Abstract

perché

Acknowledgments

Thanksss

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AFSPC   Air Force Space Command

API   Application Programming interface

ARES   Assessment of Risk Event Statistics

BC   Ballistic Coefficient

COESA   Committee On Extension to the Standard Atmosphere

CROC   Cross-section of Complex bodies

DMU   Drag Make Up

DOPRI   Dormand-Prince

DRAMA   Debris Risk Assessment and Mitigation Analysis

ECEF   Earth-Centered Earth-Fixed

ECI   Earth-Centered Inertia

EOL   End-Of-Life

ESA   European Space Agency

GEO   Geostationary Orbit

GMAT   General Mission Analysis Tool

HF2	Hyperfield 2
IAM	Inclination Adjust Maneuvers
JB2008	Jacchia-Bowman 2008
JIT	Just-In-Time
LEO	Low Earth Orbit
MASTER	Meteoroid and Space Debris Terrestrial Environment Reference
MIDAS	MASTER-based Impact flux and Damage Assessment Software
MIT	Massachusetts Institute of Technology
MLTAN	Mean Local Time of the Ascending Node
NASA	National Aeronautics and Space Administration
OSCAR	Orbital SpaceCraft Active Removal
PSW	Potential Swath Width
RAAN	Right Ascension of the Ascending Node
RTC	Revisit Time Collector
SARA	re-entry Survival And Risk Analysis
SRP	Solar-Radiation Pressure
SSO	Sun-Synchronous Orbit
TLE	Two-Line Element
UTC	Coordinate Universal Time
VIS-NIR	Visible-to-Near-Infrared
VIS-SWIR	Visible-to-Short-Wave-Infrared

Chapter 1

Introduction

1.1 Earth Observation and Remote Sensing

1.1.1 Hyperspectral imaging

1.2 Kuva Space

1.2.1 Company Vision

1.2.2 Infrastructure

1.2.3 Distribution

1.2.4 Applications

1.3 Thesis Purpose

1.4 Organization

Chapter 2

Background

This chapter aims to provide an overview on the fundamentals of Space Flight Dynamics that constitute the theoretical basis of this thesis, with a specific focus on Earth Observation applications in Low Earth Orbit (LEO), as well as a literature review on orbit management methods addressed by this work.

2.1 Space Flight Dynamics Introduction

Orbital Mechanics, Astrodynamics, Astronautics and Space Flight Dynamics are all titles of university courses whose principal topic is two-body orbital motion, that involves orbit determination, orbital flight time, and orbital maneuvers [1]. In this context, a proper definition of the subject can be the following: the study of the motion of man-made objects in space, subject to both natural and artificially induced forces [2].

2.1.1 Satellite State Representations

To define the *state* of a satellite in space six quantities are required, and they may take on many equivalent forms. Whatever the form, the collection is called either a *state vector*, usually associated with position and velocity vectors, or an *element*

set, typically used with scalar magnitude and angular representations of the orbit called *orbital elements*. Either set of quantities completely specify the two-body orbit and provide a complete set of initial conditions for solving an initial value problem class of differential equations. Time is always associated with a state vector and is often considered a seventh component. State vectors and element sets are referenced to a particular coordinate frame [3].

This thesis will use both state vectors and a specific element set. The latter, which is also the most common one in this field, is represented by the ***classical orbital elements*** [4]:

- Semimajor axis, a: the orbit size is defined by one half of the major axis dimension

- Eccentricity, e: the ratio of minor to major dimensions of an orbit defines the shape

- Inclination, i: the angle between the orbit plane and the reference plane or the angle between the normals to the two planes

- Longitude of the ascending node or Right Ascension of the Ascending Node (RAAN), Ω: the angle between the vernal equinox vector and the ascending node measured in the reference plane in a couterclockwise direction as viewed from the northern hemisphere

- Argument of periapsis, ω: the angle from the ascending node to the periapsis, measured in the orbital plane in the direction of spacecraft motion. The ascending node is the point where the spacecraft crosses the reference plane headed from south to north. the line of nodes is the line formed by the intersection of the orbit plane and the reference plane

- True anomaly, ν: the sixth element locates the spacecraft position on the orbit

This thesis work makes also use of the *argument of latitude, u*, which is the

angle measured between the ascending node and the satellite's position vector in the direction of satellite motion. A relation that is always valid exists between classical orbital elements and the argument of latitude [3]:

$$u = \omega + \nu \tag{2.1}$$

One more element that is often used is the *mean anomaly, M*, which is the angle between the periapsis of an orbit and the position of an imaginary body that orbits in the same period as the real one but at a constant angular speed (circular orbit). The angular speed assigned to the imaginary body is the satellite's average angular velocity over one orbit, and is called *mean motion, n* [5].

This research utilizes also another element set which results necessary to address case studies of past missions: the *Two-line element set* (TLE). A two-line element consists of a satellite identifier, an epoch, six orbital elements and a *B** term related to the ballistic coefficient [6]. These elsets are available to the general public through Air Force Space Command (AFSPC) [3].

The elements of a TLE are shown in equation 2.2 [3], where:

- the bars on the mean motion and semimajor axis denote *Kozai mean values*
- the numerators of the first two elements of the second line represent mean motion rate and acceleration
- $\frac{c_D A}{m}$ corresponds to the inverse of the *ballistic coefficient, BC*
- ρ_0 is the atmospheric density at perigee of the orbit
- $R_\oplus$ defines an Earth Radius of 6378.135 km
- the epoch is expressed in Coordinate Universal Time (UTC)

$$
\begin{array}{cccccc}
\bar{n} = \sqrt{\dfrac{\mu}{\bar{a}^3}} & e & i & \Omega & \omega & M \\[2ex]
\dfrac{\dot{n}}{2} & \dfrac{\ddot{n}}{6} & B^* = \dfrac{1}{2}\dfrac{c_D A}{m}\rho_0 R_\oplus & & UTC &
\end{array}
\tag{2.2}
$$

2.1.2   Reference Frames

Two types of reference frames are adopted by this research: the geocentric-inertial coordinate system and the geographic-body-fixed system. The origin of both systems is the center of mass of the central body, which in all case studies of this work is the Earth. They will therefore be labeled Earth-centered inertial (ECI) and Eart-centered Earth-fixed (ECEF) coordinate frames, respectively.

Earth-centered inertial

The ECI system is shown in FIGURE???. The equatorial plane is the reference plane. The X axis is the vernal equinox vector, and the Z axis is the spin axis of the Earth; north is positive. The axes are fixed in inertial space or fixed with respect to the stars [4].

Earth-centered Earth-fixed

FIGURE shows a representation of the ECEF reference frame. The system is Earth-centered and fixed to the rotating Earth [3]. Considering the ECI, the Z axis is the same, while the X axis always points towards the Greenwich meridian. Satellite ground track is commonly plotted in this coordinate system [4].

2.1.3   Two-Body Problem

The two-body problem is an indispensable theory to soak up the concepts of astrodynamics [3]. The Newton's law of universal gravitation (equation 2.3) states that all celestial bodies are attracted to each other by a force proportional to the product of their masses and inversely proportional to the square of the distance between their centers of mass.

$$F_g = \frac{MmG}{r^2} \tag{2.3}$$

6

G is the *universal gravitational constant*, while the other parameters clearly represents the masses and the distance. The motion of every particle in the universe is therefore determined by an infinite network attractive forces to all celestial bodies.

A real study of this scenario would be unfeasible, but since the motion of a spacecraft is dominated by one central body at time, is possible to define some plausible assumptions which lead to a simplification of the problem [4].

1. The motion of the satellite is governed by attraction to only one central body.

2. The mass of the spacecraft can be neglected with respect to the central body one.

3. The two bodies are spherically symmetrical with uniform density.

4. Only gravitational and centrifugal forces act on the bodies, along the line of their centers.

Under these assumptions, it is possible to develop a second-order, nonlinear, vector, differential equation called the *two-body equation* [3]:

$$\ddot{\vec{r}} + \frac{\mu}{r^3}\vec{r} = 0 \tag{2.4}$$

where μ is equal to the product between the central body and the universal gravitational constant G.

2.1.4 Orbital Perturbations

Orbital perturbations are deviations from a normal, idealized, or undisturbed motion. Introducing an alteration from two-body problem assumptions, the actual motion will vary due to perturbations caused by other bodies, and additional forces

not considered in Keplerian motion [3]. This subsection provides an overview of the main perturbations for an Earth orbiting spacecraft.

Earth's Gravity Field

Spinning celestial bodies are nor perfect spheres, but they are much more similar to oblate spheroids. For such a planet, the spin axis can be considered as the axis of rotational symmetry and the gravitational field will vary with the latitude as well as radius. The planet's oblateness provides a rotationally symmetric perturbation Φ, which does not depend on the longitude [7].

In particular, Φ is given by an infinite series characterized by the so-called *zonal harmonics* J_k of the planet of reference. Considering a spherical coordinate system for convenience, with origin at the planet's center of mass and third axis as the axis of rotational symmetry, in this series, shown in equation 2.5, r and ϕ are the distance from the origin and the polar angle respectively, μ is the Earth's standard gravitational parameter, R the equatorial radius and P_k represent the Legendre polynomials [7].

$$\Phi(r, \phi) = \frac{\mu}{r} \sum_{k=2}^{\infty} J_k \left(\frac{R}{r} \right)^k P_k(\cos \phi) \tag{2.5}$$

The zonal harmonics are non-dimensional quantities which are evaluated from satellite observation mission around the planet. The summation starts from $k = 2$, and the Earth's set of zonal harmonics is highly dominated by J_2. Taking into account only J_2 and starting from equation 2.5, it is possible to derivate the perturbing gravitational acceleration due to the respective zonal harmonic [7]:

$$\vec{a}_{J_2} = \frac{3}{2} \frac{J_2 \mu R^2}{r^4} \left[\frac{x}{r} \left(5\frac{z^2}{r^2} - 1 \right) \hat{\mathbf{i}} + \frac{y}{r} \left(5\frac{z^2}{r^2} - 1 \right) \hat{\mathbf{j}} + \frac{z}{r} \left(5\frac{z^2}{r^2} - 3 \right) \hat{\mathbf{k}} \right] \tag{2.6}$$

The right ascension Ω and the argument of perigee ω are significantly affected

by oblateness [7].

It is necessary to underline that the Earth's gravitational field vary not only with latitude but also with longitude due to irregularities in geometry and mass distribution [7]. However, with negligible approximation, it is definitely reasonable to consider only oblateness with J_2 effects in LEO missions, which is the major force for this range of altitudes, second only to Earth's gravity [4]. In light of the above, this thesis will take into account only J_2 forces with respect to the Earth's gravity field perturbations.

