measurements.

- **Estimation model**
  The estimation model estimates the actual states of the participating satellites. This is done using an EKF that performs a time update and a measurement update each iteration. Using an a-priori defined integrator, the states are updated each time step. Thereafter, a measurement update is performed. The measurement update takes the nominal measurement and the reference measurement into account to correct the time updated states. In simulations, the reference measurement is computed by taking the norm of the position state elements. In reality, the reference measurement would be obtained from actual ranging operations.

The performance of the autonomous orbit determination model is measured by the progression of the estimated errors and their uncertainty. The estimated errors are the difference in nominal states and estimated states and the 3σ uncertainties are obtained from the diagonal of the covariance matrix, which is the variance. Based on the convergence of the 3σ uncertainties with the estimated errors within their boundaries, one can determine whether the autonomous orbit determination model provides an accurate enough performance to meet mission requirements.

3. What are the state integrator settings in the estimation filter to provide a 3-D position model error of less than 100 m*?*

The state integrator in the estimation filter differs from the dynamic model. In the dynamic model every Solar System planet, the Sun (incl. SRP) and the Moon are added to the acceleration function. Additionally, the Earth and the Moon were modelled as a spherical harmonic gravity with a maximum order and degree of 200. Based on an acceleration breakdown, it was found that several Solar System planets exert a negligible acceleration on the satellites and could therefore be considered left out. Based on a spherical harmonic gravity analysis, the maximum order and degree of Earth and the Moon do not have to be as high as 200 to provide accurate states. Besides, such a high fidelity model requires a long computation time and it was therefore chosen to reduce the state integrator in the estimation filter in accuracy with a 3-D position model error of maximum 100 m.

Besides spherical harmonic gravity of the Earth and the Moon, the integration time step is an important variable and a higher time step could lead to high model errors. The state integrator in the estimation filter has the following settings and a maximum model error over the simulation time of 53.3 m for EML2O and 67.7 m for ELO:

Table 8.4: State integrator settings in the estimation filer.

State integrator setup	
Measurement time[2]	300 s
State update time	30 s
Acceleration settings (both EML2O and ELO)	
Earth	Spherical harmonic gravity (2, 2)
Moon	Spherical harmonic gravity (40, 40)
Sun	Point mass gravity & SRP
Mercury	Point mass gravity
Venus	Point mass gravity
Jupiter	Point mass gravity

4. What are the average 3-D position and velocity estimation accuracy's of EML2O per considered antenna configuration?

The average 3-D state estimation accuracy's are the absolute 3-D 3σ uncertainties from 6 days. It was observed that until 6 days, the estimation filter is strongly converging and thereafter the convergence flattens. Therefore, the average 3-D state estimation accuracy's are determined from day 6 until simulation end time and are shown below:

- 1 Antenna (6.5 dBi): $3\sigma_{3D.pos} = 2775\,\text{m}$ & $3\sigma_{3D.vel} = 1.387 \times 10^{-2}\,\text{m/s}$
- 2 Antenna (9.5 dBi): $3\sigma_{3D.pos} = 1431\,\text{m}$ & $3\sigma_{3D.vel} = 6.473 \times 10^{-3}\,\text{m/s}$
- 4 Antenna (12.5 dBi): $3\sigma_{3D.pos} = 781.5\,\text{m}$ & $3\sigma_{3D.vel} = 3.583 \times 10^{-3}\,\text{m/s}$
- 8 Antenna (15.5 dBi): $3\sigma_{3D.pos} = 315.9\,\text{m}$ & $3\sigma_{3D.vel} = 1.630 \times 10^{-3}\,\text{m/s}$
- 16 Antenna (18.5 dBi): $3\sigma_{3D.pos} = 122.3\,\text{m}$ & $3\sigma_{3D.vel} = 6.538 \times 10^{-4}\,\text{m/s}$

Based on the average 3-D state estimation accuracy, EML2O equipped with an array of 4 antennas with together an antenna gain of 12.5 dBi would be sufficient in order to comply with LUMIOs position and velocity requirements.

At which simulation time a 3-D position and velocity accuracy is achieved that is conform with LUMIOs requirements?

The five simulations performed with a different ranging error comply from a certain simulation time with the position and velocity requirements of LUMIO, except for configuration 1. The velocity requirement of 1 cm/s 3-D 3σ accuracy is met in a shorter amount of time than the position requirement of 1000 m 3-D 3σ position accuracy. Table 8.5 shows the simulation time required to meet the requirements.

Table 8.5: Simulation time required to comply to COMMS-070 and COMMS-070*.

No. of antennas	**T_{req} to comply with COMMS-070**	**T_{req} to comply with COMMS-070***
1	N/A	N/A
2	10.8 days	4.1 days
4	8.1 days	3.4 days
8	6.3 days	2.8 days
16	5.1 days	2.3 days

If one want to perform stationkeeping between 7 to 14 days of orbit, a 3-D position of less than 1000 m must be acquired within 7 days. Therefore, an array of 8 antennas with a gain of 15.5 dBi is required to comply with LUMIOs mission requirements.

[2] Given visibility

8.1. Discussion

This research investigated the possibilities of autonomous orbit determination using only inter-satellite ranging for EML2O that formed a radio link with ELO. Both satellites are based on the existing planned missions LUMIO and Lunar Pathfinder. This research shows that autonomous orbit determination using only inter-satellite ranging is (theoretically) possible using a high fidelity environment for obtaining the nominal states and a lower fidelity environment as integration environment for the estimation process. The ranging error plays a crucial role in order to comply with position and velocity accuracy requirements and must be mitigated in order to provide reliable estimated states.

It is shown that eight antennas, with an antenna gain of 15.5 dBi, are required on EML2O to comply with the position and velocity accuracy of LUMIO. However, these results are based on one simulation. The ranging error is normal distributed and therefore, the simulations will slightly differ each time. To improve the credibility of the results, it is advised to run the same simulation multiple times and take the average of each simulation as results (Monte Carlo simulations). In this research, this is not done due to timing issues.

The simulations performed were executed using a lower fidelity state integrator in the estimation filter, leading to a 3-D model error for EML2O and ELO of 53.3 m and 67.7 m respectively. Such a model error could have an effect on the estimation performance and could be improved. It is assumed that there exist an optimal state integrator in the estimation filter based on computation time and model error. This optimization problem could be solved in an attempt to improve the estimation model.

In general, the 3-D state estimation accuracy for ELO were higher than for EML2O. This was explained by the increasing uncertainty every time ELO was near its perilune where it experiences the highest velocity of its orbit. Due to a relative short orbit of 10.5 hours with respect to a 14 days halo orbit, the spikes occur often (every time ELO passes its perilune). One could try to mitigate this effect by applying a certain ranging strategy in an attempt to avoid inter-satellite ranging during its perilune. Improvement of the work could be to apply a variable time step for obtaining measurements.

The simulations are executed under the assumption of a constant ranging error that is calculated based on the highest inter-satellite distance. The free space losses will be smaller when the inter-satellite distance decreases, which improves the signal to noise ratio and therefore the ranging error. The ranging error will actually vary over time. Taking this into account could be beneficial for the performance of LiAISON based orbit determination.

In this research only a ranging error is considered. However, other errors or uncertainties could affect the accuracy of the model. A logic next step would be to implement measurement bias and look into clock related uncertainties on both satellites. Implementing such errors and uncertainties provide more insight in the feasibility of LiAISON based orbit determination. Basically, an in-depth research regarding the potential errors in such a model would improve the credibility of the estimated values.

In conclusion, the work executed provides knowledge about the possibilities of autonomous orbit determination in Cislunar space. However, several assumption made during this research affect the performance of the simulations. To provide more accurate output, the following aspects could be applied on top of this work: 1) Monte Carlo simulations, 2) In-depth link budget analysis / in-depth error analysis, 3) Ranging strategy incl. variable time step.

8.1.1. Future work

In order to improve the credibility and accuracy of this work, future research is required. Improvements are listed and shortly discussed.

