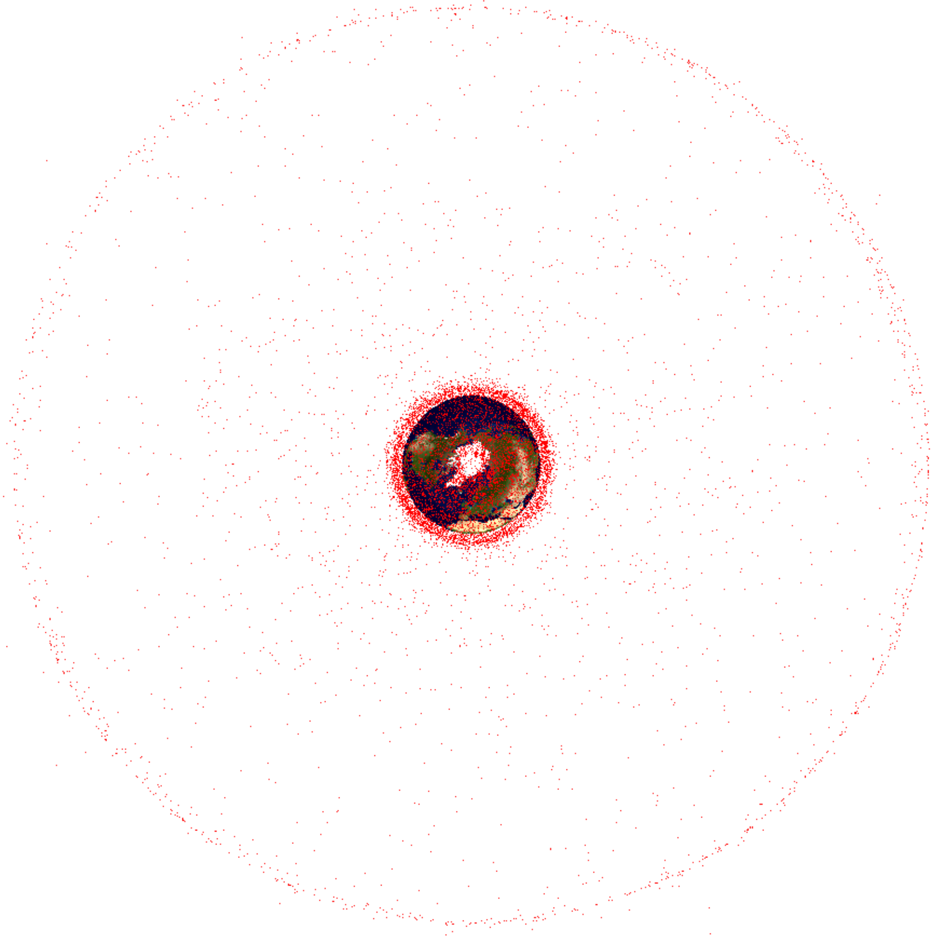
MSc Thesis Report

Filtering Techniques for Orbital Debris Conjunction Analysis

applied to SSN TLE catalog data



Jonatan Leloux, BSc

February 8, 2012

Delft University of Technology

Delft Aerospace – Faculty of Aerospace Engineering

Astrodynamics & Satellite systems

MSc Thesis Report

Filtering Techniques for Orbital Debris Conjunction Analysis

applied to SSN TLE catalog data

Jonatan Leloux, BSc

February 8, 2012

Jonatan Leloux, BSc
E-mail: j.leloux@student.tudelft.nl, jleloux@gmail.com

Delft University of Technology
Faculty: Aerospace Engineering
Department: Space Engineering
Chair: Astrodynamics and Space Missions

Building 62, Room 9.06
Kluyverweg 1, 2628 CX Delft
The Netherlands

Cover figure:

All 14,394 objects of the catalog of 2011.05.05 am, propagated to the time instance 2011.05.05 12:00:00, with a North pole view, clearly showing the crowded Geostationary-Earth Orbit (GEO) ring. Note: the objects are merely drawn to position, not to scale.

This document was generated with LaTeX 2_ε code used by the TeX typesetting system, including the following packages: `amsmath`, `booktabs`, `graphicx`, `hyperref`, `longtable`, `mathabx`, `natbib`, `pdflscape` and `pdfpages`.

Preface

This document describes the Final Thesis Work (AE5-006, 42 ECTS) on improvements to filtering techniques for orbital debris conjunction analysis, performed as part of my Master of Science (MSc) study in Space Engineering (SpE) at the Faculty of Aerospace Engineering (AE) of Delft University of Technology (DUT). Included is also a summary of the Literature Research (AE4-005, 18 ECTS) (Leloux, 2010a) which was performed beforehand on the same subject. The thesis work mainly consists of setting up a new and more efficient orbital debris conjunction analysis system in C++, which is partly integrated in the TUDelft (Dutch acronym for DUT) Astrodynamics Toolbox (Tudat) as well.

The first two chapters might be read by a general audience, but for the chapters thereafter knowledge of physics and mathematics and a basic knowledge of astrodynamics is required. The latter can also be gained in Chapter 3, in which a general overview of the fundamental concepts of astrodynamics needed for this research is given. Chapter 2 might be skipped if the reader already has sufficient general knowledge of orbital debris. The literature research also contained a chapter on collision probability (not included here), which might be particularly useful for (DUT) students who want to do thesis work on orbital debris collision probability, following up on and using my work on conjunction analysis.

I would like to take this opportunity to express my gratitude to my thesis supervisor ir. Ron Noomen, who has readily provided me with valuable help and information during the weekly appointments with an hour of feedback, guiding me through the entire process of completing this thesis. Furthermore, I want to thank my colleague MSc students in rooms 9.06 and 9.03 for the fun we had and the fresh insight and help we exchanged during our work on the ninth floor of our faculty. My performance surely accelerated from the moment I started working there during the course of my literature research. Finally, I would like to express a thank-you to my girlfriend and family for their unconditional support outside the work, during good times and bad.

Jonatan Leloux
Delft, February 8, 2012.

Contents

Abstract

The steadily growing amount of orbital debris increases the probability and amount of collisions between two objects in orbit about the Earth. These collisions in turn create even more debris, and it is therefore important to keep track of future conjunctions. The U.S. Space Surveillance Network (SSN) uses ground- and space-based sensors to observe and track objects of about 10 cm and larger, of which the orbital information is coded in Two-Line Element (TLE) sets and listed in a catalog currently containing about 20,000 objects, which is partly distributed to the public.

Using this TLE data, the Simplified General (and Deep-space) Perturbations 4 (SGP4/SDP4) analytical propagator is used to propagate the orbits of these objects, and includes secular, long-period and short-period perturbations due to the Earth's gravity field including J_2, J_3 and J_4 and resonance effects for 12- and 24-hr orbits, as well as perturbations due to atmospheric drag, solar radiation, and gravitational attraction of the Sun and the Moon.

The propagated orbits are used to predict conjunctions of pairs of objects. However, due to the large and increasing amount of objects in the catalog, numerically analysing all pairs would be too time-consuming. Therefore, numerous fast filters and sieves were designed to limit the search space of conjunction analysis, by discarding object pairs that are proven to never be able to conjunct.