Atmospheric Drag

The residual atmosphere present at a few hundred kilometers of altitude strongly influences the motion of satellites in LEO. The basic equation of the perturbing specific force (force per unit mass) due to drag is the following [3]:

$$\vec{a}_{drag} = -\frac{1}{2}\frac{C_D A}{m}\rho V_{rel}^2 \frac{\vec{V_{rel}}}{|\vec{V_{rel}}|} \tag{2.7}$$

In equation 2.7, $\frac{m}{C_D A}$ is again the ballistic coefficient, that already appeared in the TLE representation (equation 2.2): m is the mass of the spacecraft, A represents the cross-sectional area of the satellite with respect to the atmosphere and C_D is the drag coefficient, a dimensionless parameter which takes into account every aerodynamic configuration aspect of the body with respect to the drag forces [8]. The value of the drag coefficient for satellites in the upper atmosphere is generally around 2.2 [3]; this number has been considered for all the case studies of the thesis. Finally, the velocity vector $\vec{V_{rel}}$ is relative to the atmosphere, as well as the cross section.

The main effects provided by aerodynamic drag are changes in the semimajor axis and eccentricity of the orbit and an accurate description of the atmospheric properties is crucial for the evaluation of drag on satellites [3]. However,

uncertainties in the time variance of upper atmosphere make perfect prediction of spacecraft drag impossible [4]. More in depth, the density changes because of a complex interaction between three factors: the nature of the atmosphere's molecular structure, the incident solar flux, and geomagnetic interactions. Several atmospheric models can be found in literature, either static or time-varying. The static ones are less accurate, but definitely simpler than the time-varying models, thanks to the assumption of all constant parameters [3]. This thesis exploits three different models, depending on the applications addressed in the case studies.

The first model considered by this research is static: the exponential model, valid in the range of altitudes between 0 and 1000 km. It assumes a spherically symmetrical distribution of particles of the atmosphere, in which the density decreases exponentially with increasing the altitude according to equation 2.8 [3]

$$\rho = \rho_0 \exp\left[-\frac{h - h_0}{H}\right] \tag{2.8}$$

where ρ_0 and h_0 represent a reference density and altitude respectively and H is the scale height.

The second atmospheric model, still static, is the Standard Atmosphere published in 1976 by the U.S. Committee on Extension to the Standard Atmosphere (COESA), valid drom 0 to 1000 km of altitude. It is an ideal, steady-state model of the Earth's atmosphere at a latitude of 45°N in moderate solar activity conditions [3].

Finally, Jacchia-Bowman 2008 (JB2008) is an empirical time-varying atmospheric density model. It revises and improves the earlier Jacchia-Bowman 2006 which is based on the diffusion equations of the Jacchia 71 model. JB2008 takes into account factors like solar irradiances, computed from driving solar indices based on orbit-based sensor data. Density variations are described by semiannual density equations based on 81-day average solar indices, as well as from

temperature equations that include corrections for diurnal and latitudinal effects. Geomagnetic effects are modeled too. The model is validated in the altitude range of 175 to 1000 km through comparisons with accurate daily density drag data previously collected for numerous satellites, existing atmospheric models and other measurements from several Earth orbiting satellite missions [9, 10].

Third-Body Perturbations

All the orbital elements are periodically affected by the gravitational forces of the Sun and the Moon. Similarly to what is triggered by the Earth's equatorial bulge, they apply an external torque to the orbits and cause the angular momentum to rotate. This perturbation is highly negligible in LEO, where the main effects are provided by J_2 and aerodynamic drag [11]. However, low Earth orbits characterized by a constant geometry with respect to the perturbing body can be affected by emphasized long-period effects. This is the case of Sun-synchronous orbits (SSO), for which the constant pattern with the Sun causes a long-term phenomenon of variation of the inclination (around 0.05 degrees per year) [12]. Although this number might appear meaningless, inclination is critical to SSO, as will be explained in paragraph 2.2.1.

Solar-Radiation Pressure

The last perturbation treated by this thesis is the solar-radiation pressure (SRP), which induces periodic variations in all the orbital elements. SRP generally becomes significant above 800 km of altitude, as drag becomes less important. Below this threshold, it might be neglected [11]. The perturbing acceleration can be approximated by the following equation [3]

$$\vec{a}_{SRP} = -\frac{p_{SRP} c_R A_{\odot}}{m_{sat}} \frac{\vec{r}_{\oplus \odot}}{|\vec{r}_{\oplus \odot}|} \tag{2.9}$$

p_{SRP} is the solar pressure, which is a quantity representing the change in momentum per unit area, derived by the ratio between the solar flux and the speed of light. The reflectivity, c_R, a value between 0.0 and 2.0, models the type of interaction between radiation and surface exposed to the Sun, $A_{\odot}$ [3]. m_{sat} is trivially the mass of the satellite.

2.1.5   Mean Orbital Elements

The previous subsection has described how orbital perturbations generate continuous variations and oscillations in the orbit elements. The values of the elements at a single point in time, which are periodically and secularly affected by the perturbing forces, are called *osculating elements*. On the other side, it is possible to define the *mean orbit elements*, which represent the average motion over a span of time [11]. In this way it is possible to obtain a representation of the orbital motion removing short and long-periodic effects induced by perturbations. And for most operational purposes, conversion from osculating to mean elements is indispensable [13]. Indeed, applications addressed by this work, like station-keeping and differential drag, require the secular behavior of the satellite to be implemented.

The oblateness of the Earth is the main guilty of periodic effects, provoking variations in all osculating elements, as well as all the other zonal harmonics. On the other hand, only even zonal gravitational harmonics and atmospheric drag give raise to secular effects in osculating elements, which are constant or non-periodic. Aerodynamic drag secular perturbation plays a crucial role in LEO [14].

Figure 2.1 shows the comparison between osculating and mean semi-major axis of a LEO satellite motion over the course of four months. The orbital decay induced by the atmospheric drag secular effect is evident, as well as the periodic consequences of J_2 perturbation.

Several perturbation techniques have been proposed to perform the conversion
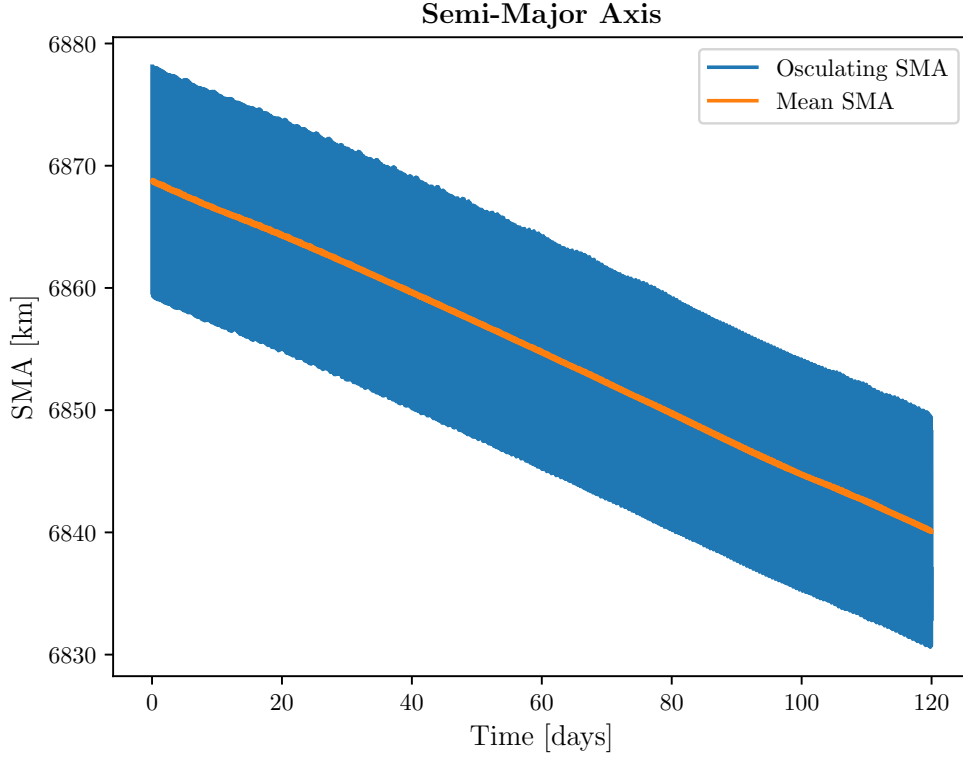
Figure 2.1: Osculating vs. mean semi-major axis for a LEO satellite under atmospheric drag and J_2 perturbations.

between osculating and mean elements. Brouwer and Kozai works were ones of the first methods appeared in literature, both consisting of first order solutions which take into account the Earth's asphericity but neglect drag effects [15]. This thesis uses a refinement of Brouwer's approach, suggested by Lyddane, which solves zero eccentricity and inclination singularities that are involved in the original theory. Moreover, the algorithm will account only first order J_2 terms [16].

2.1.6 Orbit Propagation

Orbit propagation is the principal problem in perturbation analysis. There are numerous methods to analyse the perturbing force models, and they can be classified in three categories [3]:

- *Special perturbation techniques*: they are numerical solutions, and they represent the most accurate way.

- *General perturbation techniques*: they are analytical methods based on approximations, which means the accuracy is lower, but the computation is speeded up.

- *Semianalytical techniques*: these techniques combine the best features of the previous categories, which means both good accuracy and computational efficiency.

The technique selected to design the tools shown in chapter 4 falls into the first category. Actually, the purposes of the thesis would even be achieved by a general perturbation method, saving time in the calculation phase. Indeed, there are not strict requirements of accuracy, since the research look of this work. However, a special perturbation technique has been chosen in order to provide more valuable tools which might be useful in the advanced stages of future Kuva Space missions. Furthermore, this is also a reasonable choice due to the long-time propagations needed by operations that will be discussed later.

Cowell's Technique

The Cowell's special perturbation technique is implemented in the algorithms produced by this thesis. The idea of this method is to add the perturbing accelerations to the two-body equation:

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \vec{a}_p \tag{2.10}$$

$\vec{a}_p$ represents the total acceleration given by the perturbing forces, like that ones seen in paragraph 2.1.4 (e.g., equation 2.7), while $\vec{r}$ is clearly the state vector of the satellite. Equation 2.10 is called *Cowell's formulation*. This second-order differential equation of motion needs a numerical integration method to be solved [3].