- **Monte Carlo simulations**
  To improve credibility of the results, one should rerun the simulation multiple times to compensate for the normal distributed ranging error. Taking the average of multiple runs, would increase credibility.

- **Error and uncertainty analysis**
  This research only considered a constant ranging error, while measurement bias, clock related uncer-

tainties and other possible errors or uncertainties are not taken into account. An 'error-analysis' provides knowledge to the origin of errors and uncertainties, which could then be implemented into the autonomous orbit determination model. Figure D.4 shows the progression of configuration 8 including 10 m bias. It is shown that 10 m is not 'killing' the system, but the value is not substantiated. In the error-analysis, errors and uncertainties can be substantiated and thereafter implemented in the system.

- **Variable errors**
  The ranging error is not constant over the simulation time and varies over time due to the inter-satellite distance. Adding a variable ranging error in the system would improve the performance and the credibility. Additionally, measurement bias and clock uncertainties are most likely to grow over time, which makes them variable as well.

- **Extensive link budget analysis**
  An extensive link budget analysis aligns with the error analysis and could be executed to determine the required radio parameters.

- **Multiple observations** This research only took range observations into account. One could investigate the effect of adding multiple observations types into the model, e.g., range-rate observations only or range and range-rate observations. The model used for this research is capable of performing the simulation by including range-rate observations, but since the range-rate errors were not substantiated they have not been considered. However, by substantiating the estimated range-rate errors, one could investigate the performance of range-rate only observations or several range and range-rate observation configurations. Multiple types of measurements could lead to more measurement updates and therefore an improvement in states estimation.

- **Include stationkeeping**
  A simulation span of 14 days without stationkeeping was considered. For a 14 days halo orbit, stationkeeping manoeuvres are executed every 7 to 14 days to remain near the planned trajectory. However, after a ΔV is applied the orientation of the orbit will slightly differ and therefore a new reference orbit is required after each stationkeeping manoeuvre. Implementing stationkeeping in the model lead to the possibility to simulate the autonomous orbit determination performance over the full operational phase instead of only for a limited time span.

- **Testing in space** Eventually, the goal is to have a reliable autonomous orbit determination system using only inter-satellite measurements in space. Therefore, actual autonomous orbit determination should be executed in space. This data must be collected and send back to Earth. In the same time frame, ground-based orbit determination must be performed, so that the estimated states can be compared with the actual states.

Bibliography

V. H. Adams and M. Peck. Interplanetary optical navigation. In *2016 AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2016. ISBN 9781624103896. doi: 10.2514/6.2016-2093.

S. Akhlaghi, N. Zhou, and Z. Huang. Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. In *2017 IEEE Power & Energy Society General Meeting*, pages 1–5, 2017. doi: 10.1109/PESGM.2017.8273755.

K. Andrews, J. Hamkins, S. Shambayati, and V. Vilnrotter. Telemetry-based ranging. In *IEEE Aerospace Conference Proceedings*, 2010. ISBN 9781424438884. doi: 10.1109/AERO.2010.5446926.

Anywaves. S-band tt&c antenna, 2020.

I. S. Board. *IEEE Standard Definitions of Terms for Antennas*. The Institute of Electrical and Electronics Engineers, Inc., 2004. ISBN 1559373172.

A. Cervone, F. Topputo, S. Speretta, A. Menicucci, J. Biggs, P. D. Lizia, M. Massari, V. Franzese, C. Giordano, G. Merisio, D. Labate, G. Pilato, A. Taiti, E. Bertels, B. Bosman, K. Woroniak, J. Rotteveel, A. Thorvaldsen, A. Kukharenka, and R. Walker. Phase a design of the lumio spacecraft: a cubesat for observing and characterizing micro-meteoroid impacts on the lunar far side. In *71st International Astronautical Congress (IAC)*, October 2020.

A. Cervone, F. Topputo, S. Speretta, A. Menicucci, E. Turan, P. D. Lizia, M. Massari, V. Franzese, C. Giordano, G. Merisio, D. Labate, G. Pilato, E. Costa, E. Bertels, A. Thorvaldsen, A. Kukharenka, J. Vennekens, and R. Walker. Lumio: A cubesat for observing and characterizing micro-meteoroid impacts on the lunar far side. *Acta Astronautica*, 195:309–317, 2022. ISSN 0094-5765. doi: https://doi.org/10.1016/j.actaastro.2022.03.032. URL https://www.sciencedirect.com/science/article/pii/S0094576522001424.

K.-M. Cheung. The role of margin in link design and optimization. *2015 IEEE Aerospace Conference*, page 1, August 2015. doi: 10.1109/AERO.2015.7119220.

eoPortal. Lunar pathfinder minisatellite mission, 2022. URL https://directory.eoportal.org/web/eoportal/satellite-missions/l/lunar-pathfinder.

ESA. Lunar pathfinder, 2021. URL https://www.esa.int/ESA_Multimedia/Images/2021/09/Lunar_Pathfinder.

V. Franzese, P. D. Lizia, and F. Topputo. Autonomous optical navigation for lumio mission. In *Space Flight Mechanics Meeting, 2018*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2018. ISBN 9781624105333. doi: 10.2514/6.2018-1977.

V. Franzese, F. Topputo, F. Ankersen, and R. Walker. Deep-space optical navigation for m-argo mission. *Journal of the Astronautical Sciences*, 2021. ISSN 21950571. doi: 10.1007/s40295-021-00286-9.

Y. Gao, J. Chen, B. Xu, and J. Zhou. Research on the effectiveness of different estimation algorithm on the autonomous orbit determination of lagrangian navigation constellation. *International Journal of Aerospace Engineering*, 2016:1–8, October 2016. doi: 10.1155/2016/8392148.

Y. Gao, T. Zhao, B. Jin, J. Chen, and B. Xu. Autonomous orbit determination for lagrangian navigation satellite based on neural network based state observer. *International Journal of Aerospace Engineering*, 2017: 9734164, 2017. ISSN 1687-5966. doi: 10.1155/2017/9734164. URL https://doi.org/10.1155/2017/9734164.

A. Gelain. Opinion: Will spacex spur another wave of smallsat innovation?, 2021. URL https://aviationweek-com.tudelft.idm.oclc.org/aerospace/commercial-space/opinion-will-spacex-spur-another-wave-smallsat-innovation.

N. O. Harlé and J. Friend. Lunar pathfinder, data relay satellite in orbit around the moon (service user guide). Technical report, ESA, 2020.

K. Hill and G. H. Born. Autonomous orbit determination from lunar halo orbits using crosslink range. *Journal of Spacecraft and Rockets*, 45:548–553, 2008. ISSN 15336794. doi: 10.2514/1.32316.

K. A. Hill. *Autonomous Navigation in Libration Point Orbits*. PhD thesis, University of Colorado, Boulder, 2007.

ISISpace. Isis on-board computer, 2022. URL https://www.isispace.nl/product/on-board-computer/.

V. L. Kashyap, J. Hong, S. Romaine, L. Sethares, V. Cotroneo, D. Spiga, and L. Nittler. Designing miniature x-ray optics for the smallsat lunar science mission concept cubex. *Applied Optics*, 59:5560, June 2020. ISSN 1559-128X. doi: 10.1364/ao.393554.

F. L. Markley. Autonomous navigation using landmark and intersatellite data. In *AIAA Paper*. AIAA, 1984. doi: 10.2514/6.1984-1987.

O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer, 2nd edition, 2001. ISBN 3-540-67280. URL http://www.springende/phys/.

NASA. Nasa's mission operations and communications services. Technical report, NASA, 2014.

NASA. What is capstone?, 2022. URL https://www.nasa.gov/directorates/spacetech/small_spacecraft/capstone.

I. J. O'Neill. Nasa's deep space network looks to the future, 2021. URL https://www.jpl.nasa.gov/news/nasas-deep-space-network-looks-to-the-future.

I. Out. Formation flying in the sun-earth/moon perturbed restricted three-body problem in partial fulfillment of the requirements for the degree of master of science in aerospace engineering. Master's thesis, Delft University of Technology, 2017.

