TÉCNICO LISBOA

Orbital Collision Warning and Avoidance

João Pedro Rosa da Silva

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. Paulo Jorge Soares Gil

Examination Committee

Chairperson: Prof. Paulo Jorge Coelho Ramalho Oliveira
Supervisor: Prof. Paulo Jorge Soares Gil
Member of the Committee: Prof. Rita Maria Mendes de Almeida Correia da Cunha

January 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would like to pay my special regards to Bruno Carvalho, for giving me the opportunity to explore and develop my skills on a topic that interested me a lot and allow me to have professional experience in such a challenging industry. It was a pleasure to work with one of the persons that seeks to develop the space industry, being a pioneer in Portugal.

I also wish to show my gratitude to D-Orbit PT team, Hugo Gomes, André Pires, Miguel Morujão, Luís Teixeira and João Fragoso for the help given and availability shown. I would also like to thank the opportunity to attend ESA's Space Debris Course, which was an incredible experience and a major help to write this thesis.

Last, but not least, I would like to thank my brother, Roberto Silva, and sister in law, Mafalda Moço, for always be there for me when I needed, as well as my friends and family for listening to me talking about my thesis, even without understanding anything of it.

Abstract

The growth of the number of objects in space increases the probability of collisions that will generate debris. This thesis will contribute for increasing the SSA of D-Orbit and to tackle the problem of space debris by mitigating it. The aim is to develop a collision warning tool to be integrated in the mission control software developed by D-Orbit.

Based on TLEs retrieved from SpaceTrack and on SGP4 theory, the positions of every space objects for several time instants were calculated. Filtering techniques were implemented to select only the objects that may cause a collision with the object of value. The calculation of these positions have associated uncertainties, these were estimated in the form of a covariance matrix and a probability of collision was calculated, allowing conjunction analysis and assess if an avoidance manoeuvre is needed or not.

The developed filtering techniques reduced the execution time of the code, allowing to predict conjunctions, for D-Orbit's satellite. The covariances estimation revealed large uncertainties for LEO objects especially in the along-track direction, due to the limitations of SGP4 theory in drag modelling. Analysis and comparisons with other collision warning services demonstrated that a more refined position calculation, with more accurate data, is needed. With TLEs it's possible to develop a collision warning tool, however with high uncertainties. The SSA of D-Orbit was increased and the objective completed.

Keywords

Space Debris, TLE, SGP4, Covariance, Probability of Collision, Conjunction Analysis

Resumo

O crescimento do número de objetos no espaço aumenta a probabilidade de colisões que irão gerar detritos. Esta tese contribuirá para aumentar a SSA da D-Orbit e combater o problema dos detritos espaciais, mitigando-os. O objetivo é desenvolver uma ferramenta de alerta de colisão para ser integrada no software de controlo de missão desenvolvido pela D-Orbit.

Com base em TLEs provenientes do SpaceTrack e na teoria SGP4, foram calculadas as posições de todos os objetos espaciais para vários instantes de tempo. Técnicas de filtragem foram implementadas para selecionar apenas os objetos que podem causar uma colisão com o objeto de valor. O cálculo dessas posições tem incertezas associadas, estas foram estimadas sob a forma de matriz de covariância e foi calculada uma probabilidade de colisão, permitindo uma análise da aproximação dos satélites e avaliar se uma manobra de evasão é necessária ou não.

As técnicas de filtragem desenvolvidas reduziram o tempo de execução do código, permitindo prever aproximações entre objectos, para o satélite da D-Orbit. A estimativa de covariâncias revelou grandes incertezas para objetos em LEO, especialmente na direção do vector velocidade, devido às limitações da teoria SGP4 na modelação do *drag*. Análises e comparações com outros serviços de alerta de colisão demonstraram que um cálculo de posição mais refinado é necessário, complementando os TLEs com dados mais precisos. É possível desenvolver uma ferramenta de alerta de colisão com TLEs, porém com grandes incertezas. No final, a SSA da D-Orbit foi aumentada e o objetivo concluído.

Palavras Chave

Detritos Espaciais, TLE, SGP4, Probabilidade de Colisão, Análise de Encontros de Satélites, Covariância

Contents

List of Figures

List of Tables

Nomenclature

A	Area
B	Ballistic Coefficient
B^*	Drag Term
C_D	Drag Coefficient
D_{th}	Distance Threshold
J_i	Zonal Harmonics
Q	Smaller Apogees
R_{cr}	Critical Radius
R_{th}	Threshold Radius
Δt	Time interval
ρ	Density
ρ_0	Atmospheric Density at $1\,\mathrm{atm}$
$\mathbf{r}$	Relative Position Vector
a	Semi-Major Axis
e	Eccentricity
g_0	Gravity Acceleration at Sea Level
h	Orbit Altitude
m	Mass
q	Larger Perigee

r_a Apogee Radius

r_p Perigee Radius

v_{app} Approach Velocity

v_{esc} Escape velocity

Subscripts

i, j, k Computational indexes

x, y, z Cartesian components

Superscripts

T Transpose

Acronyms

ADR	Active Debris Removal
CA	Collision Avoidance
CDM	Conjunction Data Message
CRASS	Collision Risk Assessment Software
CSpOC	Combined Space Operations Center
CW	Collision Warning
ECI	Earth-Centered Inertial
EOL	End of Life
ESA	European Space Agency
EUSST	European Space Surveillance and Tracking
GEO	Geosynchronous Equatorial Orbit
GMV	Grupo Mecánica Vuelo
GP	General Perturbations
GPS	Global Positioning System
HBR	Hard Body Radius
IADC	Inter-Agency Space Debris Coordination Committee
ID	Identification
ION-MK01	In-Orbit Now Mark 01 Satellite
ISS	International Space Station
LEO	Low Earth Orbit
MD	Miss Distance
MEO	Medium Earth Orbit
NASA	National Aeronautics and Space Administration

NSDTC Space Debris Office, “Notes from Space Debris Training Course,” ESA, Jun. 2020.

NORAD North American Aerospace Defense Command

NTW Normal In-track Cross-track

OD Orbit Determination

PC Probability of Collision

PD Payload Debris

PDF Probability Density Function

PF Payload Fragmentation Debris

PM Payload Mission Related Objects

PL Payloads

RAAN Right Ascension of the Ascending Node

RB Rocket Body

RD Rocket Debris

RF Rocket Fragmentation Debris

RM Rocket Mission Related Objects

RTN Radial Tangential Normal

SDP Simplified Deep-Space Perturbations

SGP Simplified General Perturbations

SOCRATES Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space

SP Special Perturbations

SSA Space Situational Awareness

SSN Space Surveillance Networks

TCA Time of Closest Approach

TEME True Equator Mean Equinox

TLE Two Line Element

UI Unidentified

USA United States of America

UTC Universal Time Coordinated

18SPCS 18th Space Control Squadron

1

Introduction

1.1 Objectives and Motivation

This thesis has the objective of developing a Collision Warning (CW) tool that is going to work as an add-on to the mission control software, named Aurora, developed at D-Orbit PT, in order to increase Space Situational Awareness (SSA) and the safety of a specific satellite towards all the other tracked space objects (one-on-all collision assessment), contributing to debris mitigation, which is currently a part of the solution for the increasing number of space debris. Having a CW system allows the operator of the satellite to be aware of possible future collisions that would cause loss of the mission and eventually turn the satellite inoperable and could even generate more debris.

This tool aims at a reduction of the execution time to detect conjunctions by implementing filtering techniques and allow the operator to assess a conjunction by analysing the calculated parameters, such as: Time of Closest Approach (TCA), Miss Distance (MD), relative velocity, covariance, scaling factor, Probability of Collision (PC) and maximum PC. These will help in the decision of perform, or not, an avoidance maneuver. Several methods and tools to determine and predict the positions of the satellite will be reviewed, as well as the processes to estimate the uncertainties of predicted positions using

only Two Line Element (TLE) data. After this, an approach on the calculation of PC will be studied and implemented.

1.2 Spaceflight and Space Debris

Since 1957, with the launch of Sputnik, man-made space debris has been increasing, becoming a danger for the continuity of spaceflight sustainability. Even a small object, in the order of centimeters, can cause a catastrophic collision, due to the high relative velocities involved. In order to have some intuition of the energies involved in a collision, an example is presented. Imagine a collision in Low Earth Orbit (LEO), on $h = 500\,\mathrm{km}$, then the orbital velocity is approximately $8\,\mathrm{km/s}$. The maximum relative velocity between both objects is twice the orbital velocity, i.e. $16\,\mathrm{km/s}$, if a head-on collision happens, which is very unlikely. Considering a relative velocity of $14\,\mathrm{km/s}$, it is possible to verify that the kinetic energy involved is very high. The energy involved in a collision of a $10\,\mathrm{g}$ object is comparable to a car of $2000\,\mathrm{kg}$ at $112\,\mathrm{km/h}$ crushing into a wall, so even small objects should be considered in the collision warning process.

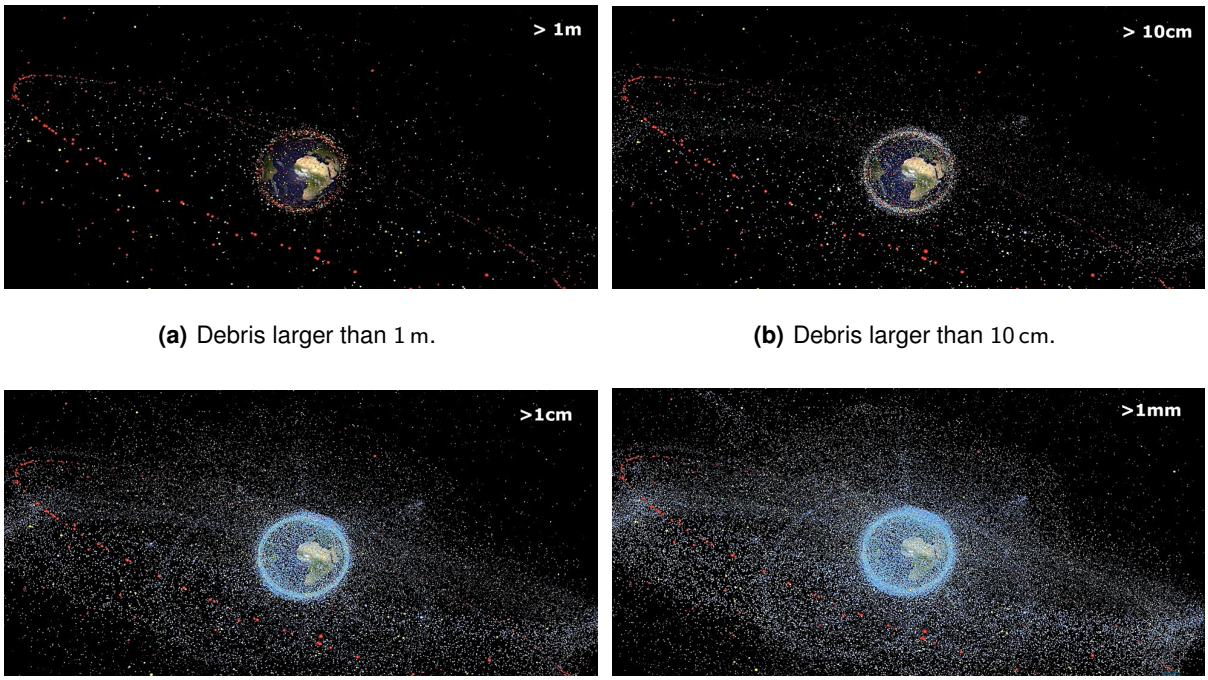
In the course of the spaceflight history, there are some recorded events that generated large amounts of space debris. The most serious event was in 2009, a collision between Iridium-33 and Cosmos 2251 occurred, forming a large cloud of debris and resulting in a wake-up call for the space community to invest in solving the problem of space debris. It's of the best interest of everybody to keep sustainability in space, to continue their business or at least to maintain their assets safe.

Relating this collision to the Kessler Syndrome [8] that states that once past a certain critical mass, the total amount of space debris will keep on increasing: collisions give rise to more debris and lead to more collisions, in a chain reaction. A collision generates debris that will increase the probability of colliding with each other, or other objects, generating even more debris. With this we can conclude that the collision of 2009 can only be considered as catastrophic.

Figure 1.1 illustrates the environment of space debris. The first considers only tracked objects larger than approximately $1\,\mathrm{m}$, the second $10\,\mathrm{cm}$ and the last two, enter in the field of trying to simulate and model the population of debris larger than $1\,\mathrm{cm}$ and $1\,\mathrm{mm}$. The images in Fig. 1.1 suggest that there are different amounts of debris, either by size or orbital region, for example it is possible to note an outer belt, the Geosynchronous Equatorial Orbit (GEO) region.

A study about the evolution of debris for the several size populations was performed by Klinkrad [1]. The evolution of the number of objects for the next 100 years, if no measures are adopted, is presented in Fig. 1.2, for the size classes of $1\,\mathrm{m}$, $10\,\mathrm{cm}$, $1\,\mathrm{cm}$ in the cases of LEO, Medium Earth Orbit (MEO) and GEO. The LEO region is the most critical one, since it is already very crowded.

There are two main approaches to solve the space debris problem: debris mitigation and debris removal. The first is concerned about establishing guidelines for operations in space, for example pas-



(a) Debris larger than 1 m.

(b) Debris larger than 10 cm.

(c) Debris larger than 1 cm.

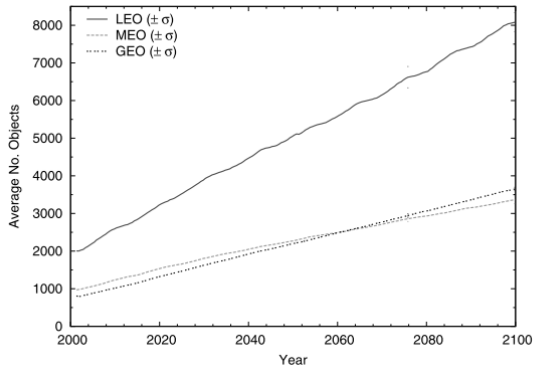
(d) Debris larger than 1 mm.

Figure 1.1: Space debris population with dimensions of 1 m, 10 cm, 1 cm and 1 mm. The images are not to scale. [1]

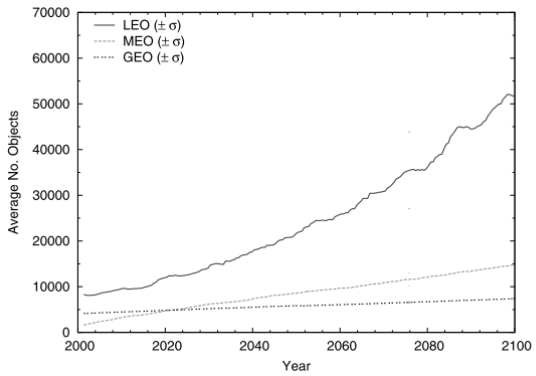
sivation of spacecrafts, i.e., when an aircraft finishes its mission, it should deorbit and/or empty their tanks to avoid explosions. The Active Debris Removal (ADR) approach is concerned about finding ways and technologies to remove debris from space, which is a more difficult approach in technical terms, but still gained some strength in the past few years. The effect of the application of these measures and approaches in the amount of space debris is presented in Fig. 1.3. The passivation techniques are not sufficient to solve the debris problem, because the amount of debris will continue to increase. It's only possible to solve the debris problem with a strong compliance of the mitigation guidelines together with ADR processes.

The United States of America (USA) observes and tracks space objects through their Space Surveillance Networks (SSN), which is composed by a group of facilities and ground stations with several types of sensors like radars and telescopes, see Fig. 1.4. Europe also has its own surveilance system network, European Space Surveillance and Tracking (EUSST), with their own catalog and collision avoidance services. However, in this thesis, the USA catalog was used because it's publicly available on SpaceTrack. USA lists orbital information of about 20 000 objects of 10 cm, or larger, in LEO and objects of 30 cm, or larger, in GEO, maintaining and updating this information daily in a catalog. There are two types of catalog: the Special Perturbations (SP) catalog and the General Perturbations (GP) catalog

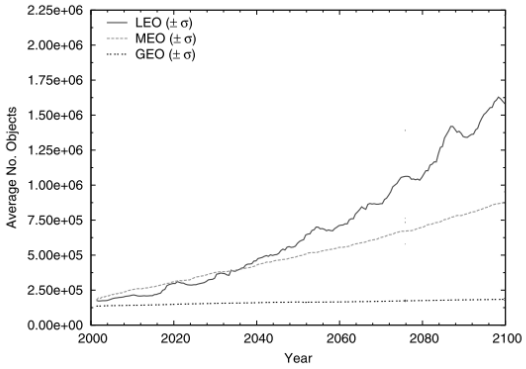
[1] Space Debris Office, "Notes from Space Debris Training Course," ESA, Jun. 2020. (NSDTC)



(a) Evolution of the amount of objects larger than 1 m.



(b) Evolution of the amount of objects larger than 10 cm.



(c) Evolution of the amount of objects larger than 1 cm.

Figure 1.2: Average amount of objects and their evolution with time, in a prediction of 100 years, for the LEO, MEO, GEO region. [1]

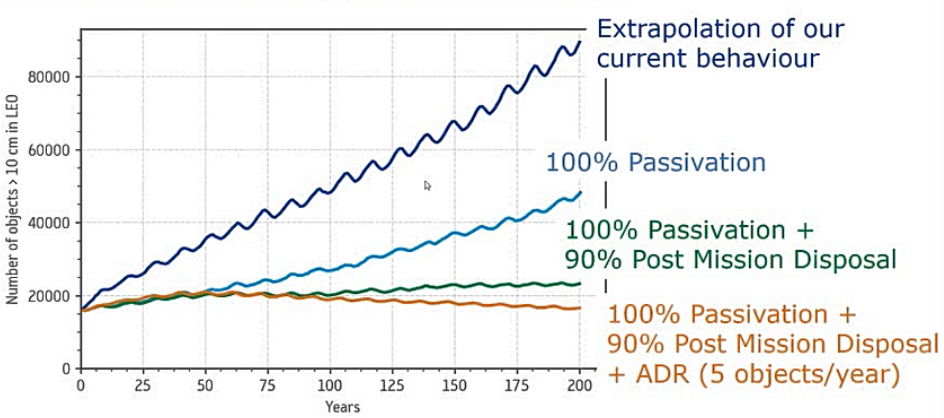


Figure 1.3: Effect of measures adopted in the evolution of the number of objects larger than 10 cm, in the LEO region, over a prediction of 200 years. Figure taken from NSDTC.

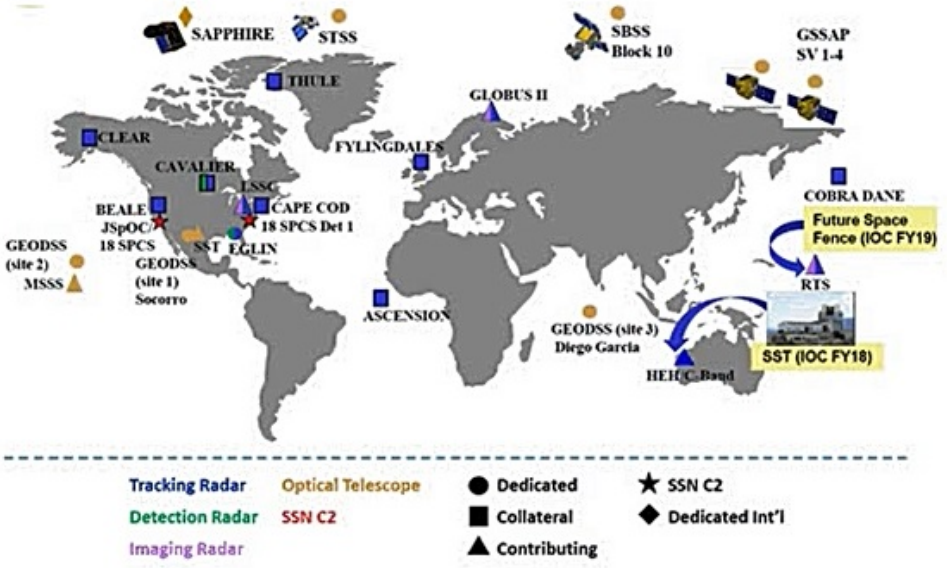


Figure 1.4: Characterization of the SSN tracking systems and their location in Earth. [2]

The first is based on SP techniques that combine numerical integration of the perturbations, resulting in higher fidelity compared to GP catalog. The GP catalog was created on the 60s and it is processed by analytical approaches, namely the Simplified General Perturbations (SGP)4 theory used to generate TLEs and propagate them. [9]

When it became computationally affordable to run intensive SP techniques, USA started to use them to maintain their catalogs. This SP catalog is not public, USA kept only the GP catalog public, which contains TLEs [10]. TLEs can be used to compute and predict space objects positions recurring to SGP4 theory. If one knows where the objects will be in the future, it is possible to know if there will be close approaches between them. If the positional errors are known, it is possible to calculate a collision probability and if this probability is too high, the operator should perform an avoidance maneuver to prevent collisions and the generation of more space debris. This way, collision warning and avoidance can be considered part of the mitigation approach to solve the debris problem.

There are errors in the calculation of the actual position of space objects due to measurement errors in the tracking systems (radars, lasers, etc) and in their conversion to TLEs. There are also additional errors associated with the propagator used, in this thesis it will be used SGP4 [11]. After calculating the positions of the objects with the respective errors estimated, a conjunction analysis may be performed, involving the calculation of parameters that characterize the conjunction, helping in the decision of performing an avoidance maneuver or not.

Conjunction analysis is a relatively new subject. In 2005, National Aeronautics and Space Administration (NASA) started an operational Collision Avoidance (CA) service for its A-train constellation [12].

[2] NSDTC

Before 2009, only low-precision catalogs in the form of TLE were available, nonetheless, in 2009, USA improved the quality of the tracking data after the collision between Iridium-33 and Cosmos 2251. They started to provide Conjunction Data Message (CDM) to the operators, with significant information for conjunction analysis, if conjunctions were detected [12]. In 2013, TLEs started to be generated differently for some objects, increasing their accuracy [13].

Comparing the operational aspects of several agencies [12], some depend on CDMs provided by Combined Space Operations Center (CSpOC), others create their own CDMs through TLEs or tracking data not publicly available. In this thesis, TLEs will be used to create CDMs. The full catalog of TLEs is available on SpaceTrack website and the tool developed will be based on those TLEs.

The CW systems of the several agencies differ in the algorithms, analysis methods and their implementation. There are different tools to analyse conjunctions for each space agency. For example, the standard threshold probability to emit a collision warning may be equal between agencies, but a conjunction may be considered safe for one agency and dangerous to another agency due to differences in algorithms. The algorithms used in conjunction analysis are 10-20 years old [12], in spite of existing better ones, they are not used because changes in operational systems are difficult and slow, which creates a big difference in the methods developed in the research community and the ones that are really implemented in CW systems.

1.3 Thesis Outline

In chapter 2, an introduction to the environment of space debris is made, the numbers relative to space debris are presented and an analysis of the sources and distribution of debris is performed. This is followed by an explanation of the methods and technologies used to track space objects orbiting the Earth.

In chapter 3, a background about TLEs and its relation to the SGP4 theory for its propagation is presented. It is also made a description of the SGP4 theory history and a brief explanation of the models included in it. Since the output of the SGP4 theory is a state vector in a True Equator Mean Equinox (TEME) reference frame, the details related to this frame are presented. Finally, the SGP4 implementation used in this thesis is validated.

In chapter 4, the problem of calculating the position for large amount of objects in several time instants is explained and the techniques to filter objects and reduce the computational effort are presented. After this, a combination of these techniques is implemented. The implementation is explained and validated.

In chapter 5, the uncertainties on the calculation of the position are estimated in the form of a covariance matrix. The need for calculating these uncertainties is explained and some statistical background is given. Afterwards, the methods in literature to determine covariance for TLEs are presented and the

implementation of the chosen one is explained and validated.

In chapter 6, the method for calculating a probability of collision and its maximum is explained and implemented, followed by the analysis of three tests: extreme and real cases, comparison with LEOLabs conjunction and analysis of the collision between Iridium-33 and Cosmos 2251.

In chapter 7, conclusions, achievements and topics requiring some improvements for future work are presented.

2

Space Debris Environment

2.1 Debris By the Numbers

According to European Space Agency (ESA), space debris is defined as all non-functional human-made objects, orbiting Earth or re-entering Earth's atmosphere. This does not include natural meteoroids. [3]

The majority of space objects are space debris due to growing space activity and orbital collisions between debris-debris and spacecraft-debris. Table 2.1 presents important numbers for an overview of space debris, in February 2020, since the launch of the first satellite, in 1957.

Table 2.1: Numbers related to space debris, by ESA's Space Debris Office, in February 2020, [2]. The number of debris with size smaller than 10 cm are obtained by statistical models.

Satellites			**Debris**			
Placed into orbit	Stil in space	Still Functioning	1 mm to 1 cm	1 cm to 10 cm	$\geq$ 10 cm	Regularly tracked
9600	5500	2300	128 000 000	900 000	34 000	22 300

The number of debris regularly tracked by SSN is important because these are the debris whose

[3] "Frequently Asked Questions," ESA, Space Debris Office, accessed on 2020/05/27. [Online]. Available: https://www.esa.int/Safety_Security/Space_Debris/FAQ_Frequently_asked_questions

position is possible to know and these are the ones taken in consideration for the purpose of this thesis, as well as the satellites still in space. Note that these numbers are frequently updated by ESA. Ideally all objects should be considered, but this is not possible in practice due to the tracking capacity of the SSN, that only tracks objects larger than 10 cm approximately. [4]

2.1.1 Sources

One of the most obvious sources of space debris is the decommission of satellites that are considered debris since the moment they're not useful anymore, but remain on orbit. Rocket upper stages also contribute to the increasing numbers of space debris. The upper stages need to have the orbital velocity of the deployed payload, so it will also stay in orbit, in some cases. Mission related objects are another source, some objects are only needed at the beginning of the mission and after that, they are released, for example launch adapters that connect the satellite to the rocket.

Sometimes, there is also lost equipment during extra-vehicular activities, explosions from residual fuel of decommissioned spacecrafts that are degraded by the harsh space environment, collisions between satellite-satellite, satellite-debris and debris-debris, but also anti-satellite missile tests. Another concern is the micro space debris like slag and dust particles from solid rocket motors and the release of reactor coolant, that results in frozen droplets.

Space objects can be divided in two categories: the ones that can be traced back to a launch event and know their nature, and the ones for which this is not possible. The later will be called Unidentified (UI) and the others will be divided into the following subcategories [2]:

- Payloads (PL) - space objects designed to perform specific function in space excluding launch functionality. Includes operational satellites and calibration objects.
- Payload Mission Related Objects (PM) - space objects which served a purpose for the function of a payload. Covers for optical instruments are an example of it.
- Payload Fragmentation Debris (PF) - space objects fragmented or unintentionally released from a payload whose genesis can be traced back to a unique event. Includes objects created when a payload explodes or collides with another object.
- Payload Debris (PD) - space objects fragmented or unintentionally released from a payload whose genesis is unclear, but orbital and physical properties enable correlation with a source.
- Rocket Body (RB) - space object designed to perform launch related functionality. This includes the various orbital stages of launch vehicles, but not payloads that release smaller payloads themselves.

[4] NSDTC

- Rocket Mission Related Objects (RM) - space objects intentionally released and served a purpose for the function of a rocket body. Engines are an example of it.

- Rocket Fragmentation Debris (RF) - space objects fragmented or unintentionally released from a rocket body whose genesis can be traced back to a unique event. This class includes objects created when a launch vehicle explodes.

- Rocket Debris (RD) - space objects fragmented or unintentionally released from a rocket body whose genesis is unclear but orbital or physical properties enable a correlation with a source.

Considering that only PL objects are useful, Fig. 2.1 shows that currently most catalogued objects are debris, approximately 6000 are PL objects and 20 000 are debris. The amount of objects has been rising since the beginning of spaceflight due to generation of more debris and increased space activity, which can be seen in Fig. 2.4, but also due to better tracking technologies, increasing the number of unidentified objects.

Inside the debris category, most of the objects are PF and RF that suffered a significant increase in the years between 2007 and 2010 probably because of the Chinese anti-satellite missile test in 2007 and the collision of Iridium-33 and Cosmos 2251 in 2009.