Dormand-Prince Method

The topic of numerical integration would require in-depth treatment due to its complexity, but only the method used by this work is briefly described in the next lines. The propagators developed to accomplish the purposes of the thesis take advantage of an eigth-order *Dormand-Prince* (DOPRI) method to solve the Cowell's equation. It is a single-step and variable step-size integrator. This means that it determines the subsequent steps exclusively from the information of the latest previous step. On the other hand, while fixed-step techniques divide the propagation into equal time steps, variable step-size algorithms dynamically adjust the step size. This is carried out including tolerance criteria to identify the accuracy at each step by considering two methods [17]. For instance, the thesis uses a DOPRI8$_{(5,3)}$, which means order 8 for the integration, with a 5th order error estimator combined with a 3rd order correction.

2.2   Specialized Orbits

Specialized orbits are the results of specific orbital features selected during the design of a mission, which can be simply related to the orbit period or, more commonly, to one of the main orbit perturbations [11]. In this section, two special Earth orbits will be analysed, both of interest for the purposes of the thesis. In fact, in general methods, the higher the propagation times, the lower the accuracy.

2.2.1   Sun-Synchronous Orbit

Sun-synchronous orbits are specialized orbits characterized by a constant geometry with the Sun over time [3]. They are used for several reasons, from technical needs like that ones deriving from thermal and electric power subsystems of the spacecraft to application requirements such as remote sensing. Indeed, a constant

Sun angle is very precious for missions working with electro-optical sensors [4]. This is the case, for instance, of hyperspectral technology.

An SSO can be obtained by matching the Westward motion of the line of nodes (*regression of the nodes*) to the solar motion projected on the equator, like shown in equation 2.11 [4].

$$\dot{\Omega} = 360°/365.242 \; days = 0.9856 \; °/day \tag{2.11}$$

This will allow the satellite's line of nodes to keep a constant angular separation with respect to the Sun. This separation can be achieved with very good approximation only considering the dominant cause of the secular motion of RAAN: J_2 perturbation. The secular behavior of Ω induced by the oblateness is function of semi-major axis, eccentricity and inclination and is given by the following formula [3]:

$$\dot{\Omega}(a, e, i) = -\frac{3nR_{\oplus}^2 J_2}{2a^2(1 - e^2)^2} \cos i \tag{2.12}$$

Equations 2.11 and 2.12 suggest that to achieve the proper $\dot{\Omega}$ of an SSO, the inclination has to be more than $90°$ (*retrograde orbit*), so that the term $\cos i$ will become negative [11]. As for the other two variables, eccentricity is generally supposed to be near-zero because low-Earth orbits usually imply very small values of e. Hence, with the assumption of $e = 0$, an SSO will be characterized by a direct relationship between a and i. Figure 2.2 shows the range of inclinations and altitudes that would provide the proper regression of nodes.

In addition, it is possible to select the initial value of the RAAN between 0 and 360 degrees according to the mission requirements for example [3]. The angle between the ascending node and the direction of the Sun is commonly labeled like *Mean Local Time of the Ascending Node* (MLTAN), because of the relative time to the noon meridian. This is a crucial design parameter because, while for an arbitrary orbit the MLTAN would continually change, for a Sun-synchronous orbit
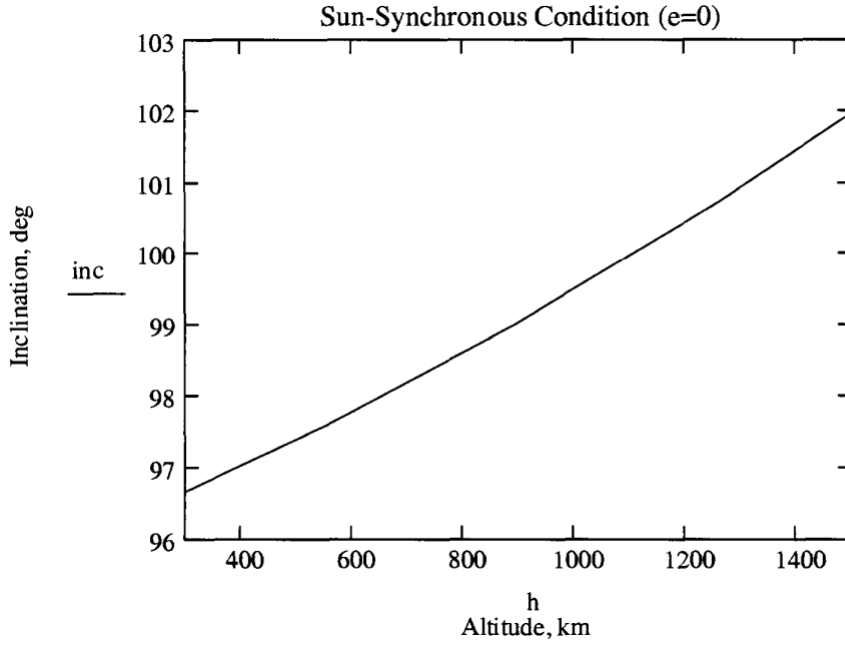
Figure 2.2: SSO condition: inclination vs. altitude ($e = 0$) [4].

it will remain constant [18] (not only at the nodes, but at any latitude).

In order to provide a better understanding of the concepts, figure 2.3 shows an example of an SSO with an LTAN of 15:00.

2.2.2 Repeat-Groundtrack Orbit

The peculiarity of a *repeat-groundtrak orbit* is the repetitive transit of the satellite to the same location relative to the surface of the Earth after a certain time interval [11]. Hence, they are used by missions which need to periodically revisit a specific point on the Earth [3], playing an interesting role in remote sensing applications.

Considering j orbit periods and the relative k days spent, an orbit is repetitive if and only if the ratio Q between j and k is a rational number, where Q represents the number of orbits per day. This means that j and k are integer and prime to each other. For instance, if a satellite performs 15.25 orbits per day, it will make a 4-day repeating ground track, because the respective rational Q is given by $j = 61$ orbits in $k = 4$ days [11]. k is typically known as the *revisit time* of

17
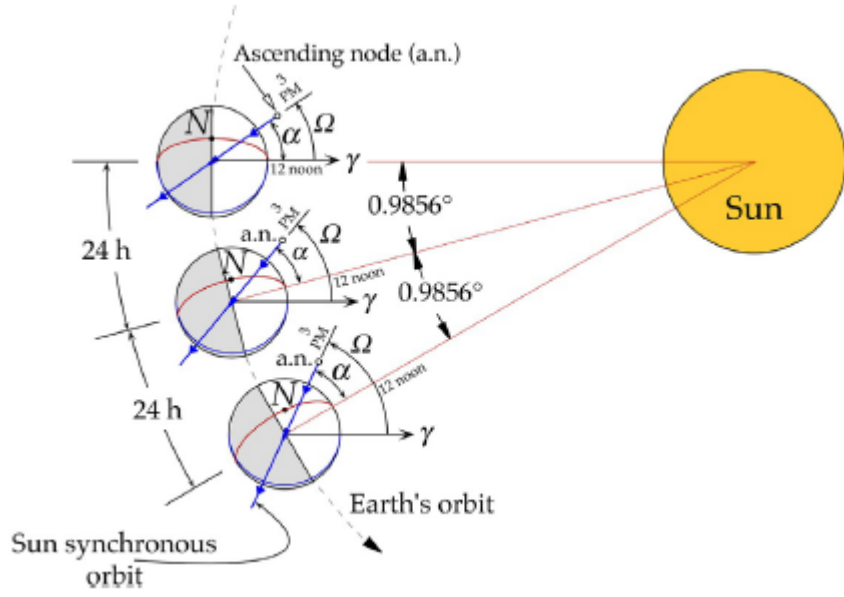
Figure 2.3: Sun-synchronous orbit 15:00 LTAN illustration [7].

the spacecraft, defined as the time elapsed between two successive observations of the same ground point [19]. It is necessary to underline that the effects of perturbations should be taken into account in order to obtain accurate repetition properties, since the actual period appears more complex [11].

In light of the above, it can be easily understood that the minimum achievable revisit time for a generic orbit is 1 day. However, mission objectives might require shorter revisit times. The solution to this issue is the design of a constellation of satellites, as will be discussed later.

2.3 Satellite Constellations

A satellite constellation can be defined as a group of similar satellites, with comparable functions and properties, designed to operate in similar, complementary orbits to undertake common objectives, under shared control [20]. In the context of this thesis, Earth-orbiting constellations are of notable interest thanks to the global coverage of the world they can enhance with respect to a single satellite mission. In recent years, the availability of smaller, lower-cost satellites, and the

easier access to space have led to a drastic evolution in the design of constellations. Thanks to this trend, applications like observation, communication and navigation have drastically expanded their potential [11].

2.3.1 Walker Delta Constellation

Walker constellations are the most symmetric of the satellite pattern families. The *Walker Delta* is the most popular model within the latter, which is also the one selected for the future Kuva Space constellation. This satellite pattern can be fully determined through four parameters. T represents the total number of satellites, uniformly distributed in each of the P orbit planes. i is the inclination of the orbits, which is supposed to be the same for all. The last term is F, and is defined as the relative spacing between satellites in adjacent planes. In a constellation $\Delta\phi$ is defined like the *phase difference*, which is the angle in the direction of motion from the ascending node to the nearest satellite at a time when a satellite in the next most westerly plane is at its ascending node. $\Delta\phi$ must be an integral multiple of $360°/T$, and can be any integer number from 0 to $P-1$: this value is F [11].

$$i : T/P/F \tag{2.13}$$

Equation 2.13 shows the shorthand notation used to indicate a Walker Delta constellation.

An example of a Walker Delta configuration is displayed in figure 2.4. The complete symmetry in longitude is one of its main advantages. In addition, there is a finite number of Walker constellations, and all of them can be identified and investigated. This allows the mission designer a full analysis of the constellation on every aspect of interest [11].

Figure 2.4: Illustration of a 65:15/3/2 Walker Delta configuration [11].

2.4 Constellation Orbit Management

Over the life of a space mission, the need to adjust or maintain the orbit elements might arise. In general, orbit maintenance is required because of two main reasons: targeting to reach an end orbit or position, or to maintain absolute or relative orientations [11].

This thesis focuses on two different phases of constellation orbit management, proposing a possible strategy for the main case study. The first maneuver addresses the *initial orbit phasing* of a satellite fleet after the deployment from the launcher. The plan is to dispose the spacecraft into the desired pattern through a *differential drag* method. The second action consists of the *relative station-keeping*. It aims to maintain the relative positions of the vehicles within the same orbit, taking advantage of low-thrust propulsion systems.

Downstream of this two-step strategy, an analysis of space debris mitigation is carried out in order to assess the importance of collision avoidance in a satellite constellation.

2.4.1   Initial Orbit Phasing (Differential Drag)

The initial orbit phasing treated by this work consists of the achievement of the desired formation of a group of satellites deployed within the same orbit. In particular, the final scheme considered by the thesis is composed by a nominal altitude (lower than the deployment's one) and a symmetric pattern with equal relative angular separation between the spacecraft. A differential drag method is proposed to accomplish this purpose.