T. Qin, D. Qiao, and M. Macdonald. Relative orbit determination using only intersatellite range measurements. *Journal of Guidance, Control, and Dynamics*, 42:703–710, 2019. ISSN 15333884. doi: 10.2514/1.G003819.

M. Scotti, C. Botteron, M. Rico, F. Basile, V. Massaras, B. Kieniewicz, J. Svatoň, P. Giordano, and J. Ventura-Traveset. Navimoon-ultra-high sensitivity gnss receiver for lunar navigation. In *In NAVITEC*, 2022.

S. Speretta. Satellite communication systems. University Lecture, 2022.

S. Speretta, A. Cervone, A. Menicucci, E. Turan, E. Bertels, B. Bosman, and F. Topputo. Designing the radio link for a lunar cubesat: the lumio case. *72 th International Astronautical Congress (IAC)*, pages 7–8, 2021.

J. Stupl and J. H. Harvard. Cubex: A compact x-ray telescope enables both x-ray fluorescence imaging spectroscopy and pulsar timing based navigation, 2018.

J. Stupl, M. Ebert, D. M. SGT, N. Ames, J. Hong, S. Romaine, A. Kenter, J. Evans, R. Kraft, L. Nittler, I. Crawford, D. Kring, N. Petro, K. Gendreau, J. Mitchell, L. Winternitz, R. Masterson, G. Prigozhin, B. Wickizer, K. Bonner, A. Clark, A. Dave, A. Dono-Perez, A. Kashani, D. Larrabee, S. Montez, K. Ronzano, T. Snyder, J. Mueting, L. Plice, Y.-L. Shen, and D. N. B. A. Hamilton. Cubex: A compact x-ray telescope enables both x-ray fluorescence imaging spectroscopy and pulsar timing based navigation. In *32nd Annual Small Satellite Conference*, August 2018.

T. P. Tanis. Literature study: Autonomous orbit determination in the lunar vicinity, 2022.

B. Tapley, B. Schutz, and G. Born. *Statistical Orbit Determination*. Elsevier Science Publishing, May 2004. ISBN 9780080541730.

M. R. Thompson, A. Forsman, S. Chikine, B. C. Peters, T. Ely, D. Sorensen, J. Parker, and B. Cheetham. Cislunar navigation technology demonstrations on the capstone mission. *Institute of Navigation, International Technical Meeting (ITM 2022)*, 2022.

F. Topputo. Lumio mission concept, 2017. URL https://www.youtube.com/watch?v=s3st5Y4PpRE&ab_channel=FrancescoTopputo.

F. Topputo, A. Cervone, P. Sundaramoorthy, S. Speretta, S. Mestry, and R. Noomen. Lumio: a cubesat at earth-moon l2. In *The 4S Symposium 2018, Sorrento, Italy*, 2018.

Z. J. Towfic, D. P. Heckman, D. D. Morabito, R. M. Rogalin, C. M. Okino, and D. S. Abraham. Simulation and analysis of opportunistic mspa for multiple cubesat deployments. In *15th International Conference on Space Operations, 2018*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2018. ISBN 9781624105623. doi: 10.2514/6.2018-2396.

Tudat. Api reference, 2022a. URL https://py.api.tudat.space/en/latest/.

Tudat. Default environment models, 2022b. URL https://docs.tudat.space/en/stable/_src_user_guide/state_propagation/environment_setup/create_bodies/default_settings.html?highlight=ephemeris#ephemeris.

Tudat. Acceleration: cannonball radiation pressure, 2022c. URL https://py.api.tudat.space/en/latest/acceleration.html#tudatpy.numerical_simulation.propagation_setup.acceleration.cannonball_radiation_pressure.

E. Turan, S. Speretta, and E. Gill. Autonomous crosslink radionavigation for a lunar cubesat mission. *Frontiers in Space Technologies*, 3, 2022. ISSN 2673-5075. doi: 10.3389/frspt.2022.919311. URL https://www.frontiersin.org/articles/10.3389/frspt.2022.919311.

UNOOSA. Online index of objects launched into outer space, 2022. URL https://www.unoosa.org/oosa/osoindex/search-ng.jspx.

K. F. Wakker. *Fundamentals of Astrodynamics*. Institutional Repository, January 2015. ISBN 9789461864192. URL http://repository.tudelft.nl.

W. Wang, L. Shu, J. Liu, and Y. Gao. Joint navigation performance of distant retrograde orbits and cislunar orbits via liaison considering dynamic and clock model errors. *Navigation*, 66, December 2019. doi: 10.1002/navi.340.