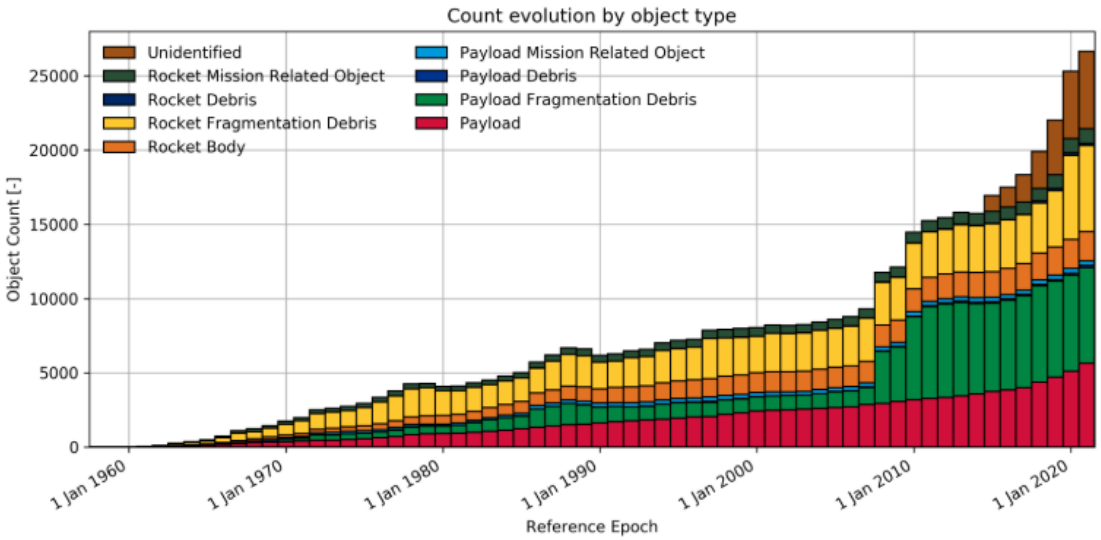


Figure 2.1: Number of cataloged objects in space since 1960 until 2020. The colors in the bars show the type of objects. [2]

Since fragmentation is the most common source of space debris, it is important to know why fragmentation events happen: [2]

- Accidental - subsystems which showed design flaws ultimately leading to breakups in some cases.

- Aerodynamics - break-ups occurs due to interaction with the Earth's atmosphere. For example, on Molniya orbits.
- Anomalous - unplanned separation of one or more detectable objects from a satellite that remains intact.
- Collision
- Deliberate - satellites deliberately destroyed.
- Electrical - stored energy for non-passivated batteries might lead to explosions.
- Propulsion - stored energy for non-passivated propulsion-related subsystems might lead to explosions.
- Unknown

Figure 2.2 shows that most fragment events are due to propulsion, deliberate and anomalous causes. Although deliberate causes have been decreasing their share in the number of fragmentation events, maybe due to a higher awareness of the space industry in recent years to the space debris problem, propulsion causes don't show a significant reduction meaning that mitigation guidelines, like passivation techniques of emptying fuel tanks after decommissioning, are not yet being fully complied.

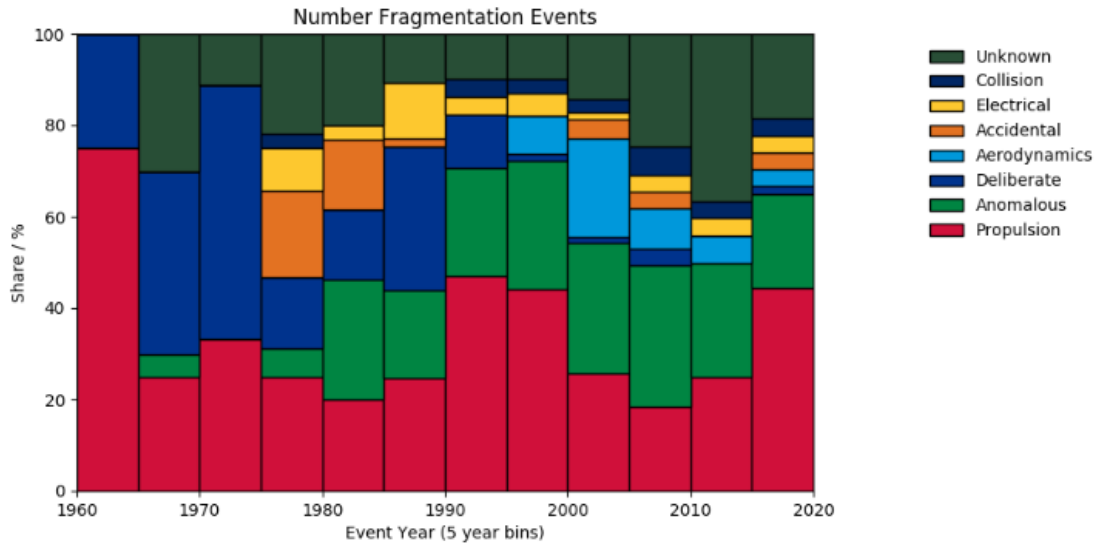


Figure 2.2: Share of each fragmentation cause in the number of fragmentation events. The colors in the bars show the cause of fragmentation. [2]

2.1.2 Distribution

An important factor to consider when analysing the space debris environment is its distribution. Debris are not evenly distributed through all types of orbits. [2], because there are types of orbits more attractive to the space industry, like the LEO and GEO regions. Figure 2.3 confirms that most cataloged objects are in the LEO region. Note that the number of objects has been increasing in every type of orbit.

The characteristic that makes LEO region so attractive is the low latency for applications that require real-time responsiveness, like communications or trading, LEO satellites will be closer to Earth reducing the time of communication. Another obvious reason for the attractiveness of LEO region is that it costs much less to put a satellite in orbit.

The main characteristic of the GEO region is that satellites, in this orbit, have a period of 24 hours, the same period of Earth rotation, allowing a satellite to always be visible in the same position relative to a point in the Earth surface, what is useful for TV broadcast, for example.

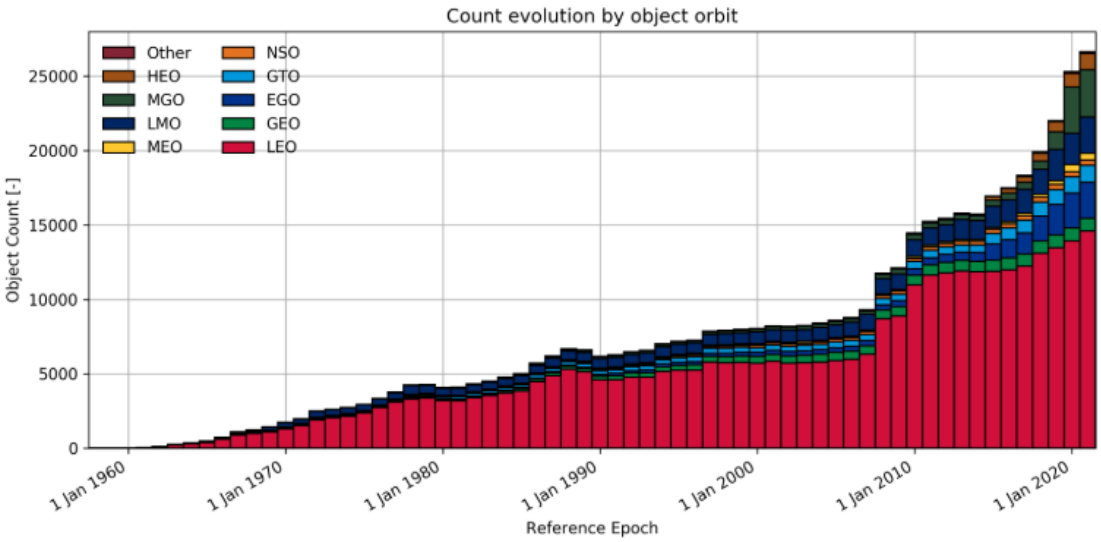
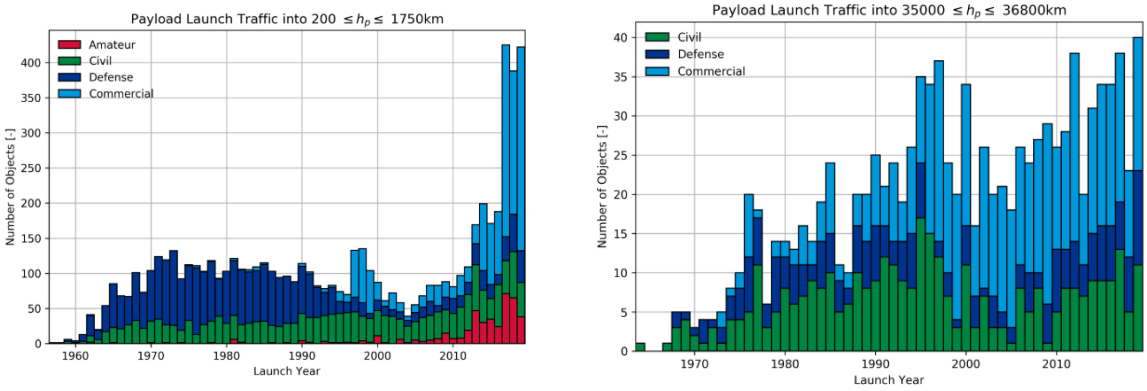


Figure 2.3: Number of cataloged objects and their evolution in time. The colors in the bars show the type of orbit where the objects are in. [2]

Figure 2.4 shows that, in the LEO region, civil activity maintained approximately the same, the defense activity has been decreasing giving place to commercial activity which increased a lot in the past few years, as well as amateur activity, fuelled by the proliferation of small satellites [2]. In this region, there is a peak before the year 2000 due to the launching of the Iridium and Globalstar constellations. There is also a peak is the past few years due to the Iridium Next Generation constellation.

In the GEO region, the number of launched objects is much less than in the LEO region. This region is also dominated by commercial activity and there is no amateur activity mainly due to launch costs.



(a) Type of activity on LEO region.

(b) Type of activity on GEO region.

Figure 2.4: Number of cataloged objects launched each year. The colors of the bars show the type of activity. [2]

2.1.3 Growth Mitigation

As a priority for space sustainability, debris mitigation guidelines started to be proposed, including the passivation of spacecrafts and End of Life (EOL) disposal. Figure 2.2 shows that explosions are a big part of the number of fragmentation events thus, measures like passivation can be taken to minimize them. For example, burn up or release the residual fuel of spacecrafts and upper stages, depressurize vessels and discharge batteries.

EOL disposal can be performed in both LEO and GEO orbits. LEO satellites can be propelled into a decaying orbit, in order to de-orbit before a guideline time of 25 years. Some of these solutions requires more propellant, increasing costs, others don't require extra fuel, for example deploying an electrodynamic tether or a sail to increase drag. GEO satellites can be placed into a graveyard orbit, above the geostationary ring, in order to not interfere with other missions but, on the long term, this solution can become a problem due to collisions in this orbit, generating debris that may interfere with GEO missions. [5]

There are also some proposed solutions for de-orbiting non-operating spacecrafts or debris using lasers [14], or actively removing debris, capturing it [15]. These guidelines are not mandatory, but space agencies and nations are implementing their own regulations that address debris mitigation. [5]

Two protected regions were established by the Inter-Agency Space Debris Coordination Committee (IADC), as depicted in Fig. 2.5. In these regions, the mitigation guidelines should be strongly applied. [5] The LEO protected region is a shell that extends from the surface of a spherical Earth with an equatorial radius of 6378 km, up to an altitude of 2000 km. The GEO protected region is a segment of a spherical shell defined by a lower altitude boundary of the geostationary altitude (35 786 km) minus 200 km and an

[5] NSDTC

upper altitude boundary of the geostationary altitude plus 200 km, in a latitude sector between 15° South and 15° North.

Statistics show that [16]:

- Around 78 % of small payloads launched to LEO protected region, during the current decade, operate in orbits which adhere to space debris mitigation measures, by decaying naturally before 25 years.
- Between 15 % and 25 % of payloads reaching EOL, during the current decade, in the LEO protected region and in a non-compliant orbit, attempt to comply with the space debris mitigation measures. Of these, between 5% and 15% do it successfully.
- Between 85 % and 100 % of all payloads reaching EOL, during the current decade, in the GEO protected region, attempt to comply with the space debris mitigation measures. Of these, between 60 % and 85 % do it successfully.

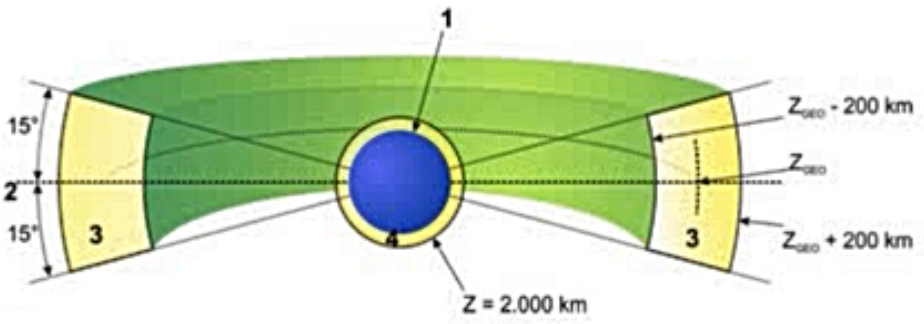


Figure 2.5: LEO and GEO protected regions. [6]

2.2 Space Surveillance and Tracking

Tracking debris is possible by ground-based and space-based telescopes, as well as radars. As shown in Table 2.1, about 22 300 objects are regularly tracked and cataloged by the SSN. There are two types of performance for sensors: tracking and surveillance. [6]

On the first, the sensor follows a single object at a time with a very narrow field of view focusing in one point, what requires *a priori* knowledge about the object's position. This type of performance is used to refine information about existing orbits and give more information for a high-priority target, for example in close approaches.

[6] NSDTC

On the second, the sensor scans regions of the sky with a larger field of view, so it's used to detect new objects and track multiple objects simultaneously. These are the ones used by USA to maintain its catalog. [7]

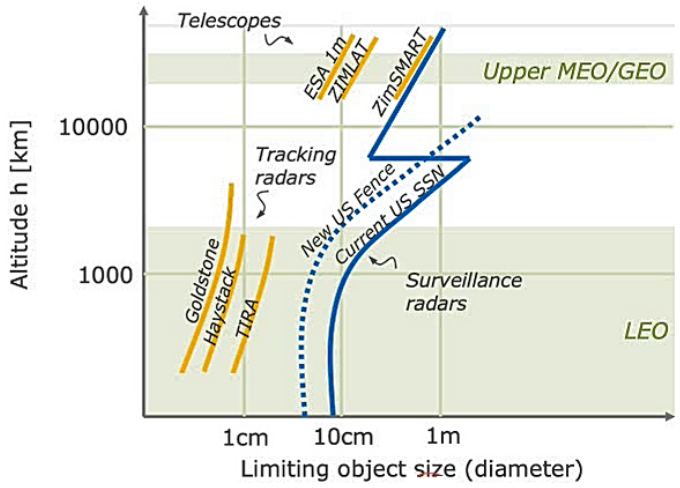


Figure 2.6: Performance of radars and telescopes indicating the size of the smallest object detectable by altitude. [7]

Figure 2.6 shows that the smallest objects tracked and maintained in the USA catalog are of approximately 10 cm in LEO and 30 cm before MEO/GEO. Smaller objects in the order of 2 mm can be observed in LEO using tracking radars, but they cannot be regularly tracked because their orbits are too unstable.[7]

Telescopes and radars are positioned at different longitudes and latitudes around the world to observe space objects from different places. These are mostly operated by Europe, Russia, Japan and USA, however, not all telescopes and radars contribute for the catalog maintained by the USA. Space surveillance aims to build-up and maintain a catalogue of orbital states and other information about objects in space. For that to happen, observations from sensors are required and then orbit determination processes are performed using those observations to obtain orbital states with associated covariances. This is how TLEs are generated, although they are not available with associated covariances.

[7] NSDTC

3

Characteristics of TLEs

3.1 TLE Propagation

TLE propagation means that an orbit is going to be propagated. To propagate an orbit means that the state vector (position and velocity) of the object is going to be predicted (calculated in the future), using the starting conditions of the object at a particular time, the mean orbital elements in TLEs. In Appendix A, a description about TLE format and data is made.

By the Newton's laws, if one knows all forces that act on an object, it's possible to determine its position in the future. In this case, the main force is Earth's gravity, which is assumed to be spherical and homogeneous at first instance, however, there are perturbations. The more complete is the model of these perturbations, the better is the predicted result. Some examples of perturbations are: gravitational perturbations because Earth is not spherical and homogeneous; gravity forces due to Moon and Sun; drag due to Earth's atmosphere and radiation pressure from the Sun.

Perturbations can be divided into three categories: secular, long-periodic, short-periodic. Secular means that the perturbation is constant in time, resulting in a linear behaviour of orbital elements. Periodic means that the perturbation repeats itself in time due to the periodic characteristics of orbits.

An important factor to consider, when propagating orbits, is how long in the future will the state vector be calculated. There are two different ways to propagate orbits: numerical integration and analytical. In the first, small time steps are taken and the state vector is calculated for all of these time steps, integrating them numerically. In the second, the state vector can be calculated far in the future by inserting the time instant desired into the analytical equations.

The accuracy of the prediction will always depend on the complexity of the model, which also increases the computational effort. It needs to exist a trade-off between the accuracy required and the computational effort. Using numerical integration techniques for one object takes more time than an analytical approach, so using it for all objects in a catalog would take a lot of time, considering the computational capabilities in the 50's and 60's. This way analytical methods like SGP4 and Simplified Deep-Space Perturbations (SDP)4 were developed.

3.1.1 Analytical General Perturbations Models

TLE's orbital elements are mean values obtained by removing periodic variations, hence if the intention is to predict the state vector of a space object, these periodic variations must be reconstructed by the prediction model, in exactly the same way they were removed to achieve the best accuracy possible. Hence, if TLEs are used as input to a different propagator than SGP4, even if the propagator models are more accurate, the predictions will be poor [17]. The development of the analytical models like SGP4 passed through several modifications and Sec. 3.1.1.A presents the history of their development.

3.1.1.A History

In 1959, Brouwer [18] developed the first model for the motion of a near-Earth object that included zonal harmonics $J_2 - J_5$. In the same year, Kozai developed a similar model [19]. In 1963, the model of Brouwer was re-examined and a singularity in eccentricity and inclination was removed by Lyddane. [20]

SGP was developed in 1966 and uses a simplification of Kozai's work for the gravitational model and considers the drag effect on mean motion as linear in time, which impose a quadratic variation of the mean anomaly with time. The drag effect on eccentricity considers that perigee height is constant in time. SGP uses a pseudo-drag term that comes in TLEs as the derivative of the mean motion divided by two. Back then, this was the model used in most of the tracking systems in the world. [21]

The work of Brouwer was improved by Lane and Cranford [22], including atmospheric effects with an analytical model rather than an empirical one. However, the size of the catalog had increased and this model became too computationally heavy for the computers of that time. Hence, a simplified version including only secular effects of drag was developed, giving origin to SGP4, in 1970. The derivation of SGP4 was shown in 1979 [23], becoming the model used in tracking systems. SGP4 has an atmospheric

model with a power density function. Besides this, orbital maneuvers, dynamic atmosphere and solar radiation pressure effects are not considered in SGP4, which generates deviations to the output.

SDP4 is an extension of SGP4 for deep-space objects, with period greater than or equal to 225 minutes, that includes the gravitational effects of the Moon and Sun. In 1980, SpaceTrack Report Number 3 [17] described, implemented and tested five orbital propagation models compatible with TLE data: SGP, SGP4, SDP4, SGP8 and SDP8. The last two use the same gravitational and atmospheric models as SGP4 and SDP4, but they're used for special cases of orbital decay and reentry [3]. For this reason they are not going to be studied in this thesis.

3.1.1.B SGP4/SDP4

TLEs are the only publicly available source of orbital data for all tracked space objects. Since SpaceTrack Report Number 3 [17] was published, several versions of SGP4 were used by the operator community with no documentation or coordination, which lead to a state of confusion. As an example, using TLEs generated with the operational SGP4 code will not reproduce the same state vectors as the original code. [3]

In order to clarify and synchronize the community with respect to the implementation of SGP4 and the TLE data released, Hoots published the derivation of the equations used in SGP4/SDP4 implementation [21]. In 2006, Vallado published a revision of the SpaceTrack Report Number 3, where the SGP4/SDP4 implementation code was published for the first time [3]. From this paper resulted a new publicly available SGP4 source code with test cases compatible with the code used by North American Aerospace Defense Command (NORAD) in the generation of TLEs. Issues related with the data format, coordinate system and time system were solved in this paper and the SGP4/SDP4 code was merged and restructured to increase efficiency, accuracy and clearness of implementation. This paper served has a new basis for the use of SGP4/SDP4 code.

In this thesis, it's used the SGP4/SDP4 implementation of a library called Orekit with a wrapper for python programming language. The input of the SGP4/SDP4 propagator is a TLE and a time instant to propagate the orbit to. The output is a state vector of dimension 6 (3 dimensions for position and 3 for velocity) in a Earth-Centered Inertial (ECI) reference frame called TEME, explained in Sec. 3.2. 'Orekit in python' is a space dynamics library that contains several algorithms for flight dynamics applications, designed to be easily used in very different contexts, from quick studies up to critical operations in several companies and agencies such as Airbus Space and Defense or ESA [8]. The SGP4/SDP4 implementation of this library is based on the official source code of Vallado and passes all test cases, however a check on this was made in Sec. 3.3.

[8] "About Orekit," Orekit, accessed on 2020/11/02. [Online]. Available: https://www.orekit.org/

3.2 Description of TEME Reference Frame

As explained in Sec. 3.1.1.B, the output state vector of the SGP4 model is defined in the TEME reference frame, which is an ECI frame, meaning that the frame is not rotating with the Earth but 'fixed' with respect to distant stars. An ECI frame is not really inertial because it's still moving with the Earth orbiting the sun, which is not a concern when dealing with a two-body problem. This introduces errors when a third body (Sun or Moon) is considered and the accelerations induced by these should also be taken into account.

The TEME frame is a cartesian coordinate system, meaning that its axes are orthogonal with respect to each other. The X axis is pointing in the direction of the Vernal Equinox, the Z axis coincides with the Earth rotational axis pointing North and the Y axis is obtained by the cross product between the previous axis, see Fig. 3.1.

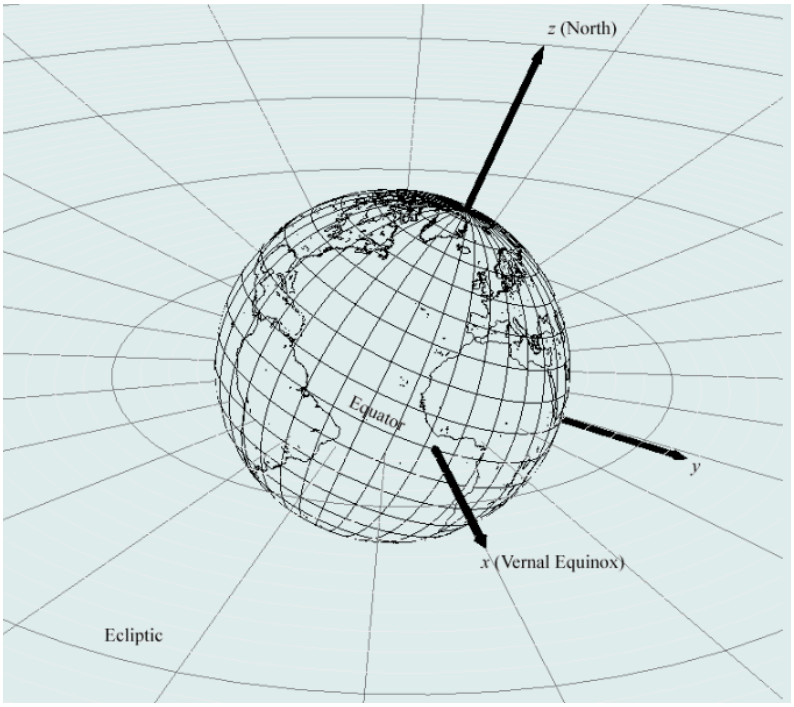


Figure 3.1: TEME reference frame and its axis orientation and alignment. [9]

The Z axis is aligned with the true instantaneous North pole, but due to gravitational pertubations, the Earth's rotational axis changes its direction, thus the Vernal Equinox changes as well. There are two types of motion for the Earth's rotational axis: precession and nutation. These motions are what make the Vernal Equinox to change in time, see Fig. 3.2.

Since the direction of Vernal Equinox is not tied to the Earth's surface, but rather to the Earth's orientation in space, an approximation of its true direction must be made. The approximation made in this case is to account for the precession of the Vernal Equinox but to ignore the nutation [9] , so the Vernal

[9] T. S. Kelso, "Orbital Coordinate Systems, Part I," Celestrak, Dec. 2019, accessed on 2020/08/11.[Online]. Available: https://celestrak.com/columns/v02n01/

Equinox direction is averaged out, and the frame is called 'mean' equinox as opposed to 'true' equinox[10]. The choice of a TEME frame is logical because the observations of SSN are done with sensors fixed to the Earth's surface referenced to the true equator, while the position of the Earth in inertial space relative to the vernal equinox must be averaged.

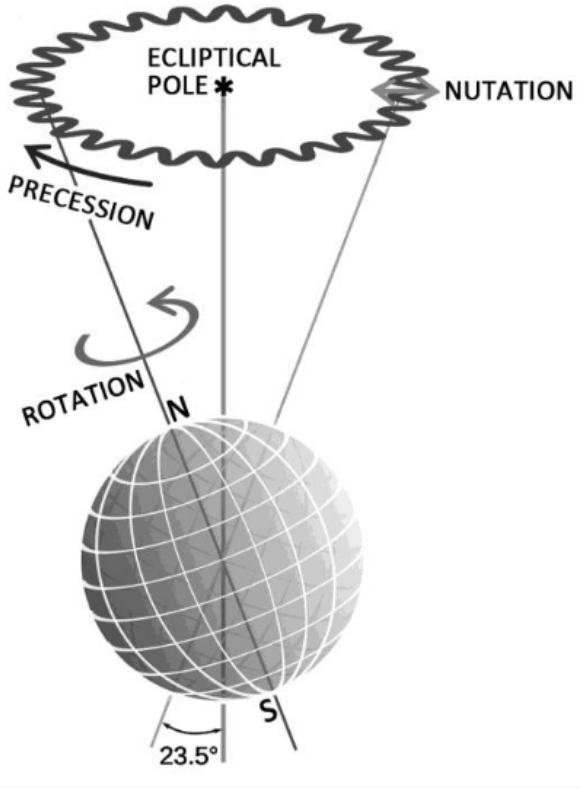


Figure 3.2: Precession and nutation motion of the rotational Earth axis.[11]

Another problem in the definition of the TEME frame is the following: there is a TEME frame for the date desired to predict the state vector (TEME of date) and a different TEME frame for the epoch of the TLE (TEME of epoch). If the later is used, rotation matrices should be used to transform the state vector for the TEME frame of the prediction date. If the first is used, the TEME frame epoch is always the same at the epoch of the state vector, so the research community believes that the "TEME of date" approach is correct and it's the one used by 18th Space Control Squadron (18SPCS), the military unity that provides data to SpaceTrack. [3]

It's not clear in Orekit documentation if the SGP4/SDP4 library uses the 'TEME of date' approach, so this will be verified in Sec. 3.3.

[10] T. S. Kelso, "FAQs: Two-Line Element Set Format," Celestrak, Dec. 2019, accessed on 2020/08/10. [Online]. Available: http://celestrak.com/columns/v04n03/

[11] T. S. Kelso, "Orbital Coordinate Systems, Part I," Celestrak, Dec. 2019, accessed on 2020/08/11.[Online]. Available: https://celestrak.com/columns/v02n01/

3.3 Validation of SGP4 Implementation in Orekit

In order to verify if the implementation of SGP4 in Orekit is the same as Vallado's code [3], it was made a comparison between the results obtained by Vallado with the TLE of Fig. 3.3.

```
1 00005U 58002B   00179.78495062 +.00000023 +00000-0 +28098-4 0  9994
2 00005 034.2682 348.7242 1859667 331.7664 019.3264 10.82419157413667
```

Figure 3.3: TLE used by Vallado for validation of SGP4. [3]

The TLE of Fig. 3.3 was propagated 3 days in the future and the results obtained by Vallado are shown in Table 3.1.