Four implementations of the classical perigee-apogee filter, next to six sieves with a new fine conjunction detection method, were analysed, implemented, and tested in terms of performance. The filter makes use of the altitude band of an object, and can be applied pre-hand. A method based on minimum and maximum radius determined from ephemerides, was found to be the most accurate and reliable in long-term application, while being able to be fine-tuned to the performance needs of a conjunction analysis process.

Increasingly complex sieves are subsequently applied to the ephemerides at each time instance in a time interval, in order to efficiently discard object pairs. Eight possible improvements to the underlying theory and application of these sieves were made, resulting in one optimal combination of these improvements, and eventually resulting in a conjunction analysis system that is almost three times as fast as the best found reference method.

Nomenclature

The nomenclature used in this research include symbols, subscripts and acronyms.

For the symbols, which are typeset in *italic*, a description is given, next to the corresponding physical unit or dimension in SI-units between brackets, if applicable. Vectors are typeset non-italic and **bold**, of which only the scalar version is explained here. Hats ($\wedge$) on top of a certain vector indicate unit vectors, unless indicated otherwise. First and second time derivatives of a parameter are represented by one or two dots on top respectively (for example: $\dot{x}$ and $\ddot{x}$).

Furthermore, subscripts that are added to symbols to give them a certain meaning, or to indicate to which other parameter they are subjected, are described below as well.

Finally, the meaning of all acronyms and abbreviations used in this document is listed. They are explained in the text only once, at the moment they are used first.

Note that sometimes multiple symbols exist for the same meaning, this is due a literal dependency on the respective references found. The notations used in the references are used in the same form in this document, when treating the theory developed in these respective publications.

Symbols

$-$	TLE column with either a plus or minus sign
$+$	TLE column with either a minus or plus sign or a space
a	Acceleration [m/s^2]
a	Semi-major axis of an ellipse [m]
a	Analysed
A	Area, Area of a surface element [m^2]
A	Point on long axis of symmetry of an ellipse
A	TLE column with any character A-Z or a space
A	Azimuth angle [rad]
AR	Ellipse Aspect Ratio
B	Point on short axis of symmetry of an ellipse

b	Semi-minor axis of an ellipse [m]
B	Ballistic coefficient [kg/m^2]
B	Conjunction plane
B^*	SGP4-type drag coefficient [R$_E^{-1}$]
c	Integration constant
c	Speed of light = 299,792,458 [m/s]
c	Threshold distance, Critical distance [m]
C	Centre of an ellipse
C	Coefficient
C	TLE Classification
d	Mathematical derivative
d	Distance of closest approach of two ellipses [m]
d	Ellipsoid diameter [m]
d	Distance [m]
D	Conjunction distance [m]
e	Eccentricity (of a conic section)
E	Energy [J]
E	Eccentric anomaly [rad]
f	Perturbing acceleration [m/s^2]
$f()$	Function of the variable(s) between brackets
F	Force [N]
F	Focus (of an ellipse)
g	Gravitational acceleration / field strength [m/s^2]
G	Universal gravitational constant = 6.672-6.676$\cdot 10^{-11}$ [m^{3}kg^{-1}s^{-2}]
G	Projection point of P on a of an ellipse
h, H	(Orbital) angular momentum [Nms], [kgm^{2}s^{-1}], [Js]
h	Height above the Earth's surface [m]
h	Time interval time step [s]
h	Elevation angle [rad]
i	Inclination [rad]
i, j, k	Summation counters (of bodies)
J	Gravity field coefficient
k	Distance from F to l [m]
k	Ellipsoid threshold factor
k	Integer counter
l	Fixed line
m, M	Mass (of a satellite) [kg]
m	Object diameter [m]
M	Mean anomaly [rad]
n	Number (of objects)
n	Mean (angular) motion [rad/s], [rev/day]
n	Number of objects

N	Position of ascending node [m]
N	TLE column with any number 0-9 or a space
N	Nutation
N	Time step number
p	Semi-latus rectum [m]
p	Amount of object pairs
P	Patch distance
P	Point
P	Legendre polynomial (one subscript) or Legendre function (two subscripts) of the first kind
P	Precession
P	Probability
q	Larger of the two perigees of a conjunction pair [m]
Q	Smaller of the two apogees of a conjunction pair [m]
r	Rejected
r, R	Radius, Distance, Position [m]
R	Perturbing potential [m^2/s^2]
R	Mean equatorial radius of the Earth [m]
R	Rotational axis
S	Cross-sectional area perpendicular to $\mathbf{V}$ [m^2]
S	TLE Classified data
S^*	Effective cross-sectional area in the direction of the radiation [m^2]
t	Time [s]
T	Orbital period [s]
T	Time step [s]
T	Time interval [s]
T	Mathematical transpose of a matrix
u	Argument of latitude [rad]
U	Gravitational potential [m^2/s^2]
U	TLE Unclassified data
$u, v, w/U, V, W$	Radial, Along-track and Out-of-plane direction
v, V	Velocity [m/s]
W	Power density of incoming radiation [W/m^2]
x	Variable in the Legendre polynomial and function
x	Scalar of state-vector of a satellite [m], [m/s]
x, y, z	Coordinates in a reference frame [m]
X, Y, Z	Reference frame directions
z	Scalar of the unit vector perpendicular to the equatorial plane of the Earth [m]

α	Angle between $\mathbf{r}_d$ and $\mathbf{r}_s$ [rad]
α	Orbital element
∂	Mathematical partial

Δ	Increment or change of a certain parameter
∇	Mathematical nabla operator
θ	True anomaly [rad]
Λ	Geographical longitude [rad]
μ	Gravitational parameter [km^{3}s^{-2}]
π	Mathematical constant
ρ	Atmospheric density [kg/m^3]
ρ	Range between conjunction object pair [m]
σ	Standard deviation
τ	Time of passage of pericenter [s]
ϕ	Angle between $\mathbf{r}_i$ and a reference direction [rad]
ϕ	Geocentric latitude [rad]
ω	Integration constant
ω	Argument of perigee [rad]
Ω	Longitude or Right ascension of the ascending node [rad]
♈	Vernal Equinox (direction vector)

Subscripts

0	Initial epoch
1,2,..	Counter
a	Apocenter
acc	Acceleration
alb	Albedo
app	Approach
c	Circular
c	Chaser object of a conjunction pair
c	Chaser and Risk object combined
c	Collision
$closest, con$	Closest approach of a conjunction
cr	Critical
d	Perturbing body
dfe	Days from (TLE) epoch
D	Aerodynamic drag
E	Earth
esc	Escape
i, j, k	Summation counters (of bodies)
IR	Infrared Radiation
k	Kinetic
k	Iteration counter
$loop$	Looping with SGP4
m	Main

max	Maximum
$mean$	Mean of an orbital element
min	Minimum
n, m	Summation counters of the Earth's gravitational potential
p	Potential
p	Pericenter, Perigee
p	Primary object
R	Reflectivity
r	Risk object of a conjunction pair
rel	Relative
s	Satellite
s	Secondary object
S	Sun
$skip$	Skipping of a variable
sph	Sphere
t	Target object of a conjunction pair
tca	Time of closest approach
th	Threshold
$th, fine$	Fine threshold
x, y, z	Components of a vector in the direction of X, Y, Z
$*$	Summation excluding $j = i$ and/or $j = k$