A

Read Me

```
Last edited 13-08-2022 by TP Tanis
###### Table of contents ######
1. General Info
2. Setup
3. Code use
        3.1 Satellites
        3.2 Nominal trajectory
        3.3 Measurements
        3.4 Estimation process
4. File description


##### 1. General Info #####
The code provided is developed for the master thesis of TP Tanis about autonomous orbit
determination using only inter-satellite ranging in the Lunar vicinity. As use case,
two theoretical satellites are
used that are based on characteristics of the Lunar Pathfinder and LUMIO. This file
describes how to setup and how to work with the directory.


##### 2. Setup #####
Coding language: Python
IDE: Pycharm any other IDE supporting Python would work as well
Virual environment: tudatpy
Environment manager: Conda

i. Download Pycharm from https://www.jetbrains.com/pycharm/ and follow the instructions.

ii. Download Miniconda from https://docs.conda.io/en/latest/miniconda.html and follow the
instructions.

iii. Open the Miniconda terminal and create the tudatpy environment by following the
steps on
https://docs.tudat.space/en/stable/_src_getting_started/installation.html#installing-anaconda

iv. Update all required packages
        - Make sure tudat-space environment is activated in the command prompt
        - Make sure PIP_REQUIREMENTS.txt and CONDA_requirements.txt are in the path as
          the tudat-space environment
        - Type in the Miniconda prompt: pip install -r PIP_REQUIREMENTS.txt
        - Type in the Miniconda prompt: conda install -r CONDA_REQUIREMENTS.txt
                requirement textfiles are found in the ReadMe_and_REQUIREMENTS map

v. Open project in Pycharm many errors will occur
        - Configure your IDE to use the python interpreter in your virtual environment
        For Pycharm:
        - Go to File -> Settings - > Project:[Name] -> Python interpreter
        - Select Python Interpreter: Python 3.9 tudat-space
        - Project is ready for use
```

```
##### 3. Code use #####
This code does not contain a single main script, but specific scripts need to be ran to
obtain the desired data. This description provide a step-by-step plan to obtain the simulated
data.

3.1 Satellites
A class is defined to create satellites and add required properties to them.


3.2 Nominal trajectory
i. Open Initials\\Simulation_Time_Setup.py
        Define the simulation time settings
        - Define t0_mjd line 14: use a value of 59091.50 to 61325.00 MJD
          as decimals .00; .25; .50;, .75 are possible
        - Define simulation duration in days
        - Define desired fixed time step line 17 in seconds

        Define the path for saving the data and the dynamic model
        - Define the DIRECTORY_NAME relative to the parent path. NAME should be equal to the
          last part of the DIRECTORY_NAME
        - OVERWRITE at line 22 allows you to overwrite existing data, leave this at 0 at all
          time, unless you are sure

Note: NAME and last path part of DIRECTORY_NAME must be equal to the name of one of the
environments that can be found in Nominal_Dynamic_Model\\Environments\\environmentname. If
you want to use custom names, go to Nominal_Dynamic_Model//Integrator_output_states and
add an if statement alowing you directory name to be used.

ii. Open Nominal_Dynamic_Model\\Nominal_States_Obtainer.py
        - Check if you are happy with the settings.

iii. Open Nominal_Dynamic_Model\\Nominal_States_Saver.py
        - Run this script and the states and output is saved as .json files in the defined
          DIRECTORY_NAME

iv. Open figure_scripts\\Nom_Sim_figures.py
        - Run this script to see the work you've done

3.3 Measurements
i. Open Initials\\Simulation_Time_Setup.py and scroll to line 38
        - Define measurement interval line 41. Note that this is relative to the
          fixed_time_step in line 17
        - Choose a number 1, 2, 4, 8, 16 of antennas to define the ranging error and
          filename line 51
        - Define bias, range rate error and bias_dot in 71-73
        - Define the type of measurements in line 95

ii. Open Measurement_Model\\Measurement_Array_Saver.py
        This saves the state vectors at the measurement interval and is therefore a smaller
        file. The idea of this file is to obtain smaller size trajectory arrays from the
        nominal states.
        - Define the file_name in line 26
        - Run file to save measurement arrays

iii. Open Measurement_Model\\Nominal_Observation_Cooker.py
        - This file cooks the nominal measurements, which are used as input for the
          estimation model.
        - It is automatically run when the estimation model is executed.

3.4 Estimation process
i. Open Initials\\Simulation_Time_Setup.py and scroll to line 76
        - If desired, tune the initial estimated error, initial covariance matrix and state
          noise compensation matrix

ii. Open Estimation_Model\\Estimation_Setup.py, \\AEKF.py, \\Executor.py
        - In the Estimation_Setup you can check all initial settings
        - The AEKF describes the Extended Kalman Filter. The state integrator is imported
          in line 18 and can be adjusted. Check Estimation_Model\\Integration_Environments\\
          and choose an integration environment.
        - Run the Estimation_Saver, which saves the output containing of [ephemeris time[0],
          state_errors[1:13], std_P[13:25, visibility[25]]
```

```
          first eml2o then elo in state errors
        - Now the estimation data is saved in the defined DIRECTORY_NAME relative to project
          path as 'estimation_data_n_antenna.json' n = number of antennas

iii. Open Estimation_Model\\3D_position_and_velocity and \\States_Estimator_figures.py
        - Run both scripts and observe the output and figures

figure_scripts\\Estimation_Simulation_figures.py and run this script. After running the
figures, both for both eml2o and elo the estimated errors and uncertainties are shown.


##### 4. File description ######
This section gives a brief description of each file in the project. The files are
discussed per directory in alphabetic order

+ Estimation_Model
        + Integration_Environments
          Contains several 7 state integrators, which differ in acceleration setup
        - 3D_position_and_velocity.py
          Shows the 6day average of the 3sigma uncertainties and the 3sigma plots
        - AEKF.py
          Function file, containing the estimation filter
        - Estimation_Saver.py
          Saves the estimation output
        - Estimation_Setup.py
          Describes the initial conditions for the estimation filter
        - estimator_functions.py
          Describes functions used in the estimation filter
        - Executor.py
           Executes the Estimation process
        - States_Estimator_figures.py
          Provide plots of saved simulation data
+ figure_scripts
        - environment_error_figures.py
          Displays the 3D-error between the dynamic model and state integrator in the
          estimation model Turn measurements and initial state error off to execute
        - Estimation_Simulation_figures.py
          Provide plots of the estimation simulations directly from the Executor.py
        - link_budget_figures.py
          Shows relations between radio parameters and ranging error
        - Nom_Sim_figures.py
          Plots of the nominal trajectories
+ Initials
        - initial_states_obtainer.py
          Several functions linked to the database and to obtain the trajectory of the
          Moon based on ephemerides
        - Simulation_Time_setup.py
          Setup script for time and measurement settings
+ Measurement_Model
        - Measurement_Array_Saver.py
          Saves the nominal trajectory in a different timestep
        - measurement_functions.py
          Contains several functions to simulate range and range-rate observations
        - Nominal_Observations_Cooker.py
          Provides the nominal observations
+ Milano_Data
        - LUMIO_states.csv
          csv-file containing the planned trjaectory of LUMIO
        - Milano_Data_reader.py
          Reads the csv files
        - Moon_states.csv
          csv-file containing the trajectory of the Moon
+ Nominal_Dynamic_Model
        + Environments
              - 12 Virtual environments used for the dynamic model
        - Integrator_output_state.py
          Function file of the dynamic integrator
        - Nominal_States_Obtainer
          Obtains the nominal states from the dynamic model
        - Nominal_States_Saver
          Saves the nominal trajectories and output
```

```
+ Ranging_System
        - Link_Budget_Analysis.py
           Performs the link budget analysis, calculates the ranging error and displays
          relations between them
        - Notes.py
          Notition file
+ ReadMe_and_REQUIREMENTS
        - CONDA_REQUIREMENTS.txt
          List of required packages for Conda
        - PIP_REQUIREMENTS.txt
          List of required packages for Pip
        - ReadMe.txt
          This file:+ Satellites_list
        - ELO.py
          Contains properties of ELO
        - EML2O.py
          Contains properties of EML2O
        - satellite.py
          Class file of a satellite ELO and EML2O are defined using this class
+ Saved_data
        + Contains per DIRECTORY_NAME defined in Initials\\Simulation_Time_Setup data
+ Trash
        + Several old files
+ Verifications
        - Biascheck_figures.py
          Plots containing 10m bias to compare without bias
        - int_environment_check.py
          Shows 3D position error for both satellites due to different integrator in
          estimation model
        - Model_error.py
          Compares the accuracy of different environments with each other
          Saved_Data is required for each environment
        - Model_verification.py
          Trajectories plotted against LUMIOs trajectory
        - symbolic_H.py
          Calculation of symbolic values of the observation state matrix
        - Unit_Tests.py
          Contains unit tests of self defined functions
```