Table 3.1: State vector obtained by Vallado, for a propagation of 3 days, in a TEME of date reference frame, of the TLE presented in Fig. 3.3.

Propagation Time	x [km]	y [km]	z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
3 days	-9060.473736	4658.709525	813.686732	-2.232833	-4.110453	-3.157345

Using SGP4 implementation of Orekit, the TLE of Fig. 3.3 was propagated for several days in the future. The results of the state vectors are presented in Table 3.2.

Table 3.2: State vector obtained with SGP4 of Orekit library for several propagation times, using the TLE of Fig. 3.3.

Propagation Time	x [km]	y [km]	z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
0.5 days	-7134.593401	6531.686413	3260.271865	-4.113793	-2.911922	-2.557328
1 day	-938.559239	-6268.187488	-4294.029248	7.536105	-0.427128	0.989878
3 days	-9060.473736	4658.709525	813.686732	-2.232833	-4.110453	-3.157345
10 days	-2275.835983	8449.246701	3243.894655	-5.777536	0.516852	-2.370718
20 days	-8607.021890	814.008041	-5428.048891	0.311499	-5.564348	-0.930918

Comparing the results obtained in Table 3.2 for 3 days in the future and the results obtained by Vallado, shown in Table 3.1, it's possible to verify that the state vector obtained is the same in every component, at the millimeter level (10^{-6} km). Hence the implementation of SGP4 is the same as Vallado's and the reference frame used in the implementation of Orekit is the TEME of date approach, since the state vector in Table 3.1 is in a TEME of date reference frame.

It was also verified that the SGP4 Orekit implementation applies the SDP4 theory if the satellite has a period greater than or equal to 225 minutes.

4

Selection of Dangerous Objects

4.1 Problem Statement

The objective is to find all conjunctions of the satellite of value for the operator, from now on referred as primary object, against all the other objects in the TLE catalog, from now on referred as secondary objects. A conjunction is defined as an event where both object's centers of mass are within a specified distance from one another. When propagating all objects for different time instances and calculating Miss Distance (MD) and Probability of Collision (PC) for each conjunction, the program becomes computationally heavy. Thus, to turn the task of conjunction detection into a more manageable one, pre-filters and sieves are used to filter out secondary objects that cannot possibly collide with the primary, based on simple flight dynamics principles.

The difference between pre-filters and sieves is that the first only has to be applied per object pair once, while the later has to be applied per object pair in each time instance.

4.2 Overview of Pre-Filters and Sieve Methods

The development of pre-filters started with three methods [4]: perigee/apogee filter, orbit path filter and time filter. These are analytical and geometrical methods used to discard objects which will never be within a defined distance of each other. Initially, it is considered that the objects follow Keplerian orbits with no perturbations, which is a fair assumption if the threshold distance is set large enough, as it was done in Sec. 4.3.1.C. Some improvements, modifications and analysis were made in literature upon these pre-filters, and some other approaches appeared, namely the sieves methods. [5]

4.2.1 Perigee/Apogee Filter

The most basic and used pre-filter is the perigee/apogee filter, also known by altitude filter, that discards objects that do not cross each other's altitude and so can never collide. Being q the larger perigee of the objects pair and Q the smaller of the two apogees, if

$$q - Q > D_{th} \ , \tag{4.1}$$

then the object is ruled out and the objects that passed the pre-filter move on to the next, see Fig. 4.1.

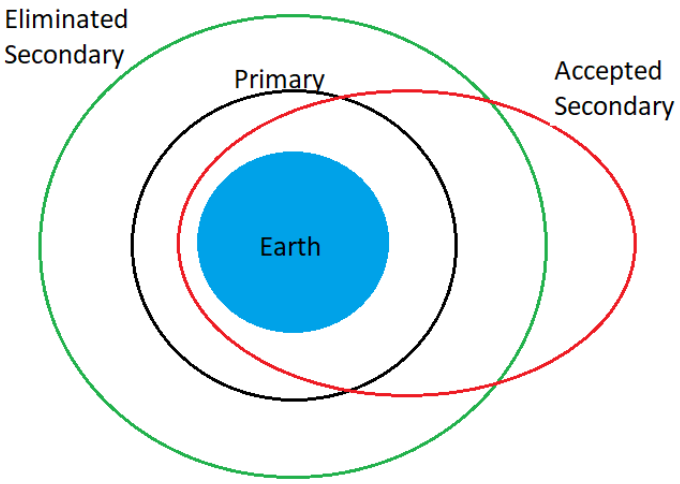


Figure 4.1: Geometry of the perigee/apogee filter. The black orbit is the orbit of the primary object. The green orbit is an example of a secondary object eliminated by the filter. The red orbit is an example of a secondary object that passes the filter.

When using this filter, it's not necessary to propagate any object. The decision to discard, or not, an object is only based on its orbit characteristics (perigee and apogee), which can be easily taken from TLEs. This filter can be used in two different ways: by taking into account perturbations or not. For the later, a large value of D_{th} has to be used. A bad application of this filter may cause false negatives,

meaning that objects are discarded by the filter but may cause a conjunction later. For this filter to work well, a large value for D_{th} needs to be used, hence this problem will be addressed in Sec. 4.3.1.

4.2.2 Orbit Path Filter

The orbit path filter, also know by geometrical pre-filter, eliminates objects whose orbits never get within a defined distance, independently of where they are in their orbits [4]. The method computes the two closest points of the orbits paths, d_1 and d_2, and compares them to a distance threshold to check if the pair is eliminated or not, see Fig. 4.2. A disadvantage of this filter is that it doesn't work with near-coplanar objects, where the relative angle between orbital planes is close to zero, becoming an indeterminate problem.

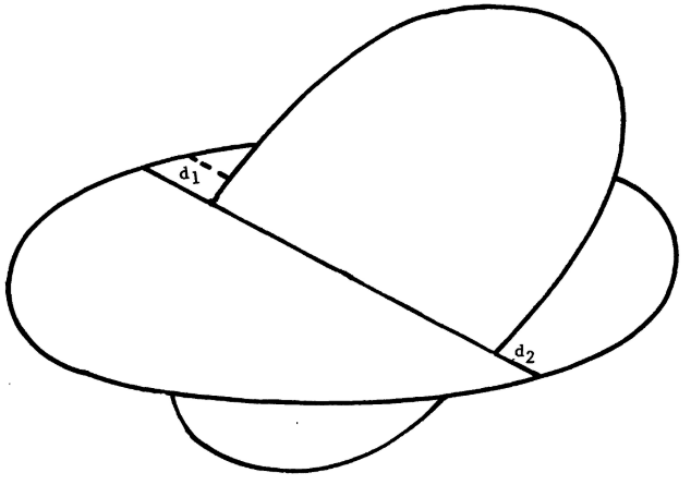


Figure 4.2: Geometry of the orbit path filter. The d_1 and d_2 parameters represent the closest points between the orbit paths of the objects. [4]

It was recommended to not use this filter due to the generation of many false negatives and little computational time savings. The computational benefits in a complex version of the filter does not justify the effort to improve its robustness. [24]

4.2.3 Time Filter

Even if the orbit paths lie within a distance threshold of one another, it doesn't mean that the objects are in the region of close approach simultaneously, so a time filter can be applied to eliminate pairs based on the time they pass the region of close approach. An angular window is defined and then it's converted to time and checked for overlaps. If it overlaps, the point of closest approach is computed and if it's smaller than a defined threshold, the object passes the filter, if not, it's eliminated.

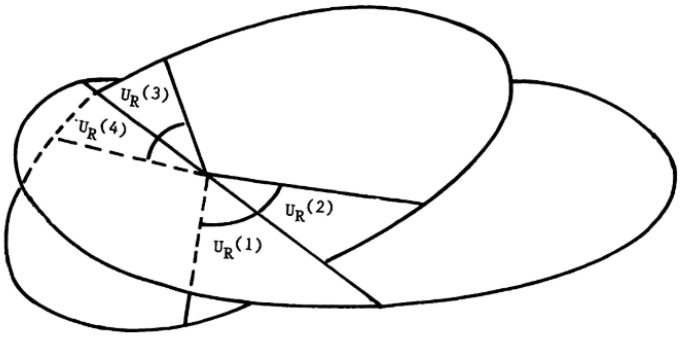


Figure 4.3: Definition of the angular windows for the geometry of the time filter. [4]

An advantage of this filter is that it finds by itself the time intervals when the conjunction happens [24]. The disadvantages of this filter are similar to the orbit path filter, it assumes that the orbits are non-coplanar and requires information about the variations of the nodal period.

4.2.4 Smart Sieve

This method was used in a software tool, called Collision Risk Assessment Software (CRASS), developed by Grupo Mecánica Vuelo (GMV) for ESA. Later, it was also implemented in GMV's software *Closeap* and contains a group of sieves for conjunction detection. This group is called Smart Sieve [5].

Comparing it to pre-filters, the smart sieves method includes acceleration effects when determining the conjunction volume and it works whether the orbits are coplanar or non-coplanar. In this method, the state vectors of the objects are computed for several time instances equally spaced in time, by a defined time step, during a certain time interval. If the distance between a pair of objects is higher than a threshold distance, R_{th}, the pair is eliminated from further consideration. Another distance should be defined, a critical distance, R_{cr}, the distance that defines a conjunction. If an object is inside this critical distance, then a conjunction is happening.

To define R_{th}, it's used the fact that objects orbiting Earth can't exceed the escape velocity, v_{esc}, of $11.186\,\mathrm{km/s}$. In spite of existing objects with higher velocities (launches or meteoroids), these are not in the TLE catalog, so they are not taken into account. Thus, the relative velocity between two orbiting objects can never exceed twice the escape velocity, $22.3726\,\mathrm{km/s}$. Hence, during a time step duration, Δt, the secondary object cannot enter the critical volume if it's at a larger distance than R_{th} and this object can be discarded. This way, R_{th} can be defined as

$$R_{th} = R_{cr} + \frac{d_{esc}}{2} = R_{cr} + v_{esc}\Delta t \ , \tag{4.2}$$

where d_{esc} is the distance travelled during Δt with maximum relative velocity of $2v_{esc}$, as described in Fig. 4.4.

Note that the threshold volume varies with the time step chosen, which is something not used in pre-filters where the threshold is fixed. Equation (4.2) doesn't account for effects of gravity and assumes

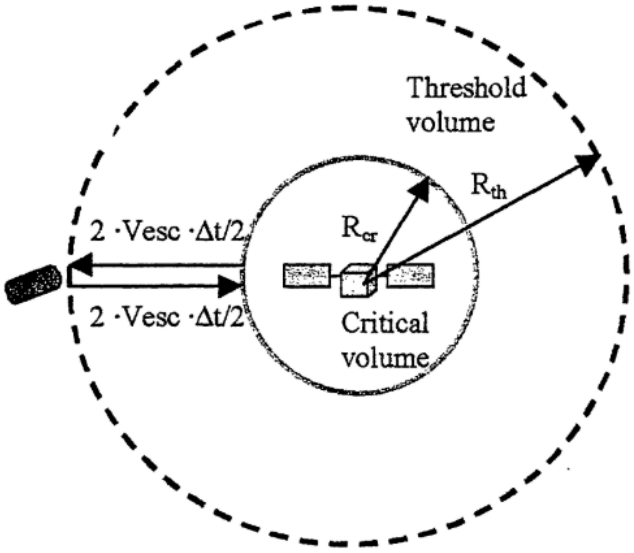


Figure 4.4: R_{th} and R_{cr} definition. The velocity escape of the Earth is defined by v_{esc} and the time step chosen in the propagation of the orbit is Δt. [5]

linear relative motion, so in order to account for acceleration effects and neglecting any maneuvers that may occur, the maximum acceleration between a pair of objects can never exceed twice the gravity acceleration at sea level, g_0. Thus, the maximum deviation from a straight constant velocity path during the time step Δt is

$$d_{acc} = \frac{1}{2} 2 g_0 \Delta t^2 \ , \tag{4.3}$$

hence, a maximum distance R_{acc} can be defined, ensuring that the secondary object will be detected, accounting for acceleration effects,

$$R_{acc} = R_{cr} + d_{acc} = R_{cr} + g_0 \Delta t^2 \ . \tag{4.4}$$

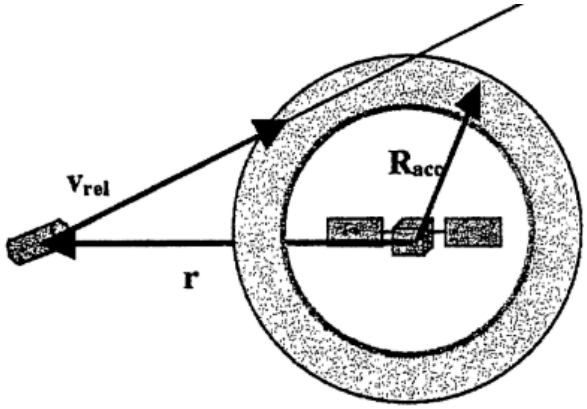


Figure 4.5: Definition of R_{acc}. The relative velocity vector is defined by v_{rel} and the relative position vector by r. [5]

Taking these definitions into account, the overall smart sieve process has the following steps:

XYZ Sieves Uses R_{th} defined in Eq. (4.2) and discards objects that have one component of their relative position vector, **r**, larger than R_{th}, because this implies that the relative position vector is larger than R_{th},

$$r_i > R_{th} \Rightarrow |\mathbf{r}| > R_{th} \quad \text{for } i = x, y, z \; . \tag{4.5}$$

Step Skipping If the relative position vector, **r**, is much larger than R_{th}, then the secondary object will not enter in the threshold volume at a number of following steps and these can be skipped. A safe amount of steps to be skipped is

$$N = int\left(\frac{|\mathbf{r}| - R_{th}}{2 v_{esc} \Delta t} \right) . \tag{4.6}$$

The distance that will take for the secondary to reach the threshold volume, $|\mathbf{r}| - R_{th}$, is divided by the maximum relative velocity, $2v_{esc}$, times one time step and then rounding the result for the integer below gives us the number of time steps that can be skipped, ensuring that the secondary doesn't reach the threshold volume.

r^2 **Sieve** Some objects may pass the XYZ sieve and still have a distance larger than R_{th}. For those, the r^2 sieve is applied,

$$|\mathbf{r}|^2 = r_x^2 + r_y^2 + r_z^2 > R_{th}^2 \; . \tag{4.7}$$

The square of the distance is used instead of using the norm of the vector in order to avoid calling the square root operator, reducing computational time.

Minimum Distance Sieve Rejects an object from further analysis in the current time step if the square of the minimum distance between two objects is greater than R_{acc}^2,

$$|r_{min}|^2 > R_{acc}^2, \tag{4.8}$$

where $|r_{min}|^2$ is computed by Eq. (4.9), which is equivalent to Eq. (4.11) [5].

$$|r_{min}|^2 = |\vec{r}|^2 - \left| \vec{r} \cdot \frac{\vec{v_{rel}}}{|\vec{v_{rel}}|} \right|^2 \tag{4.9}$$

r^2 **Fine Sieve** Similar to r^2 sieve but uses the approach velocity, v_{app} , instead of two times the escape velocity, v_{esc}, to define $R_{th_{fine}}$. This fine definition of the threshold distance also takes into account R_{acc},

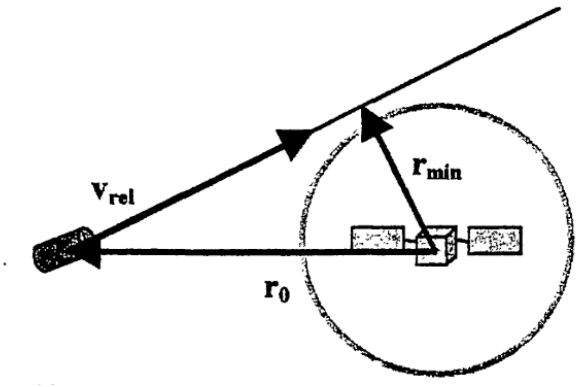


Figure 4.6: Definition of the minimum distance, r_{min}. The relative velocity vector is defined by v_{rel} and the relative position vector by r_0. [5]

$$R_{th_{fine}} = R_{acc} + \frac{1}{2} v_{app} \Delta t = R_{acc} + \frac{1}{2} \left| \vec{v_{rel}} \cdot \frac{\vec{r}}{|\vec{r}|} \right| \Delta t \ . \tag{4.10}$$

So if $r^2 > R^2_{th_{fine}}$ the object pair is rejected.

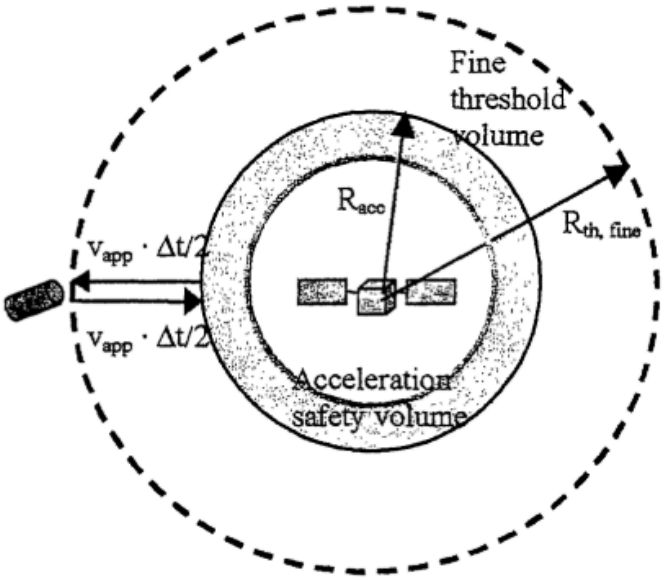


Figure 4.7: Geometry of the fine r^2 sieve and $R_{th_{fine}}$ definition. R_{acc} defines the acceleration safety volume, v_{app} defines the approach velocity between the objects and Δt the time step chosen for orbit propagation. [5]

Fine Conjunction Detection The point of closest approach is calculated numerically by finding null values of v^2_{app} for the pairs that passed all previous sieves.

4.3 Chosen Methodology

The methodology chosen follows the recommendations to not use the orbit path and time filter because these may lead to false negatives, discarding objects that will cause a conjunction later [24]. Modifications on the time filter could be done in order to improve its robustness but there is no detailed

descriptions about how to do it. Since the objective of this thesis is not to improve previous filters or develop new ones, the chosen methodology will be based on a group of filters that are considered robust by the research community but not extensively complex, because only analysing and implementing the improvements in all filters would require an extensive analysis and testing, beyond the scope of this work.

Taking into consideration the description of all filters and approaches in Sec. 4.2, it was decided to use the smart sieve techniques and the perigee/apogee filter. Comparing the first with the orbit path and time filter, smart sieve techniques don't make assumptions about the type of orbit (coplanar or non-coplanar), making it a more robust technique that works for every type of orbit. Moreover, the smart sieve technique saves computational time and it's easier to implement than the orbit path and time filters. The general conjunction detection method can be seen in Fig. 4.8, with the following processes:

- **Catalog Download and Reading** - The catalog with the more recent TLEs is downloaded from SpaceTrack and a dictionary of TLEs is created.
- **Out-of-date TLE Filter** - In first place, the objects, whose TLEs epoch is too old, are not further considered, because applying SGP4 to these objects would lead to large errors in the propagation process, especially for LEO objects, as explained in Sec. 5.4. Hence, all objects whose TLE epoch is older than 20 days are eliminated. The reasons why these are not often tracked are because some were lost, some are decaying and others are not considered important to track that often. This issue can only be solved if NORAD, or other entities, track these objects more often and improve the catalogs.
- **Perigee/Apogee Filter** - In order to account for variations of the perigee and apogee during propagation due to perturbations, it's necessary to estimate the maximum variation of the perigee and apogee to define a good value for D_{th}. This value needs to be tuned because if too large, the filter will not reject many objects that could be rejected, if it's too small the filter may reject some objects that can cause a conjunction in the future. The tuning of this value will be done in Sec. 4.3.1.
- **Propagation** - All objects that passed previous filters are propagated, generating ephemerides for each object in the time interval analysed. The code is configurable to choose the limits of the time interval, but the initial time instant was defined as the download date of the more recent catalog and then the user can define the duration of the propagation, i.e., how long in the future the user wants to propagate.
- **Smart Sieve Techniques** - The ones implemented were the step skipping method, the XYZ sieves, the r^2 sieve and the r^2 fine sieve. The minimum distance sieve was not used because it was found that it would only be valid for small time steps, in the order of 10 seconds. It is intended to have a large time step, in the order of 180 seconds, to reduce the time it takes to propagate. If larger

time steps were used with this filter, the approximation for calculating the minimum distance is not valid, leading to wrong values of the minimum distance and consequently to false positives or false negatives. The fine conjunction detection was also not used because it's not described in detail [5] and it's not very conclusive about its implementation, so another method called 'refined propagation and TCA' was developed.

- **Generation of Conjunction Data** - After all sieves are passed, for one conjunction event there is more than one time instance where the secondary object is at a relative distance smaller than the defined thresholds and only the data where the conjunction reaches the minimum distance is needed. Moreover, the secondary object may have more than one conjunction with the primary during the time interval analysed, because in the next revolution the objects may have a conjunction again. These factors need to be taken into account when implementing the code. After this, only state vectors with minimum distance between them are selected. Note that this is not the true minimum distance due to discretization of time performed when propagating. It will be a distance in a time instance near to the one found.

- **Refined Propagation and TCA** - In order to find the time where the real minimum distance (TCA) occurs , it is used a Taylor series approximation to extrapolate the motion of the objects, see Eqs. (4.11) and (4.12) [25],

$$\mathbf{r} = \mathbf{r_0} + (t - t_0)\dot{\mathbf{r_0}} + \frac{1}{2}(t - t_0)^2\ddot{\mathbf{r_0}} + ... \,, \tag{4.11}$$

$$TCA = t_0 - \frac{\mathbf{r_0} \cdot \dot{\mathbf{r_0}}}{\dot{\mathbf{r_0}} \cdot \dot{\mathbf{r_0}}} \,, \tag{4.12}$$

where **r** is the relative position vector and the variables with the subscript 0 are the ones found during the propagation that have a minimum relative distance. To obtain the relative state vector at TCA or, by other words, the MD, it's necessary to replace t by TCA, in Eq. (4.11). This is only valid for small time steps, so to turn around this problem, a refined propagation of both objects around the TCA found is performed, with a time step of 1 second. This process is done in a loop always with smaller time steps until the TCA calculated differs less than 0.001 seconds from the previous TCA. With this process the calculation of the TCA will always be valid.

This method is similar to the minimum distance sieve, which also uses a Taylor series approximation, but requires to use small time steps. The refined propagation method allows to use the Taylor approximation with a larger time step initially set up, reducing the overall time of propagating objects. If MD $> R_{cr}$, then the object is discarded. If MD $\leq R_{cr}$ then a conjunction was found. For the objects that passed this last filter, all the conjunction data such as: MD, relative velocity and state vectors of both objects are calculated at TCA and stored in a file.

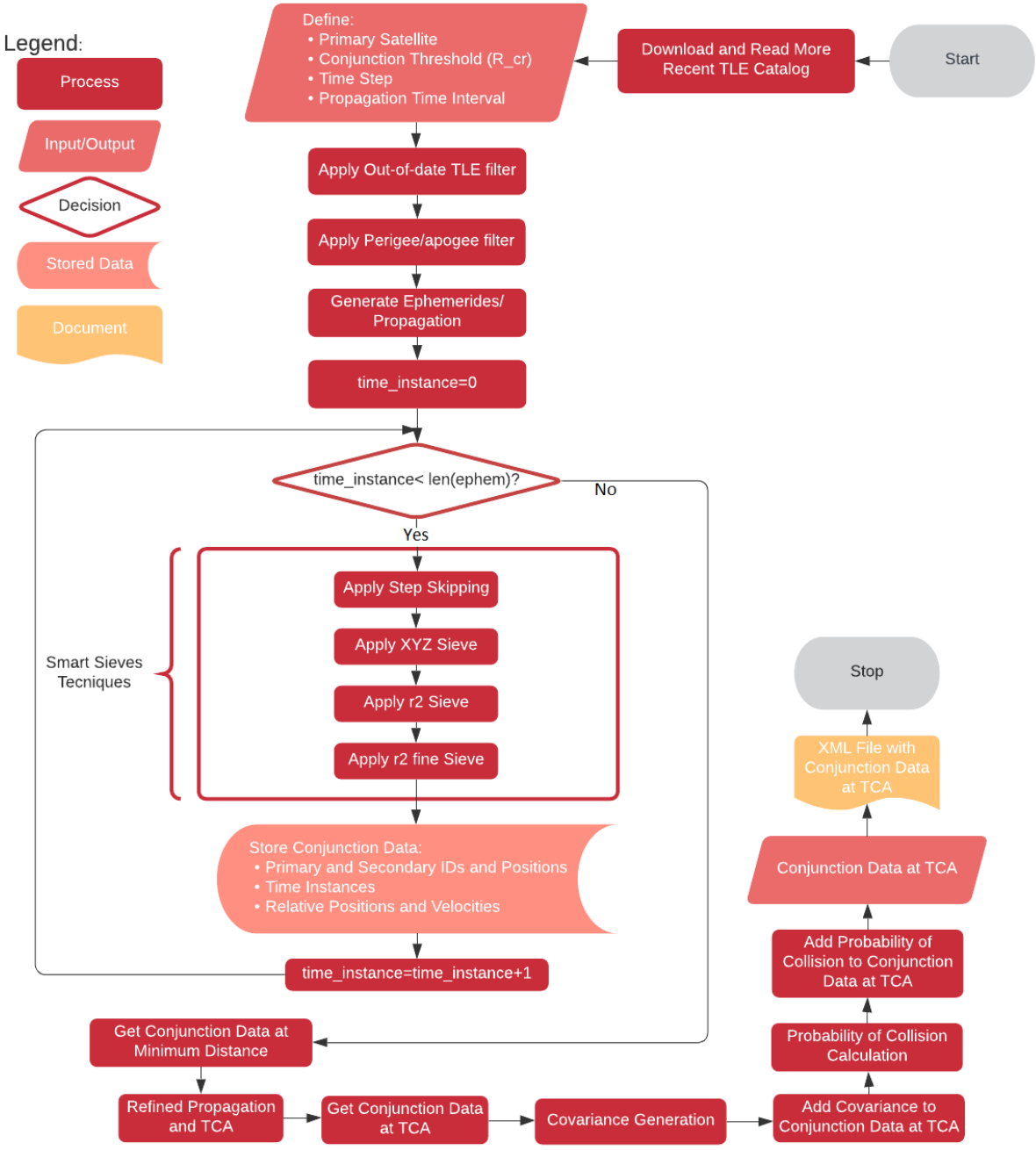


Figure 4.8: Flowchart of the general developed code for the orbital collision warning tool. The code starts with the download and reading of the most recent TLE catalog from SpaceTrack. This catalog contains all tracked objects. After this, the user configures the code by choosing the primary satellite, the conjunction threshold R_{cr}, the time step for the propagation of TLEs and the propagation time interval. The beginning of the time interval is set as the download timestamp of the TLE catalog. The Out-of-date TLE filter and the perigee/apogee filter are applied. The objects that pass these filters are propagated, generating ephemerides (state vectors at each time instance). The code runs through the ephemerides applying the smart sieves techniques and storing all state vectors and time instances that passed all sieves. Several state vectors represent only one conjunction, but we are only interested on the state vectors whose distance between objects is minimum, for each conjunction, so the 'Get Conjunction Data at Minimum Distance' is applied. After this, the exact TCA and MD is calculated by performing a refined propagation and all the conjunction data (Objects' IDs, positions and velocities, relative position, MD and relative velocity) is obtained at TCA. In the end, there is a covariance generation process that allows to calculate the PC. These parameters are added to the conjunction data at TCA and a file is written with it.