Differential drag can be defined as the difference in atmospheric drag acting on each of a couple of satellites. Assuming that the spacecraft travel through similar atmospheric density areas, the differential drag is almost fully determined by differences in ballistic coefficient of them [21]. Considering identical satellites, and remembering that BC depends on the area perpendicular to the velocity vector, differential drag can be achieved controlling the attitude of the body, leading to different values of surface exposed to the aerodynamic flux.

To clarify ideas and simplify the problem, only two configurations will be identified as shown in figure 2.5: low-drag and high-drag modes.

Moreover, the satellite on the lower orbit will be indicated as the *leading satellite*, whereas the other one *trailing satellite*. Supposing of using differential drag only to equalize the altitudes, with both spacecrafts travelling in low-drag profile, the trailing satellite shall be set to high-drag mode until the leader's orbit is reached. Indeed, the acceleration of its motion is the same as if the air drag force, reversed, were pushing the satellite. This phenomenon is known as *satellite paradox* [23].

Several applications can take advantage from differential drag. It was first proposed for *formationkeeping* to keep a satellite in a specific position with respect to another. Even rendezvous maneuver strategies have been combined with differential drag actions. This method has also been selected for collision avoidance
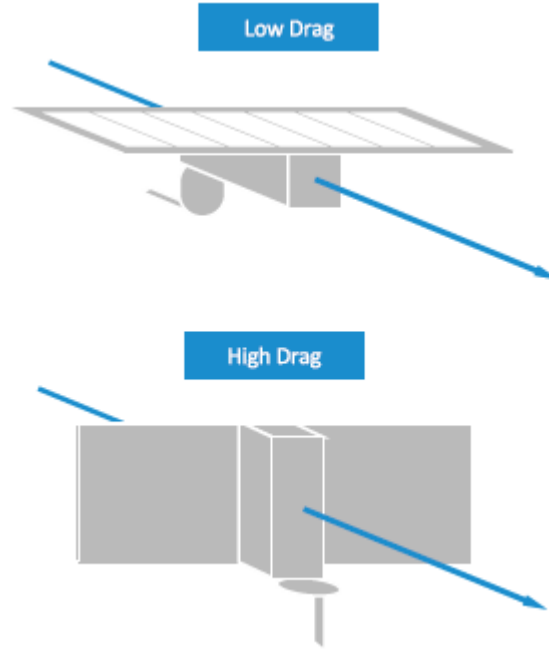
Figure 2.5: Low drag vs. high drag mode of a generic satellite shape [22].

scenarios [22].

What is of huge interest for the purposes of the thesis is the experience provided by Planet company (formerly Planet Labs). They applied differential drag control to fleets of propulsion-less satellites deployed in the same orbit with successful results for several missions. The goal of Planet's controller was to bring all the spacecrafts at the same altitude and perform phasing with only differential drag after deployment from launcher. Three examples of constellation phasing are shown in figure 2.6. The initial conditions subsequent to the deployment contain small but significant variations in SMA: achieving the same orbital altitude means zero-relative speed between them. On the other hand, the phasing aims to symmetrically separate the vehicles in terms of angular distance, along the same orbit. Once the desired constellation pattern is achieved, the differential drag control continues to operate as station-keeping to maintain the formation, since the propulsion-less nature of the satellites [22].

Based on the Planet's controller, this thesis proposes an algorithm for the

Figure 2.6: Generic cases of desired relative placement of four satellites [22].

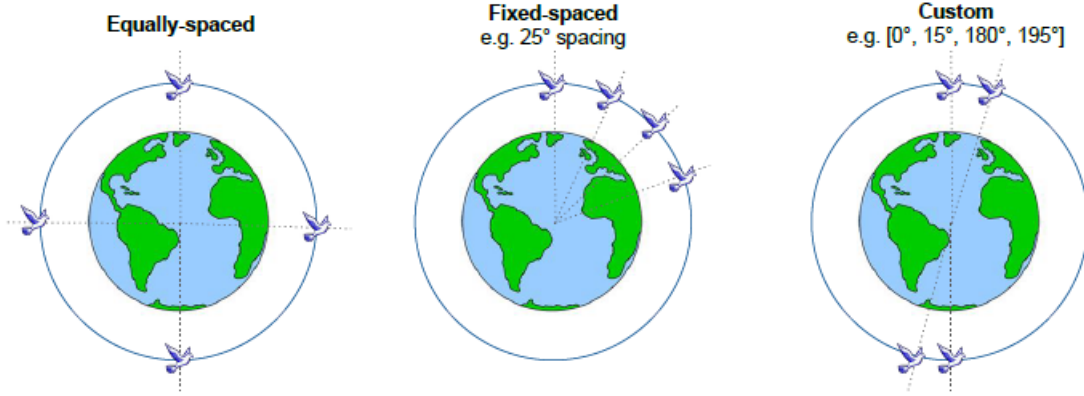phasing of a group of satellites after the initial dispersion from the launcher only using differential drag.

2.4.2 Absolute Station-Keeping

Station-keeping can be defined as the procedure of controlling a satellite in order to maintain its position within a specific imaginary box or with respect to inertial space or other spacecraft. In constellation maintenance, the only goal is to keep the relative placement between satellites. Orbit perturbations and variations in initial conditions for each vehicle are the major causes which make necessary maintenance maneuvers. A long-lived LEO constellation must principally fight the atmospheric drag in order to avoid orbit decay.

The strategy addressed by the thesis follows the approach of constantly negating the perturbing force, which requires continuous propellant usage. More in detail, each vehicle is kept in a well-defined mathematical box under *absolute station-keeping*. The idea is to periodically put back the ΔV taken out by drag. This is equivalent to keep the satellite in a box by letting it drift to one side of the box, hitting it with a thruster, letting it drift across the box and back, and repeat the cycle, as shown in figure 2.7 The objective is to maintain a desired pattern as illustrated by figure 2.8. Station-keeping boxes must typically be 10 to 50

kilometers long, and accounting for low-thrust propulsion systems, this strategy will need a large number of small burns [11].
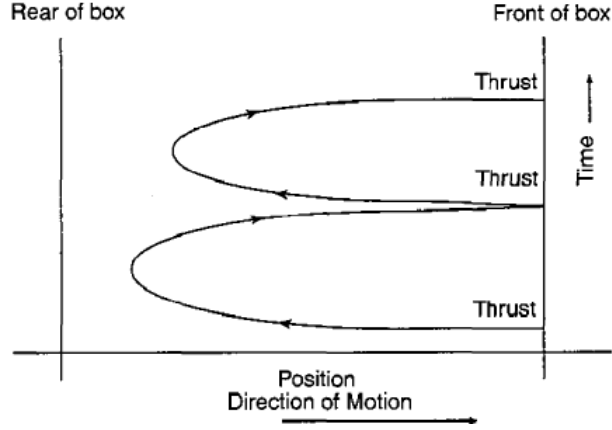
Figure 2.7: Satellite's drift in the station-keeping box [11].
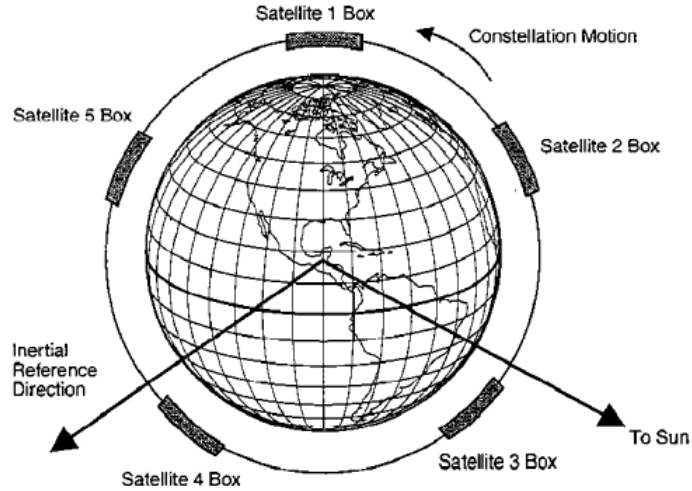
Figure 2.8: Absolute station-keeping of a group of satellites within the same orbit [11].

In conclusion, this thesis takes into account the regular execution of annual Inclination Adjust Maneuvers (IAM) and frequent Drag Make Up (DMU) maneuvers to counteract perturbing effects [24]. The simulations of these scenarios, which naturally involve forces generated by thrusters, rely on the *Edelbaum-Kechichian* formulas.

Edelbaum-Kechichian Theory

An analytic solution for the optimum low-thrust transfer between inclined circular orbits of different radius is provided by the equations of Edelbaum reformulated by Kechichian in 1992 [25, 26]. This thesis takes advantage of this theory to compute ΔV, burning time and related acceleration needed to satisfy the requirements of station-keeping.

The problem assumes only tangential and out-of-plane acceleration, and circular orbits during the transfer ($e = 0$). These assumptions are widely acceptable in this context, because all the case studies of this research are characterized by near-circular orbits, and the thrust direction of the satellites is ideally intended to be parallel to the velocity vector.

The first input parameter is the thrust force acceleration f, deriving from the ratio between the thrust provided by the thruster and the spacecraft mass, which is broken down into three mutually orthogonal components: the first one is tangential to the velocity vector of the satellite, the second component is normal to the orbital plane, and the last one is in the out-of-plane direction.

They are expressed in terms of two angles: α is the angle between the velocity vector and the component of the thrust vector in the plane of the orbit, while β represents the angle between the thrust vector and the plane of the orbit [25].

$$f_T = f \cos \beta \cos \alpha \qquad f_N = f \cos \beta \sin \alpha \qquad f_z = f \sin \beta \qquad (2.14)$$

According to the hypotheses of this theory, α is always assumed to be zero and consequently f_N is null. Initial and final desired orbit are only assigned by the variations in semimajor axis an inclination (Δa, Δi). From the initial and final semimajor axis, the velocities V_0 and V_f are computed

$$V_0 = \sqrt{\frac{\mu}{a_0}} \qquad V_f = \sqrt{\frac{\mu}{a_f}} \qquad (2.15)$$

where μ is always the Earth's gravity constant. Kechichian's reformulation derives the initial angle β_0 from the velocities in 2.15 and Δi:

$$\beta_0 = \tan^{-1} \frac{\sin \frac{\pi}{2}\Delta i}{\frac{V_0}{V_f} - \cos \frac{\pi}{2}\Delta i} \qquad (2.16)$$

It is then possible to write the total ΔV

$$\Delta V = V_0 \cos \beta_0 - \frac{V_0 \sin \beta_0}{\tan \left(\frac{\pi}{2}\Delta i + \beta_0\right)} \qquad (2.17)$$

and consequently achieve the transfer time

$$t_f = \frac{\Delta V}{f} \qquad (2.18)$$

Given the the initial β_0 and the burning time, Kechichian finally formulates the following control law for the yaw angle β:

$$\beta = \tan^{-1} \frac{V_0 \sin \beta_0}{V_0 \cos \beta_0 - ft_f} \qquad (2.19)$$

In case of coplanar transfers ($\beta = 0$), Δi is equal to 0 and ΔV will be simply given by the difference of the velocities in equation 2.15 [26].