B

Requirements

B.1. Conda Requirements

```
# This file may be used to create an environment using:
# $ conda create --name <env> --file <this file>
# platform: win-64
basemap=1.2.2=py39h381b4b0_4
boost-cpp=1.72.0=h5b4e17d_7
brotli=1.0.9=h8ffe710_7
brotli-bin=1.0.9=h8ffe710_7
bzip2=1.0.8=h8ffe710_4
ca-certificates=2022.4.26=haa95532_0
certifi=2022.5.18.1=py39haa95532_0
cspice-cmake=66=h0a763a3_5
cycler=0.11.0=pyhd3eb1b0_0
fonttools=4.33.3=py39hb82d6ee_0
freetype=2.10.4=h546665d_1
geos=3.9.1=h39d44d4_2
icu=69.1=h0e60522_0
intel-openmp=2022.0.0=haa95532_3663
jbig=2.1=h8d14728_2003
jpeg=9e=h8ffe710_1
kiwisolver=1.4.2=py39h2e07f2f_1
krb5=1.19.3=h1176d77_0
lcms2=2.12=h83e58a3_0
lerc=3.0=hd77b12b_0
libblas=3.9.0=14_win64_mkl
libbrotlicommon=1.0.9=h8ffe710_7
libbrotlidec=1.0.9=h8ffe710_7
libbrotlienc=1.0.9=h8ffe710_7
libcblas=3.9.0=14_win64_mkl
libclang=13.0.1=default_h81446c8_0
libcurl=7.83.0=h789b8ee_0
libdeflate=1.10=h8ffe710_0
libffi=3.4.2=h8ffe710_5
liblapack=3.9.0=14_win64_mkl
libpng=1.6.37=h1d00b33_2
libssh2=1.10.0=h680486a_2
libtiff=4.3.0=hc4061b1_3
libwebp=1.2.2=h2bbff1b_0
libwebp-base=1.2.2=h8ffe710_1
libxcb=1.13=hcd874cb_1004
libzlib=1.2.11=h8ffe710_1014
lz4-c=1.9.3=h8ffe710_1
m2w64-gcc-libgfortran=5.3.0=6
m2w64-gcc-libs=5.3.0=7
m2w64-gcc-libs-core=5.3.0=7
m2w64-gmp=6.1.0=2
m2w64-libwinpthread-git=5.0.0.4634.697f757=2
matplotlib=3.5.1=py39haa95532_1
matplotlib-base=3.5.1=py39hd77b12b_1
```

```
mkl=2022.0.0=h0e2418a_796
mpmath=1.2.1=py39haa95532_0
msys2-conda-epoch=20160418=1
munkres=1.1.4=py_0
nrlmsise-00=0.1=h0a763a3_1
numpy=1.22.3=py39h0948cea_2
openjpeg=2.4.0=hb211442_1
openssl=1.1.1o=h2bbff1b_0
packaging=21.3=pyhd3eb1b0_0
pandas=1.4.2=py39h2e25243_1
pillow=9.1.0=py39ha53f419_2
pip=22.0.4=pyhd8ed1ab_0
proj=8.0.1=h1cfcee9_0
pthread-stubs=0.4=hcd874cb_1001
pybind11=2.9.2=py39h2e07f2f_1
pybind11-global=2.9.2=py39h2e07f2f_1
pyparsing=3.0.8=pyhd8ed1ab_0
pyproj=3.3.1=py39ha996c60_0
pyqt=5.12.3=py39hcbf5309_8
pyqt-impl=5.12.3=py39h415ef7b_8
pyqt5-sip=4.19.18=py39h415ef7b_8
pyqtchart=5.12=py39h415ef7b_8
pyqtwebengine=5.12.1=py39h415ef7b_8
pyshp=2.3.0=pyhd8ed1ab_0
python=3.9.12=h9a09f29_1_cpython
python-dateutil=2.8.2=pyhd3eb1b0_0
python_abi=3.9=2_cp39
pytz=2022.1=pyhd8ed1ab_0
qt=5.12.9=h556501e_6
scipy=1.8.0=py39hc0c34ad_1
setuptools=62.1.0=py39hcbf5309_0
six=1.16.0=pyhd3eb1b0_1
sofa-cmake=1.0.0=h7dfa325_6
sqlite=3.38.3=h8ffe710_0
sympy=1.10.1=py39haa95532_0
tabulate=0.8.9=py39haa95532_0
tbb=2021.5.0=h2d74725_1
tk=8.6.12=h8ffe710_0
tornado=6.1=py39hb82d6ee_3
tudat=2.10.8=hc0c6cd2_0
tudat-resources=1.1.2=h7dfa325_8
tudatpy=0.6.3=py39he20d1bb_3
tzdata=2022a=hda174b7_0
ucrt=10.0.20348.0=h57928b3_0
unicodedata2=14.0.0=py39hb82d6ee_1
vc=14.2=hb210afc_6
vs2015_runtime=14.29.30037=h902a5da_6
wheel=0.37.1=pyhd3eb1b0_0
xorg-libxau=1.0.9=hcd874cb_0
xorg-libxdmcp=1.1.3=hcd874cb_0
xz=5.2.5=h62dcd97_1
zlib=1.2.11=h8ffe710_1014
zstd=1.5.2=h6255e5f_0
```

B.2. Pip Requirements

```
basemap @ file:///D:/bld/basemap_1637007602558/work
certifi==2022.5.18.1
cycler @ file:///tmp/build/80754af9/cycler_1637851556182/work
fonttools @ file:///D:/bld/fonttools_1651018206941/work
kiwisolver @ file:///D:/bld/kiwisolver_1648854553193/work
matplotlib @ file:///C:/ci/matplotlib-suite_1647423638658/work
mpmath==1.2.1
munkres==1.1.4
numpy @ file:///D:/bld/numpy_1651020773031/work
packaging @ file:///tmp/build/80754af9/packaging_1637314298585/work
pandas @ file:///D:/bld/pandas_1649588095250/work
Pillow @ file:///D:/bld/pillow_1649818211002/work
pybind11 @ file:///D:/bld/pybind11-split_1648909175557/work
pybind11-global @ file:///D:/bld/pybind11-split_1648909175557/work
```

```
pyparsing @ file:///home/conda/feedstock_root/build_artifacts/pyparsing_1649603503565/work
pyproj @ file:///D:/bld/pyproj_1650803294603/work
PyQt5==5.12.3
PyQt5_sip==4.19.18
PyQtChart==5.12
PyQtWebEngine==5.12.1
pyshp @ file:///home/conda/feedstock_root/build_artifacts/pyshp_1651509119669/work
python-dateutil @ file:///tmp/build/80754af9/python-dateutil_1626374649649/work
pytz @ file:///home/conda/feedstock_root/build_artifacts/pytz_1647961439546/work
scipy @ file:///C:/bld/scipy_1644357595180/work
six @ file:///tmp/build/80754af9/six_1644875935023/work
sympy @ file:///C:/ci/sympy_1647853873858/work
tabulate==0.8.9
tornado @ file:///D:/bld/tornado_1648827405403/work
unicodedata2 @ file:///D:/bld/unicodedata2_1649112114189/work
```