σ	Standard deviation

Acronyms

AAS	American Astronautical Society
AE	Aerospace Engineering
AFSPC	Air Force Space Command
AIAA	American Institute of Aeronautics and Astronautics
AS	Astrodynamics and Satellite Systems (Chair of DUT, now called Astrodynamics and Space Missions)
ASAT	Anti-SATellite weapon
ASCII	American Standard Code for Information Interchange
ATV	Automated Transfer Vehicle
BBS	Bulletin Board System
BSc	Bachelor of Science
CCD	Charge-Coupled Device
Closeap	Close Approach tool
CNES	Centre National d'Etudes Spatiales (French Space Agency)
COPUOS	Committee on the Peaceful Uses of Outer Space
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit (computer processor)

CRASS	Collision Risk Assessment Software
DEB	Debris
DEBIE	Debris in-Orbit Evaluator
DISCOS	Database and Information System Characterising Objects in Space
DoD	United States Department of Defense
DUT	Delft University of Technology
ECI	Earth-Centered Inertial (reference frame)
ECTS	European Credit Transfer and Accumulation System
EOL	End-Of-Life
ERS	European Remote Sensing satellite
ESA	European Space Agency
ESOC	European Space Operations Centre (ESA)
ESTEC	European Space Research and Technology Centre (ESA)
EURECA	EUropean REtrievable CArrier
FGAN	ForschungsGesellschaft für Angewandte Naturwissenschaften (Research Establishment for Applied Sciences)
GEO	Geosynchronous/Geostationary Earth Orbit
GHA	Greenwich Hour Angle
GLONASS	GLObal'naya NAvigatsionnaya Sputnikovaya Sistema (GLObal NAvigation Satellite System)
GMV	Grupo Mecanica Vuelo (Flight Mechanics Group)
GORID	Geostationary Orbit Impact Detector
GPS	Global Positioning System
GRAVES	The French Space Surveillance System
GSFC	Goddard Space Flight Center (NASA)
GTO	Geosynchronous/Geostationary Transfer Orbit
HAX	Haystack Auxiliary Radar
HEO	Highly Elliptical Orbit
HST	Hubble Space Telescope
IADC	Inter-Agency Space Debris Coordination Committee
ID	Object Identification Number
Int.	International
IRAS	Infra-Red Astronomical Satellite
ISO	International Organization for Standardization
ISS	International Space Station
JGM	Joint Gravity Model
JD	Julian Date
LAGEOS	LAser GEOdynamics Satellite
LDEF	Long Duration Exposure Facility
LEO	Low Earth Orbit
LMT	Liquid Mirror Telescope
LRIR	Haystack Long-Range Imaging Radar
MASTER	Meteoroid and Space Debris Terrestrial Environment

	Reference
MEO	Medium Earth Orbit
MIT	Massachusetts Institute of Technology
MSc	Master of Science
MSX	Mid-Course Space Experiment
NASA	National Aeronautics and Space Administration (U.S.)
NAVSPASUR	Naval Space Surveillance System
NODO	NASA Orbital Debris Observatory
No.	Number (of)
NORAD	North American Aerospace Defense Command
NSSCC	National Space Surveillance Control Center (U.S.)
NSSDC	National Space Science Data Center
PPT	Position and Partials as functions of Time
PROBA	Project for On-Board Autonomy
R/B	Rocket Body
RMS	Root Mean Square
SatCat	Satellite Catalog
SBSS	Space-Based Space Surveillance System
SBV	Space-Based Visible instrument (of MSX)
SDP	Simplified Deep-space Perturbations satellite orbit model
SeaWiFS	Sea-viewing WIde Field-of-view Sensor
SGP	Simplified General Perturbations satellite orbit model
SMM	Solar Maximum Mission
SOCRATES	Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space
SpE	Space Engineering
SPADATS	Space Detection and Tracking System
SRM	Solid Rocket Motor
ss	smart sieve
SSN	Space Surveillance Network
STR	SpaceTrack Report
STS	Space Transportation System
STSC	Scientific and Technical Subcommittee
TEME	True Equator and Mean Equinox
TIRA	Tracking and Imaging RAdar
TLE	Two-Line Element (set)
Tudat	TUDelft Astrodynamics Toolbox
TUDelft	Technische Universiteit Delft (Dutch for DUT) (Delft University of Technology)
U.S./USA	United States (of America)
UCS	Union of Concerned Scientists
UN	United Nations
UNOOSA	United Nations Office for Outer Space Affairs
USSTRATCOM	United States Strategic Command

UT	Universal Time
UTC	Coordinated Universal Time
WDC	World Data Center
WGS	World Geodetic System

Chapter 1

Introduction

Since the beginning of spaceflight, October 1957, the amount of man-made debris in orbit about the Earth has been increasing in number as well as in mass and volume. This so-called orbital debris poses a hazard to operating satellites, because, due to the relative velocities involved, a collision with debris could prove to be catastrophic for a mission.

Just recently, on the 10th of February, 2009, the operating Iridium 33 satellite collided with the deactivated Cosmos-2251 satellite, in turn creating large clouds of new debris. A depiction of the evolution of these debris clouds can be seen in Figure 1.1. Another recent event which caused a significant increase in orbital debris was a Chinese anti-satellite missile test, on the 11th of January, 2007. The quantification of these events will soon follow below.

The United States (U.S.) Space Surveillance Network (SSN) observes and tracks orbital debris and lists information on the orbits of the objects in a catalog, which currently consists of about 20,000 objects (Stansbery, 2009) of 10 cm and larger. This orbital information is listed using Two-Line Element (TLE) sets, which follow a certain code to compactly state all orbital elements and other information. A depiction of the relative position of these objects can be seen in Figure 1.2, for the Low Earth Orbit (LEO) region.

The current ground-based network of radar and optical telescopes can track objects of about 10 cm and larger in LEO and objects of about 30 cm and larger in GEO, respectively (Klinkrad, 2009).

The catalog increased by 60% from 2007-2010, of which 80% was due to the aforementioned two collision events (Kelso, 2010b). Due to these and other events, the amount of orbital debris steadily increases if no mitigation measures are taken, see Figure 1.3.

The trend shown in the figure was even too optimistic, since it predicted that currently there would be about 16,000 objects of 10 cm and larger,

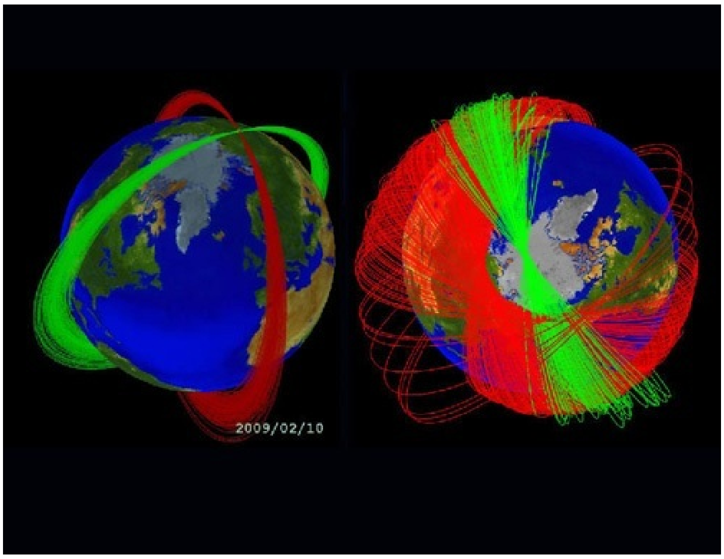


Figure 1.1: Orbital evolution of the debris fragment cloud created by the collision between the Iridium 33 and Cosmos 2251 satellites on the 10th of February, 2009. (Stansbery, 2009)