2.4.3   Space Debris Mitigation

Space debris mitigation consists of a series of design and operational provisions, with the aim of limiting the number of debris in orbit, the probability and effects of on-orbit fragmentation and collision events, and the risks associated to re-entry, whether expected or planned. The space debris population includes spacecraft and launch vehicle orbital stages becoming non-functional, as well as all the other mission-related objects. They are all candidates at triggering fragmentation events due to on-orbit break-ups and collisions. The short and long-term survivability

of any other operational space object is put at risk by fragmentation debris [27]. Mitigation actions taken into account by this thesis are mainly related to the tools provided by the *Debris Risk Assessment and Mitigation Analysis* (DRAMA) software, developed under ESA contract [28]. Section 3.3 presents a detailed explanation of DRAMA and its features.

In the context of this thesis, there is one more essential point to make. Constellations involve narrowly defined intersecting orbits and often respond to high levels of symmetry which lead to patterns where, in principle, two spacecrafts will end up at the same place at the same time in orbit. This scenario fully represents the main case study of this research. The above considerations suggest how fundamental the issue of collision avoidance is in the systems engineering of a satellite constellation.

Typically, two key elements are estimated to assess this topic. The first one is the collision probability. It is determined by the collision opportunities, defined as the events between two satellites in which they pass close to each other.

The second key element consists of the consequences of a collision. They are extremely deleterious for two main reasons. First, the amount of energy released by a collision is enormous, because this energy is proportional to the square of the relative velocity between the colliding objects. Orbital velocities in LEO are in the order of 7 km/s. Secondly, the destruction of a satellite generates a huge number of fragments, giving rise to a *debris cloud* of various sizes particles which practically travel in the same orbit where the satellite exploded. As a result, the collision probability of other operating spacecrafts definitely increases, since each one of the fragments might potentially collide with them. Their decay process will obviously depend on the level of atmospheric drag [11].

Chapter 3

Methodology

Exclusively open-source resources have been used to carry out the thesis work. In particular, the orbital scenarios under examination have been simulated through dynamic propagation models created in Python, taking advantage of existing free Python libraries. In order to validate the results, the *General Mission Analysis Tool* (GMAT) has been chosen as the reference software to make comparisons. Even space debris mitigation has been performed thanks to the availability of an open-source software, DRAMA, published by ESA.

In the following paragraphs a brief description of the tools mentioned previously is presented.

3.1 Python for Astrodynamics Applications

Due to the computationally intensive nature of astrodynamics tasks, astrodynamicists have relied on compiled programming languages such as Fortran for the development of astrodynamics software. Interpreted languages like Python on the other hand offer higher flexibility and development speed thereby increasing the productivity of the programmer. While interpreted languages are generally slower than compiled languages, recent developments such as just-in-time (JIT) compil-

ers or transpilers have been able to close this speed gap significantly. Another important factor for the usefulness of a programming language is its wider ecosystem which consists of the available open-source packages and development tools including integrated development environments and debuggers.

In light of the above, Python can be considered as a suitable language for scientific computing due to the existence of well established libraries like *NumPy* and *SciPy* for mathematical calculations. Not only do these libraries make use of compiled code from existing accelerated libraries, but it is also possible in Python to interface with compiled code to speed up specific algorithms where required. This allows a user to define a problem in Python, while benefiting from the speed of compiled languages for computationally expensive algorithms. Indeed, JIT-compiled dynamic languages such as Python with *Numba* have reached a competitive level of performance while still offering the advantages of lower complexity and better programmer productivity [29].

And not for nothing, nowadays, by a wide margin, Python is the most popular interpreted language in astronomy [30].

3.1.1 Numba

Arguably, the most important library in the scientific Python stack is NumPy, which implements n-dimensional arrays and its related methods in C programming language, and wraps them using the CPython API (Application Programming interface). It is a fundamental piece of software that powers most numerical codes written in Python nowadays. However, while it is possible to vectorize certain kinds of numerical operations, there might be other cases where this may not be feasible and where the dynamic nature of Python leads to a performance penalty, especially when the algorithm involves several levels of nested looping. To overcome these limitations it is possible to use Numba, an open-source library

which can infer types for array-oriented and math-heavy Python code and generate optimized machine instructions using the LLVM compiler infrastructure [31]. In brief, Numba is a JIT compiler for scientific Python, which allows to optimize the running time.

3.1.2 poliastro

poliastro is an open-source Python library for astrodynamics and orbital mechanics released under the MIT (Massachusetts Institute of Technology) license. It features algorithms which are written in pure Python and compiled using Numba. It is dedicated to astrodynamics applications, such as orbit propagation, resolution of the Kepler and Lambert problems, conversion between position and velocity vectors and classical orbital elements and orbit plotting [31]. In addition, thanks to Astropy, poliastro can perform seamless coordinate frame conversions and use proper physical units and timescales. At the moment, poliastro is the longest-lived Python library for astrodynamics, has contributors from all around the world, and several New Space companies and people in academia use it [32].

3.2 General Mission Analysis Tool

The General Mission Analysis Tool is a software system for trajectory optimization, mission analysis, trajectory estimation, and prediction developed by NASA (National Aeronautics and Space Administration), the Air Force Research Lab, and private industry. GMAT is designed to model, optimize, and estimate spacecraft trajectories in flight regimes ranging from LEO to lunar applications, interplanetary trajectories, and other deep space missions. GMAT's design and implementation are based on four basic principles: open-source visibility for both the source code and design documentation; platform independence; modular design; and user extensibility [33].

3.3 Debris Risk Assessment and Mitigation Analysis

The *Debris Risk Assessment and Mitigation Analysis* software tool supports space missions in the analysis of every kind of debris related problems. It is provided with the necessary computational and data reference by another tool: MASTER (*Meteoroid and Space Debris Terrestrial Environment Reference*). DRAMA includes five individual components, aiming at aiding the verification of the compliance of space missions with mitigation guidelines and standards, like that ones mentioned in subsection 2.4.3 [28].

The first one is the *Assessment of Risk Event Statistics* (ARES) tool which evaluates the annual rates of close approaches between an operational satellite of interest and debris in Earth orbits along with statistics on the necessary number of collision avoidance maneuvers and related ΔV and propellant mass.

The *MASTER-based Impact flux and Damage Assessment Software* (MIDAS) assists the assessment of debris and meteoroid impact rates over the satellite's lifetime.

DRAMA also provides the estimation of the orbital lifetime and the evaluation of different possible disposal strategies after the End-of-Life (EOL) thanks to the *Orbital SpaceCraft Active Removal* (OSCAR) tool.

The *re-entry Survival And Risk Analysis* (SARA) tool allows the assessment of the re-entry survivability of all the different spacecraft components. It then provides the combined on-ground casualty risk given a world population model and the impact footprint of the surviving fragments.

Finally, the *Cross-section of Complex bodies* (CROC) intends to compute the cross-sectional areas of complex satellites and different possible attitudes laws after the EOL [34].

Chapter 4

Satellite Constellation Management Tools

This chapter presents all the scripts designed along this work to achieve the purposes of the thesis. The first paragraph contains the primary functions needed for the development of the main tools proposed from section 4.4 to 4.2.

4.1 Orbit Propagators

The study of the applications and topics covered by this thesis clearly require an orbital propagator. The core of all the propagators produced along this work exploit the Cowell's model shown in subsection 2.1.6. Both undisturbed and perturbed motion have been analysed.

Orbit propagators must take in input the following data:

- **Initial orbit**. The initial state of the staellite orbit must be defined before propagation. All the six orbital elements (see subsection 2.1.1) and the epoch are required. The scripts can also work defining the six components of position and velocity vectors. Even the TLE can be set as input, which will be appropriately converted into the classical elements representation.

33

The reflectivity coefficient must be added if the problem takes into account the SRP.

- **Time settings**. These input parameters include *time frame* and *time step*. The first one simply consists of the total temporal period of propagation requested by the user. On the other hand, the time step is strictly related to the output density. Lower values imply more detailed results, as well as longer computation times. In a nutshell, the outputs will be discretized in as many points as those deriving from the ratio between time frame and time step. Small time steps provide narrower plots. Figure 4.1 shows the SMA of a LEO satellite under drag perturbation during one orbit (around 50 minutes). The assigned time step is of 5 minutes (the curve is therefore discretized into 10 points). It is important to underline that the accuracy of the integrator used by thesis does not depend on this time step value at all.

- **Perturbations parameters**. Propagators that involve any perturbing acceleration need the parameters which describe that force. For example, the perturbing specific force of atmospheric drag (equation 2.7) requires the air density, which will be given by a specific atmospheric model. Even for the acceleration of an eventual thruster it is necessary to specify the force it generates.

- **Spacecraft specifications**. According to the features of the selected propagator, one or more satellite characteristics shall be provided when considering perturbations. For instance, in presence of atmospheric drag, C_D and BC (which means cross-sectional area and mass) are needed.

- **Relative tolerance**. It represents the admitted percent error at any step of the simulation. For instance, a relative tolerance of 0.01 indicates a 1% allowable error limit relative to each state value at each step. This setting controls the accuracy of the solution of the selected integrator [17] (DOPRI8$_{(5,3)}$ is the method used by the thesis).
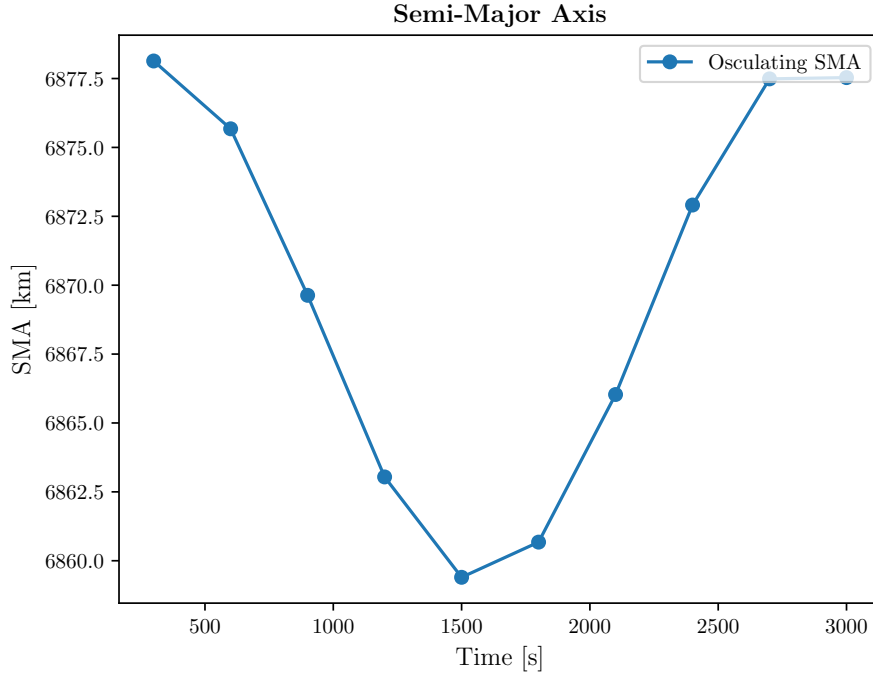
Figure 4.1: Osculating SMA of a LEO satellite during one orbit. Propagation time settings: time step = 5 minutes, time frame = 50 minutes.