C

Unit tests

```
"""
Some unit tests
"""
#general
import random
import unittest
import numpy as np

#own libraries
from Initials import initial_states_obtainer
from Initials import Simulation_Time_Setup
from Saved_Data import Data_Loader
from Measurement_Model import measurement_functions
from Estimation_Model import estimator_functions

NAME = "Solar_System_200_200"
DIRECTORY_NAME = "Saved_Data\\Solar_System_200_200"

'Initial states from Database'
initial_states = initial_states_obtainer.initial_states_eml2(Simulation_Time_Setup.
    t0_mjd)
'Trajectory obtained from the dynamic model'
states_nom = Data_Loader.json_2states_reader(Simulation_Time_Setup.DIRECTORY_NAME)
states_meas = Data_Loader.json_measurementarray_reader(DIRECTORY_NAME)
'Trajectory moon from ephemeris'
x_moon = initial_states_obtainer.moon_ephemeris(Simulation_Time_Setup.
    measurement_span_ephemeris)

class TestCalc(unittest.TestCase):

    'Time implementation check, MJD to ET from Database'
    def test_t0(self):
        self.assertEqual(initial_states_obtainer.simulation_start_epoch(59091.50),
            652060869.1826417446136475)
        self.assertEqual(initial_states_obtainer.simulation_start_epoch(59137.00),
            655992069.1826417446136475)
        self.assertEqual(initial_states_obtainer.simulation_start_epoch(60390.00),
            764251269.1826417446136475)
        self.assertEqual(initial_states_obtainer.simulation_start_epoch(60418.00),
```

```
        766670469.1826417446136475)

    'Check␣whether␣correct␣initial␣states␣are␣obtained␣from␣Database'
    def test_states_loader(self):
        self.assertEqual(states_nom[0,0], -310537.9975687619880773 * 10 ** 3)
        self.assertEqual(states_nom[0,1], 249423.1565183288475964 * 10 ** 3)
        self.assertEqual(states_nom[0,2], 174937.7572135815862566 * 10 ** 3)
        self.assertEqual(states_nom[0,3], -0.9931718419758050 * 10 ** 3)
        self.assertEqual(states_nom[0,4], -0.7664085138876902 * 10 ** 3)
        self.assertEqual(states_nom[0,5], -0.5251732804449779 * 10 ** 3)

        self.assertEqual(states_nom[0,0], initial_states[0])
        self.assertEqual(states_nom[0,1], initial_states[1])
        self.assertEqual(states_nom[0,2], initial_states[2])
        self.assertEqual(states_nom[0,3], initial_states[3])
        self.assertEqual(states_nom[0,4], initial_states[4])
        self.assertEqual(states_nom[0,5], initial_states[5])

    def test_moon(self):
        'Check␣Moon␣ephemeris␣and␣states␣Moon␣used␣in␣Database'
        """Database Moon states and the ephemeris Moon states lie within 1 m
            position accuracy and 1 mm/s velocity
        accuracy if this condition is met"""
        mjd_time1 = 60390
        et_time1 = initial_states_obtainer.simulation_start_epoch(mjd_time1)
        moon_database = initial_states_obtainer.states_moon_data(mjd_time1,
            mjd_time1)[0]
        moon_tudat = initial_states_obtainer.moon_ephemeris(np.linspace(et_time1,
            et_time1,1))[0]
        self.assertAlmostEqual(moon_database[0], moon_tudat[0], 0, "x-direction␣is␣
            way␣too␣off")
        self.assertAlmostEqual(moon_database[1], moon_tudat[1], 0, "y-direction␣is␣
            way␣too␣off")
        self.assertAlmostEqual(moon_database[2], moon_tudat[2], 0, "z-direction␣is␣
            way␣too␣off")
        self.assertAlmostEqual(moon_database[3], moon_tudat[3], 3, "vx-direction␣is␣
            way␣too␣off")
        self.assertAlmostEqual(moon_database[4], moon_tudat[4], 3, "vy-direction␣is␣
            way␣too␣off")
        self.assertAlmostEqual(moon_database[5], moon_tudat[5], 3, "vz-direction␣is␣
            way␣too␣off")
        mjd_time2 = 60400
        et_time2 = initial_states_obtainer.simulation_start_epoch(mjd_time2)
        moon_database2 = initial_states_obtainer.states_moon_data(mjd_time2,
            mjd_time2)[0]
        moon_tudat2 = initial_states_obtainer.moon_ephemeris(np.linspace(et_time2,
            et_time2, 1))[0]
        self.assertAlmostEqual(moon_database2[0], moon_tudat2[0], 0, "x-direction␣is
            ␣way␣too␣off")
        self.assertAlmostEqual(moon_database2[1], moon_tudat2[1], 0, "y-direction␣is
            ␣way␣too␣off")
        self.assertAlmostEqual(moon_database2[2], moon_tudat2[2], 0, "z-direction␣is
            ␣way␣too␣off")
        self.assertAlmostEqual(moon_database2[3], moon_tudat2[3], 3, "vx-direction␣
            is␣way␣too␣off")
        self.assertAlmostEqual(moon_database2[4], moon_tudat2[4], 3, "vy-direction␣
```

```
            is␣way␣too␣off")
        self.assertAlmostEqual(moon_database2[5], moon_tudat2[5], 3, "vz-direction␣
            is␣way␣too␣off")

    def test_measurementsarray_loader(self):
        '''Measurement array is states each 5 minutes instead of 10s. This check is
            to see whether the right states are
        loaded'''
        a = Data_Loader.json_measurementarray_reader(DIRECTORY_NAME)
        self.assertEqual(a[0, 1], states_nom[0, 1])
        self.assertEqual(a[1, 2], states_nom[30, 2])
        self.assertEqual(a[2, 3], states_nom[60, 3])
        self.assertEqual(a[18, 5], states_nom[540, 5])
        self.assertEqual(a[123, 4], states_nom[3690, 4])

    'Range␣observation␣function,␣bias␣and␣ranging␣error␣are␣set␣on␣zero'
    def test_nominal_ranging_error1(self):
        observations = measurement_functions.nominal_range_observation(states_meas,
            x_moon, 0, 0)
        '''It is known, that at the very start there is no visibility and in the
            middle of simulation time there is
        visibility. No visibility gives an observation value of 0'''
        self.assertEqual(observations[0], 0)
        self.assertAlmostEqual(max(observations)*10**-3, 87237, 0, 'Maximum␣inter-
            satellite␣distance')

    def test_nominal_range_observations2(self):
        dummy = measurement_functions.nominal_range_observation(states_meas, x_moon,
             0, 0)
        self.assertGreater(1800e3, dummy[1])
        self.assertGreater(dummy[20], 1800e3)
    'Range␣rate␣observations␣(not␣used,␣but␣available)'
    def test_rangerateobservations_dummyvalues(self):
        a = 108/np.sqrt(108)
        X = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
        b = measurement_functions.rangerate_observation_row(X, 0, 0)
        self.assertEqual(a, b)

    'Observation␣state␣matrix␣H'
    def test_observation_state_matrix_H(self):
        x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
        ID = 1 # Range observation (ID = 2 is range rate observation)
        dummyH = estimator_functions.H(x, ID)
        self.assertAlmostEqual(dummyH[0], -0.57735027, 8, 'Something␣wrong␣with␣H')
        self.assertAlmostEqual(dummyH[1], -0.57735027, 8, 'Something␣wrong␣with␣H')
        self.assertAlmostEqual(dummyH[2], -0.57735027, 8, 'Something␣wrong␣with␣H')
        self.assertAlmostEqual(dummyH[6], 0.57735027, 8, 'Something␣wrong␣with␣H')
        self.assertAlmostEqual(dummyH[7], 0.57735027, 8, 'Something␣wrong␣with␣H')
        self.assertAlmostEqual(dummyH[8], 0.57735027, 8, 'Something␣wrong␣with␣H')
        x2 = np.array([random.randrange(20), random.randrange(50), random.randrange
            (123), random.randrange(654),
                       random.randrange(20), random.randrange(20), random.
                           randrange(20), random.randrange(34),
                       random.randrange(23), random.randrange(34), random.
                           randrange(45), random.randrange(9),])
        dummyH2 = estimator_functions.H(x2, ID)
```

```
        self.assertEqual(dummyH2[3], 0)
        self.assertEqual(dummyH2[4], 0)
        self.assertEqual(dummyH2[5], 0)
        self.assertEqual(dummyH2[9], 0)
        self.assertEqual(dummyH2[10], 0)
        self.assertEqual(dummyH2[11], 0)


if __name__ == '__main__':
    unittest.main()
```

D

Measurement bias

The following figures show the position error and uncertainty of each configuration for EML2O. The performance of the estimation filter with a measurement bias of 10 m is compared to the performance without bias and it is shown that there are no significant differences in performance.

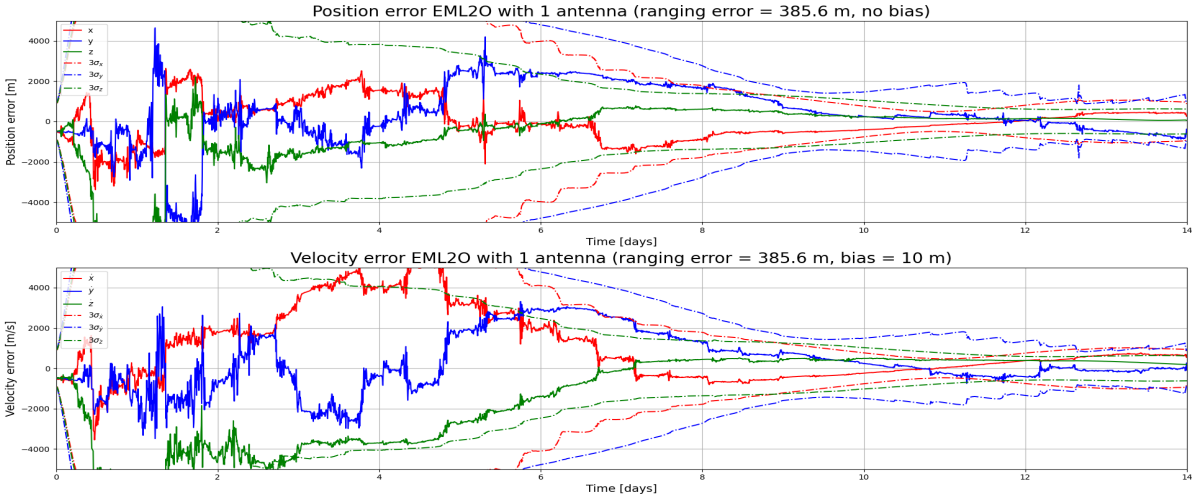


Figure D.1: Comparison of the estimation performance with and without bias for configuration 1.