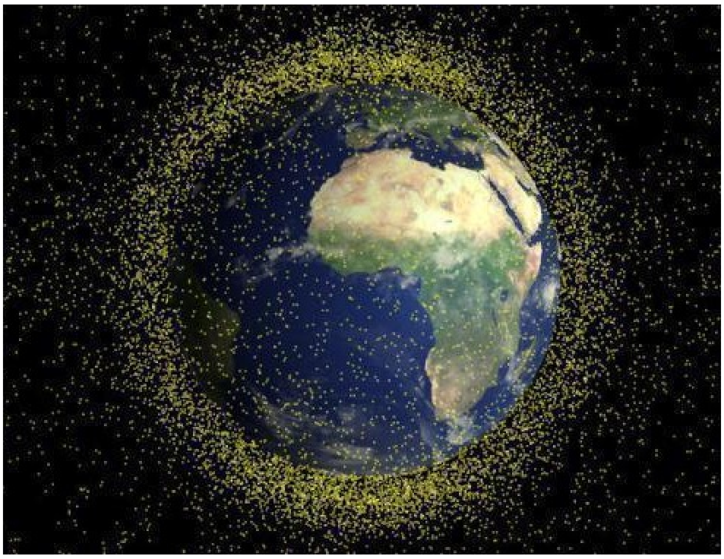


Figure 1.2: Orbital debris in Low-Earth Orbit (LEO), objects' size not to scale. (ScienceDaily, 2008)

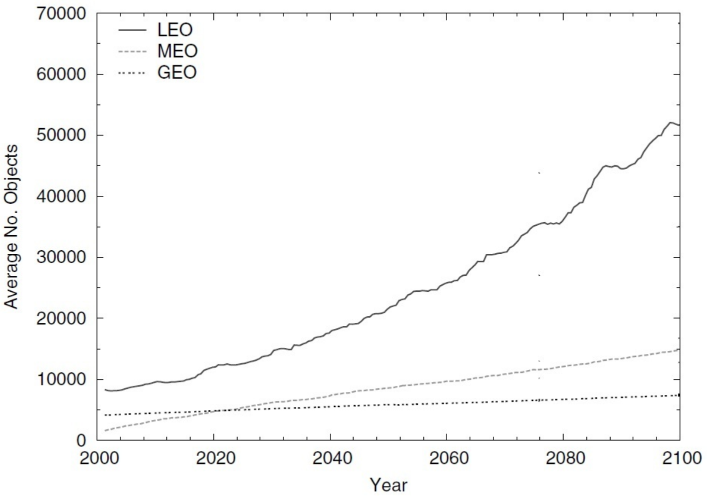


Figure 1.3: Expected evolution of the orbital debris population of objects larger than 10 cm in the LEO, Medium Earth Orbit (MEO), and GEO regimes, for a business-as-usual scenario, over a 100-year prediction timespan. (Klinkrad, 2006)

while there is a significant larger amount (20,000) with that same size range in the catalog at present.

A depiction of the evolution of the amount of objects in the GEO region in 2112, can be seen in Figure 1.4. The term 'business-as-usual' used here, is what happens when no future orbital debris growth mitigation procedures are followed, which will be explained more deeply in the following chapter.

The catalog will also grow due to ever increasing accuracy and capability of the SSN. Very recently, on the 25th of September 2010, the first 'Pathfinder' satellite of the Space-Based Space Surveillance System (SBSS) was launched, which will increase the capability of the SSN, in terms of the amount of orbital debris observed, by a factor two, see also the essay on SBBS Mechatronics by Leloux (2010b).

The orbital information of the objects in the catalog can be used to propagate the simulated position of these objects forward in time. Knowing where the objects will be in the future, this information can be used to predict conjunctions of pairs of objects.

If the propagated positional error of the objects is also known, the probability of collision of the object pair which is in conjunction might even be determined. If this collision probability is above a certain threshold, a satellite operator could decide to execute an avoidance manoeuvre to decrease this probability, for satellites with thrusting capability.

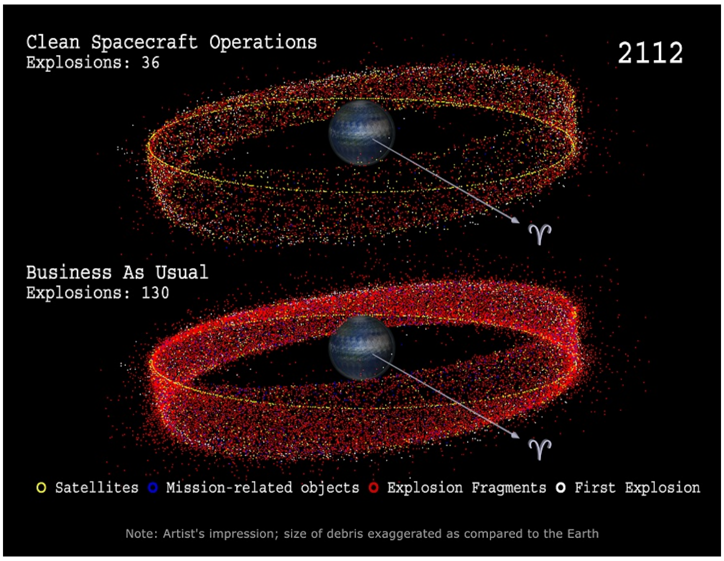

Clean Spacecraft Operations
Explosions: 36
2112
♈
Business As Usual
Explosions: 130
♈
○ Satellites ○ Mission-related objects ○ Explosion Fragments ○ First Explosion
Note: Artist's impression; size of debris exaggerated as compared to the Earth

Figure 1.4: Evolution of the Geostationary-Earth Orbit (GEO) debris ring in 2112 for a business-as-usual scenario compared to clean spacecraft operations. (Klinkrad, 2009)

A depiction of a conjunction event can be seen in Figure 1.5, including the path, velocity vectors and error ellipsoids of the objects.

Conjunction analysis could also be used to predict the conjunction and thus collision trend in the future. Figure 1.6 shows such a trend, in which background events are collisions involving only launch- and mission-related objects, and explosion fragments, while uncontrollable feedback events are collisions involving previously generated collision fragments.

This prediction is not far off at present, since the first catastrophic collision of an operating satellite now has taken place. However, 1 collision at the beginning of this trend does not really say anything yet, being a very poor statistic.

However, due to the size of the catalog which is ever increasing, a very large and increasing number of object pairs need to be analysed for conjunctions. This could be done by simply propagating all objects in time and numerically stepping forward in time to determine the closest approach distance of any object pair, but this would take too much computational time.

So, in order to make this process more efficient, some clever filters or sieves can be set up which decrease the search space by discarding certain object pairs which are proven to never be able to have a conjunction in the time interval under analysis.