4.3.1 Tuning the Threshold Distance of the Perigee/Apogee Filter

The perigee/apogee filter has the advantage of not being necessary to propagate the objects in order to discard them from further analysis. These can be discarded simply by parameters that come directly from TLEs, the eccentricity and the semi-major axis (obtained by the mean motion) and with these compute the perigee and apogee by $r_p = a(1-e)$ and $r_a = a(1+e)$. It is important to notice that the perigee and apogee obtained directly from the TLE will be different than the one using SGP4 because, in the first way, mean elements are used and the second is the actual value at a certain time instance.

The desire is to discard as much objects as possible, in order to reduce the computational time propagating the objects that pass this filter. To achieve this, it's necessary to tune the threshold distance D_{th}, presented in Sec. 4.2.1. If D_{th} is too large, then almost certainly that any discarded object will cause a conjunction but objects that don't need further analysis will be propagated, increasing computational processing time. If D_{th} is too small, then some objects that will cause a conjunction in the future can be discarded, which is not desired.

A good way to choose a value for D_{th} is to estimate the variation of the perigee and apogee during a certain time interval *a priori*, meaning that this estimation is not included in the process of conjunction detection and analysis, it will only be used to define the value of D_{th}.

All objects in the catalog with a TLE epoch not older than 20 days from the catalog epoch are propagated during a certain time, obtaining its eccentricity and semi major axis, which allows to compute the perigee and apogee of each object's orbit at a certain time instant. The maximum value of the variation of the perigee and apogee $\Delta d_{p,a}$ is given by Eqs. (4.13) and (4.14),

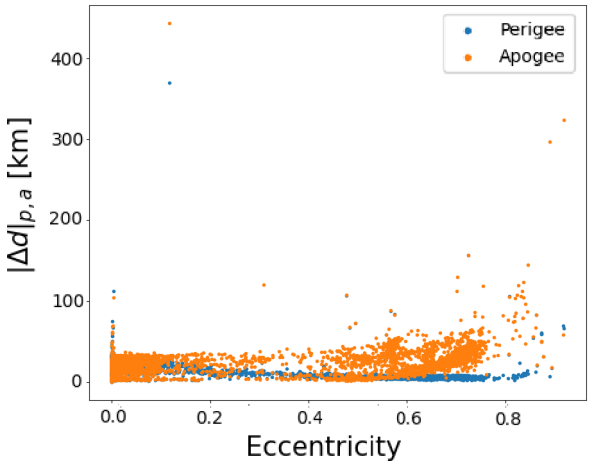
$$|\Delta d_p| = |\max(r_{p_{max}} - r_{p_{min}})| \ , \tag{4.13}$$

$$|\Delta d_a| = |\max(r_{a_{max}} - r_{a_{min}})| \ . \tag{4.14}$$

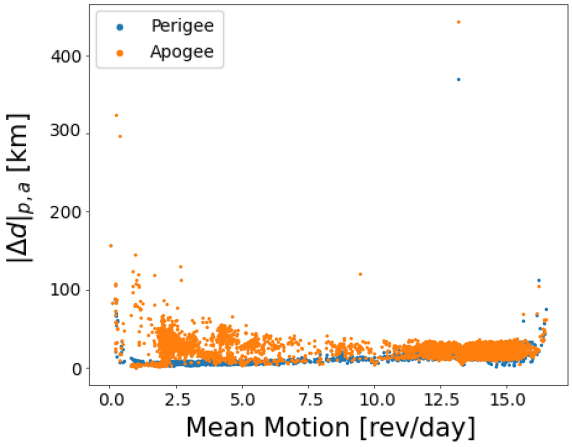
To compute $|\Delta d_{p,a}|$, three different catalogs were used, one from $2020/03/24$ and two closer in time from $2020/08/20$ and $2020/08/21$. For each catalog, all objects were propagated during 1 day, 3 days, 6 days and 10 days. The maximum variation of the perigee and apogee for each object was plotted against variables present in TLEs. The results obtained for $|\Delta d_{p,a}|$ for each object of the catalog of $2020/03/24$ are presented in Figs. 4.9 and 4.10. The results obtained for the other catalogs are presented in Appendix B. The main objectives of this section are:

- To find a dependency of the variation of perigee and apogee with variables present in TLEs.
- To quantify the variation of perigee and apogee with how long in the future the TLE is propagated.
- To choose a good value for D_{th}.

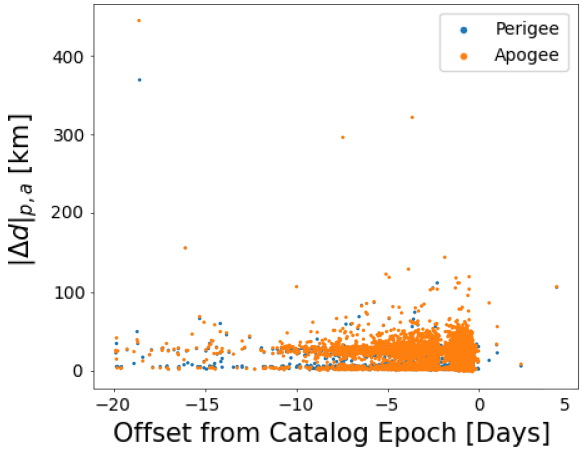
4.3.1.A Variation of Perigee/Apogee Dependency with TLE Variables



(a) Maximum variation of perigee and apogee plotted against eccentricity.



(b) Maximum variation of perigee and apogee plotted against mean motion.



(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

Figure 4.9: Maximum variation of the perigee and apogee for all space objects in the catalog of $2020/03/24$ and a propagation time of 1 day. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.

Figure 4.9(a) shows a high concentration of objects with low values of eccentricity. Figure 4.9(b) demonstrates a high concentration of objects with high values of mean motion, higher than $10\,\mathrm{revs/day}$, which makes sense, since the LEO region is the most populated area. Figure 4.9(c) reveals that most TLEs are 10 days old or less. Hence, most objects are in LEO with near-circular orbits and with TLEs not older than 10 days.

In Fig. 4.9(a), the maximum variation of the apogee starts to increase for $e > 0.5$, which can be

explained by the fact that, in highly eccentric orbit's perigees, objects can be slowed down by drag, which will vary the apogee radius. Figure 4.9(a) also shows a peak for circular orbits ($e = 0$), which may be explained by the fact that these orbits are in LEO regions, being more affected by drag and leading to larger variations. If a peak exists in $e = 0$, then a peak also exists in the mean motion graph for values larger than $16\,\mathrm{revs/day}$, meaning that these are LEO objects.

The peak in low values of mean motion can be explained by the fact that objects with large eccentricity often have small values of mean motion and so the peak of high eccentricities also appears in low values of mean motion.

For Fig. 4.9(c), it was expected that, for older TLEs, the maximum variation of the perigee and apogee was larger, but it seems that the larger maximum variation of perigee and apogee happens for objects with TLEs whose epoch is closer to the catalog epoch, what can be explained by the large number of objects with recent TLEs.

This analysis was performed to check which one of the variables present in TLEs (eccentricity, mean motion, TLE epoch) shows a relation with the variation of perigee and apogee, in order to help defining D_{th}. Eccentricity and mean motion are the variables showing this relation, however, eccentricity was chosen as the variable to help define D_{th} because the peak of the circular LEO orbits ($e = 0$) is better defined in eccentricity plots than in mean motion plots.

4.3.1.B Variation of Perigee/Apogee With Propagation Time Interval

The same plots as before were created but now for different propagation time intervals, 3, 6 and 10 days, see Fig. 4.10. The plots for 3 and 6 days are presented in Appendix B.

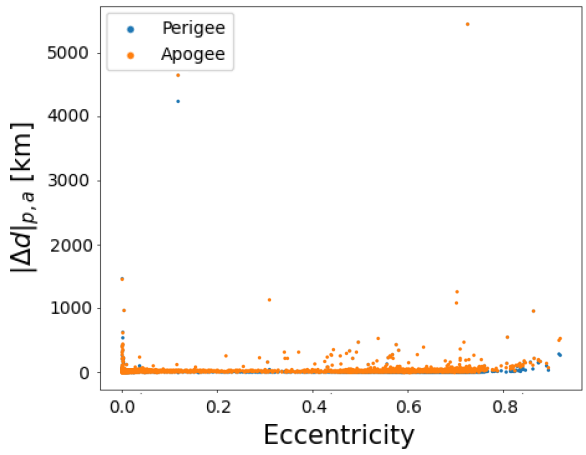
When increasing the propagation time interval, the results become more random, being difficult to find a dependency with the variables studied. The outliers that already existed in propagation time interval of 1 day are even more amplified. Maneuvers are not detected by the SGP4 theory, so there is no apparent reason why an object varies its perigee and apogee more than $1000\,\mathrm{km}$ only by the perturbation taken into account by the SGP4 theory, indicating that these are errors in the observations used to create TLEs, so these objects will be ignored for the definition of the D_{th}.

Even though these objects are ignored for the choice of the value of D_{th}, it doesn't mean that these will be automatically discarded by the filter or that they will generate false negatives, because we are looking here for the variation of the perigee/apogee and the object will only be discarded if the condition in Eq. (4.1) verifies.

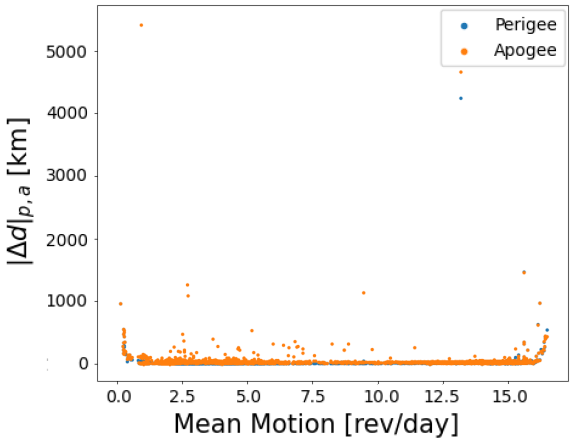
Another reason that allows to ignore these objects is the fact that the objects that generate this large variation are not always the same for each catalog, due to errors in observations used to create TLEs. For example, object 44 288 in the catalog of $2020/08/20$ generates a maximum variation of perigee and apogee in the order of $1000\,\mathrm{km}$, for a propagation time interval of 10 days, see Fig. B.6, and in the catalog

of $2020/08/21$ generates a maximum variation of perigee and apogee in the order of $40\,000\,000\,\mathrm{km}$ (this object was removed from the graphs of Fig. B.10 for the sake of visualization).

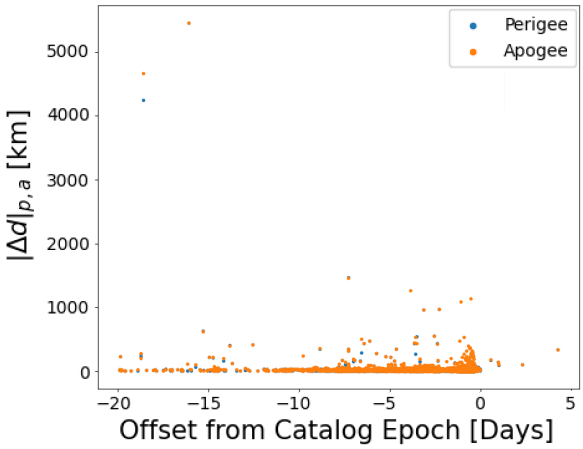
In the worst case scenario, an object is discarded by the filter and generates a conjunction in the future that it is not detected, this may happen for a catalog but not happen for the catalog of the next day, allowing to discard them for the definition of D_{th}.



(a) Maximum variation of perigee and apogee plotted against eccentricity.



(b) Maximum variation of perigee and apogee plotted against mean motion.



(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

Figure 4.10: Maximum variation of the perigee and apogee for all space objects in the catalog of $2020/03/24$ and a propagation time of 10 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.

4.3.1.C Definition of D_{th} and Filter Efficiency

The definition of D_{th} is not just one value, but several values, depending on the eccentricity of the object and the propagation time interval chosen, see Table 4.1.

In order to study the perigee/apogee filter efficiency using these definitions of D_{th}, the relation between rejected objects and the ones analysed by the filter (r/a) was calculated in terms of percentage, see Table 4.1. This parameter will be called filter efficiency from now on.

Table 4.1: Definitions of D_{th} for each propagation time interval and each eccentricity interval. The values of D_{th} were obtained by choosing a value above all points in the plots of the maximum variation of apogee and perigee, ignoring outliers. The filter efficiency (r/a) is presented for each class of eccentricities and for each propagation time interval class for the catalog of $2020/08/20$, considering the primary satellite as the International Space Station (ISS).

Propagation Time	Eccentricity	D_{th} [km]	r/a [%]	r/a_{total} [%]
≤ 1 day	$0 \leq e < 0.2$	150	71.82	76.87
	$0.2 \leq e \leq 1$	200	5.05	
> 1day, ≤ 3 days	$0 \leq e < 0.2$	400	35.45	39.09
	$0.2 \leq e \leq 1$	500	3.64	
> 3 days, ≤ 6 days	$0 \leq e < 0.03$	900	17.44	22.42
	$0.03 \leq e < 0.2$	500	2.11	
	$0.2 \leq e \leq 1$	1000	2.86	
> 6 days,≤ 10 days	$0 \leq e < 0.03$	1500	10.22	13.74
	$0.03 \leq e < 0.2$	900	0.98	
	$0.2 \leq e \leq 1$	1500	2.54	

Table 4.1 shows that more objects are discarded by the filter for propagation time intervals smaller than 1 day, because D_{th} values are small. The filter efficiency decreases when the propagation time interval is increased because the values of D_{th} are larger, resulting in less objects being discarded.

It's important to notice that the eccentricity class that contains the values of e closer to zero is the one that has a larger filter efficiency because most objects have near-circular orbits. For propagation time intervals that have three classes of eccentricity, the last one rejects more objects than the second one, even with a larger D_{th}. This occurs because the last class contains much more objects.

The filter efficiency varies with the object chosen as the primary object. This will be shown by calculating the filter efficiency for several objects at different altitude regimes. The chosen objects are:

- 42 794, D-Sat launched by D-Orbit in 2017.
- 46 274, In-Orbit Now Mark 01 Satellite (ION-MK01) launched by D-Orbit in 2020.
- 20 510, which is an object that is going to be used further to validate the pre-filters implementation.
- 37 214, 7530, 24 876, 20 776.

The results of the filter efficiency for each primary object and propagation time interval are presented in Table 4.2. The filter efficiency decreases with the propagation time interval for every type of primary

Table 4.2: Perigee/Apogee filter efficiency for primary objects in different altitude regimes and different propagation time intervals. The catalog of $2020/09/07$ was used.

Primary Object	Altitude Regime [km]	Propagation Time	r/a_{total} [%]
ISS	420-430	≤ 1 day	78.75
		> 1day, ≤ 3 days	43.44
		> 3 days, ≤ 6 days	22.63
		> 6 days, ≤ 10 days	13.71
D-Sat	500-700	≤ 1 day	45.67
		> 1day, ≤ 3 days	24.98
		> 3 days, ≤ 6 days	14.46
		> 6 days, ≤ 10 days	13.20
ION-MK01	500-600	≤ 1 day	67.26
		> 1day, ≤ 3 days	31.35
		> 3 days, ≤ 6 days	19.23
		> 6 days, ≤ 10 days	13.52
20510	500-600	≤ 1 day	61.59
		> 1day, ≤ 3 days	27.91
		> 3 days, ≤ 6 days	16.53
		> 6 days, ≤ 10 days	13.42
37214	800-900	≤ 1 day	41.86
		> 1day, ≤ 3 days	23.42
		> 3 days, ≤ 6 days	13.83
		> 6 days, ≤ 10 days	13.08
7530	1400-1500	≤ 1 day	77.35
		> 1day, ≤ 3 days	68.56
		> 3 days, ≤ 6 days	19.80
		> 6 days, ≤ 10 days	12.78
24876	20000-20300	≤ 1 day	99.85
		> 1day, ≤ 3 days	91.88
		> 3 days, ≤ 6 days	91.16
		> 6 days, ≤ 10 days	90.16
20776	35700-35800	≤ 1 day	91.32
		> 1day, ≤ 3 days	89.26
		> 3 days, ≤ 6 days	87.85
		> 6 days, ≤ 10 days	87.25

object, which was already expected due to the increase of D_{th}. The filter efficiency is much higher for the last two objects, 24 876 and 20 776, which are MEO and GEO satellites respectively, because of the less density of objects in these regions. Hence, the filter is more efficient if the primary is a MEO or GEO object.

The rest of the objects are in LEO region and for those, with a propagation time interval between 6 and 10 days, the filter efficiency is similar for all of them, in the order of $13\,\%$. This happens because the defined D_{th} covers all the LEO region, so most of the objects rejected are the ones out of this region.

For altitude regime between 420 km and 430 km, there is a larger filter efficiency for less than 3 days of propagation time in relation to other LEO primary objects. This may be explained by the fact that, in this altitude regime, the density of objects is not as high as in the altitude regime between 700 km and 900 km, where D-Sat and 37 214 have the lowest filter efficiencies. For the altitude regimes of ION-MK01

and $20\,510$, the filter efficiency is slightly better, having a big drop when the propagation time is larger than 3 days, which is common to every LEO satellite.

In conclusion, the estimation of the maximum variation of the perigee and apogee yield semi-random errors originated in the creation of TLEs. However, it's possible to define conservatives values of D_{th}. The D_{th} defined is valid for the catalogs analysed but it does not guarantee that it's valid for all catalogs in the future, which is why very conservative values were chosen to ensure that the perigee/apogee filter can be executed in a safe way.

4.4 Validation of Filters Implementation

In an early phase of the pre-filters and sieves implementation, the primary satellite chosen was the ISS. For the catalog of $2020/08/17$, using propagation time intervals in the order of seconds, the same secondaries were always detected at every time instance, with the following NORAD Identification (ID)s: $45\,937$, $45\,607$, $45\,595$ and $45\,476$.

Upon checking, their relative velocity with respect to the ISS was almost zero, meaning that these were accompanying it. It was discovered that these were Soyuz and Progress spacecrafts still attached to ISS, used for transport and supply missions. This is an indicator that the pre-filters were well implemented, detecting objects that are constantly near the ISS. Of course this is not enough to validate the filters implementation, so other approaches such as a brute force method and Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES) will be used to compare with the conjunctions found by the filters implemented in this thesis.

4.4.1 Brute Force Method

A brute force method was used to validate the filters implementation and assure that no false negatives or false positives were introduced. In this method, no filters are used and a time step of 1 second is chosen to assure that, during the propagation, there are no time instances with a conjunction not detected. This time step is very small, making this method extremely heavy computationally, which is why it will only be used for validation purposes.

It is also necessary to choose the distance that defines a conjunction, the conjunction distance threshold, R_{cr}. After the propagation is made, it's verified if the secondary object lies within the conjunction distance threshold, $|r|^2 < R_{cr}^2$, if this happens, the close approach is considered a conjunction. Since there are no objects orbiting Earth with velocity larger than the escape velocity of Earth ($11.186\,\mathrm{km/s}$), the relative velocity can never exceed twice this value. Considering that the primary is in rest, the secondary, in a time step of 1 second, can only travel $2v_{esc}\Delta t = 2 \times 11.186 \times 1 = 22.3726\,\mathrm{km}$. It

was used a conjunction threshold distance of $26\,\mathrm{km}$ to be conservative. This value will guarantee that a state vector is calculated during the conjunction and that no conjunction is missed.

This is the same line of thought of the smart sieve technique but, in here, the time step is fixed and the conjunction distance threshold is chosen accordingly. In the smart sieve technique, the threshold varies with the time step, allowing to choose an optimal time step for the conjunction detection. The results of the brute force method are shown in Table 4.3.

Table 4.3: Number of conjunctions obtained for several analysis intervals with the brute force method. The primary object is $20\,510$ and it was used a R_{cr} of $26\,\mathrm{km}$. The catalog epoch is $2020/08/26$. The Catalog Reading executing time is the time it takes to read the catalog and to apply the out-of-date TLE filter. The propagation executing time is the time it takes to propagate all objects in the propagation time interval. The Brute Force executing time is the time it takes to compare the relative distances of every object pair at each time instance. The Conj Data Gen executing time is the time it takes to apply the refined propagation and TCA process for calculating TCA, MD and relative velocity for every conjunction found.

Analysis interval [s]	# Conjunctions	Executing Time [s]				
		Catalog Reading	Propagation	Brute Force	Conj Data Gen	Total
180	1	1	158	42	0.01	202
600	4	1	539	168	0.03	708
1800	9	1	2511	1390	0.2	3902
3600	13	1	7286	4473	0.3	11761
7200	40	1	18614	10004	0.4	28620

The number of conjunctions increases with the propagation time interval and the process that takes more time is the propagation because, for each object, it will be calculated a state vector at each second. For a propagation time interval of 7200 seconds, the execution time is $28\,629$ seconds, almost 8 hours. This shows that, from the operational point of view, a brute force method is useless. Imagine a conjunction happening in 2 hours, if it takes 8 hours to detect it, then the conjunction would only be detected after it happened and the operator could not perform an avoidance maneuver, not to mention other technical constraints like computer memory. This is why the brute force was used only for validation purposes, confirming that the problem of conjunction detection is only practical if filters and sieves are used.

For the same propagation time intervals used in the brute force method and with a time step of 60 seconds, the filters and sieves method was implemented. The results of its performance can be seen in Table 4.4.

The catalog reading time is always the same, approximately, because the catalog used is always the same ($2020/08/26$). In here, it's possible to notice the importance and power of the perigee/apogee filter. It only takes 1 second regardless of the duration of the analysis interval because it is applied directly to the catalog and not to the ephemerides generated by the propagation.

The propagation time is much smaller than the one of brute force because there are less objects to propagate due to the perigee/apogee filter and the time step used is sixty times larger. For a propagation time interval of 2 hours (7200 seconds), the brute force method takes almost 8 hours to run, while the filters and sieves method takes only 1 minute approximately, in the same computational environment.

Table 4.4: Number of conjunctions obtained for several analysis intervals with a time step of 60 seconds, with the filters and sieves method. The primary object is 20 510 and it was used a R_{cr} of 26 km. The catalog used was the one of $26/08/2020$. The Perigee/Apogee filter executing time is the time it takes to apply the Perigee/Apogee filter. The Sieves executing time is the time it takes to apply the sieves for each time instant and the Conj Data Gen executing time is the time it takes to apply the refined propagation and TCA process, calculating TCA, MD and relative velocity for every conjunction found and discarding all conjunctions whose MD is larger than R_{cr}.

Analysis interval [s]	# Conjunctions	Executing Time [s]					
		Catalog Reading	Peri/Apo Filter	Propagation	Sieves	Conj Data Gen	Total
180	1	1	1	2	0.5	0.3	5
600	4	1	1	4	1	1	9
1800	9	1	1	11	3	2	18
3600	13	1	1	22	6	5	35
7200	40	1	1	43	10	9	65
43200	233	1	1	256	66	52	376
86400	463	1	1	519	148	116	786

The sieves executing time increases with the duration of the propagation because more time instances need to be checked, but the time needed is much less than in the brute force method because now there are less objects being compared. The generation of the conjunction data only depends on the number of objects that passed all filters and sieves at each time instance, so it increases with the number of conjunctions detected.

Comparing the conjunction data generation process with the brute force method, this takes more time because more objects reach this part of the code, since the threshold before the conjunction data generation process is $R_{th_{fine}}$ which is higher than the R_{cr} used directly in the brute force method. In the filters and sieves method the R_{cr} is only used after the conjunction data is generated.

The most important result obtained when comparing Table 4.3 and 4.4 is that, for each propagation time interval, the number of conjunctions detected and their conjunction data is the same, this means that no object that will cause a conjunction is discarded by the filters and sieves method and the assumptions made in their development and implementation are valid.

4.4.2 Comparison with SOCRATES

Another way to validate the filters and sieves implementation is to compare the conjunctions obtained with Celestrak tool, SOCRATES [26]. It's a some-vs-all conjunction detection public service that looks for conjunctions in the upcoming week, using TLEs and SGP4 for propagation. The parameters calculated by SOCRATES will be compared to the ones obtained by the developed warning tool. These are: MD, relative velocity and TCA.

SOCRATES uses a conjunction threshold of 5 km hence, for comparison reasons R_{cr} will be set to 5 km as well. Object 46 274 (D-Orbit's ION-MK01) will be analysed as the primary satellite to compare with the conjunction data of SOCRATES, shown in Fig. 4.11. The conjunctions detected by the developed collision warning tool are presented in Table 4.5.

Action	NORAD Catalog Number	Name	Days Since Epoch	Max Probability / Start (UTC)	Dilution Threshold (km) / TCA (UTC)	Min Range (km) / Stop (UTC)	Relative Velocity (km/sec)
Analysis	46274	OBJECT C [+]	1.681	8.833E-08	2.761	4.262	15.030
	21359	DELTA 1 DEB [-]	1.716	2020 Sep 09 04:24:45.011	2020 Sep 09 04:24:45.185	2020 Sep 09 04:24:45.359	
Analysis	46274	OBJECT C [+]	1.838	1.291E-06	0.720	1.119	14.967
	36432	COSMOS 2251 DEB [-]	3.924	2020 Sep 09 08:10:56.273	2020 Sep 09 08:10:56.598	2020 Sep 09 08:10:56.924	
Analysis	31304	AIM [+]	2.540	1.496E-06	0.674	1.032	15.037
	46274	OBJECT C [+]	2.184	2020 Sep 09 16:28:51.086	2020 Sep 09 16:28:51.412	2020 Sep 09 16:28:51.737	
Analysis	46274	OBJECT C [+]	2.348	4.998E-07	0.754	2.761	8.149
	44862	PSLV R/B [-]	3.285	2020 Sep 09 20:24:39.545	2020 Sep 09 20:24:40.057	2020 Sep 09 20:24:40.568	

Figure 4.11: Conjunctions detected for ION-MK01, by SOCRATES, from 2020/09/07 to the upcoming 3 days.

Table 4.5: Conjunctions detected for ION-MK01 by the developed collision warning tool, with conjunction threshold of 5 km and a time step of 60 seconds, for a propagation time interval of 3 days, using the catalog of 2020/09/07 20:35:13 [Universal Time Coordinated (UTC)].

Secondary Object ID	TCA [UTC]	MD [km]	Relative Velocity [km/s]
44252	2020-09-09T02:18:48.029	2.854	13.794
21359	2020-09-09T04:24:45.185	4.262	15.030
36432	2020-09-09T08:10:56.598	1.119	14.967
31304	2020-09-09T16:28:51.414	1.018	15.037
44862	2020-09-09T20:24:40.053	2.673	8.149

Table 4.5 presents one more conjunction than Fig. 4.11. An explication for this might be that SOCRATES produces false negatives with the application of the orbit path and time filter whose usage is not recommended [24]. Another explanation might be the fact that SOCRATES not always use the more recent TLE for each object and the method developed in this thesis does. This was not verified when the screenshot of the SOCRATES data was taken, so there is no way of knowing that because after one day all conjunction data changes and SOCRATES don't keep record of it.

It is also possible to notice that for the objects 31 304 and 44 862, the TCA is not exactly the same as the one of SOCRATES and because of that the MD and relative velocity also differs. This might also be due to the fact that SOCRATES is not using the most recent TLEs.

Another run was performed with the catalog of 2020/09/09 and the conjunctions presented by SOCRATES were collected. It was verified that the TLEs used by SOCRATES were the same as those in the most recent catalog of 2020/09/09 used. The number of conjunctions and conjunction data obtained was exactly the same as SOCRATES, as it can be seen in Fig. 4.12 and Table 4.6, indicating that the extra conjunction detected and the differences in the TCAs, calculated before, are due to the fact that different TLEs were being used by SOCRATES. These comparisons suggest that the implementation of the filters and sieves method is correct.