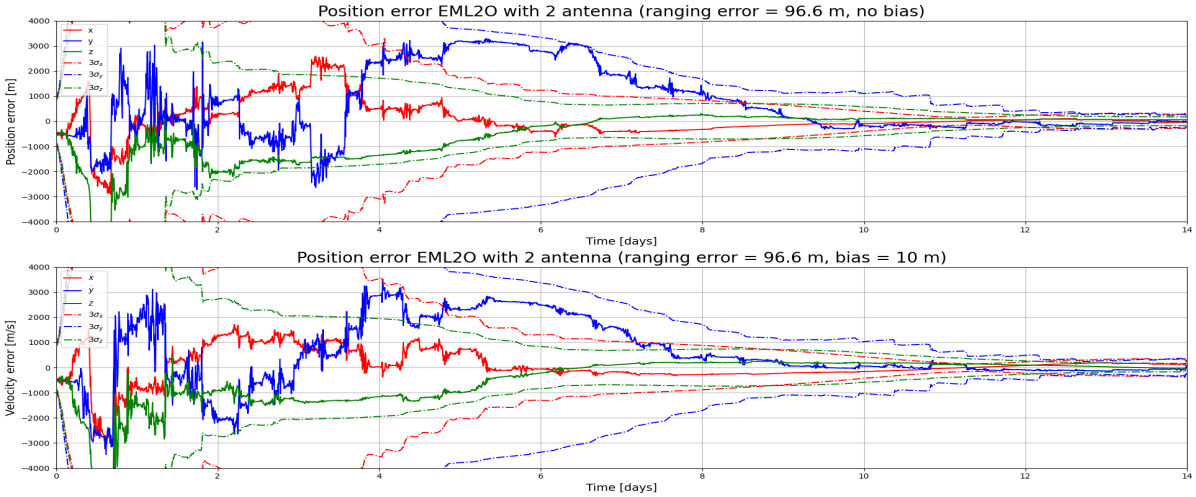


Figure D.2: Comparison of the estimation performance with and without bias for configuration 2.

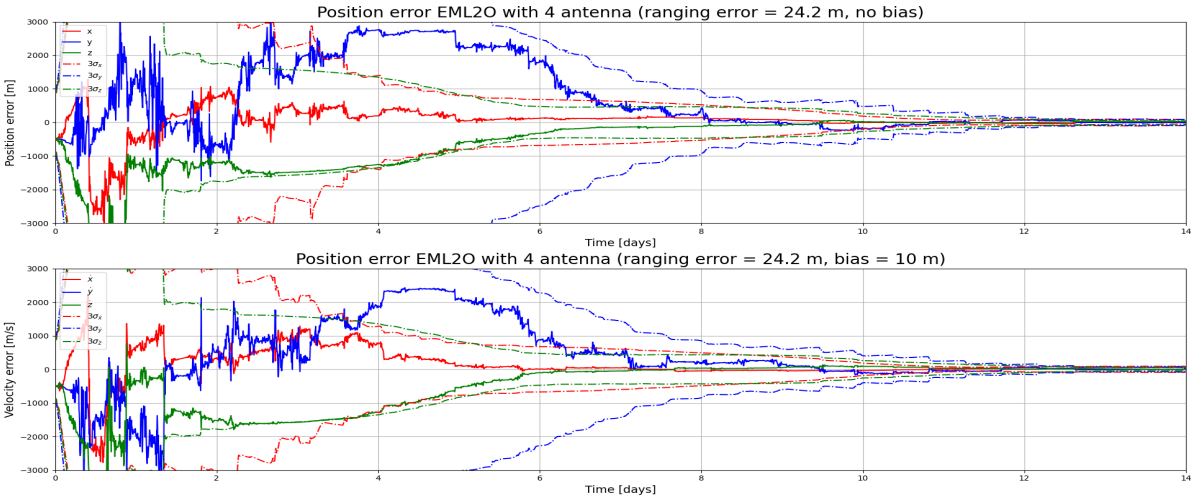


Figure D.3: Comparison of the estimation performance with and without bias for configuration 4.

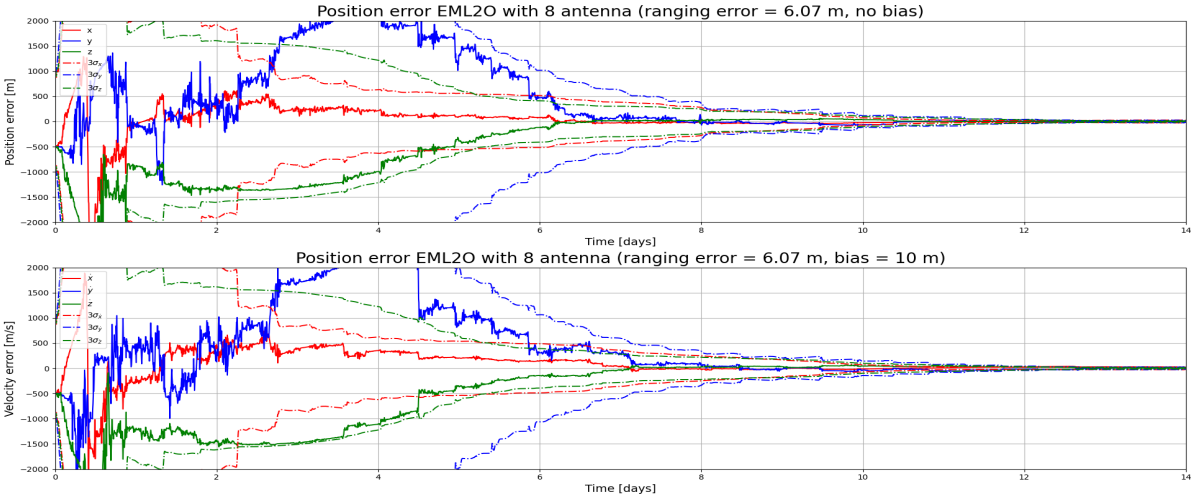


Figure D.4: Comparison of the estimation performance with and without bias for configuration 8.

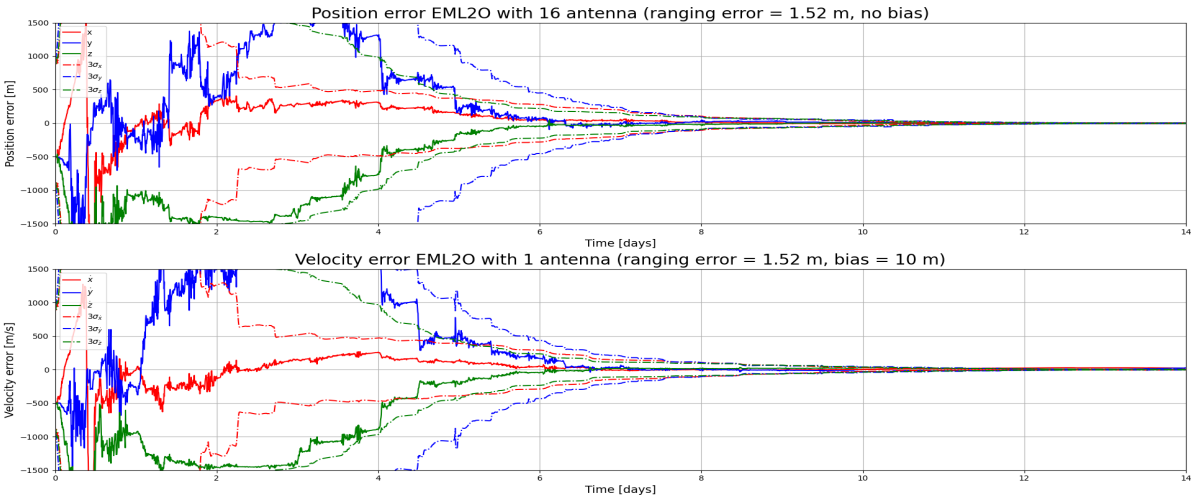


Figure D.5: Comparison of the estimation performance with and without bias for configuration 16.