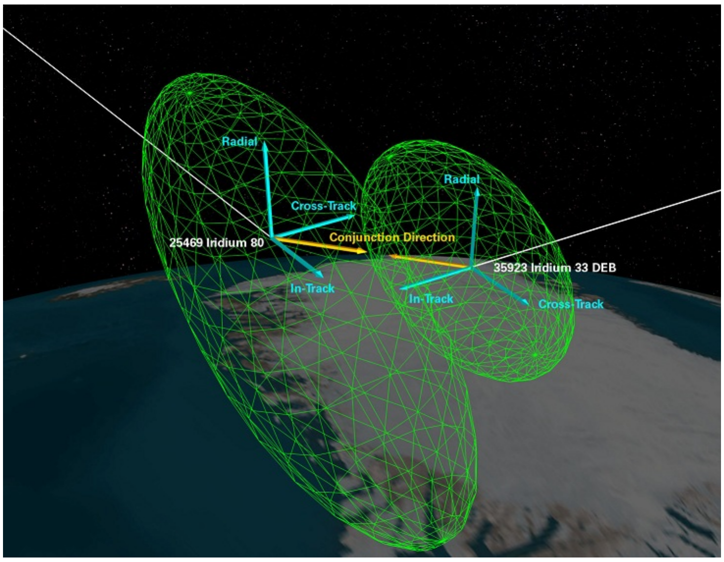


Figure 1.5: Conjunction geometry of an Iridium satellite and an Iridium debris object including their 1-σ position error ellipsoids. (Hall et al., 2010)

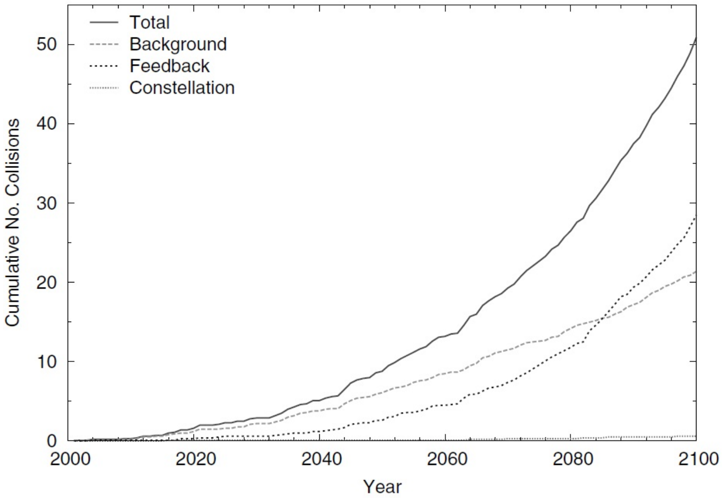


Figure 1.6: Expected cumulative number of catastrophic collisions in LEO for a business-as-usual scenario, over a 100-year prediction timespan, discriminated by collision type. (Klinkrad, 2006)

The purpose of this thesis is to set up a complete orbital debris conjunction analysis system, while implementing as many improvements as possible with respect to current systems, resulting in a method with unprecedented efficiency and thus a smaller computational effort.

To do so, insight is needed into orbital debris in general, as well as into the SSN, its catalog and the orbital propagation of the objects. Furthermore, the fundamental concepts of astrodynamics used in the theory of this propagation should be understood. Subsequently, a large amount of orbital debris conjunction filters and sieves that are designed up until now, and are in use today, should be thoroughly examined and their performance evaluated.

The structure of this document is as follows.

Orbital debris will be discussed in Chapter 2, including general information like its history, sources, distribution, re-entry and consequences. Furthermore, observing and tracking debris using ground-based, space-based and in-situ sensors as well as measuring its consequences will be included. The chapter will finish with solutions to the problems orbital debris poses, covering protection, growth mitigation and international cooperation.

Chapter 3 then follows with a description of the fundamental concepts and theory of astrodynamics, including its history, the many-body and two-body problem, elliptical orbits and their perturbed version due to various perturbing forces, as well as elementary analysis thereof.

This description includes all aspects needed to understand the propagator of the objects in the catalog set up by the SSN, which is treated in Chapter 4. This chapter comprises information on the history of the SSN, how to retrieve and to interpret TLE data, the SGP4 TLE propagator currently in use by the SSN, and the reference frame and accuracy of the propagated TLE data.

Orbital debris conjunction analysis is treated in Chapter 5, in which all methods designed and in use until now are discussed, as well as their performance.

Then, in Chapter 6, the implementation of a TLE reader is discussed, which can read TLE catalog files, save their parameters and check their validity, as well as determine new variables from these parameters, and save them in memory. Furthermore, it is able to selectively write any of these (derived) variables to a file, and do statistical analysis.

Following the TLE reader is a description of the implementation of the SGP4 Propagator, in Chapter 7. A handler with all the required functionality for the conjunction analysis process was written, to function with the SGP4 core code, which was found online. Input options like the time parameter type and interval, the SGP4 version and constants used, as well as output parameters, like time, Cartesian state vector and orbital parameters, are all specified here.

Finally, the implementation and performance of the filters and sieves is treated in Chapter 8. An overview of the four variations of the perigee-apogee filter designed in this thesis is given. Furthermore, the performance of the original sieves is compared with respect to several improvements made to their underlying theory and application. At last, the time step of both the setup of the perigee-apogee filter as well for the sieves is optimised, and the results are compared to references.

Chapter 2

Orbital Debris

Orbital debris is defined as all objects, or parts thereof, created by humans, that are currently orbiting the Earth or re-entering Earth's atmosphere, and serve no useful purpose any more (Klinkrad, 2009)(Stansbery, 2009). Other phrases frequently used include: space debris, space junk or space waste.

Orbital debris thus does not include micrometeorites, which mostly have smaller dimensions but higher relative velocities, but will be treated here shortly as well.

This chapter will provide general information about orbital debris in Section 2.1, after which the observing, tracking and measuring of this debris is described in Section 2.2, and concludes with possible solutions to the problems this debris poses in Section 2.3.

2.1 General Information

This section will first discuss the history and current status of orbital debris, after which its sources are explained. Then the spatial distribution of orbital debris and its decay and re-entry will be treated, after which the section finishes with a discussion of the negative consequences of orbital debris.

2.1.1 History

Since the launch of the first satellite, Sputnik 1, on the 4th of October 1957, in total 5092 launch attempts (McDowell, 2010) have placed 6236 satellites (Kelso, 2010a) in orbit, of which only 928 (UCS, 2010) are functioning at the moment of writing. Due to these launches, an estimated number of $2 \cdot 10^4$ objects larger than 10 cm, $6 \cdot 10^5$ objects larger than 1 cm, more than $3 \cdot 10^8$ objects larger than 1 mm, and $3 \cdot 10^{13}$ objects larger than 1 μm are orbiting the Earth at present (Klinkrad, 2009).

Of these objects currently only less than 15,000 are regularly tracked and cataloged by the Space Surveillance Network (SSN) of the United States, of which data is made publicly available (USSTRATCOM, 2004). However, the catalog only includes objects of about 5-10 cm or larger in LEO and 30-100 cm or larger in GEO, since smaller debris cannot be tracked by current technology, see Section 2.2.