4.1.1 Undisturbed Motion

The first orbit propagator proposed by this thesis consists of the ideal motion of a spacecraft neglecting all the possible perturbations that might affect it. Therefore, this kind of propagation is governed by the two-body equation shown in subsection 2.1.3.

This model is mainly useful for two reasons. First, it requires computation times definitely shorter than a perturbed propagation, allowing quick results when perturbations are not considered decisive for the purposes of the user. Secondly, it provides an educational method to better understand the perturbing effects when compared to more realistic propagators.

Figure 4.2 shows the orbital elements values of the motion of a satellite in LEO. Only SMA and true anomaly are reported for matters of layout. The graph of
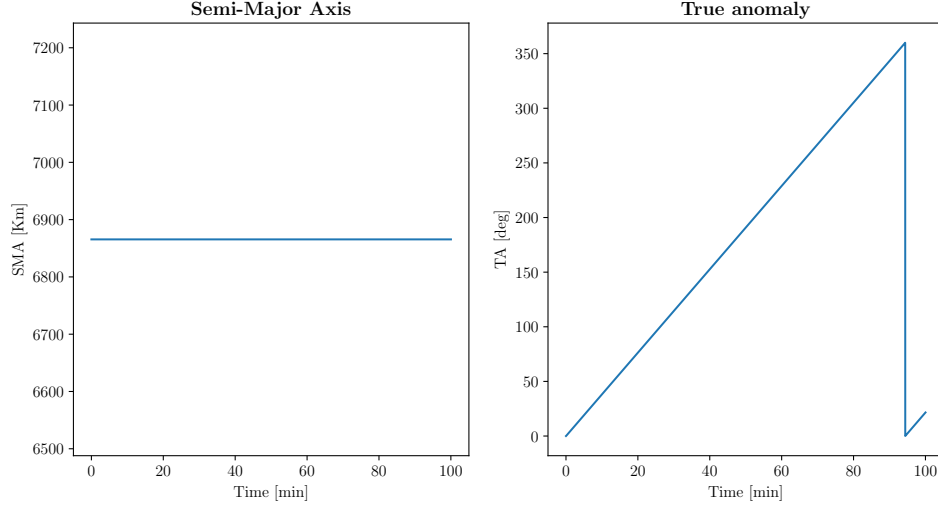
Figure 4.2: SMA and true anomaly variation along around one LEO satellite orbit with eccentricity of 0.001

the other elements is constant like the semi-major axis, because no perturbation affects their behavior. The true anomaly is the only time-variant parameter, and in this case its curve looks like varying constantly. The reason of this aspect lies in the eccentricity: this example is characterized by a near-circular orbit, therefore the velocity of the satellite is almost the same along the entire orbit. Increasing the eccentricity value, like in figure 4.3, the true anomaly changes more rapidly when is in proximity of the perigee and vice-versa.

All the case studies of this research consist of near-circular orbits. Despite the fact that zero-eccentricity condition is not feasible, this comparison demonstrates that very low values of eccentricity would make the assumption of circular orbit highly acceptable. Therefore, the Edelbaum-Kechichian formulas (paragraph 2.4.2) adopted for the station-keeping maneuvers are largely suitable for the problems addressed along the thesis.
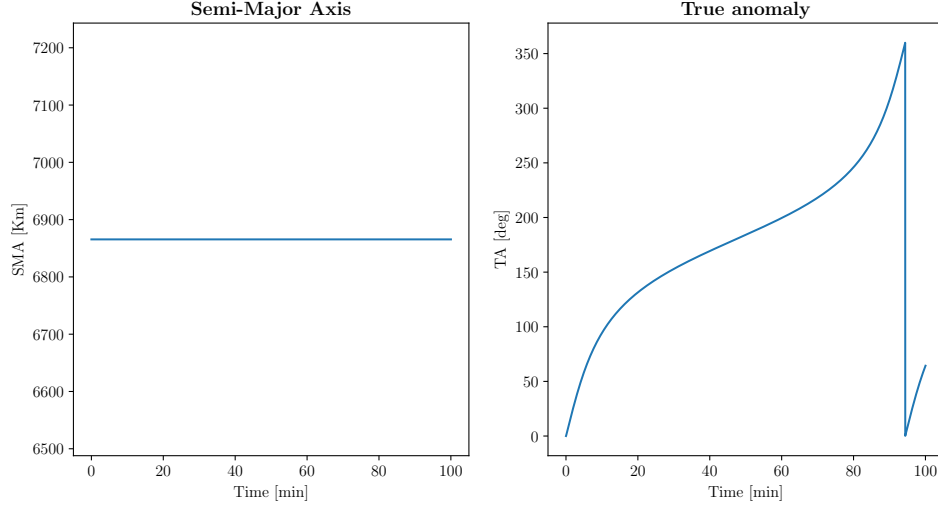
Figure 4.3: SMA and true anomaly variation along around one LEO satellite orbit with eccentricity of 0.5

4.1.2 Perturbed Motion

The propagators developed by this work which take into account perturbing effects are based on the Cowell's technique of subsection 2.1.6.

Since Sun-synchronous orbits at low altitudes denote the main discussion point of this thesis, Earth's oblateness and atmospheric drag represent the perturbations of greatest interest. Despite this, propagations with SRP and 3rd-body forces have been examined as well for assessing long-term effects.

The perturbing acceleration functions performed by this thesis come from the poliastro's library.

Atmospheric Models

The atmospheric drag exponential model provided by poliastro has been found to be too approximate compared to the results obtained by GMAT of equal examples. For this reason, two more accurate atmospheric models have been implemented into the scripts designed along this work. The first one consists of COESA76

model, while the second is the JB2008 one. Their theoretical background has been already exposed in paragraph 2.1.6. JB2008 is definitely the most accurate model

Mean Orbital Elements Converter

Sun Synchronous Orbits Functions

4.1.3   Satellite Constellation Propagator

4.2   Differential Drag Controller

The differential drag algorithm represents the main effort of this thesis. It consists of a method to perform the phasing through the differential drag of two or more satellites, of any shape and mass, after their deployment at the beginning of the mission. The goal is to achieve an assigned orbital slot and same mean semi-major axis respectively on all spacecraft forming the constellation, resulting in zero relative velocity between them. Since the algorithm designed by this thesis takes inspiration from the controller adopted by Planet (Foster et al. [35]), a description of the latter is presented.

4.2.1   Planet's Control Algorithm

The objective of Planet's algorithm is the same as that of the thesis. The technique consists of labeling the satellite in the lowest orbit as the leading satellite. Then, a two-component state composed by the mean Earth-centered angular distance to its desired slot θ_i, and the angular velocity $\dot{\theta}_i$ relative to the leader, is computed for each trailing satellite. The procedure to achieve the wanted pattern is reported below in the identical form presented by the company [35].

Planet's Control

1: **procedure** CREATE HIGH-DRAG WINDOWS
2:     remove any previously issued high-drag windows
3:     compute θ_i, $\dot{\theta}_i$ of each satellite i relative to previous leader (mean, not osculating)
4:     compute $\ddot{\theta}$ of a satellite in high-drag vs. low-drag mode
5:     assign leader to be satellite with greatest $\dot{\theta}$ (i.e. lowest altitude)
6:     **for** each satellite i that is not the leader **do**
7:         offset θ_i, $\dot{\theta}_i$ to be relative to new leader: $\theta_i = \theta_i - \theta_{leader}, \dot{\theta}_i = \dot{\theta}_i - \dot{\theta}_{leader}$
8:         get required high-drag duration to null relative speed with leader: $\Delta t_{i,hd} = -\frac{\dot{\theta}_i}{\ddot{\theta}}$
9:         get angle travelled during desired high-drag window: $\theta_{i,hd} = \frac{1}{2}\ddot{\theta}\Delta t_{i,hd}^2$
10:         get wait time to target desired slot: $\Delta t_{i,wait} = \frac{\theta_{i,hd} - \theta_i}{\dot{\theta}_i}$
11:         create high-drag window for satellite i starting at $t_{now} + \Delta t_{i,wait}$ for duration $\Delta t_{i,hd}$
12:     **end for**
13: **end procedure**

The strategy is to first obtain the high-drag time $\Delta t_{i,hd}$ (high-drag window) needed by the trailing satellite for matching the leader's semi-major axis, and then the amount of time $\Delta t_{i,wait}$ each spacecraft has to wait in the low-drag configuration before the high-drag maneuver is initiated.