E

CRTBP to ICRF

E.1. The circular restricted three-body problem (CRTBP)

The CRTBP describes the motion of the satellite in a rotating normalized reference frame, where the two primaries lay on the x-axis, the z-axis points perpendicular on the rotating frame and the y-axis finishes the reference frame using the right hand rule. The origin of the reference frame is the Earth-Moon barycenter and the distances of the primaries to the barycenter are based on their masses and represented as a dimensionless parameter *three-body gravitational parameter* μ ($\mu \in [0,1]$), which is the real mass of the secondary divided by collective mass of the primary and secondary as shown in Equation E.1. Then using the fact that both primaries are placed on the x-axis the positions are: $P_{barycenter}$: [0, 0, 0], P_1: [-μ, 0, 0] and P_2: [(1-μ), 0, 0]. From the positions of the primaries, the absolute distance of the primaries to the barycenter is easily obtained and consists of $OP_1 = \mu$ and $OP_2 = 1 - \mu$.

$$\mu = \frac{m_{P2}}{m_{P1} + m_{P2}} \tag{E.1}$$

From Equation E.1 and the distances of the primaries, the following can be concluded that:

- $OP_1 = m_2 = \mu$
- $OP_2 = m_1 = 1 - \mu$
- $m_1 + m_2 = 1$ (normalized masses)
- $OP_1 + OP_2 = 1$

The absolute distance of the primaries to the satellites in the non-dimensional rotating reference frame are defined as r_1 and r_2 as shown in Equation E.2 and Equation E.3

$$r_1 = \sqrt{(\mu + x)^2 + y^2 + z^2} \tag{E.2}$$

$$r_2 = \sqrt{(1 - \mu + x)^2 + y^2 + z^2} \tag{E.3}$$

Using r_1 and r_2 the equations of motion (EOM) can be defined for the CRTBP. Note that the EOM are given in $\ddot{x} =$, $\ddot{y} =$ and $\ddot{z} =$ and the full derivation can be found in [wakker book].

$$\ddot{x} = x + 2\dot{y} - (1-\mu)\frac{x+\mu}{r_1^3} - \mu\frac{x+\mu-1}{r_2^3} \tag{E.4}$$

$$\ddot{y} = -2\dot{x} + \left(1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3}\right) y \tag{E.5}$$

$$\ddot{z} = \left(\frac{\mu-1}{r_1^3} - \frac{\mu}{r_2^3}\right) z \tag{E.6}$$

Then by integrating $[\dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]^T$ over a certain time span, the state $[x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ can be determined, which describes the nominal trajectory of the satellite in the normalized rotating reference frame.

New units for length, mass, time and velocity are used based on the distance and the masses of the primaries. Using the assumption that the Earth and Moon circular rotate their barycenter, the Earth-Moon distance is constant set on 384 400 km, which is the characteristic length (l^*). The collective mass of the two primaries is the *characteristic mass (m^*) in kg* and is obtained by adding the mass of the two primaries: $m^* = m_{P1} + m_{P2}$. The mass of the third body, the satellite, is negligible small and therefore not taken into account for the characteristic mass. Using the characteristic length, the characteristic mass and the universal gravitational constant G, the characteristic time (t^*) in s and velocity (v^*) in m/s can be determined as described in Equation E.7 and Equation E.8

$$t^* = \sqrt{\frac{(l^*)^3}{G \cdot m^*}} \tag{E.7}$$

$$v^* = \frac{l^*}{t^*} \tag{E.8}$$

E.2. The International Celestial Reference System and Frame (ICRF)

The conversion from the CRTBP to the ICRF is shortly elaborated.

Before starting the conversion, one must have obtained the following:

1. The non-dimensional states in CRTBP frame in $\mathbf{x}_{norm.bc_based} = [x_{norm}, y_{norm}, z_{norm}, \dot{x}_{norm}, \dot{y}_{norm}, \dot{z}_{norm}]^T$
2. The three-body gravitational parameter μ as defined as in Equation E.1
3. The characteristic mass m^*, which is the collective mass of the two primaries in kg
4. The state vector of the secondary based with respect to the center of the primary in a dimensional non-rotating reference frame $\mathbf{x}_{P2} = [x_{P2}, y_{P2}, z_{P2}, \dot{x}_{P2}, \dot{y}_{P2}, \dot{z}_{P2}]^T$
5. Based on the state vector of the primary, the characteristic length l^* in meter ($l^* = \sqrt{x_{P2}^2 + y_{P2}^2 + z_{P2}^2}$)
6. Based on the characteristic length and mass, the characteristic time (t^* and velocity v^* can be determined using Equation E.7 and Equation E.8.

First, the non-dimensional state vector with respect to the barycenter, is transformed to the state vector with respect to the center of mass of the primary. This is done by subtracting the normalized state vector of P1 of the normalized state vector of the satellite, as done in Equation E.9

$$\mathbf{x}_{norm.P1\_based} = \mathbf{x}_{norm.bc_based} - \mathbf{x}_{P2} = \mathbf{x}_{norm.bc\_based} + [-\mu, 0, 0, 0, 0, 0]^T. \tag{E.9}$$

Second, the non-dimensional primary based state vector is transformed to dimensional primary based state vector, as done in Equation E.10

$$\mathbf{x}_{dim.P1\_based} = \begin{bmatrix} l^* & 0 & 0 & 0 & 0 & 0 \\ 0 & l^* & 0 & 0 & 0 & 0 \\ 0 & 0 & l^* & 0 & 0 & 0 \\ 0 & 0 & 0 & v^* & 0 & 0 \\ 0 & 0 & 0 & 0 & v^* & 0 \\ 0 & 0 & 0 & 0 & 0 & v^* \end{bmatrix} \begin{bmatrix} x_{norm.P1\_based} \\ y_{norm.P1\_based} \\ z_{norm.P1\_based} \\ \dot{x}_{norm.P1\_based} \\ \dot{y}_{norm.P1\_based} \\ \dot{z}_{norm.P1\_based} \end{bmatrix} \tag{E.10}$$

Now that the state vector is written in primary centered dimensional reference frame, it is needed to add the Moons motion in order to complete the state transition. In order to do so, the full transition matrix is compute. The attitude matrix $\mathbf{A}$ is computed between the inertial and rotating reference frame using $\mathbf{r}_{P2} = [x_{P2}, y_{P2}, z_{P2}]$ and $\mathbf{v}_{P2} = [\dot{x}_{P2}, \dot{y}_{P2}, \dot{z}_{P2}]$. Using the following equations

$$\begin{aligned} \mathbf{X}_{ref} &= \frac{\mathbf{r}_{P2}}{|\mathbf{r}_{P2}|} \\ \mathbf{Z}_{ref} &= \frac{\mathbf{r}_{P2} \times \mathbf{v}_{P2}}{|\mathbf{r}_{P2} \times \mathbf{v}_{P2}|} \\ \mathbf{Y}_{ref} &= Z_{ref} \times X_{ref}, \end{aligned} \tag{E.11}$$

the attitude matrix **A** can be obtained, which is a 3x3 matrix.

$$\mathbf{A}_{ref} = \begin{bmatrix} \mathbf{X}_{ref} \\ \mathbf{Y}_{ref} \\ \mathbf{Z}_{ref} \end{bmatrix} \tag{E.12}$$

Then, the instantaneous angular velocity is obtained using Equation E.13

$$\omega = \frac{|\mathbf{r}_{P2} \times \mathbf{v}_{P2}|}{|\mathbf{r}_{P2}|^2}, \tag{E.13}$$

which is used to compute angular velocity matrix **B** in Equation E.14

$$\mathbf{B}_{ref} = \begin{bmatrix} \omega \cdot A_{ref}[1,2] & -\omega \cdot A_{ref}[1,1] & 0 \\ \omega \cdot A_{ref}[2,2] & -\omega \cdot A_{ref}[2,1] & 0 \\ \omega \cdot A_{ref}[3,2] & -\omega \cdot A_{ref}[3,1] & 0 \end{bmatrix} \tag{E.14}$$

Then, using A_{ref} and B_{ref}, the full transformation matrix can be obtained, as done below:

$$\mathbf{T} = \begin{bmatrix} A_{ref} & \emptyset_{3\times3} \\ \emptyset_{3\times3} & B_{ref} \end{bmatrix} \tag{E.15}$$

and using the full transformation matrix, the dimensional, primary based state vector including the motion of the secondary with respect to the primary is obtained by

$$\mathbf{x}_{ICRF} = \mathbf{T}\mathbf{x}_{dim.P1\_based}. \tag{E.16}$$