The sources and amount of cataloged objects with respect to time can be seen in Figures 2.1 and 2.2, sorted by country and category respectively. These figures also show that the total amount of large objects has increased drastically since the beginning of spaceflight. The total amount of orbital debris mass and cross-sectional area in orbit with respect to time show similar results (Klinkrad, 2006).

In Figures 2.3 and 2.4 the amount, type (rocket bodies and satellites are depicted differently) and spatial distribution of the debris with respect to the Earth are depicted. Since most satellites are launched to LEO and GEO, most debris can also be found there.

2.1.2 Sources

Apart from the operative satellites, there are a number of sources and events responsible for the total amount of orbital debris orbiting the Earth. Summarizing the most important sources and events found in (Klinkrad, 2009), (Klinkrad, 2006) and (Wikipedia, 2010c):

Decommissioned satellites When a satellite is not useful any more its mission is ended and no contact will be made with it any more. However, it will still keep on orbiting the Earth.

Rocket upper stages To inject a satellite into orbit the upper stage of a rocket will need to have the same speed as the satellite at the start of its first orbit. The upper stage will thus also stay in orbit.

Mission related objects Some objects are needed only at the start of the mission and are then released into space. Examples include: launch adapters that connect the satellite to the upper rocket stage, apogee kick-motors that propel a satellite into Geostationary Transfer Orbit (GTO), yo-yo de-spin weights that are rolled out in order to stabilize the initial spin of the satellite, covers to protect optical instruments from contamination during launch, clamp-bands, etc.

Lost equipment During extra-vehicular activities of astronauts, some equipment has been lost over the years like screwdrivers, protective gloves, cameras, wrenches, pliers, tool bags, garbage bags jettisoned from Mir and even a toothbrush!

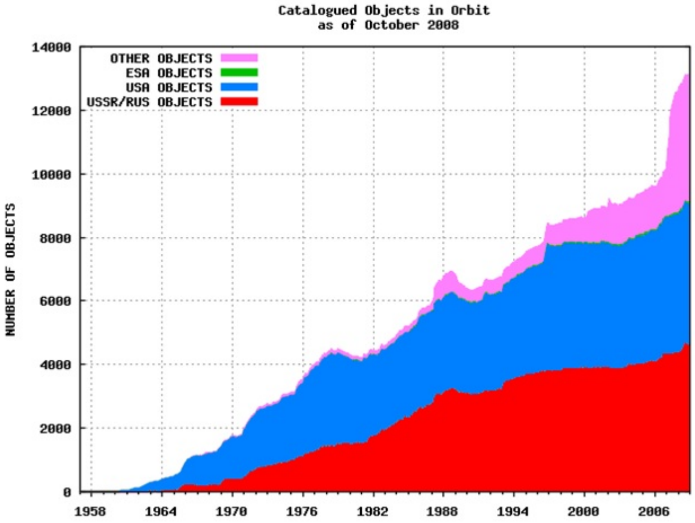


Figure 2.1: Amount and distribution of the cataloged objects in orbit according to source country versus time. (Klinkrad, 2009)

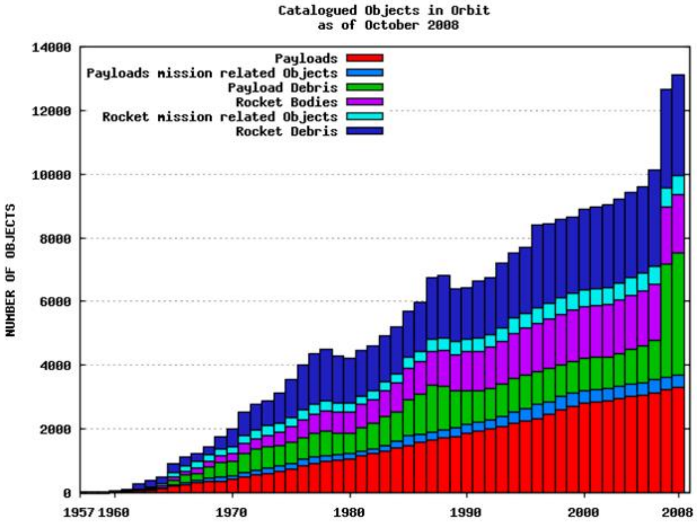
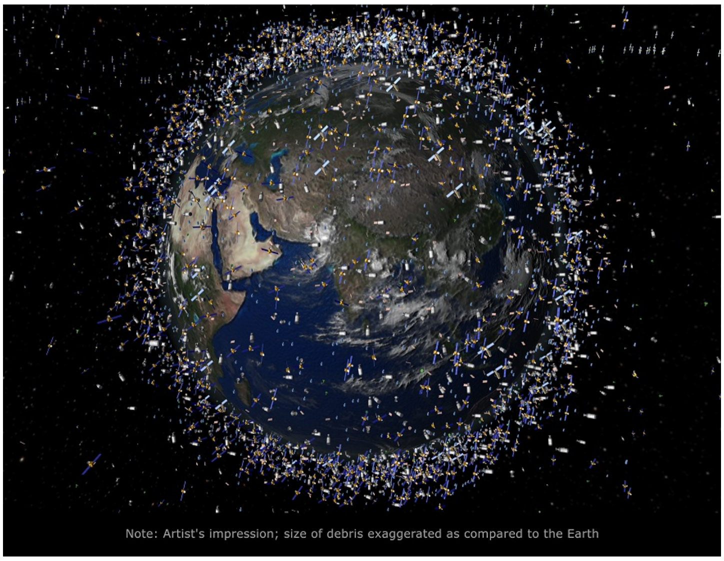
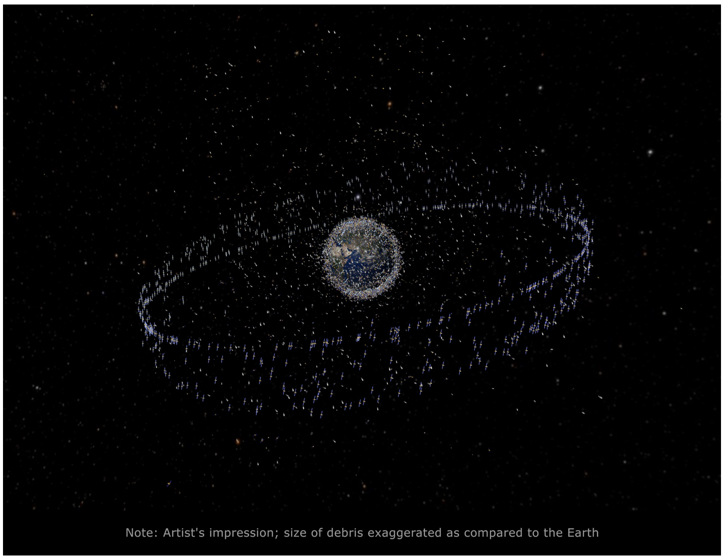


Figure 2.2: Amount and distribution of the cataloged objects in orbit according to source category versus time. (Klinkrad, 2009)



Note: Artist's impression; size of debris exaggerated as compared to the Earth

Figure 2.3: Depiction of the amount, type (rocket bodies and satellites are depicted differently) and spatial distribution of the debris with respect to the Earth for the LEO region. (Klinkrad, 2009)



Note: Artist's impression; size of debris exaggerated as compared to the Earth

Figure 2.4: Depiction of the amount and spatial distribution of the debris with respect to the Earth for the GEO region. (Klinkrad, 2009)

Explosions Residual fuel in tanks and fuel lines of rocket stages in combination with the degradation of these parts due to the harsh space environment has led to leakage and mixing of these fuels which have caused almost 200 explosions resulting in large clouds of small debris fragments. Other sources of explosions are satellite self-destructs.