Action	NORAD Catalog Number	Name	Days Since Epoch	Max Probability	Dilution Threshold (km)	Min Range (km)	Relative Velocity (km/sec)
				Start (UTC)	TCA (UTC)	Stop (UTC)	
Analysis	31304	AIM [+]	0.756	1.666E-06	0.638	0.978	15.037
	46274	OBJECT C [+]	0.533	2020 Sep 09 16:28:51.060	2020 Sep 09 16:28:51.386	2020 Sep 09 16:28:51.712	
Analysis	46274	OBJECT C [+]	0.697	5.348E-07	0.729	2.669	8.149
	44862	PSLV R/B [-]	0.701	2020 Sep 09 20:24:39.502	2020 Sep 09 20:24:40.021	2020 Sep 09 20:24:40.540	
Analysis	46274	OBJECT C [+]	2.231	1.580E-07	2.153	3.059	15.202
	39313	PEGASUS DEB [-]	3.395	2020 Sep 11 09:13:57.715	2020 Sep 11 09:13:57.976	2020 Sep 11 09:13:58.236	
Analysis	46274	OBJECT C [+]	2.723	2.900E-07	0.949	3.782	5.519
	46198	H-2A DEB [-]	2.748	2020 Sep 11 21:03:17.472	2020 Sep 11 21:03:18.064	2020 Sep 11 21:03:18.657	

Figure 4.12: Conjunctions detected for ION-MK01, by SOCRATES, from $2020/09/09$ to the upcoming 3 days.

Table 4.6: Conjunctions detected for ION-MK01 by the developed collision warning tool, with conjunction threshold of 5 km and a time step of 60 seconds, for a propagation time interval of 3 days, using the catalog of $2020/09/09$ 15:59:35 [UTC].

Secondary Object ID	TCA [UTC]	MD [km]	Relative Velocity [km/s]
31304	2020-09-09T16:28:51.387	0.978	15.037
44862	2020-09-09T20:24:40.021	2.669	8.149
39313	2020-09-11T09:13:57.975	3.059	15.202
46198	2020-09-11T21:03:18.064	3.782	5.519

4.5 Sieves Efficiency and Time Step Choice

In order to study the influence of the time step in the sieves efficiency and executing times, the code was executed for several propagation time intervals (1, 3, 6 and 7 days) with different time steps. The sieves will be applied at each time instance and the number of rejected objects will be different in every time instance so, in order to analyse the sieves efficiency, it will be calculated the average of all time instances. The overall smart sieve efficiency and the efficiency of the specific sieves were calculated. The results are shown in Table 4.7 to 4.10. Only the propagation, sieves and conjunction data generation times are shown because they are the ones influenced by the propagation time interval and time step. The computer used to obtain these results has a processor Intel (R) Core (TM) $i5-7200U$ CPU @2.50 Hz, 2.71 Hz.

Table 4.7: Executing time and sieves efficiency with a propagation time interval of 1 day for several time steps. The primary object is ION-MK01 and the catalog epoch is $2020/09/09$.

Time Step [s]	Executing Time [s]				Smart Sieves r/a [%]	XYZ r/a [%]	r^2 r/a [%]	r^2 fine r/a [%]
	Propagation	Sieves	Conj Data Gen	Total				
60	497	121	51	670	99.62	86.38	93.76	39.96
90	303	77	96	476	99.23	83.46	90.37	39.83
100	266	76	119	461	99.08	82.42	89.30	39.99
120	217	60	162	439	98.71	80.32	87.11	39.52
150	190	44	238	472	98.06	77.03	83.85	39.59
180	178	40	419	648	97.25	73.63	80.85	38.59
210	127	38	469	631	96.28	70.12	78.09	37.02
240	123	37	597	757	95.17	66.52	75.38	35.73

Table 4.8: Executing time and sieves efficiency with a propagation time interval of 3 days for several time steps. The primary object is ION-MK01 and the catalog epoch is $2020/09/09$.

Time Step [s]	Executing Time [s]				Smart Sieves r/a [%]	XYZ r/a [%]	r^2 r/a [%]	r^2 fine r/a [%]
	Propagation	Sieves	Conj Data Gen	Total				
120	1978	1246	1298	4524	98.63	79.88	86.60	41.52
150	1373	808	1892	4076	97.89	76.42	83.11	40.47
180	1012	489	2525	4028	96.98	72.79	79.85	39.27
210	817	398	3477	4695	95.90	69.06	76.81	38.01
240	686	250	4516	5455	94.66	65.19	73.94	36.71

Table 4.9: Executing time and sieves efficiency with a propagation time interval of 6 days for several time steps. The primary object is ION-MK01 and the catalog epoch is $2020/09/09$.

Time Step [s]	Executing Time [s]				Smart Sieves r/a [%]	XYZ r/a [%]	r^2 r/a [%]	r^2 fine r/a [%]
	Propagation	Sieves	Conj Data Gen	Total				
120	8642	4459	3809	16915	98.67	79.64	86.92	42.74
150	4284	4214	4322	12822	97.92	76.15	83.34	41.18
180	3597	2576	6520	12695	97.00	72.52	80.01	39.75
210	2741	2501	8314	13559	95.90	68.77	76.91	38.28
240	2224	2425	10643	15294	94.65	64.91	73.98	36.93

It is important to enhance the fact that regardless of the time step chosen, the number of conjunctions found will always be the same because, in the conjunction data generation, with the refine propagation and TCA process, the objects whose MD is larger than the R_{cr} are rejected. The influence of the time step will be in the number of objects that arrive to the conjunction data generation process, increasing its executing time if this number grows.

When the time step is increased, the propagation time decreases because there are less time instances where the state vectors are calculated. For the same reason, the sieves executing time also decreases with the increase of the time step, because there are less time instances to be analysed. On the other hand, the conjunction data generation executing time increases with the time step because the larger the time step, the larger are the thresholds defined in the smart sieves, so more objects pass all sieves, reaching the conjunction data generation process, increasing its executing time. This can be seen in the smart sieves r/a column, where the filter efficiency decreases with the time step.

The fact that the propagation and sieves executing time decreases and the conjunction data generation time increases with the time step, gives origin to an optimal time step where the total executing time is minimum. However, this optimal time step is difficult to find, because it depends on several factors such as the altitude regime of the primary object and the catalog used. It depends on the altitude regime because the perigee/apogee filter efficiency also depends on the altitude regime. For satellites with the altitude regime similar to ION-MK01 the optimal time steps will be similar, but for GEO and MEO objects, the optimal time step will be much lower, because less objects will pass the perigee/apogee filter and less objects will be propagated.

Hence, the objective of this section isn't to find the exact optimal time step for a specific case, but

Table 4.10: Executing time and sieves efficiency with a propagation time interval of 7 days for several time steps. The primary object is ION-MK01 and the catalog epoch is $2020/09/09$.

Time Step [s]	Executing Time [s]				Smart Sieves r/a [%]	XYZ r/a [%]	r^2 r/a [%]	r^2 fine r/a [%]
	Propagation	Sieves	Conj Data Gen	Total				
150	5599	5767	5475	16843	97.97	76.12	83.67	41.64
180	5313	3360	8068	16743	97.05	72.50	80.31	40.09
210	3959	4936	10468	19367	95.97	68.77	77.17	38.57

rather a time step that decreases the total executing time if the primary is a LEO object, a pseudo-optimal time step. In the context of D-Orbit, the primary satellites will be in LEO, so GEO and MEO pseudo-optimal time steps weren't studied.

This pseudo-optimal time step was found for several propagation time intervals. It was found that, for 1 day of analysis interval, the pseudo-optimal time step is 120 seconds and for the other analysis intervals it is 180 seconds. This growth has to due with the fact that, for larger propagation time intervals, the perigee/apogee efficiency is smaller and more objects will pass this filter, needing propagation. Hence, for the same time step but different duration of the analysis interval, the propagation time will have more weight in the executing time, because there will be more objects and time instances to propagate to. Therefore, the pseudo-optimal time step increases, in order to decrease the propagation weight in the total executing time.

A code that finds the optimal time step with better resolution could be developed . One that performs several runs increasing the time step at each run, storing the total executing time and returning its minimum. This would be a code needed to be ran every time the primary satellite was changed, what would be very heavy computationally. In a high precision level, to find the optimal time step, the dependency on the catalog used would also need to be taken into consideration. For all of these reasons, it was only found a pseudo-optimal time step for LEO region. Table 4.11 presents the LEO pseudo-optimal time step for each propagation time interval.

Table 4.11: LEO pseudo-optimal time steps found for each propagation time interval and their respective total executing times.

Propagation time [day]	Pseudo-optimal Time Step [s]	Total Executing Time [s]
1	120	441
3	180	4028
6	180	12695
7	180	16743

Analysing the sieves efficiency in Table 4.7 to 4.10, the sieve with larger efficiency is the r^2 sieve. This sieve compares the relative distance of the object pair with the threshold, meaning that it is a more restrictive sieve than the XYZ sieve, in spite of being more computationally heavy. The r^2 fine sieve is the one with lowest efficiency because it is used after the r^2 sieve, so there are less objects to discard, being most of them already within $R_{th_{fine}}$. In spite of this, the overall efficiency of the smart sieve method

implemented is large, always above the $94\,\%$ for the time steps studied.

With this study, it is possible to define the configurations of the collision warning tool. From the operational point of view, a propagation time interval less than 3 days would be useless since, if a conjunction is detected and a manoeuvre is required, there won't be enough time to analyse/accompany the conjunction and plan an avoidance manoeuvre. It was defined a propagation time interval of 7 days. As concluded in this section, for an analysis time interval of 7 days and a primary satellite in the LEO region, an approximation to the optimal time step is 180 seconds. The start of the propagation time interval was defined as the timestamp of the more recent catalog download, this way all objects passing the perigee/apogee filter will be propagated initially to this time instance.

5

Covariance Estimation Using TLEs

5.1 Problem Statement

TLEs don't have information about the covariance of the state vector, so this chapter will cover the estimation of it. When the state vector (position and velocity) of an object is calculated, it always have an associated uncertainty due to the kind of data used and also to the algorithms used to propagate those positions. Those uncertainties are represented in a matrix, called the covariance matrix. [27]

The covariance matrix is an important information because it allows the operator of the satellite to know the confidence in its SSA. These uncertainties will allow, in Chapter 6, to perform conjunction analysis and calculate the Probability of Collision (PC).

5.2 Theoretical Background

In Chapter 6, for PC calculation, it will be assumed that the uncertainties in position follow a Gaussian distribution. Other distributions can be used but this is the one generally used in literature, since it's well known and simplifies computations [28]. Hence, considering two random Gaussian variables, X and

Y, with unbiased measurements and random error, then the covariance is the expected value of the deviations product of those variables [29],

$$\mathrm{cov}(X,Y) = E\Big[\big(X - E[X]\big)\big(Y - E[Y]\big)\Big] \qquad \mathrm{cov}(X,X) = E\Big[\big(X - E[X]\big)^2\Big] = E[e^2] = \sigma^2 \,. \tag{5.1}$$

Considering the vector $\mathbf{X} = [X_1, ..., X_N]$, the covariance matrix, P, of dimensions $N \times N$ is an extension of the meaning of covariance but for multidimensional variables,

$$P_{ij} = \mathrm{cov}(X_i, X_j) = E\Big[\big(X_i - E[X_i]\big)\big(X_j - E[X_j]\big)\Big] \,. \tag{5.2}$$

The elements in the diagonal of the matrix, where $i = j$, are the variances of the elements in vector **X**. In this case, **X** is a vector of dimension 6 (3 components of position and 3 components of velocity) and P a matrix of dimension 6×6.

The diagonal terms of a covariance matrix are the square of the standard deviation and the non-diagonal terms are the correlation components, they describe how much the error with other directions correlate. The covariance matrix can be divided into two matrices, the position matrix and the velocity matrix. If one only considers the positional covariance, it is possible to relate these values with the probability of finding the object in a certain region, called the covariance ellipsoid, shown in Fig. 5.3. This ellipsoid is the 3D representation of a covariance matrix and the uncertainties in objects' position. It's the surface of a 3D Gaussian distribution where σ has three dimensions, the along-track, radial, and cross-track directions. It can also be seen has a confidence ellipsoid defining the region that contains $95\,\%$ of all samples, if plotting the 1σ ellipsoid.

To plot this ellipsoid, it's necessary to diagonalize and calculate the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors represent the orientation and the axis of the ellipsoid. The eigenvalues represent the shape of the ellipsoid and the length of the previous axes, basically the spread of data in the direction of the eigenvectors.

5.3 Methods for Covariance Estimation in TLEs

The accuracy of a TLE depends on several factors such as: the accuracy of the tracking sensor that originated the observations for TLE formation; the amount of measurements done; the distance of the object to the sensor; the type of orbit and its space environment conditions; the object size, shape and reflectivity, among other things.

Using TLEs to know the position of the object has the disadvantage of having low accuracy comparing to state vectors derived from more complete numerical propagators. In spite of TLEs having low

accuracy, these are the only format of data publicly available to determine the position of all tracked space debris and this is why TLEs are used in this thesis.

Covariance information of TLEs allows this type of data to be used in a larger field of applications and because of that, this topic received extensive attention and investigation throughout the years, giving origin to different methods to estimate covariance for TLEs. There are three methods to estimate the accuracy of a TLE and obtain a covariance for it, these are: TLE differencing, TLE comparison with tracking data and TLE as pseudo-obervations [27]. The first will be the one used in this thesis

TLE differencing is based only on historical TLEs and works as follows: take as much TLEs of an object as possible, propagate them to the same time instance and subtract the state vectors to obtain statistical information [29]. The covariance matrix is extracted in the most recent TLE epoch or another desired time instance, by propagating previous TLEs to that instance and differencing the values, obtaining the so called, residuals. The covariance is formed by the distribution of points in those time instances. A downside of this method is that it considers that TLEs are unbiased and TLE generation is error free. The biggest advantage is that it only relys on SGP4/SDP4 theory and it does not require additional models and Orbit Determination (OD) routines, so it can be used for any cataloged object in a simple and fast way, which is why it was used in this thesis.

TLE comparison with tracking data is used for specific objects and not for every type of object present in the catalog, because it requires high-precision tracking data like pseudo-ranges and elevation angles or Global Positioning System (GPS) data [27]. This method is out of the scope of this thesis because there is no high-precision tracking data from ground stations publicly available for all objects. Methods that require additional data besides TLEs will not be considered.

TLE as pseudo-observations uses derived states of TLEs as pseudo-observations of ground stations, meaning that these states will be considered as measurements in an OD process. Through inverted tracking station models, simulated tracking data is obtained to perform an OD process, obtaining a new state. This state may be propagated using a numerical propagator and the covariance is obtained directly from the OD process [30]. The downside is that these inverted tracking station models and special perturbations propagators are not available and because of this, it was decided to go for the TLE differencing method.

5.4 Implementation of TLE Differencing

TLEs will be downloaded for the objects involved in a conjunction for a period 20 days into the past. The more recent TLE will be called prime and the covariance will be calculated only for a time instant, the TCA of the conjunction. All TLEs are propagated to TCA, using SGP4. The prime TLE is assumed to generate the ”true” state vector, called prime state vector, which is the most logical assumption. Thus,

several state vectors are calculated at TCA and subtracted. Treating them statistically, it is possible to calculate a covariance, assuming that these state vectors are Gaussian distributed.

The state vectors generated by SGP4 are represented in a TEME coordinate system, so the obtained covariance will also be in this coordinate system. For better understanding of results, it is useful to transform the covariance for a satellite based coordinate system, Radial Tangential Normal (RTN) and Normal In-track Cross-track (NTW) [6], see Fig. 5.1.

The RTN frame moves with the satellite and it's used mainly to describe orbital error and displacements of satellite orbits. The R axis points from the center of the Earth towards the satellite. The N axis points in the direction of the angular momentum and the T axis is perpendicular to the R and N axis. The T axis is usually not aligned with the velocity vector except for circular or elliptical orbits at apogee and perigee. Radial displacements are along the R axis, along-track displacements are along the T axis and cross-track displacements are along the N axis. Given a state vector with position, r, and velocity, v, the rotation matrix for this coordinate system, $R_{\text{TEME}\to\text{RTN}}$ is,

$$\hat{R} = \frac{\vec{r}}{|\vec{r}|}\,, \quad \hat{N} = \frac{\vec{r}\times\vec{v}}{|\vec{r}\times\vec{v}|}\,, \quad \hat{T} = \hat{N}\times\hat{R}\,, \tag{5.3}$$

$$\vec{r}_{\text{TEME}} = \begin{bmatrix}\hat{R} & \hat{T} & \hat{N}\end{bmatrix}\vec{r}_{\text{RTN}}\,, \quad \vec{v}_{\text{TEME}} = \begin{bmatrix}\hat{R} & \hat{T} & \hat{N}\end{bmatrix}\vec{v}_{\text{RTN}}\,, \tag{5.4}$$

$$R_{\text{RTN}\to\text{TEME}} = \begin{bmatrix}\begin{bmatrix}\hat{R} & \hat{T} & \hat{N}\end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix}\hat{R} & \hat{T} & \hat{N}\end{bmatrix}\end{bmatrix}\,, \tag{5.5}$$

$$R_{\text{TEME}\to\text{RTN}} = R^{-1}_{\text{RTN}\to\text{TEME}} = R^{T}_{\text{RTN}\to\text{TEME}} = \begin{bmatrix}\begin{bmatrix}\hat{R} & \hat{T} & \hat{N}\end{bmatrix}^T & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix}\hat{R} & \hat{T} & \hat{N}\end{bmatrix}^T\end{bmatrix}\,. \tag{5.6}$$

The transformation from TEME to RTN is simply a cartesian transformation between orthogonal reference frames, then $R^{-1}_{\text{RTN}\to\text{TEME}} = R^{T}_{\text{RTN}\to\text{TEME}}$, which can be easily understood when observing $R_{\text{RTN}\to\text{TEME}}$ definition in Eq. 5.5.

The NTW system is mainly used to analyze drag effects because drag always acts along the velocity vector. The T axis always points to the velocity vector, the W axis has the direction of the angular momentum and the N axis is perpendicular to the T and W axis. Note that sometimes there is some confusion when referring to the displacements of position. In the NTW reference system, the displacements in the velocity direction are called in-track displacements, while in the RTN reference system, these are called along-track displacements. Given a state vector with position, r, and velocity, v, the rotation matrix for this coordinate system, $R_{\text{TEME}\to\text{NTW}}$ is,

$$\hat{T} = \frac{\vec{v}}{|\vec{v}|}\,, \quad \hat{W} = \frac{\vec{r}\times\vec{v}}{|\vec{r}\times\vec{v}|}\,, \quad \hat{N} = \hat{T}\times\hat{W}\,, \tag{5.7}$$

$$\vec{r}_{\text{TEME}} = \begin{bmatrix}\hat{T} & \hat{N} & \hat{W}\end{bmatrix} \vec{r}_{\text{NTW}} \ , \qquad \vec{v}_{\text{TEME}} = \begin{bmatrix}\hat{T} & \hat{N} & \hat{W}\end{bmatrix} \vec{v}_{\text{NTW}} \ , \tag{5.8}$$

$$R_{\text{NTW}\to\text{TEME}} = \begin{bmatrix}\begin{bmatrix}\hat{T} & \hat{N} & \hat{W}\end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix}\hat{T} & \hat{N} & \hat{W}\end{bmatrix}\end{bmatrix} \ , \tag{5.9}$$

$$R_{\text{TEME}\to\text{NTW}} = R^{-1}_{\text{NTW}\to\text{TEME}} = R^{T}_{\text{NTW}\to\text{TEME}} = \begin{bmatrix}\begin{bmatrix}\hat{T} & \hat{N} & \hat{W}\end{bmatrix}^T & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix}\hat{T} & \hat{N} & \hat{W}\end{bmatrix}^T\end{bmatrix} \ . \tag{5.10}$$

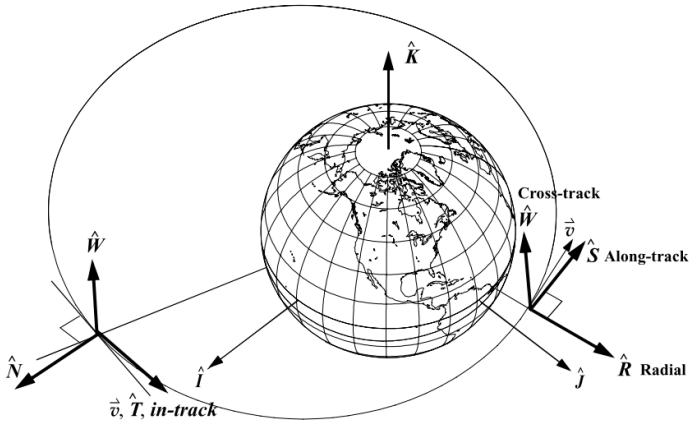


Figure 5.1: Satellite based coordinate systems. NTW and RTN, also called RSW, reference frames. [6]

Defining the state vectors by $\vec{X} = [\vec{r} \quad \vec{v}]^T = [r_x \quad r_y \quad r_z \quad v_x \quad v_y \quad v_z]^T$ and calculating the residuals between the state vectors obtained from propagated TLEs and the prime state vector,

$$\delta\vec{X}_{\text{TEME}i} = \vec{X}_{\text{propagated}_{\text{TEME}}} - \vec{X}_{\text{prime}_{\text{TEME}}} \ , \tag{5.11}$$

where i refers to the i-th TLE propagated to TCA, in a set of N TLEs. Remembering Eq. 5.2, $P_{\text{TCA}_{\text{TEME}}}$, the covariance matrix at the TCA in the TEME frame is

$$P_{\text{TCA}_{\text{TEME}}} = E[(\delta\vec{X}_{\text{TEME}} - m)(\delta\vec{X}_{\text{TEME}} - m)^T] \ , \tag{5.12}$$

being m the vector with the mean of the residuals, it follows that

$$m = \frac{\sum_{i=1}^{N-1}(\delta\vec{X}_{\text{TEME}})_i}{N-1} \ , \qquad P_{\text{TCA}_{\text{TEME}}} = \frac{\sum_{i=1}^{N-1}(\delta\vec{X}_{\text{TEME}i} - m)(\delta\vec{X}_{\text{TEME}i} - m)^T}{N-1} \ . \tag{5.13}$$

In order to convert a TEME covariance matrix to a RTN covariance matrix, it is necessary to apply a rotation matrix, $R_{TEME\to RTN}$, obtained through the prime state vector,

$$P_{\text{TCA}_{\text{RTN}}} = R_{\text{TEME}\to\text{RTN}} P_{\text{TCA}_{\text{TEME}}} R^T_{\text{TEME}\to\text{RTN}} \ . \tag{5.14}$$

5.4.1 Validation of Implementation

In order to validate the implementation of the covariance generation process, it was tried to reproduce the same results as the ones presented by Osweiler [29]. The results of the covariance are presented in a NTW reference frame so, for validation purposes, the covariance returned by the algorithm will be in NTW. For the time window 8, described in Osweiler as the time interval between $2004/10/06$ and $2004/10/21$, all TLEs, whose epoch is in this time interval, were retrevied for the LAGEOS satellite with the prime TLE shown in Fig. 5.2. The covariance obtained by Osweiller is presented in Table 5.1. Table 5.2 shows the covariance matrix obtained by the developed code for covariance generation.

```
1 08820U 76039A   04294.65281357 -.00000002 +00000-0 +10000-3 0  9996
2 08820 109.8594 349.9486 0044244 196.5047 163.4332 06.38664775408443
```

Figure 5.2: Prime TLE used in Osweiller for the LAGEOS satellite and time window 8 (from $2004/10/06$ to $2004/10/21$).

Table 5.1: Covariance matrix obtained by Osweiller for the LAGEOS satellite and time window 8 (from $2004/10/06$ to $2004/10/21$). The covariance matrix is presented in the NTW reference frame (called VNC by Osweiller) and the units of the individual cells are equal to the product of the corresponding row and column units.

	R_T [km]	R_N [km]	R_W [km]	V_T [km/s]	V_N [km/s]	V_W [km/s]
R_T [km]	4.38E-02	5.47E-02	-9.35E-04	-2.55E-06	-1.76E-05	2.58E-06
R_N [km]	5.47E-03	2.69E-03	9.16E-04	-1.25E-06	-1.21E-06	5.64E-09
R_W [km]	-9.35E-04	9.16E-04	8.05E-03	-4.25E-07	2.72E-06	-2.94E-06
V_T [km/s]	-2.55E-06	-1.25E-06	-4.25E-07	5.78E-10	5.67E-10	-3.29E-12
V_N [km/s]	-1.76E-05	-1.21E-06	2.72E-06	5.67E-10	8.09E-09	-1.84E-09
V_W [km/s]	2.58E-06	5.64E-09	-2.94E-06	-3.29E-12	-1.84E-09	1.58E-09

Table 5.2: Covariance matrix obtained by the developed code for the LAGEOS satellite and time window 8 (from $2004/10/06$ to $2004/10/21$). The covariance matrix is presented in the NTW reference frame and the units of the individual cells are equal to the product of the corresponding row and column units.

	R_T [km]	R_N [km]	R_W [km]	V_T [km/s]	V_N [km/s]	V_W [km/s]
R_T [km]	4.16E-02	5.36E-03	-6.77E-04	-2.50E-06	-1.66E-05	2.43E-06
R_N [km]	5.36E-03	2.70E-03	1.03E-03	-1.25E-06	-1.12E-06	1.65E-09
R_W [km]	-6.77E-04	1.03E-03	7.79E-03	-4.78E-07	2.59E-06	-2.78E-06
V_T [km/s]	-2.50E-06	-1.25E-06	-4.78E-07	5.79E-10	5.27E-10	-1.39E-12
V_N [km/s]	-1.66E-05	-1.12E-06	2.59E-06	5.27E-10	7.65E-09	-1.74E-09
V_W [km/s]	2.43E-06	1.65E-09	-2.78E-06	-1.39E-12	-1.74E-09	1.49E-09

Comparing Tables 5.1 and 5.2, the values obtained are in the same order of magnitude and with the same signal.

Analysing only the diagonal elements in the positional covariance matrix, which is the one used in Chapter 6 for Probability of Collision (PC) calculation, the difference obtained in the positional standard deviations is approximately $5.32\,\mathrm{m}$, $0.09\,\mathrm{m}$, $1.46\,\mathrm{m}$ in the in-track, normal and cross-track displacements, respectively. These are very small when compared to the standard deviations of LEO objects, so the

discrepancies found between the implementation of Osweiller and the one developed in this thesis are not significant. In Table 5.2, the in-track displacement has a covariance value of $\sigma^2 = 0.0416\,\text{km}^2$ which gives a standard deviation of $\sigma = 0.203\,\text{km} = 203\,\text{m}$. LAGEOS is a MEO satellite, so it won't suffer the drag effects that a satellite in LEO will. The normal ($P_{R_N R_N}$) and cross-track ($P_{R_W R_W}$) covariances lead to standard deviations of 52 m and 88 m, respectively, which are much lower than the in-track deviation. These comparisons were done for the other satellites and time windows in Osweiler, the results were similar to the presented case.

When trying to reproduce the results of Osweiller, it was ensured that the TLE set used was the same, so the differences encountered may be explained by the fact that the SGP4 version used by Osweiller is different than the one used here, although there is no way to confirm it. This is likely because the paper that served as the standard base for SGP4 implementation [3], was only published in 2006, as well as Osweiler's work, so some differences in implementation may have occurred.

This method does not take into account any manoeuvres performed by the operator, meaning that if a maneuver is performed, any previous TLEs are not valid anymore and so the covariance values may increase dramatically. This is a problem that could only be tackled if there was a completely developed report system, where each operator reports the maneuver to be made.

When integrating this covariance generation process in the collision warning tool, the difference in implementation is that covariance is not calculated for the prime TLE epoch but for TCA. To give an example of a covariance matrix in a conjunction, for the $2020/10/01$ catalog and ION-MK01 as the primary object, it was found a conjunction with object 44 414. The covariances at TCA were generated for these two objects, shown in Tables 5.3 and 5.4.

Table 5.3: Covariance matrix of ION-MK01 at TCA. The catalog of $01/10/2020$ was used.

	R_R [km]	R_T [km]	R_N [km]	V_R [km/s]	V_T [km/s]	V_N [km/s]
R_R [km]	3.80E-02	5.53E+00	1.57E-02	-6.05E-03	-4.20E-05	1.75E-05
R_T [km]	5.53E+00	1.30E+03	2.02E+00	-1.42E+00	-5.49E-03	2.45E-03
R_N [km]	1.57E-02	2.02E+00	8.61E-03	-2.21E-03	-1.98E-05	8.82E-06
V_R [km/s]	-6.05E-03	-1.42E+00	-2.21E-03	1.56E-03	6.00E-06	-2.68E-06
V_T [km/s]	-4.20E-05	-5.49E-03	-1.98E-05	6.00E-06	5.45E-08	-2.05E-08
V_N [km/s]	1.75E-05	2.45E-03	8.82E-06	-2.68E-06	-2.05E-08	9.59E-09

Table 5.4: Covariance matrix of 44414 at TCA. The catalog of $01/10/2020$ was used.