Despite this method has successfully worked for the initial orbit phasing of a satellite constellation, it presents some limitations and ambiguities. The durations $\Delta t_{i,wait}$ and $\Delta t_{i,hd}$ are predicted making assumptions related to the density of the atmosphere and the attitude of the vehicles. This leads to unavoidable errors in accuracy. In addition, it is not clear how the angular acceleration $\ddot{\theta}$ is estimated. The angle $\theta_{i,hd}$ might also contain inaccuracies because is computed via Taylor approximation. The accuracy of the algorithm is then strongly sensitive to the initial conditions of dispersion. Too small differences between satellite orbits do not allow the control to function properly, as well as too big variations lead to inaccurate results. Despite these negative observations, this algorithm has successfully worked for the initial phasing of a satellite constellation. The Taylor approximation seems to be quite effective according to the on-orbit results shown
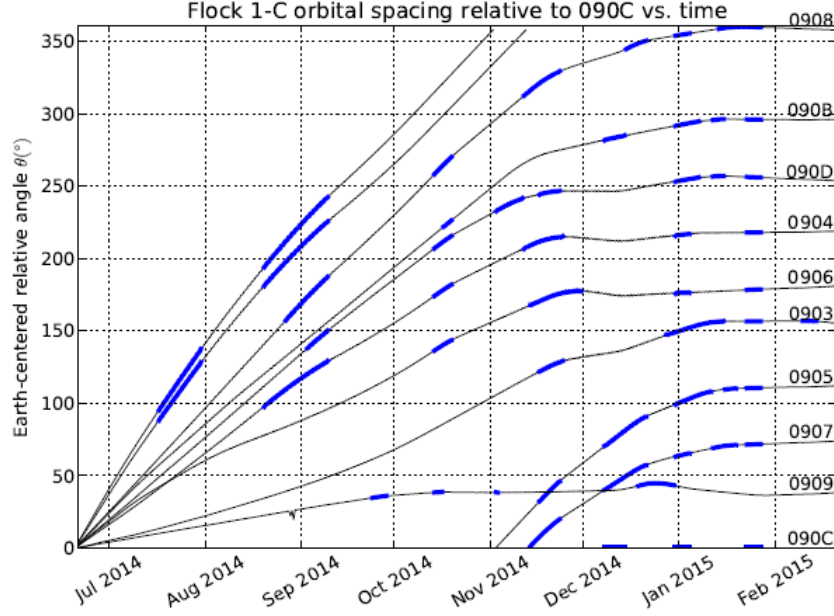
Figure 4.4: Actual Planet's fleet relative motion achieved by using differential drag. Blue lines represent high-drag windows [35].

in figure 4.4 [36]. The curves on the graph represent the relative angular distance between each satellite and the leader (which remains at zero degrees in the plot), and the blue lines are the high-drag windows. Indeed, Planet reruns its algorithm on a regular cycle to minimize the errors deriving from the forecast of the future state of the atmosphere. As for the initial dispersion, it can be adjusted always through differential drag maneuvers, in order to achieve more proper initial conditions for the controller [35]. However, there is no clear information about the appropriate circumstances which make the algorithm work correctly.

Planet has applied this tool to groups of propulsion-less spacecraft. Once the desired formation is achieved, the same differential drag control is used to maintain the requested relative angular distances between the vehicles.

4.2.2 Thesis Controller

This paragraph contains the work carried out to achieve the main goal of this thesis: the design of a differential drag controller for the initial orbit phasing of a group

40

of spacecraft. This is then combined with a low-thrust relative station-keeping. Their coupling enables the orbital management of a LEO SSO constellation of low-thrust satellites throughout the mission lifetime. The drag control derives from an improvement of the Planet's algorithm.

First essential requirement is the use of the mean orbital elements. All the space vehicles involved are supposed to have the same initial orbital elements except SMA and argument of latitude, and all the operations require the use of the mean orbital elements.

The algorithm establishes a reference vehicle, the *leading satellite*, which is identified by the spacecraft travelling in the lowest orbit (greatest mean motion n). The *trailing satellites* have the task of reaching the desired relative placement with respect to the leader. For each of them, the script computes the distance from the leader by the mean motion, the angular error θ_{err} from their position to the relative target, and the first attempt value of t_{est}. Angular separations are calculated in terms of argument of latitude. The latter parameter represents the interval of time in which the algorithm evaluates the variation of n. $\dot{n}$ derives from the acceleration generated by the switch from low-drag to high-drag configuration. This lead to the assessment of t_{hd}, the high-drag window. After that, the algorithm evaluates the path travelled in high-drag configuration θ_{hd} and finally the wait time t_{wait}. At this point, the estimation of t_{hd} and t_{wait} is repeated with a new t_{est} provided by a function which depends on the altitudes of the satellites and the time intervals computed before. The procedure is rerun for a sufficient number of iterations. The procedure is shown below.

Differential Drag Control

1: **procedure** PERFORM PHASING THROUGH DIFFERENTIAL DRAG
2:     establish the leading satellite (greatest mean motion)
3:     **for** each trailing satellite of the constellation **do**
4:         compute difference in mean motion with respect to the leader Δn
5:         compute angle to reach the desired slot relative to the leader θ_{err}
6:         assign first attempt value to t_{est} from θ_{err}
7:         **for** assigned number of iterations **do**
8:             get mean motion variation $\dot{n}$ from low to high drag mode over t_{est}
9:             get high drag time window $t_{hd} = \frac{\Delta n}{\dot{n}}$
10:             get angle travelled during high drag window $\theta_{hd} = \frac{1}{2}\dot{n}t_{hd}^2$
11:             get waiting time window $t_{wait} = \frac{\theta_{err} - \theta_{hd}}{\Delta n}$
12:             get new estimation time $t_{est} = f(t_{wait}, t_{hd}, h_{lead}, h_{trail})$
13:         **end for**
14:     **end for**
15: **end procedure**

The strong point of the proposed method lies in the last line. The estimation time is a critical factor in terms of accuracy for the final result. It is estimated thanks to an empirical equation formulated within this thesis work. This function enables a significant better prediction of the two time windows, making unnecessary the rerun of the algorithm in several cases. Despite this, regular repetition of the calculations remains a good and sometimes necessary action as for the Planet's strategy. t_{est} is evaluated for an assigned number of iterations, but the experience shows that just twice in enough. The statements above as will be proven in chapter 6. Similarly to Planet's algorithm, this method is sensitive to the initial dispersion after the deployment. However, this is issue is easily addressed implementing preliminary high-drag or wait windows. The only determining factors of this problem are the initial difference in semi-major axis and the altitude from which the launcher deploys the satellites. The first condition is managed with proper high-drag maneuvers. On the other hand, when the deployment altitude it is too high, differential drag is unavoidably too low. Therefore, a preparatory wait time window might be required for the entire fleet of spacecraft, eventually applying high-drag mode to them in order to speed up the process.

4.3   Station-Keeping Simulator

This tool is an absolute station-keeping simulator. It is able to simulate actions of IAM and DMU performed by low-thrust propelled satellites. The method combines an orbit propagator and an algorithm that keep the spacecraft within a defined range of altitudes and inclinations. The required input parameters include the thrust force of the propulsion system as well as the edge values of the orbital elements that define the station-keeping box (paragraph 2.4.2). The simulator works with the perturbations, therefore it is indispensable to deal with the mean orbital elements to carry out the maintenance.

The logic behind the method is shown below.

Station-Keeping Simulation

 1: **procedure** SIMULATE THE ABSOLUTE STATION-KEEPING OF A SPACECRAFT
 2:     **for** each time step in the time frame **do**
 3:         propagate the satellite's orbit
 4:         **if** current mean SMA < lower SMA **then**
 5:             **while** current mean SMA < upper SMA **do**
 6:                 activate thruster
 7:             **end while**
 8:         **end if**
 9:         **if** current mean inclination < lower inclination value **then**
10:             **while** current mean inclination < upper inclination value **do**
11:                 activate thruster
12:             **end while**
13:         **end if**
14:     **end for**
15: **end procedure**

The algorithm uses the Edelbaum-Kechichian theory reported in subsection 2.4.2. However, a careful observer would notice that actually the script takes advantage of this theory only to compute the yaw angle (equation 2.19) in case of out-of-plane maneuver (change in inclination). Indeed, the burning time estimated by the equations is never taken into account. The update of the satellite's position at each time step allows a better accuracy which otherwise would be affected by

the uncertainties generated by the unpredictable factors of the perturbing effects over a long time interval.

Nevertheless, the equations of the Kechichian's method are used to provide the ΔV required by the station-keeping, which is a fundamental parameter for the mission designer.

4.4 Revisit Time Collector

The revisit time collector (RTC) tool intends to compute the amount of passages of a satellite above a specific position on Earth. It collects the *access times* of the spacecraft when its target on the ground is within the area delimited by the swath width of the satellite's on-board camera. It can work with a single space vehicle or with a constellation. In the latter case, the script provides all the access times taking into account every single satellite of the pattern.

In addition to the input parameters required by the selected propagator (paragraph 4.1), the RTC needs the target coordinates in latitude and longitude, as well as the swath width of the electro-optical sensor. The algorithm computes the distance between the sub-satellite point on the ground and the target at each time step. This calculation is performed considering some meaningful aspects.

The goal of the RTC clearly requires the estimation of the satellite's position in an ECEF reference frame. If its motion was evaluated with respect to an ECI system, the ground track of the orbit would be always the same after each orbital period. ECEF coordinates shall be then converted into the geodetic latitude-longitude location of the spacecraft's subpoint in order to proceed with the distance calculation.

The computation of the distance between two points on the Earth surface shall take into account the sphericity of the planet. To accomplish this condition, the algorithm computes the so-called geodesic distance, which can be defined as

the shortest path between two points on the surface of an ellipsoidal model of the Earth. As regards the geodetics, the script takes advantage of the method proposed by Karney [37].

Summarizing, the algorithm extracts the sub-point latitude and longitude from the ECEF state vector of the satellite and, after that, calculates the geodesic distance with respect to the ground target. This path is then compared with half the value of the potential swath width (PSW) at each time step of the propagation. If the distance is smaller than PSW then the script saves the epoch related to that time step.

Revisit Time Collection

 1: **procedure** COLLECT REVISIT TIME EPOCHS OVER A SPECIFIC TARGET
 2:      propagate the satellite's orbit for the desired time frame
 3:      **for** each time step in the time frame **do**
 4:         convert satellite state vector from ECI to ECEF
 5:         extract geodetic latitude and longitude
 6:         compute the geodesic distance between target and sub-satellite point
 7:         **if** distance between the two points $< \frac{PSW}{2}$ **then**
 8:            collect the epoch of the current time step
 9:         **end if**
10:      **end for**
11: **end procedure**