Collisions About 10 accidental and intentional collisions have taken place, creating large clouds of debris. Included are collisions of large debris objects with an active satellite or other debris, and anti-satellite tests performed by the U.S., the Soviet Union and China, destroying satellites with missiles.

Mission resulting small debris Slag and dust particles are created due to the firing of solid rocket motors (SRM) of rocket upper stages, apogee kick-motors or satellites. Furthermore, the release of reactor coolant of reconnaissance satellites has resulted in debris in the form of small frozen droplets. Another source has been the release of a large number of thin copper wires for Project West Ford.

Surface degradation Due to radiation and impacts of small debris and micro-meteoroids, the surface and coatings of objects in space crack and degrade, creating very small debris particles.

Of the tracked objects in the SSN catalog 38% are decommisioned satellites, upper stages or mission-related objects, and 6% are functioning satellites. The other 56% are fragments created mainly by explosions, and for a small part by (un)intentional collisions.

Furthermore, the majority of the approximately $6 \cdot 10^5$ objects larger than 1 cm that can not be tracked are also debris fragments created by these explosions and collisions.

However, for objects of 1 mm or smaller, the amount of natural micro-meteoroids is larger than the mad-made orbital debris (Klinkrad, 2009). These micro-meteoroids which are very common in space are pieces of rock from the early stages of the solar system and can attain relatively velocities significantly higher than man-made orbital debris.

In Figure 2.5 a clear distinction of the sources of orbital debris and meteoroids is made with respect to their size regime. The white part of the bar for Sodium-Potassium droplets indicates the size domain for droplets that have already entered the Earth's atmosphere, according to computer simulations (Vorsmann, 2008).

In 2007, a Chinese anti-satellite test increased the catalog population by 25%. In February 2009, two satellites collided which created an estimated 1000 pieces of larger than 10 cm. Both events took place around 800 km

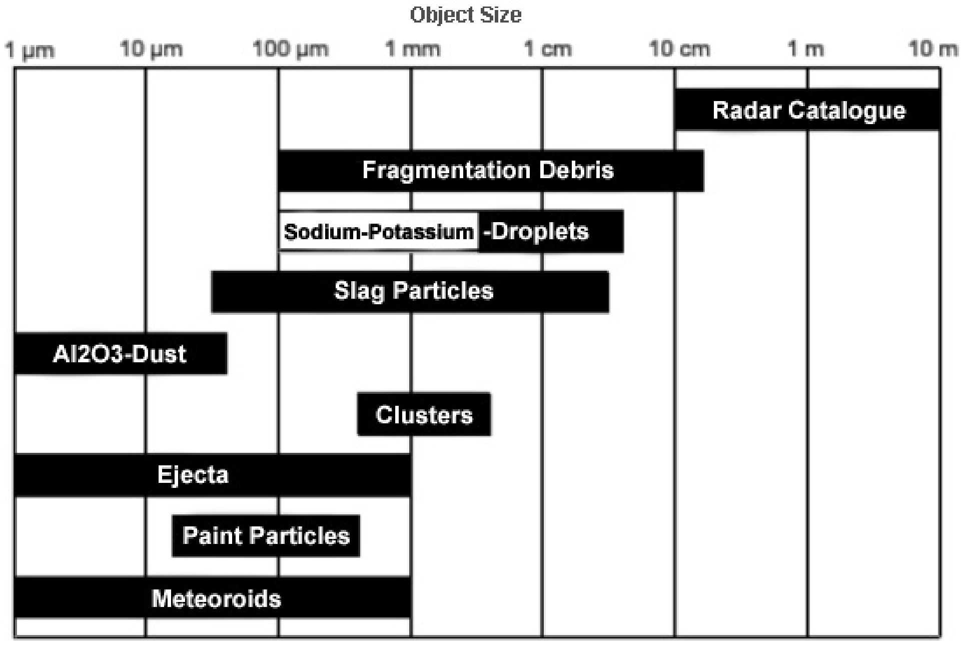


Figure 2.5: Different sources of orbital debris versus their typical size region, along with the size regime of cataloged objects. (Vorsmann, 2008)

altitude, which is a popular region for satellites. Since there is almost no aerodynamic drag at this height, the debris will stay in orbit for a very long time.

Also the United States of America (USA) destroyed a satellite with a missile in February 2008, but the resulting debris re-entered the atmosphere quite fast, since the event took place at around 250 km altitude, see Subsection 2.1.4 on orbital decay.

2.1.3 Distribution

In Figures 2.6 - 2.8 the spatial density distribution of orbital debris with respect to its altitude and source is depicted. The graphs are generated by ESA's MASTER-2001 (Meteoroid and Space Debris Terrestrial Environment Reference) software, which simulates the orbital debris environment and is updated and corrected by making use of verification data provided by tracking and statistical measurement systems and methods, see Section 2.2.

It can be seen that the majority of the large (>10 cm) objects is situated in the LEO (especially around 800 km) and GEO regions, where the main sources are larger fragments, and mission and launch objects.

This is also true for medium sized (>10 cm) objects, where the smaller fragments and SRM slag dominate. However, the distribution here is more spread out since smaller objects generally have less stable orbits and frag-

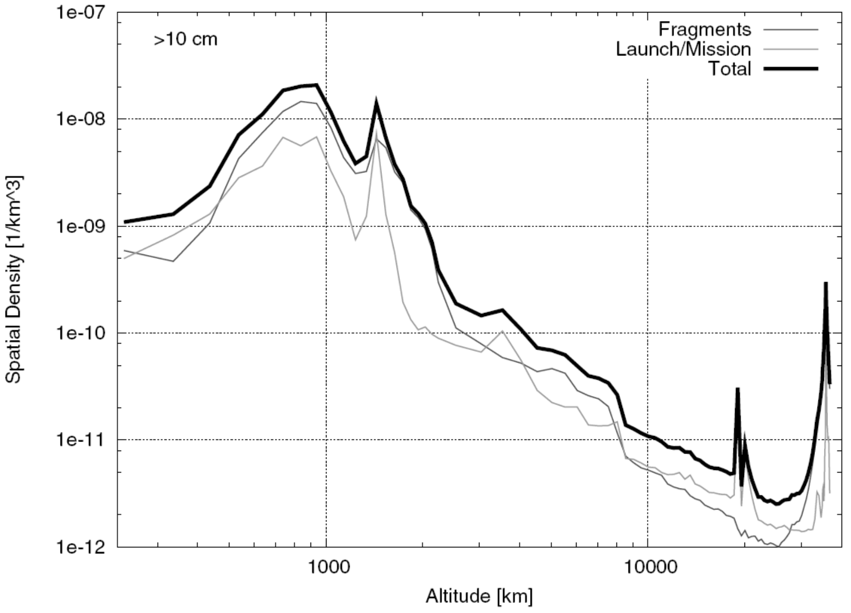


Figure 2.6: The spatial density distribution of orbital debris with a diameter larger than 10 cm with respect to its altitude and source. (Klinkrad, 2006)

ments created by explosions and collisions add significant radial velocities to the particles as well changing the original orbital altitude.