	R_R [km]	R_T [km]	R_N [km]	V_R [km/s]	V_T [km/s]	V_N [km/s]
R_R [km]	1.32E-02	-1.95E+00	-3.86E-03	2.18E-03	-2.77E-05	-4.11E-05
R_T [km]	-1.95E+00	6.33E+02	6.61E-01	-7.02E-01	6.63E-03	7.61E-03
R_N [km]	-3.86E-03	6.61E-01	1.54E-03	-7.37E-04	9.60E-06	1.43E-05
V_R [km/s]	2.18E-03	-7.02E-01	-7.37E-04	7.79E-04	-7.37E-06	-8.48E-06
V_T [km/s]	-2.77E-05	6.63E-03	9.60E-06	-7.37E-06	8.57E-08	1.08E-07
V_N [km/s]	-4.11E-05	7.61E-03	1.43E-05	-8.48E-06	1.08E-07	1.51E-07

ION-MK01 has an along-track standard deviation of 36 km, a radial standard deviation of 195 m and

a cross-track standard deviation of 93 m. The secondary object has an along-track standard deviation of 25 km, a radial standard deviation of 115 m and a cross-track standard deviation of 39 m. The along-track deviations are very large, comparing with radial and cross track deviations, these are typical values for objects in LEO and an indicator of the limitations of SGP4 and TLEs in drag modelling. In order to visualize the meaning of these covariances, the 1σ ellipsoid was plotted, see Fig. 5.3. With the covariance generation process implemented, it is possible to calculate a Probability of Collision (PC). Chapter 6 will cover this topic.

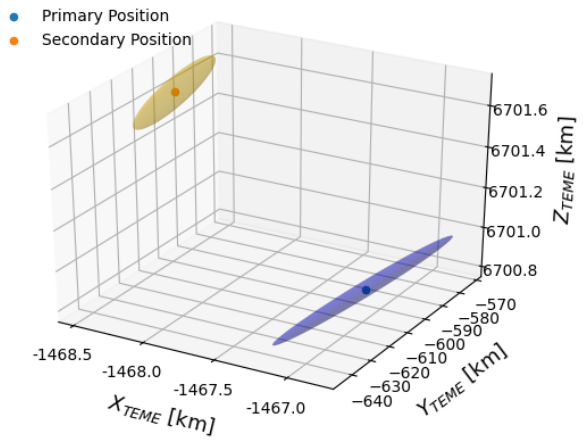


Figure 5.3: Covariance 1σ ellipsoids for the primary and secondary object at TCA. The primary object is ION-MK01 and the secondary object NORAD ID is 44 414. This was a conjunction detected when running the catalog of $01/10/2020$. This plot allows to have a better visualization of the conjunction geometry and the size of the position uncertainties presented in Tables 5.3 and 5.4.

6

Probability of Collision Calculation

6.1 Defining Probability of Collision

The Probability of Collision (PC) is of outmost importance when analysing a conjunction because the Miss Distance (MD) parameter ignores position uncertainty and may lead to an exaggerated assessment of the true risk. The PC includes position uncertainty because it is obtained through covariance information.

There are several approaches for the PC calculation, SOCRATES has a very conservative approach. It calculates the maximum PC instead of the actual PC [31], setting a standard shape and orientation for the covariance and changing the size of it until it reaches a maximum PC, leading to an overestimation. There is no issue about being conservative in the PC calculation, but the problem of SOCRATES method is the need to fix the major-to-minor axes of the projected covariance ellipse. This is done by setting radial, along-track and cross-track values of 100 m, 300 m, 100 m, respectively [12], which is not a good assumption since this won't be the case for all conjunctions. In this thesis, a better approach will be used. The PC calculation is based on covariance information, obtained in Chapter 5. Then, this covariance

[12] T. S. Kelso, "SOCRATES: Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space," CelesTrak, Dec. 2019, accessed on 2020/11/22. [Online]. Available: https://celestrak.com/SOCRATES/

will be sized in order to have the maximum PC.

The definition of the PC is shown in Eq. 6.1. It's the integration of a 2D Gaussian probability density function centered on the secondary object, over the circle of radius d centered on the primary object,

$$PC = \frac{1}{2\pi|Det(C)|^{1/2}} \iint_{x^2+y^2 \leq d^2} \exp\Big(-\frac{1}{2}(\mathbf{r} - \mathbf{r_{s/p}})^T C^{-1} (\mathbf{r} - \mathbf{r_{s/p}})\Big) \mathrm{d}x\mathrm{d}y \ , \tag{6.1}$$

where C is the 2×2 projection on the conjunction plane of the combined 3×3 covariance at TCA (see Fig. 6.1), d is the sum of the two object radius also called Hard Body Radius (HBR) (see Fig. 6.2), $\mathbf{r} = [x \ y]^T$ is any point in the conjunction plane that satisfies $x^2 + y^2 \leq d^2$ and $\mathbf{r_{s/p}} = [r_{s/p} \ 0]^T$ is the position of the secondary relative to the primary, in the conjunction plane.

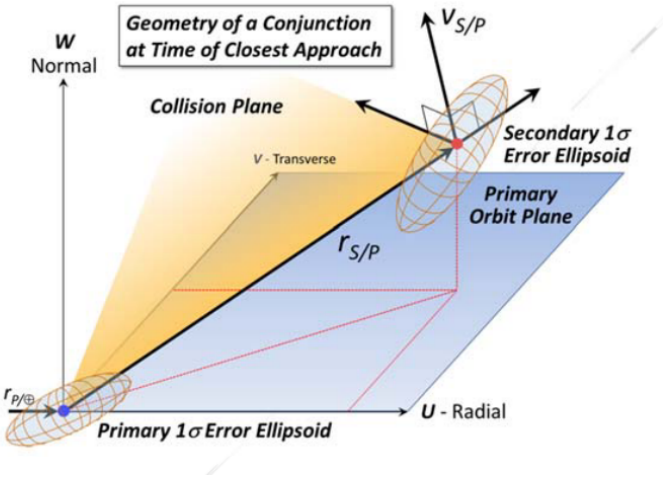


Figure 6.1: Representation of a conjunction plane, also called collision plane or B-plane, and the 1-σ error ellipsoids of each object. The conjunction plane, in orange, is the plane perpendicular to the relative velocity at TCA, $\mathbf{v_{s/p}}$, that contains the positions of both objects. The conjunction reference frame is defined as the frame centered at the primary or secondary, with the x-axis in the same direction of the relative position vector, $\mathbf{r_{s/p}}$, and the y-axis with the same direction of the relative velocity vector, $\mathbf{v_{s/p}}$. In TCA these vectors are always orthogonal, being the z-axis the orthogonal vector to these two. [7]

When calculating the PC, what is computed is not exactly the probability of collision, but the probability that two objects are less than a specified distance at TCA [7]. This approach is made because there is no information about size, shape and attitude of all catalogued objects, so a sphere of a certain radius around the objects is defined. This is a conservative assumption because if the objects are not spheres, then a simple change in attitude may avoid a collision and when considering the objects as spheres, the problem of not knowing objects' attitude is discarded.

The data needed for the PC calculation is:

- The size of the primary and secondary object.
- The TCA of the conjunction.

- The positional covariance of the primary and secondary object at TCA (the velocity covariance will not be considered here).

The last two parameters were obtained in Chapters 4 and 5, so it's only necessary to know the size of the objects. The DISCOS database [32], maintained by ESA, has information about the sizes of cataloged objects and it will be used in this thesis. This database is organized by object type and if the object size is not present in this database, it will be used a pre-assigned conservative value for the dimension of that type of object. For objects characterized as debris and mission related objects, if the object's dimensions are not available the object radius is defined as $10\,\mathrm{m}$, which is a very conservative value. For objects characterized as 'payload', 'rocket body' and 'unknown', if the object's dimensions are not available, the object radius is defined as $20\,\mathrm{m}$.

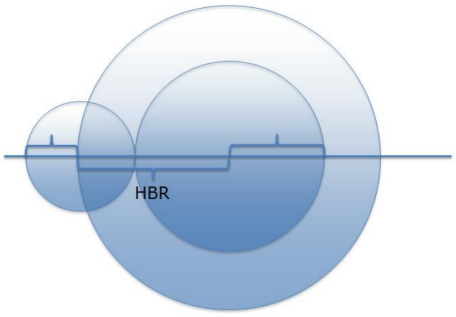


Figure 6.2: Representation of Hard Body Radius (HBR) as the sum of the radius that define both primary and secondary objects' sizes. [13]

The PC calculation is a 3D dynamic problem but this will be reduced to a 2D static problem in order to be more efficient, calculating a double integral instead of triple integral. The sum of the objects' sizes will be projected to the conjunction plane as a circle and the error ellipsoids as ellipses. In order to do this, the following assumptions are considered [7]:

- Objects will be treated as spheres with conservative radius.
- The conjunctions are very short in time, meaning that the relative velocities between objects is large, in most cases. This allows to consider the relative motion as rectilinear with constant velocities, during the conjunction. For low relative velocities, the conjunction lasts longer, so the motion may not be considered rectilinear anymore. The assumption of rectilinear motion is valid for relative velocities as low as $0.013\,\mathrm{km/s}$ [33]. Most conjunctions in LEO have relative velocities in the order of $15\,\mathrm{km/s}$, so this assumption may be generalized.
- Velocity covariance is negligible. It is very small when compared to the velocities of LEO objects and positional covariances, as it can be seen in the covariance matrices obtained in Sec. 5.4.1, for example Table 5.4.

[13] NSDTC

- Positional covariance is constant during the conjunction, allowing to calculate the PC only at TCA.

- Positional uncertainties can be described by a random Gaussian distribution. This assumption is valid for propagation time intervals smaller than 1 week. If the objects are propagated much far in the future, the uncertainties would become so large that the covariance would need to bend along the orbit path and another distribution should be used. [14]

- Primary and secondary position uncertainties are independent, allowing to simply sum both covariances in a common frame, to form a combined covariance.

The combined covariance matrix is obtained by summing the covariances of both objects, these covariance matrices need to be in the same reference frame, the conjunction reference frame described in Fig. 6.1, centered at the primary object. A covariance matrix is calculated in this reference frame by translating and rotating the state vectors in TEME. The rotation matrix is obtained by the following expressions,

$$\vec{I} = \frac{r_{s/p}}{|r_{s/p}|} \,, \qquad \vec{J} = \frac{v_{rel}}{|v_{rel}|} \,, \qquad \vec{K} = \vec{I} \times \vec{J} \,, \tag{6.2}$$

$$R_{\text{TEME}\rightarrow\text{B-plane}} = \begin{bmatrix} \begin{bmatrix} \hat{I} & \hat{J} & \hat{K} \end{bmatrix}^T & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} \hat{I} & \hat{J} & \hat{K} \end{bmatrix}^T \end{bmatrix} \,. \tag{6.3}$$

To obtain the state vector in the conjunction plane, it's necessary to multiply the rotation matrix with the state vector in TEME,

$$X_{\text{B-plane}} = R_{\text{TEME}\rightarrow\text{B-plane}} X_{\text{TEME}} \,. \tag{6.4}$$

The covariance matrix is then calculated using the state vectors in the conjunction plane. Both covariances matrices are now ready to be summed. The next step is to project the combined covariance matrix to the conjunction plane. Recurring to the expression of a projection matrix,

$$P = A(A^T A)^{-1} A^T \,, \tag{6.5}$$

where P is the projection matrix and A is the matrix whose columns are the vectors that form the basis of the plane intended to project. The covariance matrix projected in the plane is obtained by,

$$C_{projected} = PCP^T \,. \tag{6.6}$$

[14] NSDTC

6.1.1 Integral Computation

The PC is obtained by calculating the integral in Eq. 6.1. This is done by a numerical integration using a two-dimensional Simpson's rule in the x and y cartesian coordinates of the conjunction reference frame, i.e, in the conjunction plane. A grid is created over the HBR and the integral is calculated in this grid. However, it's not possible to form a perfect circle with the grid created and if the steps are very small, then the calculation of PC would take a lot of time for every conjunction found. So, in order to be conservative, the integration is performed over a square circumscribing the HBR, this simplifies the computation of PC and gives a slightly larger but safe value.

The step in the numerical integration needs to allow a good accuracy of the PC but it shouldn't take too long to calculate it. In order to determine an adequate step, a conjunction was simulated by imposing a certain Miss Distance (MD), a certain HBR and a certain covariance matrix so that PC is equal to 1. A conjunction with a PC equal to 1 is one that has a MD equal to 0 m and a combined covariance with standard deviations much smaller than HBR. In this case, HBR is set to 15 m and a identity matrix as the combined covariance, meaning that the standard deviations, in every direction, will be 1 m. With these conditions the PC is equal to 1 and the step chosen is the one that gives a probability of 1 or very close to 1, but that does not take too much time calculating PC.

The value of PC and the time it takes to calculate it depends on HBR because, the larger the HBR, the more iterations are needed to create the grid, so the step will be defined has $h = HBR/n$, where n is the number of divisions of HBR in one direction. Hence, by choosing a value for n, the step h is being defined.

For several values of n and with the conditions to result in a PC equal to 1, the time it takes to calculate PC was obtained. A value of $n = 150$ was chosen because it was the value that lead to a PC equal to 1 and the time of calculation was acceptable, 1.8 seconds.

6.1.2 Maximum Probability and Covariance Sizing

A concern, when analysing a conjunction, is the covariance realism. The covariance obtained in Sec. 6.1 is also an estimation, hence it may be overestimated or sub-estimated, originating a too small PC. The size of the covariance may be changed, this means that the entries in the covariance matrix may be multiplied by a scaling factor. This process is called covariance sizing and it's used to obtain the maximum PC.

There is a theoretical maximum for the PC when using a covariance with a certain shape and orientation, because the PC is the integral of the Gaussian distribution over the HBR, see Fig. 6.3, so there is a certain value for the standard deviation where the area below the Gaussian will be maximized. The scaling is done by maintaining the shape and orientation of the ellipse, only the size of it is changed. The

covariance matrix is multiplied by the scaling factor, k. This factor can be found analytically, assuming that PC takes its maximum when the Gaussian distribution does, [34]

$$k = \sqrt{\frac{r_{s/p} C^{-1} r_{s/p}}{2}} \ , \tag{6.7}$$

where $r_{s/p}$ is the relative position vector and C the 2×2 projection on the conjunction plane of the combined 3×3 covariance at TCA (see Fig. 6.1). The scaled covariance is obtained by

$$C_{\text{PC}_{\text{max}}} = k^2 C \ . \tag{6.8}$$

PC is calculated the same way as before but using the scaled covariance, $C_{\text{PC}_{\text{max}}}$, in order to obtain the maximum PC. If $k > 1$, then the uncertainties are small comparing to MD, if $k < 1$, it means that the covariance estimated is diluted and the uncertainties are larger than MD. If operating in this dilution region, the recommendation is to obtain better data to reduce the uncertainty and reassess the conjunction [31]. Even if PC is small, it doesn't mean that the conjunction is safe, as proven in Sec. 6.2.2 and illustrated by Fig. 6.3.

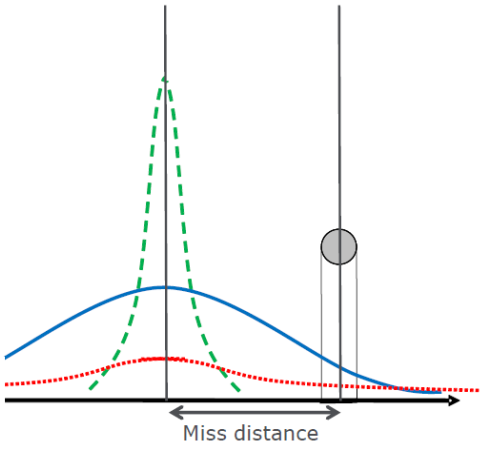


Figure 6.3: Representation of the area below the intersection of the Gaussian curve with the HBR, represented as a grey circle. The blue Gaussian illustrates the scaled covariance that maximizes the area below the Gaussian in 1D. When calculating the maximum PC, the line of thought is the same but for a 2D Gaussian. In both green and red cases, the PC is small because the area below the intersection of the Gaussian curve with the HBR is small. However, the green case represents a safe conjunction because the uncertainty is small and the red case an unsafe conjunction because the uncertainty is large. [15]

[15] NSDTC

6.2 Testing the Probability of Collision Calculation

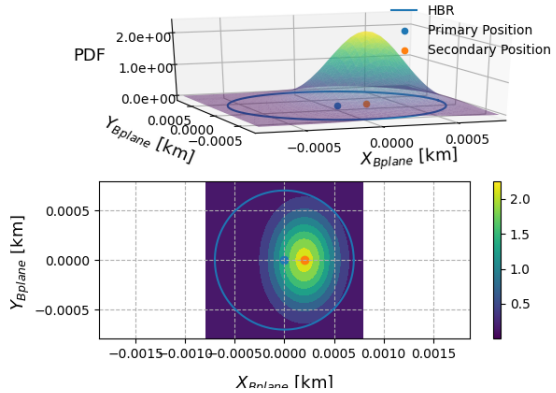
The final objective of this thesis is to test the implementation of the PC calculation. In order to do that, several cases will be studied by performing conjunction analysis in extreme and real cases, comparing with a LEOLabs conjunction and analysing the collision between Iridium-33 and Cosmos 2251. When a conjunction is detected, an analysis is made by the operator to evaluate the danger of the conjunction and decide if an avoidance manoeuvre is needed or not, building his decision upon the parameters calculated (TCA, MD, relative velocity, positional covariances, PC, HBR, PC_{max} and scaling factor).

6.2.1 Extreme and Real Cases

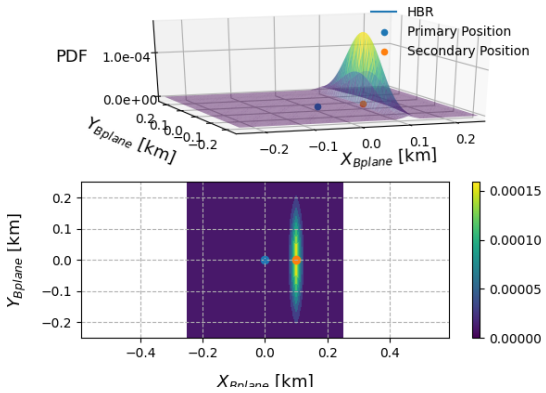
In this test, three cases will be studied: two simulated cases and one real case. The simulated cases represent the extreme cases in PC calculation, when $PC = 1$ and when $PC = 0$. The real case is a conjunction found for ION-MK01, when running the code. For every case, $HBR = 0.7\,m$ because ION-MK01's largest dimension is $0.4\,m$ and object's 44 414 is $0.3\,m$. Figure 6.4 shows the conjunctions of the three cases, in the conjunction plane, with the respective Probability Density Function (PDF). The relevant parameters for each case are defined as:

- **Case 1** - Simulated conjunction where MD was defined as $0.2\,m$ and the projected combined covariance matrix was set as $\begin{bmatrix} 0.05 & 0 \\ 0 & 0.1 \end{bmatrix}$ m^2. This case is the extreme case when PC should be equal or very close to 1, because MD is smaller than HBR and the standard deviation values are very short. This means that objects will pass very close to each other, with their radius intersecting, and the confidence in their positions is high. The goal is to verify if the PC obtained is equal to 1.
- **Case 2** - Simulated conjunction where MD was defined as $100\,m$ and the projected combined covariance matrix was set as $\begin{bmatrix} 100 & 0 \\ 0 & 10000 \end{bmatrix}$ m^2. This case is the extreme case when PC should be equal to 0, because MD is large relative to the standard deviation in the x-axis. This means that objects will pass far from each other and the confidence in their positions is high. The goal is to verify if the PC obtained is equal to 0.
- **Case 3** - Conjunction found between ION-MK01 and the object 44 414 when running the catalog of 2020/10/01. MD is 3212 m and the projected combined covariance matrix is $\begin{bmatrix} 5.8 \times 10^8 & -3.0 \times 10^8 \\ -3.0 \times 10^8 & 1.3 \times 10^9 \end{bmatrix}$ m^2. This is a real case where MD is large but the standard deviations are also large, meaning that the confidence in their positions is low. Hence, PC should be small but larger than case 2 because of the high uncertainties in position. The goal is to verify this.

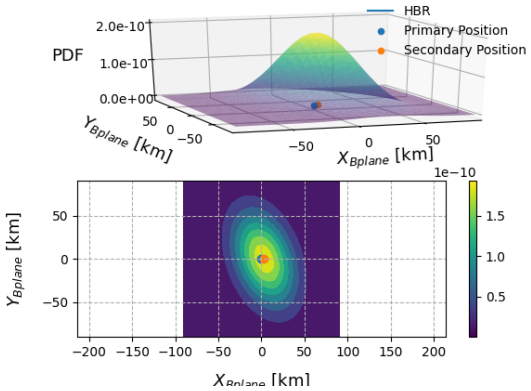
In case 1, the PC obtained was 0.96, which matched expectations. In Fig. 6.4(a), it's important to notice the shape and orientation of the PDF. The covariance matrix was defined as a diagonal matrix with standard deviations of $0.22\,m$ in the x-axis and $0.32\,m$ in the y-axis, resulting in a projected combined covariance ellipse more stretched along the y-axis. Since it is a diagonal matrix, the ellipse is not tilted with respect to the conjunction frame, as it is for example in case 3, Fig. 6.4(c). This case represents a high risk conjunction, so the operator should decide to perform an avoidance manoeuvre.



(a) Case 1 - Simulated conjunction where MD was defined as 0.2 m, with high confidence in positions.



(b) Case 2 - Simulated conjunction where MD was defined as 100 m, with high confidence in positions.



(c) Case 3 - Real conjunction found between ION-MK01 and object 44 414, when running the catalog of 2020/10/01. MD is 3212 m and the uncertainties in positions are large.

Figure 6.4: Representation of conjunctions, in the conjunction plane and frame defined in Fig. 6.1, with the respective PDF in the Z axis. The plot in 2D represents the same as the 3D plot but seen from above, where the color map represents the PDF of the projected combined covariance. It is also represented the MD, which is the distance between the primary and secondary position, in blue and orange, respectively. The HBR is represented by the blue circumference. These figures allow to have a better visualization of the combined covariance projection into the conjunction plane and a better intuition of how PC is calculated, the integral of the PDF over the HBR area.

For case 2, the PC obtained was 6.5×10^{-26} which is a very small value, matching the expectations. A detail to note in Fig. 6.4(b) is that the HBR is barely noticed in the plot, when MD is much larger than HBR. Hence, when MD increases, the PC will decrease because the PDF will have lower values in the HBR area. This case represents a safe conjunction, so the operator doesn't need to perform an avoidance maneuver.

In case 3, the PC was 3.7×10^{-10}, which is larger than case 2, as expected. The MD and standard deviations are much larger than the previous cases and so the HBR is not noticed in Fig. 6.4(c). The values are much lower than the previous cases because the standard deviations are much larger in this case. The values of PC, in most conjunctions found when running the code, will be similar to this one, because the standard deviations are larger than in the simulated cases, so the values of the PDF will be lower. The HBR is a small area, which leads to small values of PC. In case 3, the decision of perform an avoidance maneuver or not is difficult because, even if MD is large and PC is small, the uncertainties in position are large, so it's not clear if the conjunction is safe or unsafe. This is where the maximum PC, PC_{max}, and scalling factor, k, can help in the decision. The PC_{max} and k obtained was 1.39×10^{-8} and 0.1, respectively. Since k is smaller than 1, then the covariance used for PC calculation is diluted, i.e., the uncertainties are larger than MD. More accurate data is needed to better evaluate the collision and decide if an avoidance maneuver is needed. Overall, in every case, the results for the PC matched the expectations and the PC implementation passed the test.

6.2.2 Comparing with LEOLabs conjunction

Another test to the implementation of PC was a comparison with the parameters obtained by LEOLabs, in a high risk conjunction, detected on the $2020/10/13$. LEOLabs is a company dedicated to SSA and tracking of LEO objects. They built their own radars in order to have more accurate orbital data for specific targets and provide collision warning and avoidance services. The conjunction parameters found by LEOLabs are shown in Tables 6.1 and 6.2.

Table 6.1: Data of the high risk conjunction found by LEOLabs on $2020/10/13$ between objects 19 826 and 36 123.

TCA [UTC]	MD [m]	Relative Velocity [km/s]	PC
2020-10-16T00:56:40.726	25	14.659	0.038

Table 6.2: Data of the objects involved in the conjunction. The parameters σ_x, σ_y and σ_z are the standard deviations in the reference frame of the conjunction.

	Primary Object	Secondary Object
Norad ID	19826	36123
σ_x[m]	5	4
σ_y[m]	80	82
σ_z[m]	8	10

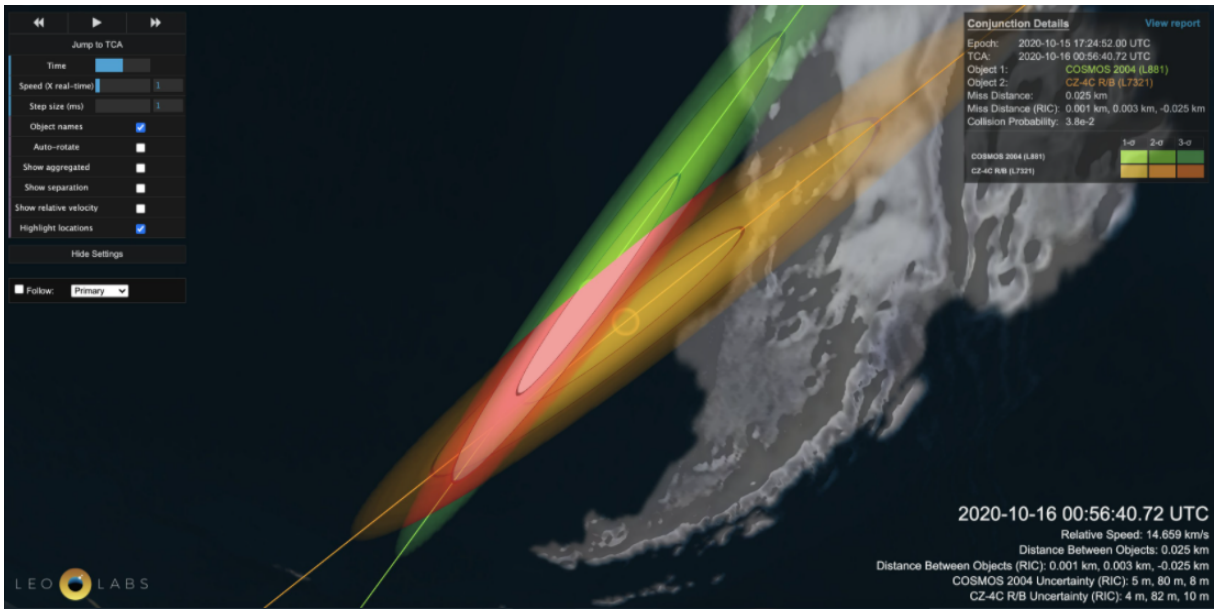


Figure 6.5: Image published by LEOLabs of the conjunction between objects 19 826 and 36 123. The software developed by LEOlabs presents the TCA, MD, relative velocity, relative position vector, PC and the uncertainties in the objects' position. The several shades of colours in the error ellipsoids represent the 1σ, 2σ and the 3σ ellipsoid. [16]

Figure 6.5 shows an image, published by LEOLabs, of the conjunction with both objects' positions and error ellipsoids, as well as the conjunction data. When running the developed code with the catalog of $2020/10/14$ and object 19 826 as primary, it was found a conjunction with object 36 123. The conjunction data found by the developed collision warning tool is shown in Tables 6.3 and 6.4. Figure 6.6 shows the plots in the conjunction plane for the PC calculation, as well as a plot to better visualize the conjunction geometry.

Analysing the results obtained in Table 6.3 and Fig. 6.6, the TCA and relative velocity are very similar to the ones of LEOLabs, in Table 6.1. The scaling factor, k, is higher than 1 so, the PC is not in a diluted region, meaning that the uncertainties are not too high. Another factor to consider is the size of the objects, which is very high when comparing to the previous conjunctions analysed in Sec. 6.2.1. In this conjunction, HBR is equal to 9.6 m, increasing the danger of the conjunction, because the larger the objects, the more debris can be generated if a collision happens.