It is important to underline that the actual sensor input value to be implemented in the RTC should be related to the concept of *potential swath coverage*. It identifies the potential path reachable by a pointable space system. If the satellite can not work off-nadir then the input parameter will be simply equal to half the SW due to the fact that the subpoint always lies in the middle of the observed area. Therefore, according to the figure 4.5, a proper use of the algorithm should get half the potential swath width (PSW) as input value.
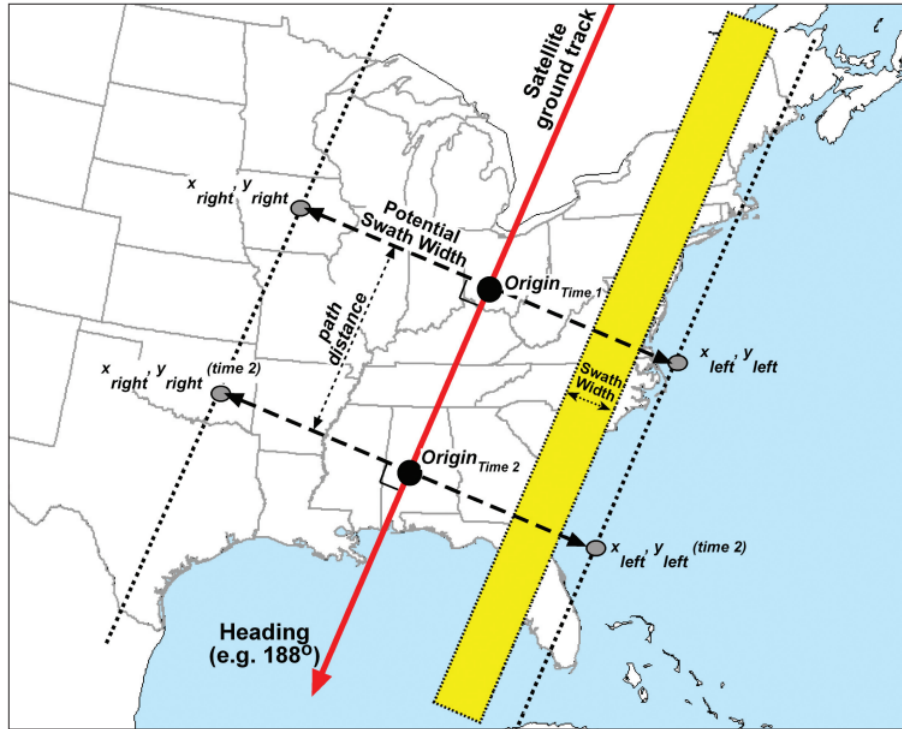
Figure 4.5: Projection of satellite ground track and potential swath width with respect to its swath width at two time steps [38].

Chapter 5

Case Studies

The tools developed by this thesis have been tested on four case studies. They can be classified in two groups: two examples of single satellite missions and two constellation scenarios respectively. For both categories, there is one mission accomplished and another planned for the next few years.

The projects in the pipeline relate to the objectives of Kuva Space and lay the foundations for the purposes of this research. The designed tools arise from the need of providing useful instruments to the management of a LEO SSO satellite constellation.

Despite this, single spacecraft simulations are still beneficial to assess the performances of the scripts. Indeed, they require much less processing time, which is advantageous in the software design phase, and serve as an optimal comparison model to evaluate the qualities of a constellation. Each case study is analysed with the tools that best suit the representative features of the respective mission.

5.1 Reaktor Hello World

5.2 Hyperfield 2nd Generation

The second generation of Hyperfield represents the evolution of Hyperfield-1a and Hyperfield 1-b. They consist of satellites whose platform features hyperspectral imagers operating in the visible-to-near-infrared (VIS-NIR) and visible-to-short-wave-infrared (VIS-SWIR) [39]. The first generation models are propulsion-less 6U CubeSats which will be operating on-orbit starting from middle year 2024. Hyperfield 2 (HF2) is still in the design phase, therefore the specifications selected within this work are not definitive yet. HF2 will be considered to be a microsatellite of around 60 kilograms, propelled by a low-thrust system of 7 millinewtons of force.

Spacecraft sizes	Mass: 60 kg
	Low-Drag Area: 0.05 m^2
	High-Drag Area: 0.15 m^2
Orbit	Type: SSO
	Altitude: 380 km
Mission life	5 years

Table 5.1: HF2 specifications

5.3 Doves Constellation

5.4 Hyperfield Constellation

The Hyperfield constellation will be the result of all the Hyperfield satellites launched starting from 2024. According to the requirements that satisfy the objective of Kuva Space of providing three times daily images from any location on Earth, the final number of spacecraft in orbit is around 100 [39].

Since the satellites will travel in different orbital planes in order to form a

Walker Delta pattern, the case study analysed consists of 15 vehicles operating in the same orbit.

Chapter 6

Analysis and Results

6.1 Reaktor Hello World Life Data

6.2 Hyperfield Orbit Maintenance Design

6.3 Planet Constellation Differential Drag Results

6.4 Kuva Constellation Management Results

Chapter 7

Conclusion

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Bibliography

[1] Craig A Kluever. *Space flight dynamics.* John Wiley & Sons, 2018.

[2] Michael Douglas Griffin. *Space vehicle design.* AIAA, 2004.

[3] David A. Vallado. *Fundamentals of Astrodynamics and Applications.* Microcosm Press, fourth edition, 2013.

[4] Charles D Brown. *Spacecraft mission design.* AIAA, 1998.

[5] Ian Ridpath. *A dictionary of astronomy.* Oxford Quick Reference, 2012.

[6] Kathleen Riesing. Orbit determination from two line element sets of iss-deployed cubesats. 2015.

[7] Howard D Curtis. *Orbital Mechanics for Engineering Students: Revised Reprint.* Butterworth-Heinemann, 2020.

[8] Mohammad Sadraey and Dr Müller. Drag force and drag coefficient. *M. Sadraey, Aircraft Performance Analysis. VDM Verlag Dr. Müller*, 2009.

[9] Bruce R Bowman, W Kent Tobiska, Frank A Marcos, and Cesar Valladares. The jb2006 empirical thermospheric density model. *Journal of Atmospheric and Solar-Terrestrial Physics*, 70(5):774–793, 2008.

[10] Bruce Bowman, W Kent Tobiska, Frank Marcos, Cheryl Huang, Chin Lin, and William Burke. A new empirical thermospheric density model jb2008 using

new solar and geomagnetic indices. In *AIAA/AAS astrodynamics specialist conference and exhibit*, page 6438, 2008.

[11] James R Wertz. Orbit & constellation design & management, 2nd printing. *Hawthorne, CA and NY: Microcosm Press and Springer*, 2009.

[12] Giorgio EO Giacaglia. Long-period perturbations in the inclination of sunsynchronous satellites. *Revista Brasileira de Ciencias Mecanicas (ISSN 0100-7386*, 16:583–589, 1994.

[13] HG Walter. Conversion of osculating orbital elements into mean elements. *The Astronomical Journal*, 72:994, 1967.

[14] Gim J Der and Roy Danchick. Conversion of osculating orbital elements to mean orbital elements. In *Flight Mechanics/Estimation Theory Symposium 1996*, 1996.

[15] David Arnas. Analytic transformation from osculating to mean elements under j2 perturbation. *arXiv preprint arXiv:2212.08746*, 2022.

[16] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Aerospace Systems*. 2002.

[17] Angel Rocha. *Numerical Methods and Tolerance Analysis for Orbit Propagation*. PhD thesis, San José State University, 2018.

[18] Ronald J Boain. Ab-cs of sun-synchronous orbit mission design. 2004.

[19] Xin Luo, Maocai Wang, Guangming Dai, Xiaoyu Chen, et al. A novel technique to compute the revisit time of satellites and its application in remote sensing satellite optimization design. *International Journal of Aerospace Engineering*, 2017, 2017.

[20] Lloyd Wood. Satellite constellation networks: The path from orbital geometry through network topology to autonomous systems. *Internetworking and computing over satellite networks*, pages 13–34, 2003.

[21] Carolina Lee Leonard. *Formationkeeping of spacecraft via differential drag.* PhD thesis, Massachusetts Institute of Technology, 1991.

[22] Cyrus Foster, James Mason, Vivek Vittaldev, Lawrence Leung, Vincent Beukelaers, Leon Stepan, and Rob Zimmerman. Differential drag control scheme for large constellation of planet satellites and on-orbit results. *arXiv preprint arXiv:1806.01218*, 2018.

[23] Blake D Mills. Satellite paradox. *American Journal of Physics*, 27(2):115–117, 1959.

[24] Megan R Johnson and Jeremy D Petersen. Maintaining aura's orbit requirements while performing orbit maintenance maneuvers containing an orbit normal delta-v component. In *International Symposium on Space Flight Dynamics*, number GSFC-E-DAA-TN14128, 2014.

[25] Theodore N Edelbaum. Propulsion requirements for controllable satellites. *Ars Journal*, 31(8):1079–1089, 1961.

[26] J KECHICHIAN. The reformulation of edelbaum's low-thrust transfer problem using optimal control theory. In *Astrodynamics Conference*, page 4576, 1992.

[27] ESA Space Debris Mitigation WG. Esa space debris mitigation compliance verification guidelines. ESA, 2023.

[28] V Braun, J Gelhaus, C Kebschull, N Sánchez-Ortiz, J Oliveira, R Dominguez, C Wiedemann, H Krag, and P Vörsmann. Drama 2.0-esa's space debris risk

assessment and mitigation analysis tool suite. In *64th International Astronautical Congress*, pages 23–27, 2013.

[29] Helge Eichhorn, Juan Luis Cano, Frazer McLean, and Reiner Anderl. A comparative study of programming languages for next-generation astrodynamics systems. *CEAS Space Journal*, 10:115–123, 2018.

[30] Ivelina Momcheva and Erik Tollerud. Software use in astronomy: an informal survey. *arXiv preprint arXiv:1507.03989*, 2015.

[31] Juan Luis Cano Rodríguez, Helge Eichhorn, and Frazer McLean. Poliastro: an astrodynamics library written in python with fortran performance. In *6th International Conference on Astrodynamics Tools and Techniques*, 2016.

[32] Juan Luis Cano Rodríguez and Jorge Martínez Garrido. poliastro: a python library for interactive astrodynamics. 2022.

[33] Darrel J Conway and Steven P Hughes. The general mission analysis tool (gmat): Current features and adding custom functionality. In *International Conference on Astrodynamics Tools and Techniques (ICATT)*, number LEGNEW-OLDGSFC-GSFC-LN-1107, 2010.

[34] Vitali Braun, Quirin Funke, Stijn Lemmens, and Silvia Sanvido. Drama 3.0-upgrade of esa's debris risk assessment and mitigation analysis tool suite. *Journal of Space Safety Engineering*, 7(3):206–212, 2020.

[35] Cyrus Foster, Henry Hallam, and James Mason. Orbit determination and differential-drag control of planet labs cubesat constellations. *arXiv preprint arXiv:1509.03270*, 2015.

[36] Zachary Thomas Lee. *CubeSat constellation implementation and management using differential drag*. PhD thesis, Massachusetts Institute of Technology, 2017.

[37] Charles FF Karney. Algorithms for geodesics. *Journal of Geodesy*, 87:43–55, 2013.

[38] Michael E Hodgson and Bandana Kar. Modeling the potential swath coverage of nadir and off-nadir pointable remote sensing satellite-sensor systems. *Cartography and Geographic Information Science*, 35(3):147–156, 2008.

[39] Tuomas Tikka, Jussi Makynen, and Michal Shimoni. Hyperfield-hyperspectral small satellites for improving life on earth. In *2023 IEEE Aerospace Conference*, pages 1–8. IEEE, 2023.