The MD obtained, in Table 6.3, is much larger than the one obtained by LEOLabs, Table 6.1. Hence, the PC will be smaller comparing to the one of LEOLabs. The PC obtained is between 1.48×10^{-13} and 2.22×10^{-5} and the one of LEOLabs is 3.80×10^{-2}. This also happens because, in PC calculation, the covariance matrices estimated are different. The standard deviations obtained by the developed method are shown in Table 6.4, which are much larger than the ones used by LEOLabs, in Table 6.2. Comparing

[16] LEOLabs, "This visualization shows our latest information on the event," published on 2020/10/15,Twitter, accessed on 2020/11/24. [Online]. Available: https://twitter.com/LeoLabs_Space/status/1316822400131104768/photo/1

these standard deviations with the ones found for ION-MK01 and the secondary object in Sec. 6.2.1 case 3, the deviations here are much smaller, because these objects orbit in higher altitudes around 1000 km and ION-MK01 altitude is around 500 km more affected by drag, leading to higher uncertainties.

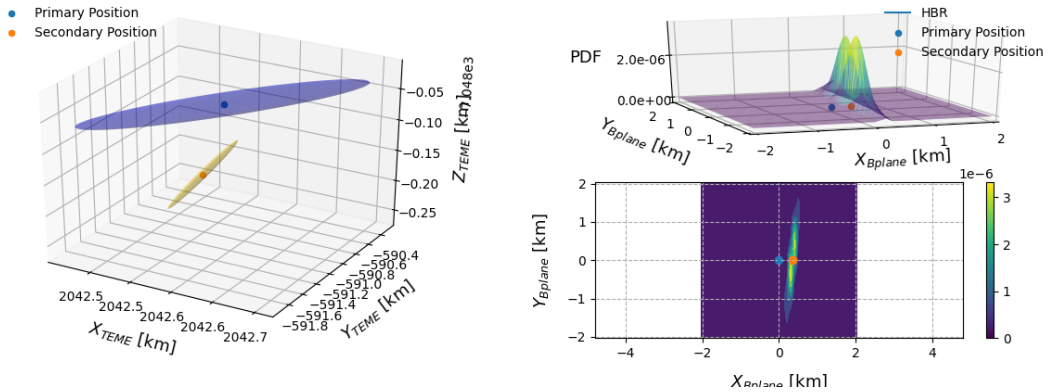
Overall, the differences found between both tools in the conjunction data are due to the type of data used. The developed collision warning tool uses TLEs and LEOLabs uses data of their own radars. This can be verified by the standard deviations obtained with TLEs, wich are much larger, but also in the MD obtained, which is 226 m different from the one of LEOLabs. This fact suggests that other data besides TLEs should be used to have a better assessment of the collision, with lower uncertainties. After the conjunction happened, LEOLabs screened for debris using their radar fence, but it was only found an intact object, indicating that the collision didn't happen.

Table 6.3: Conjunction data found by the developed collision warning tool with the catalog of 2020/10/14 . The parameter k is the scalling factor obtained and used in PC_{max} calculation.

TCA [UTC]	MD [m]	Relative Velocity [km/s]	PC	PC_{max}	k
2020-10-16T00:56:40.761	351	14.659	1.48E-13	2.22E-05	4.82

Table 6.4: Data of the objects involved in the collision. The parameter 'TLEs used' refers to the number of TLEs used to estimate the covariance of the object. The parameters σ_x, σ_y and σ_z are the standard deviations in the reference frame of the conjunction.

	Primary Object	Secondary Object
Norad ID	19826	36123
Type of Object	Payload	Rocket Body
Largest Dimension [m]	2.1	7.5
TLEs used	33	29
σ_x [m]	69	55
σ_y [m]	794	176
σ_z [m]	143	19



(a) Visualization of the error ellipsoids in the RTN frame and the positions of the objects in the TEME frame.

(b) Conjunction plane with the respective PDF for PC calculation.

Figure 6.6: Ellipsoid errors and the PDF on the plane of the conjunction found between 19 826 and 36 123.

6.2.3 Collision between Iridium-33 and Cosmos 2251

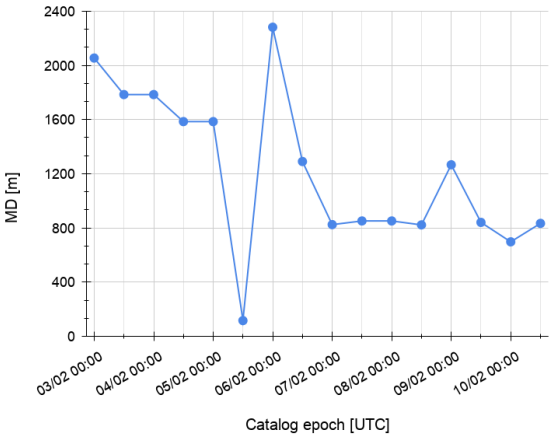
In this test, it will be analysed the collision between Iridium-33 and Cosmos 2251 in 2009, in order to have an idea of the PC threshold to consider a conjunction as a high-risk conjunction, as well as study the evolution of the conjunction data with time.

The collision between these satellites happened at $2009/02/10$ 16:55:59 [UTC] [35]. The TLEs emitted before the collision are available on SpaceTrack, hence, these were retrieved and fed into the developed collision warning tool. Several conjunction analysis were made, two times per day, starting from seven days before the collision, using the more recent catalog at noon and midnight of each day. The TCA, relative velocity, scaling factor (k), MD, PC and PC_{max} were calculated and the evolution with time of the last three is shown in Fig. 6.7.

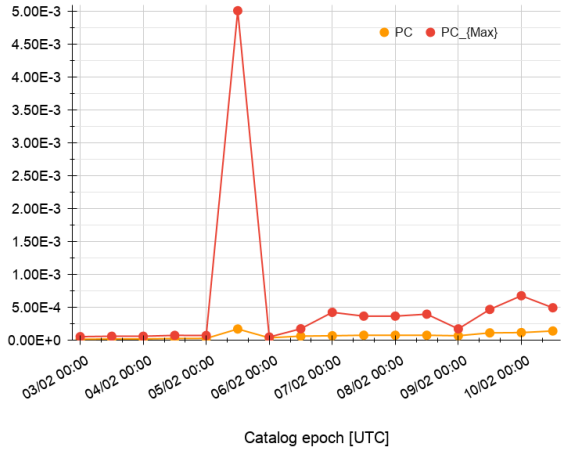
The TCAs obtained vary between $2009/02/10$ 16:55:59.662 [UTC] and $2009/02/10$ 16:55:59.928 [UTC]. The relative velocities vary between $11\,646.66\,\mathrm{m/s}$ and $11\,647.33\,\mathrm{m/s}$, which are very small variations and, in the case of TCA, similar to the one when the collision occurred. Hence, the TCA and the relative velocity are reliable parameters, even for predictions of seven days.

In Fig. 6.7(a), the MD varies between $117\,\mathrm{m}$ and $2282\,\mathrm{m}$, during the week before the collision. In spite of being possible to identify a descendent behaviour of MD with time, there are some outliers with large and small MDs. Since MD is related to PC, intuitively, when MD decreases, PC increases and that is shown by the outlier of $05/02$ 12:0, where MD decreases to $117\,\mathrm{m}$, causing a peak in PC and PC_{max}. PC varies between 1.12×10^{-5} and 1.68×10^{-4} and PC_{max} between 4.51×10^{-5} and 5.01×10^{-3}.

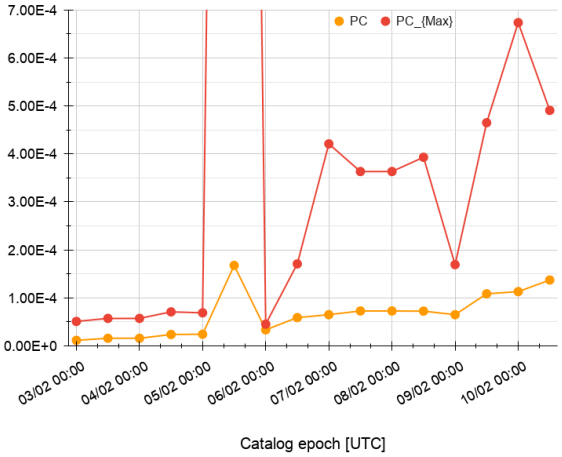
The variations of these parameters are due to the different TLEs that were emitted along that week, revealing low consistency for a good conjunction analysis. The scaling factor, k, was also calculated for each simulation, always with values smaller than 1, indicating that more accurate data needs to be used to reduce the uncertainties in position. However, this analysis suggests that even for PCs in the order of 10^{-5}, the conjunctions can be considered as high-risk conjunctions.



(a) Evolution of MD with time.



(b) Evolution of PC and PC_{max} with time. PC is represented in orange and PC_{max} in red.



(c) Zoom of Fig. 6.7(b) containing the evolution of PC and PC_{max} with time. PC is represented in orange and PC_{max} in red.

Figure 6.7: Evolution of MD, PC and PC_{max} with time, for the collision between Iridium-33 and Cosmos 2251. These calculations were performed with the catalogs of noon and midnight after 2009/02/03, until the day of the collision 2009/02/10.

7

Conclusions

7.1 Achievements

One of the objectives, from the point of view of the client, was to integrate the developed tool in their mission control software, Aurora. The developed tool is completely independent of Aurora, meaning that Aurora only needs to have access to the conjunction data messages generated by the developed collision warning tool and the TLEs of the objects involved in order to display a table with the conjunction data and a simulation of the conjunction. The initial objective of generating a warning between a chosen object and a threatening object was achieved, as well as all the processes needed to select the dangerous objects, generate covariances and calculate PCs allowing the operator to analyse conjunctions and assess the risk of collision of ION-MK01 with other objects.

The filtering techniques studied were implemented with success, allowing to run the developed code in a reasonable time and predict the ION-MK01 conjunctions for the next 7 days. For example, when predicting the conjunctions for 2 hours in the future, without using the filtering techniques (brute force method), this took 28 620 seconds (approximately 8 hours) to run and, with filtering techniques, it took 65 seconds. This reveals, in this case, a 99.8 % reduction in the execution time and the importance of using

filtering techniques to detect conjunctions. The efficiency achieved with filtering techniques allowed to reduced the processing power and the cost for the algorithms to run in a web-based environment.

The covariance generation process implemented revealed that the uncertainties when using TLEs are very large, especially for LEO objects on the in-track direction, suggesting that other type of data should be fed into the code, in order to have more accuracy in the calculated positions. In spite of that, this method allowed to generate a covariance to be used in the PC calculation.

The implementation of the PC calculation allowed to perform several conjunctions analysis and assess if the conjunction is dangerous or not, helping in the decision of performing an avoidance maneuver. The extreme cases studied to test the implementation of PC gave the expected results and the comparison with LEOLabs conjunction revelead, once again, the limitations of using TLEs, since the PC obtained was much smaller than the one obtained by LEOLabs . The detection of the collision between Iridium-33 and Cosmos 2251 allowed to set a PC threshold to consider a conjunction has a high-risk conjunction. The found threshold was 10^{-5}, meaning that a conjunction with PC higher than this should be considered as a high-risk conjunction.

7.2 Future Work

TLEs are the largest source of inaccuracy in the developed tool. The future work would involve to change the source of data from TLEs to ephemerides in an SP catalog, provided by SpaceTrack to satellite operators. Using this SP catalog will involve work with orbit determination processes and numerical propagators, that can be much more accurate than TLEs and SGP4. However, the time of execution is a disadvantage when using numerical propagators. Propagating all objects in the catalog with numerical propagators, even with the filtering techniques implemented, would take more time. Hence, the SP catalog may complement the developed tool, to better assess the conjunctions found. This way, it would only be necessary to propagate the threatening objects found by the developed tool, reassessing the conjunctions found with better data.

Another point that can be modified in the developed tool is to use the GPS data of the primary object, in this case ION-MK01, available through Aurora. With the GPS data of the primary object, it would be possible to have better orbit and covariance information. This was not done in this thesis in order to develop a tool completely independent of Aurora. Theoretically, the results using GPS data would be even better than using the data of the SP catalog. By using this, the collision warning tool would be iterative with mission operations, meaning that if a manoeuvre is performed, it would be recognisable in the GPS data. However, for this tool to be used in an operational context, it needs to pass for a more extensive test process.

In terms of implementation of the algorithm, an improvement would be the in the ephemerides gen-

eration. The algorithm developed stores all positions for all objects along the configured propagation time interval. Another solution that would not require as much memory as the implemented one, would be to make the application of the sieves in smaller time spans, removing the necessity of storing all the positions of the objects through all the configured duration of propagation.

In the future, when more accurate orbital data is fed into the system, the conjunction data will be more accurate, allowing to suggest an avoidance maneuver, upgrading this tool from a collision warning tool to a collision warning and avoidance tool.

Bibliography

[1] H. Klinkrad, *Space Debris: models and risk analysis*, 1st ed. Springer-Verlag Berlin Heidelberg, 2006.

[2] "Space Debris User Portal: Space Environment Statistics," ESA, accessed on 2020/08/06. [Online]. Available: https://sdup.esoc.esa.int/discosweb/statistics/

[3] D. A. Vallado, P. Crawford, R. Hujsak, and T. S. Kelso, "Revisiting Spacetrack Report #3," *AIAA/AAS 2006-6753 Astrodynamics Specialist Conference*, pp. 1–88, 2006. [Online]. Available: https://doi.org/10.2514/6.2006-6753

[4] F. R. Hoots, L. L. Crawford, and R. L. Roehrich, "An Analytic Method To Determine Future Close Approaches Between Satellites." *Celestial Mechanics*, vol. 33, pp. 144–158, 1984. [Online]. Available: https://doi.org/10.1007/BF01234152

[5] J. R. Rodríguez, F. M. Fadrique, and H. Klinkrad, "Collision Risk Assessment with a 'Smart Sieve' Method," in *Joint ESA-NASA Space-Flight Safety Conference 2002, Special Publication*, B. Battrick and C. Preyssi, Eds., vol. 486. ESA, Jul. 2002, pp. 159–164.

[6] D. A. Vallado, "Covariance Transformations for Satellite Flight Dynamics Operations," *Astrodynamics Specialist Conference*, pp. 1–35, AIAA/AAS Paper 03-526, 2003.

[7] *Spaceflight Safety Handbook for Satellite Operators, Version 1.4*, 18th Space Control Squadron, Combined Force Space Component Command, Vandenberg Air Force Base, California, USA Feb. 2019. [Online]. Available: https://www.space-track.org/documents/Spaceflight_Safety_Handbook_for_Operators.pdf

[8] D. J. Kessler, N. L. Johnson, J. C. Liou, and M. Matney, "The Kessler Syndrome: Implications to Future Space Operations," *Advances in the Astronautical Sciences*, vol. 137, no. 8, p. 2010, Jan. 2010.

[9] S. J. Setty, P. J. Cefola, O. Montenbruck, and H. Fiedler, "Application of Semi-analytical Satellite Theory Orbit Propagator to Orbit Determination for Space Object Catalog Maintenance,"

Advances in Space Research, vol. 57, no. 10, pp. 2218–2233, Mar. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.asr.2016.02.028

[10] M. J. Nicolls and D. Mcknight, "Collision Risk Assessment for Derelict Objects in Low-Earth Orbit," in *First International Orbital Debris Conference*, Paper no. 6096, 2019.

[11] E. Kahr, O. Montenbruck, and K. P. O'Keefe, "Estimation and Analysis of Two-Line Elements for Small Satellites," *Journal of Spacecraft and Rockets*, vol. 50, no. 2, pp. 433–439, Mar. 2013. [Online]. Available: https://doi.org/10.2514/1.A32352

[12] F. Schiemenz, J. Utzmann, and H. Kayal, "Survey of the operational state of the art in conjunction analysis," *Council of European Aerospace Societies (CEAS) Space Journal*, vol. 11, no. 3, pp. 255–268, Feb. 2019. [Online]. Available: http://dx.doi.org/10.1007/s12567-019-00242-2

[13] J. Geul, E. Mooij, and R. Noomen, "TLE Uncertainty Estimation using Robust Weighted Differencing," *Advances in Space Research*, vol. 59, no. 10, pp. 2522–2535, Mar. 2017. [Online]. Available: https://doi.org/10.1016/j.asr.2017.02.038

[14] S. Shen, X. Jin, and C. Hao, "Cleaning Space Debris with a Space-based Laser system," *Chinese Journal of Aeronautics*, vol. 27, no. 4, pp. 805–811, Mar. 2014. [Online]. Available: https://doi.org/10.1016/j.cja.2014.05.002

[15] J. Reed and S. Barraclough, "Development of Harpoon System for Capturing Space Debris," in *6th European Conference on Space Debris*, L. Ouwehand, Ed., vol. 6, no. 1. ESA SP-723, Apr. 2013. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/sdc6/paper/27/SDC6-paper27.pdf

[16] ESA Space Debris Office, "ESA's Annual Space Environment Report," Tech. Rep. 3.2, Jul. 2019. [Online]. Available: https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf

[17] F. R. Hoots and R. L. Roehrich, "Spacetrack Report No. 3–Models for Propagation of NORAD Elements Sets," Spacetrack, Tech. Rep. 3, Dec. 1980.

[18] D. Brouwer, "Solution of the Problem of Artificial Satellite Theory without Drag," *The Astronomical Journal*, vol. 64, no. 1274, pp. 378–396, Oct. 1959. [Online]. Available: https://doi.org/10.1086/107958

[19] Y. Kozai, "The Motion of a Close Earth Satellite," *The Astronomical Journal*, vol. 64, no. 1274, pp. 367–377, Oct. 1959. [Online]. Available: https://doi.org/10.1086/107957

[20] R. H. Lyddane, “Small Eccentricities or Inclinations in the Brower Theory of the Artificial Satellite,” *The Astronomical Journal*, vol. 68, no. 8, pp. 555–558, Oct. 1963. [Online]. Available: https://doi.org/10.1086/109179

[21] F. R. Hoots, P. W. Schumacher, and R. A. Glover, “History of Analytical Orbit Modeling in the U.S. Space Surveillance System,” *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 2, pp. 174–185, Mar. 2004. [Online]. Available: https://doi.org/10.2514/1.9161

[22] K. Cranford and M. Lane, “An Improved Analytical Drag Theory for the Artificial Satellite Problem,” *Astrodynamics Conference*, AIAA/AAS Paper 69-925, Aug. 1969. [Online]. Available: https://doi.org/10.2514/6.1969-925

[23] F. R. Hoots, “Theory of the Motion of an Artificial Earth Satellite,” *Celestial Mechanics*, vol. 23, no. 4, pp. 307–363, Apr. 1981. [Online]. Available: https://doi.org/10.1007/BF01230744

[24] J. Woodburn, V. Coppola, and F. Stoner, “A Description of Filters for Minimizing the Time Required for Orbital Conjunction Computations,” in *Advances in the Astronautical Sciences*, vol. 135, no. 2, AAS Paper 09-372 , 2009, pp. 1157–1173.

[25] L. M. Healy, “Close Conjunction Detection on Parallel Computer,” *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 4, pp. 824–829, Jul. 1995. [Online]. Available: https://doi.org/10.2514/3.21465

[26] T. S. Kelso and S. Alfano, “Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES),” in *Modeling, Simulation, and Verification of Space-based Systems III*, Paper no. 6221, May 2006. [Online]. Available: https://doi.org/10.1117/12.665612

[27] D. A. Vallado and P. J. Cefola, “Two-line Element Sets - Practice and Use,” in *63rd International Astronautical Congress*, vol. 7, Paper IAC-12-A6.6.11, Jan. 2012, pp. 5812–5825.

[28] S. Alfano and D. Oltrogge, “Probability of Collision: Valuation, variability, visualization, and validity,” *Acta Astronautica*, vol. 148, pp. 301–316, 2018. [Online]. Available: https://doi.org/10.1016/j.actaastro.2018.04.023

[29] V. P. Osweiler, “Covariance Estimation and Autocorrelation of NORAD Two-line Element Sets,” Master’s thesis, Air Force Institute of Technology, Mar. 2006. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a446817.pdf

[30] T. Flohrer, H. Krag, and H. Klinkrad, “Assessment and Categorization of TLE Orbit Errors for the US SSN Catalogue,” in *Advanced Maui Optical and Space Surveillance Technologies Conference*, S. Ryan, Ed., Paper E53, Sep. 2008.

[31] S. Alfano, “Relating Position Uncertainty to Maximum Conjunction Probability,” *The Journal of the Astronautical Sciences*, vol. 53, no. 2, pp. 193–205, 2005. [Online]. Available: https://doi.org/10.1007/BF03546350

[32] “Database and Information System Characterising Objects in Space (DISCOS),” maintained by ESA, accessed on 2020/11/22. [Online]. Available: https://discosweb.esoc.esa.int/

[33] D. P. McKinley, “Development of a Nonlinear Probability of Collision Tool for the Earth Observing System,” in *Astrodynamics Specialist Conference*, AIAA/AAS Paper 2006-6295, Aug. 2006. [Online]. Available: https://doi.org/10.2514/6.2006-6295

[34] N. Bérend, “Estimation of the Probability of Collision Between Two Catalogued Orbiting Objects,” *Advances in Space Research*, vol. 23, no. 1, pp. 243–247, 1999. [Online]. Available: https://doi.org/10.1016/S0273-1177(99)00009-5

[35] T. S. Kelso, “Analysis of the Iridium-33 Cosmos 2251 Collision,” in *Advanced Maui Optical and Space Surveillance Technologies Conference*, Sep. 2009.

A

TLE Format

TLEs can be used to propagate orbits with SGP4 and SDP4 theory. They contain mean orbital elements obtained by removing periodic perturbations. These perturbations are then added by SGP4/SDP4 theory to obtain state vectors (position and velocity) of objects that have TLEs routinely updated. These mean orbital elements are publicly available and stored in the format of TLEs. [17]

The generation of TLEs is not based on an established timetable for all type of objects, the rate of update depends on the orbit, object type and maneuverability. A satellite in LEO may have its TLEs updated several times a day because of the slightly unpredictable results of atmospheric drag. A satellite in GEO might only need updates once or twice a week. Space debris won't be updated as frequently either, unless there is a prediction of a close approach with an operational payload. [17]

TLE data comes in a standard format. which can be seen in Fig. A.2, where a TLE of the ISS is shown and the data included in it. It is possible to chose whether or not the TLE comes with the name of the object in a third line. Fig. A.2 presents the format without the name of the object. TLE has two lines with 69 characters, containing the orbital elements of an object, among other data. It has some valid

[17] T. S. Kelso, "FAQs: Two-Line Element Set Format," Celestrak, Dec. 2019, accessed on 2020/08/10. [Online]. Available: http://celestrak.com/columns/v04n03/

characters for specific positions, like numbers between 0 and 9, capital letters between A and Z, period (.),space ( ) and the minus (-) and plus (+) signs. Fig. A.1 represents the specific position for each type of character, where columns with a period, plus or minus sign and space can have no other character.

```
1 NNNNNU NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNN
```

Figure A.1: Specific positions for each type of character in TLEs. 'N' means that the column can have any number between 0 and 9 or a space. 'A' means that the column can have any letter between A and Z or a space. The letter 'U' is in the position that classifies the TLE as unclassified or classified data. Since TLEs are publicly available, that position always takes the letter 'U' for unclassified. The plus sign can have a minus sign or a space and vice-versa.

Referring to Fig. A.2, the object number field is generated by NORAD and it's a unique ID for each cataloged object. The maximum number this field can handle is 99 999, which will become a problem in the future when more than 99 999 objects are cataloged.

Object numbers between 70 000 and 90 000 are analyst objects, with insufficient fidelity for publication in the official catalog. The lack of fidelity may be due to infrequent tracking, cross-tagging (observation association with closely-spaced objects) or inability to associate the object with a known launch. When this is solved, the analyst object ID changes to an ID between 1 and 69 999. Currently, the new tracked objects have an ID around the 47 000, but with the growth of space activity and tracking capabilities it is expected to reach the limit of the ID numbers soon.

The launch year is designated by the two last numbers of the year. Hence, 1900 should be added to the number, if it's larger than or equal to 57 (the year of Sputnik launch), otherwise, 2000 should be added. This means that, before 2057, TLE format needs to be changed. The launch number is incremented in every launch performed and reset every year. The maximum launch number handled by a TLE is 999, which is sufficient since there are approximately 100 launches per year, currently. The piece of the launch field indicates the piece connected to that launch. 'A' refers to payload, 'I' and 'O' are not used so, a total of 13 824 pieces are supported by the format.

The epoch year field refers to the year that the TLE was emitted and the epoch day to the day of that year. These epochs are measured in mean solar time units, not sideral time units.

The first derivative of motion field is divided by two in units of revolutions per day^2, the second derivative of motion field presented is divided by six, in units of revolutions per day^3. These fields are not used by the SGP4/SDP4 theory, but they were used before in the SGP/SDP, as explained in 3.1.1.A.

The B^* drag term field represents a drag coefficient used in SGP4 theory. It's a different version of the ballistic coefficient (B) of an object, given by $B = C_D A/m$, where C_D is the drag coefficient of the object, A the area and m its mass. This parameter assimilates force model errors and sometimes it's negative, implying wrongly that energy is being added to the system, which reveals the limitations of SGP4 in force modelling. B^* is defined as $B^* = B\rho_0/2$, where ρ_0 is the atmospheric density at 1 atm.

B^* has units of Earth radii^{-1} .[18]The format of this field is a modified exponential notation with an implied leading decimal point. For example, the value $-34\,255 - 7$ is equivalent to $-0.342\,55 \times 10^{-7}$.

The inclination field can have values between $0°$ and $180°$. The Right Ascension of the Ascending Node (RAAN), the argument of perigee and the mean anomaly fields can range from $0°$ up to $360°$. The eccentricity field is a unitless field and assumes a leading decimal point. The Mean Motion field is measured in revolutions per day.

The ephemeris type field indicates the orbital model used to generate the TLE. Currently, all TLEs come with the value 0, because they are all generated through the SGP4/SDP4 model. The TLE number field is incremented when a new TLE is generated. The revolution number field represents the number of revolutions of the object since its launch. The checksum field is calculated by ignoring all letters, spaces, plus signs, assigning a value of 1 to all minus signs and add the values of all numbers in that line. The check number is the last number of that sum result. All TLE data provided in SpaceTrack passes the checksum test. [18]

[18]T. S. Kelso, "FAQs: Two-Line Element Set Format," Celestrak, Dec. 2019, accessed on 2020/08/10. [Online]. Available: http://celestrak.com/columns/v04n03/

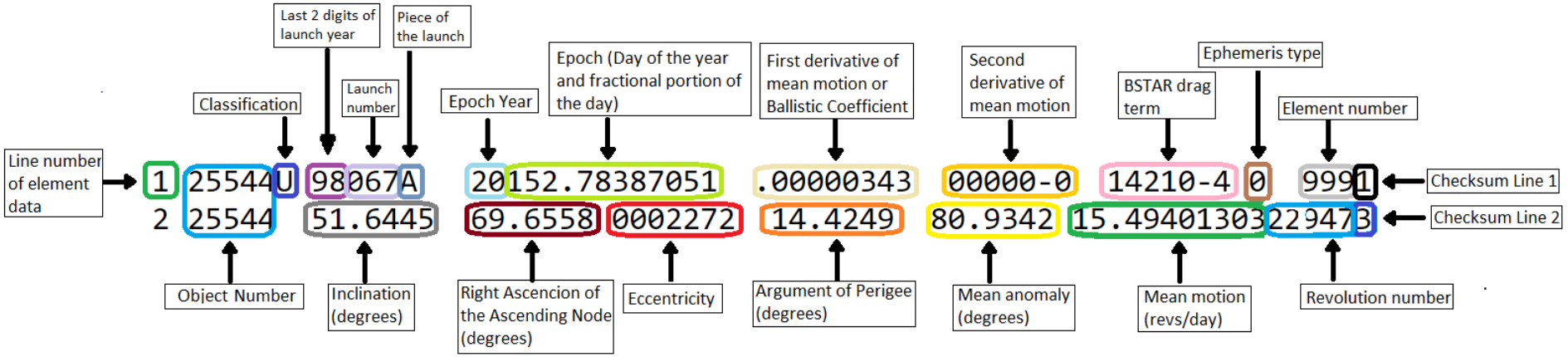
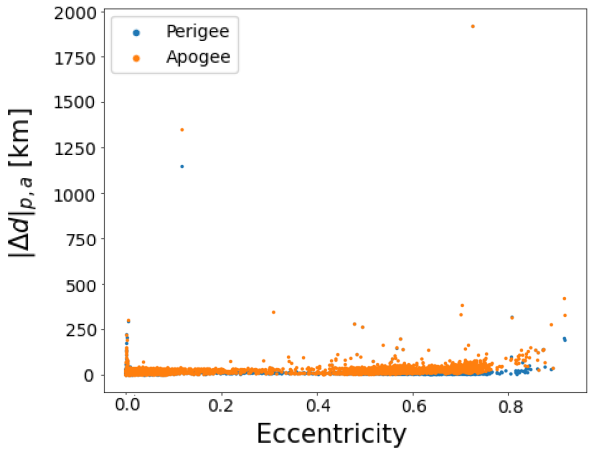


Figure A.2: TLE example of ISS with the description of each field.

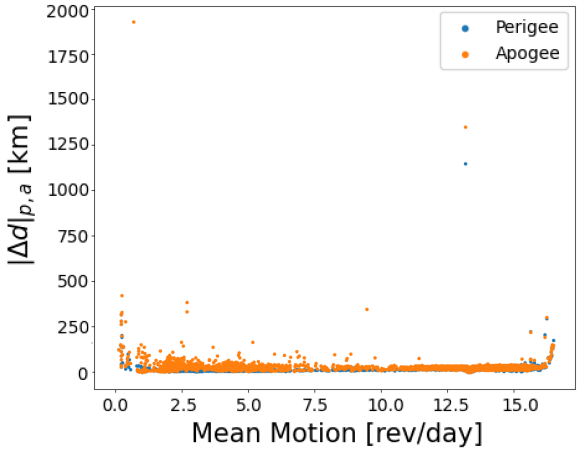
B

Maximum Variation of Perigee and Apogee Plots

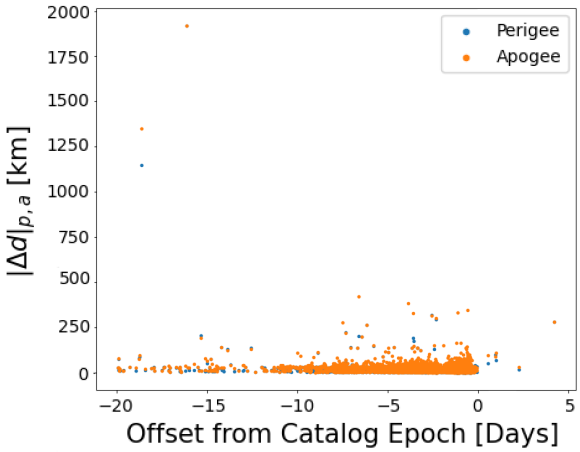
The following figures refer to Sec. 4.3.1 and the maximum variation of apogee and perigee plotted against several variables, for 1, 3, 6 and 10 days, using the catalogs of 2020/03/24, 2020/08/20 and 2020/08/21.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

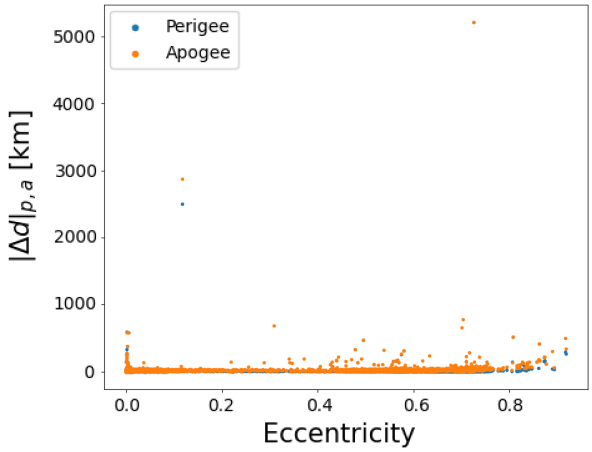


(b) Maximum variation of perigee and apogee plotted against mean motion.

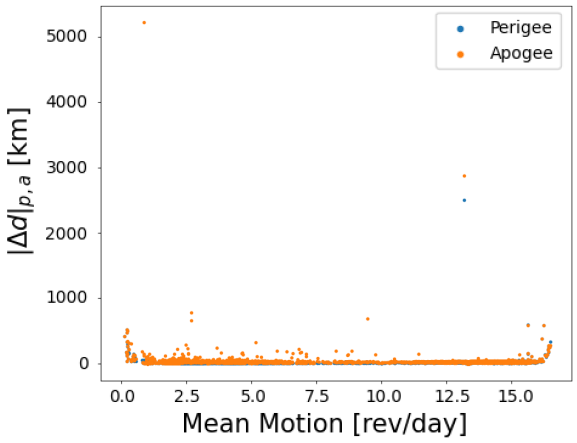


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

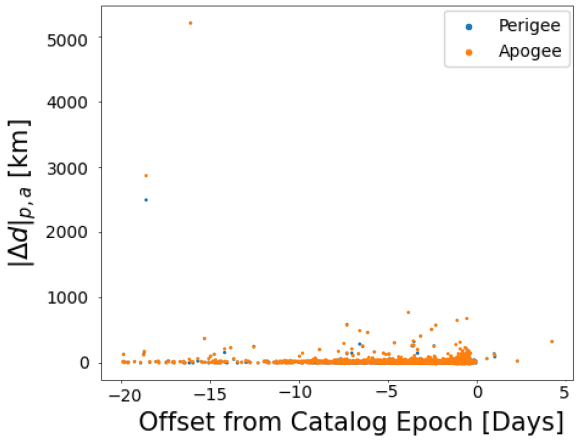
Figure B.1: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/03/24 and a propagation time of 3 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

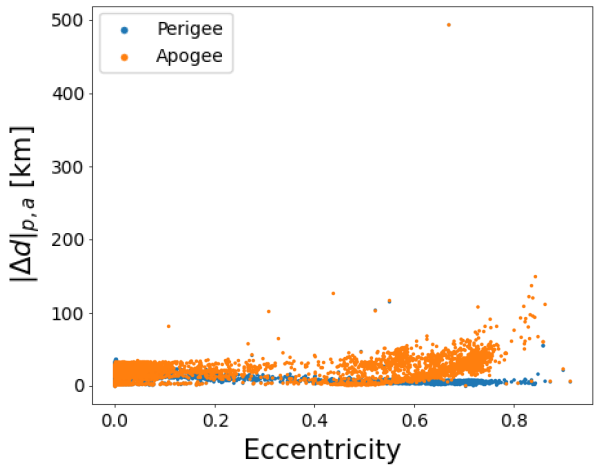


(b) Maximum variation of perigee and apogee plotted against mean motion.

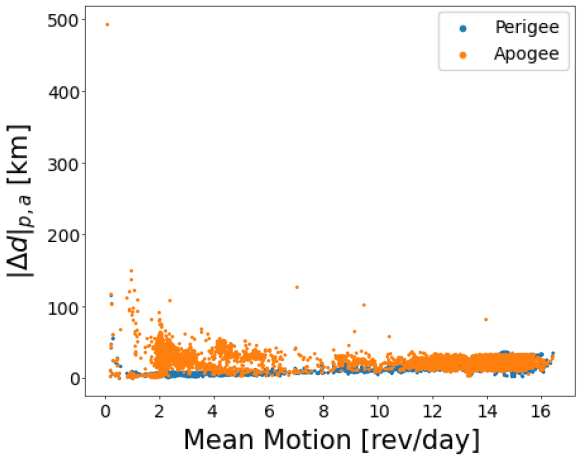


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

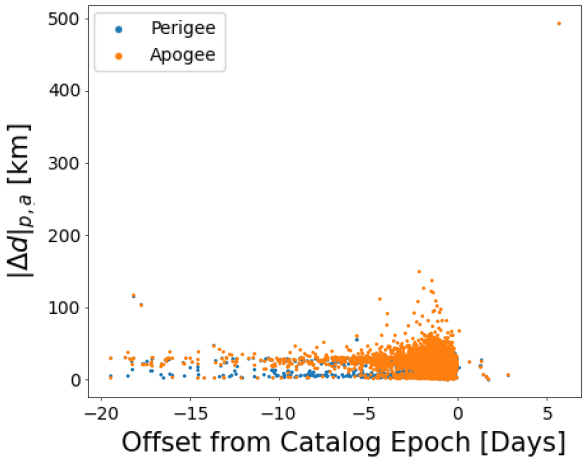
Figure B.2: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/03/24 and a propagation time of 6 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

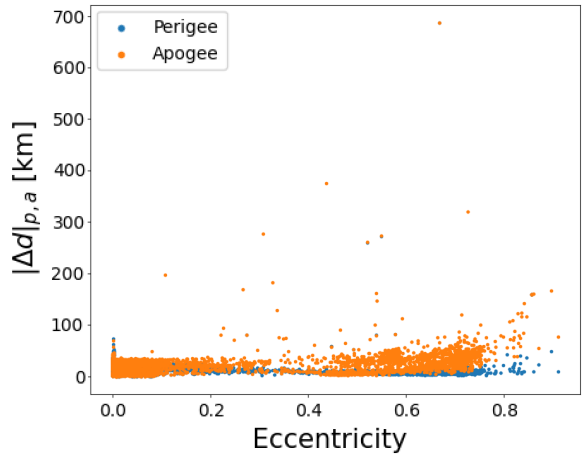


(b) Maximum variation of perigee and apogee plotted against mean motion.

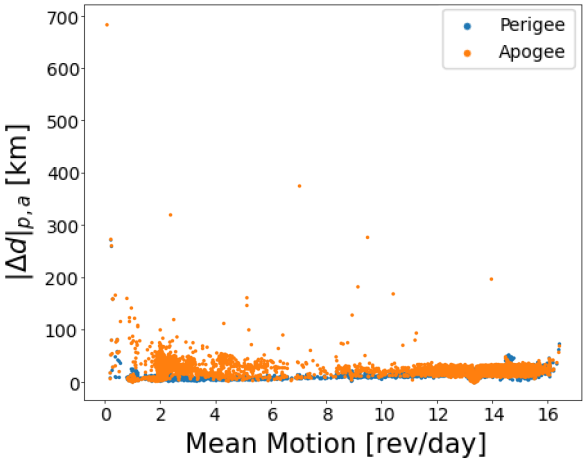


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

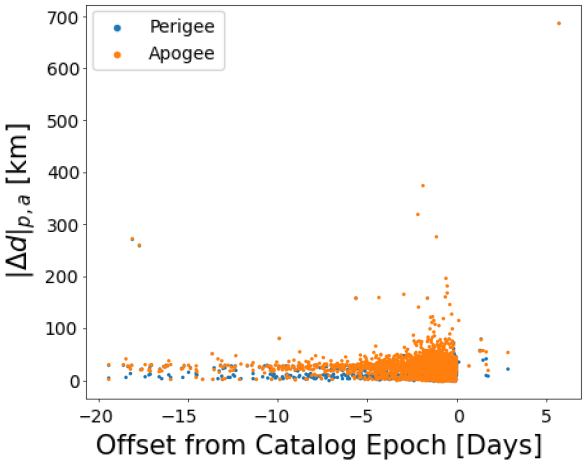
Figure B.3: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/20 and a propagation time of 1 day. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

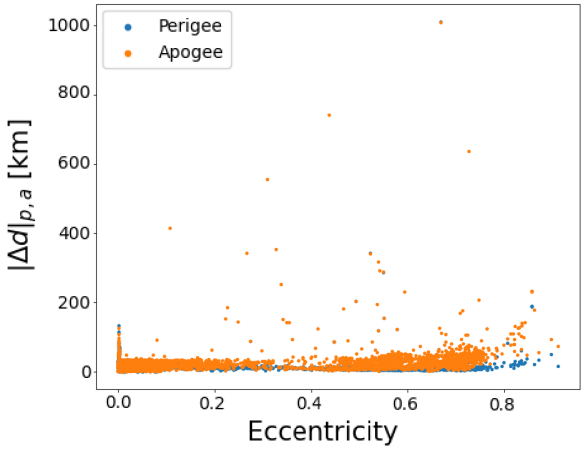


(b) Maximum variation of perigee and apogee plotted against mean motion.

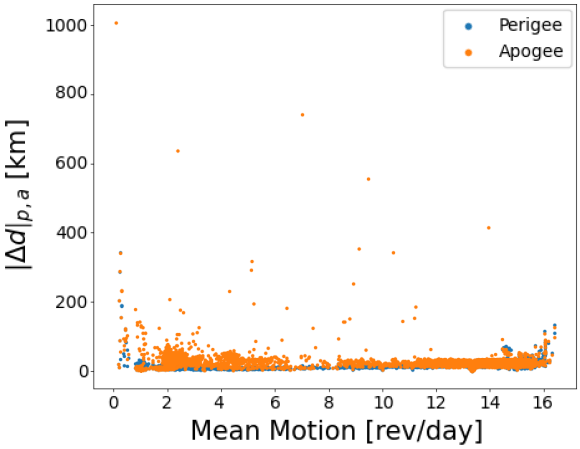


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

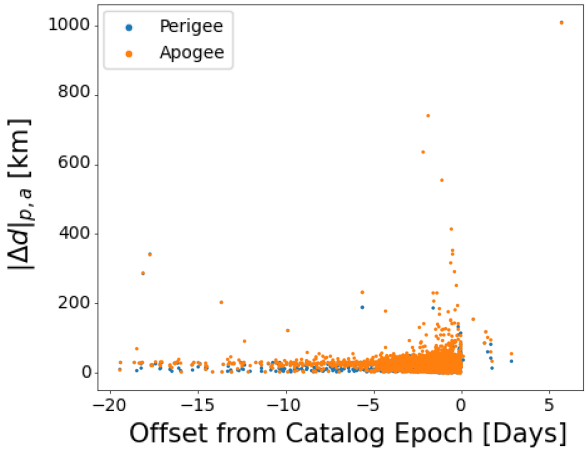
Figure B.4: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/20 and a propagation time of 3 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

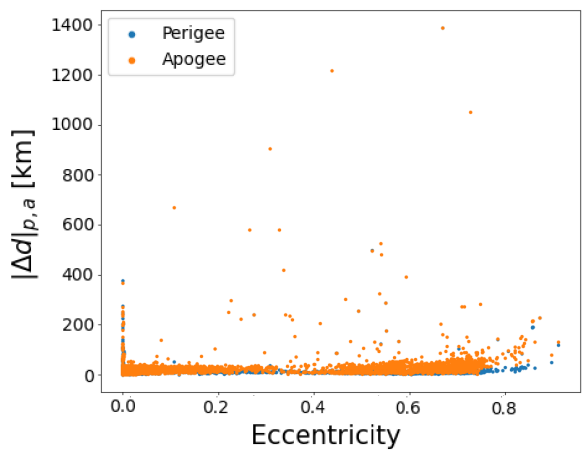


(b) Maximum variation of perigee and apogee plotted against mean motion.

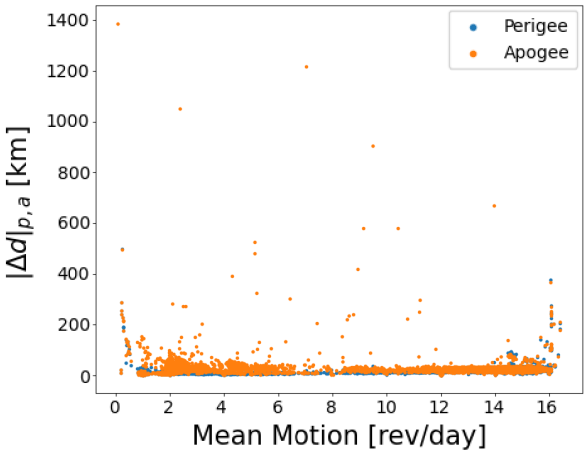


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

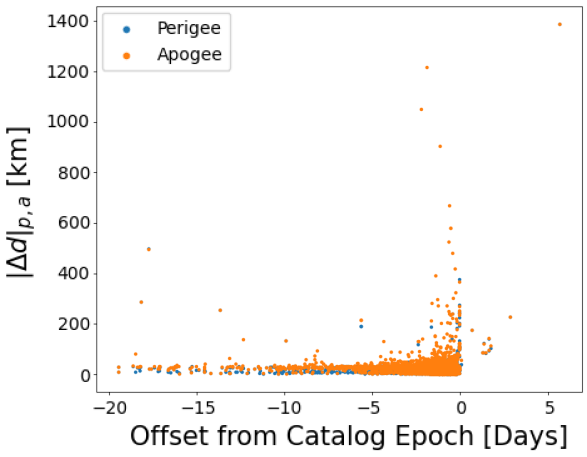
Figure B.5: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/20 and a propagation time of 6 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

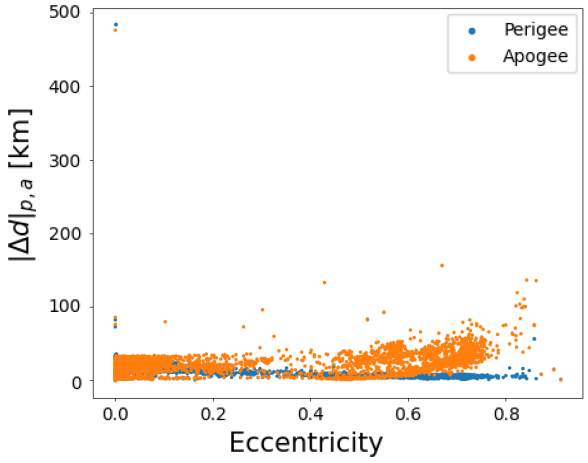


(b) Maximum variation of perigee and apogee plotted against mean motion.

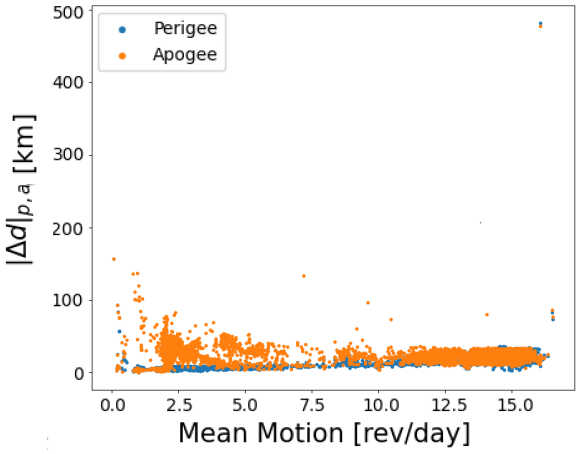


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

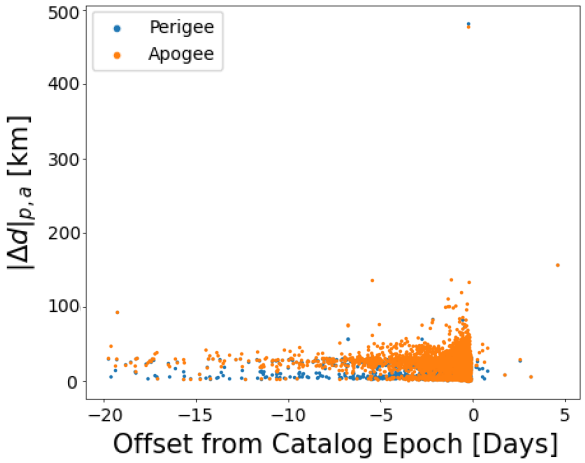
Figure B.6: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/20 and a propagation time of 10 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

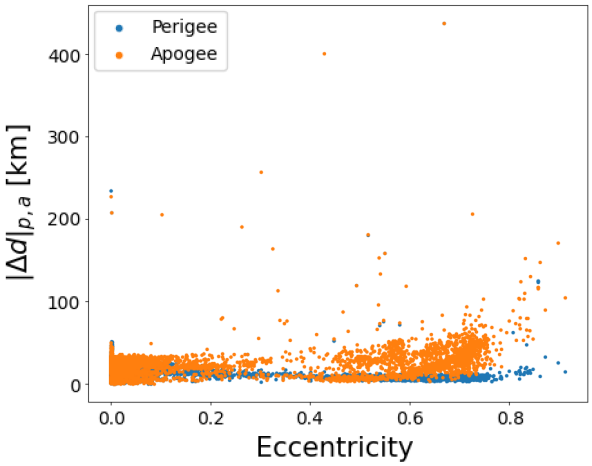


(b) Maximum variation of perigee and apogee plotted against mean motion.

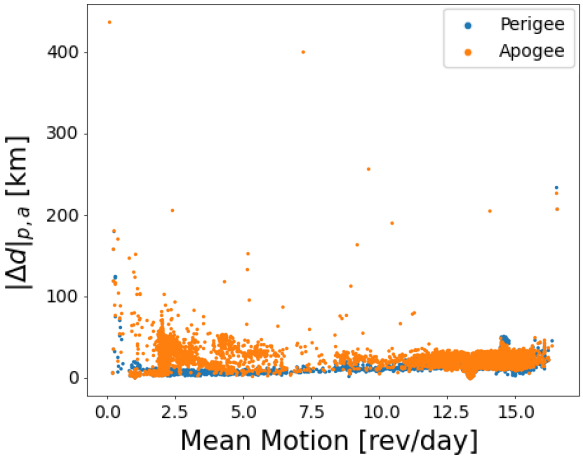


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

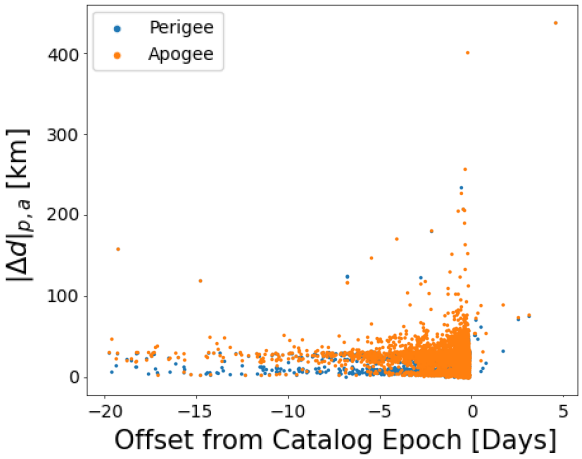
Figure B.7: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/21 and a propagation time of 1 day. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

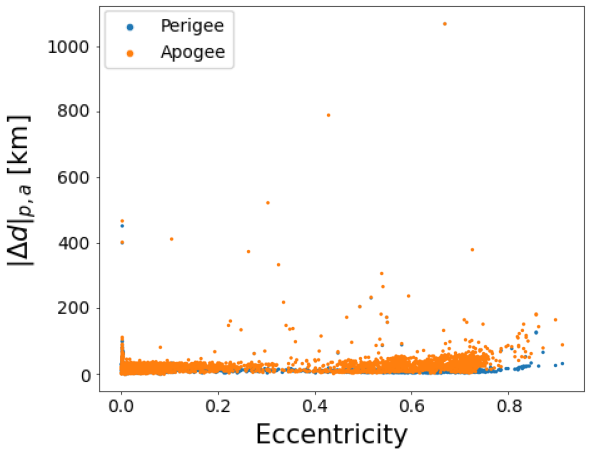


(b) Maximum variation of perigee and apogee plotted against mean motion.

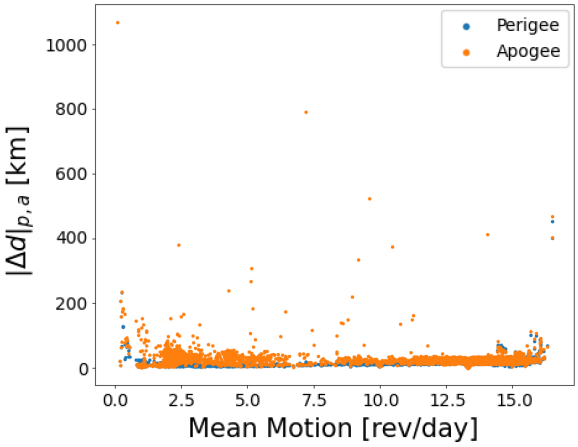


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

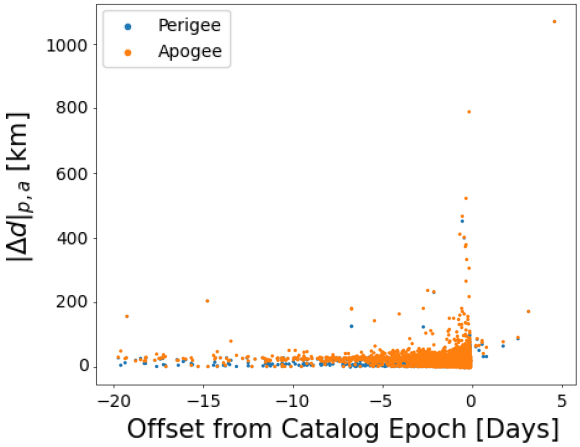
Figure B.8: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/21 and a propagation time of 3 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.

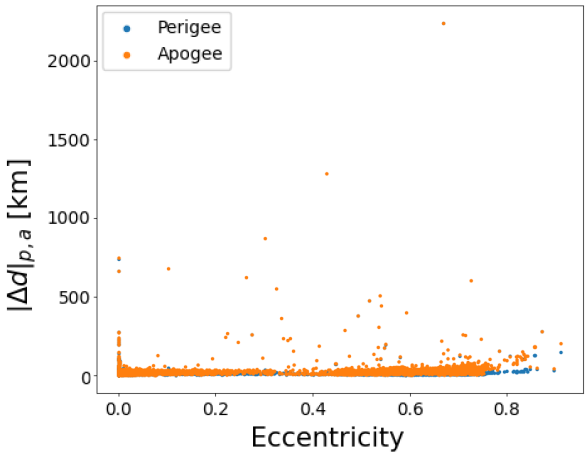


(b) Maximum variation of perigee and apogee plotted against mean motion.

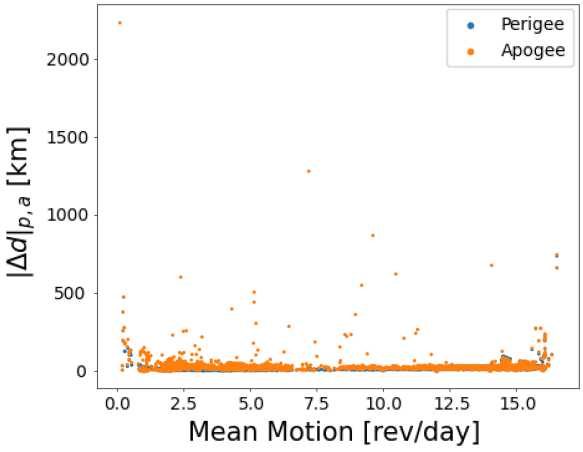


(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

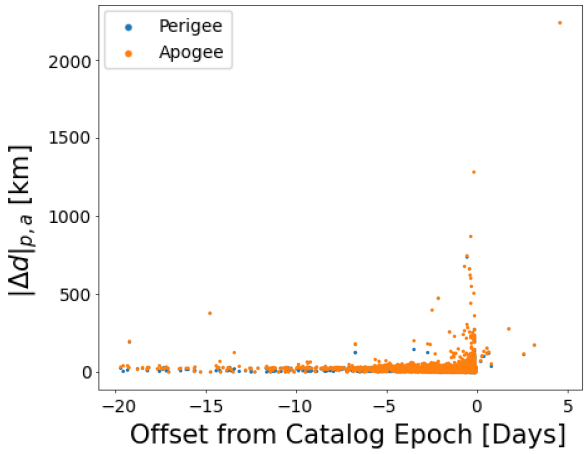
Figure B.9: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/21 and a propagation time of 6 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.



(a) Maximum variation of perigee and apogee plotted against eccentricity.



(b) Maximum variation of perigee and apogee plotted against mean motion.



(c) Maximum variation of perigee and apogee plotted against the offset from the catalog epoch.

Figure B.10: Maximum variation of the perigee and apogee for all space objects in the catalog of 2020/08/21 and a propagation time of 10 days. The orange dots correspond to the apogee maximum variation and the blue dots to the perigee maximum variation.