For very small (>1 μm) objects the distribution is completely spread out due to the fact that the main contributor is SRM dust, which receive such large ΔV during GTO because they are very light and therefore their orbits have very high eccentricities.

In Figure 2.9 the orbital debris and meteoroid cross-sectional flux versus size can be seen, which is cumulative with decreasing size. Use has been made of data at various LEO altitudes and inclinations near 28.5 deg using many measurement techniques, which will be explained in Section 2.2. Here it can be seen that micro-meteoroids dominate the smaller object regimes while man-made orbital debris dominates the larger objects.

2.1.4 Orbital Decay and Re-Entry

Orbital debris, including (decommissioned) satellites, rocket stages and fragments thereof, continuously decay into the Earth's atmosphere. The orbits of orbital debris decay until they reach denser parts of the atmosphere, which causes more drag, and this decays the orbit even faster.

Due to the high velocity involved, typically 7.8 km/s at the start of the decay process at an altitude of about 120 km, the amount of friction keeps

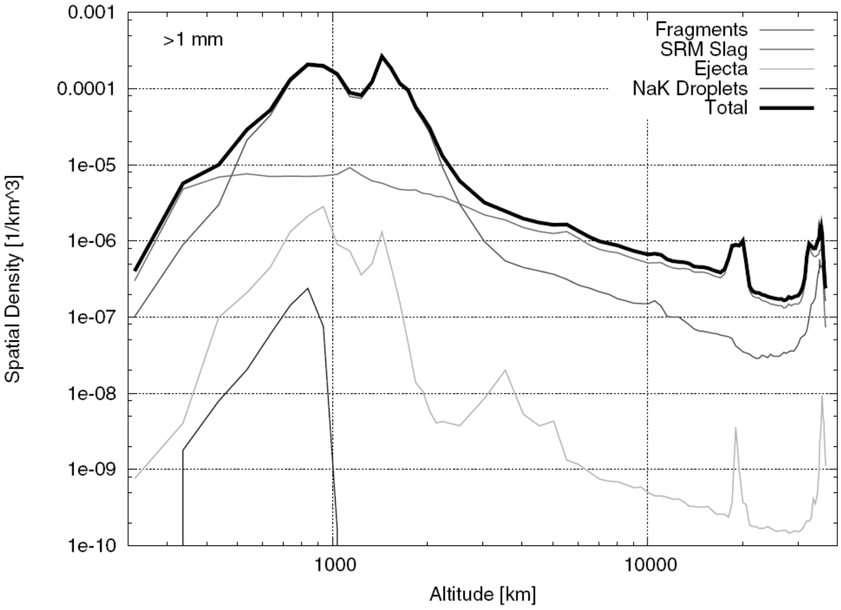


Figure 2.7: The spatial density distribution of orbital debris with a diameter larger than 1 mm with respect to its altitude and source. (Klinkrad, 2006)

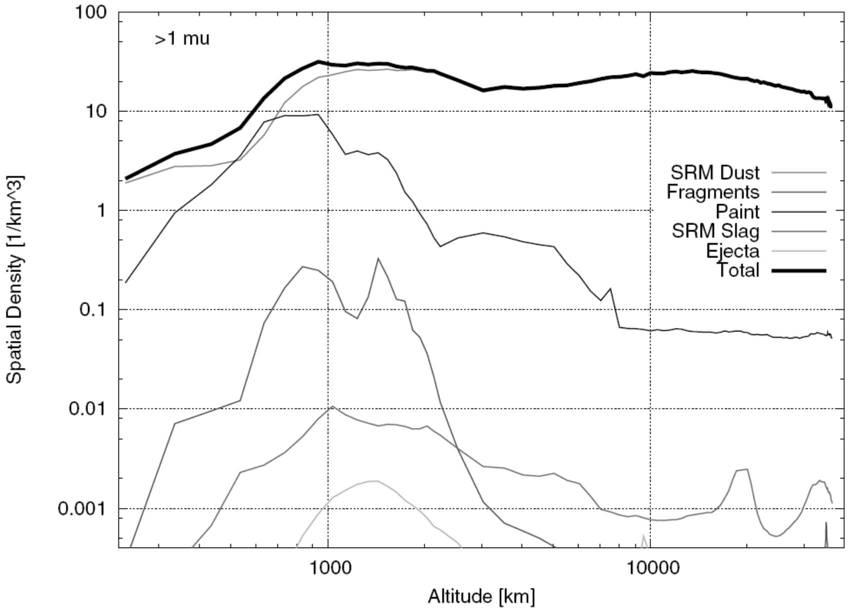


Figure 2.8: The spatial density distribution of orbital debris with a diameter larger than 1 μm with respect to its altitude and source. (Klinkrad, 2006)

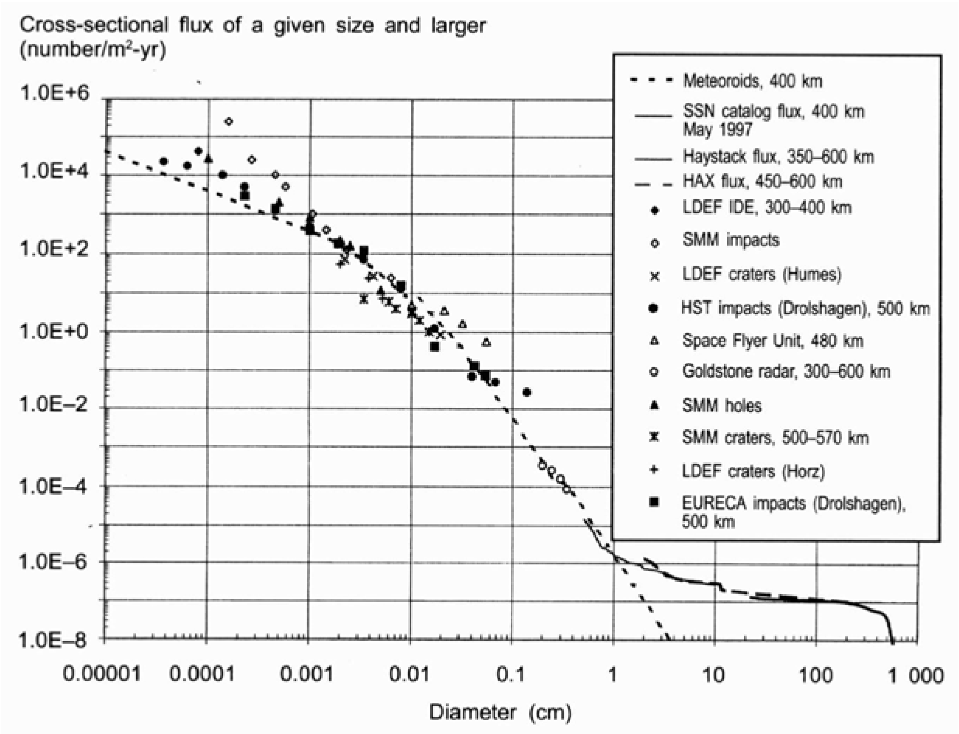


Figure 2.9: Cumulative cross-sectional flux versus decreasing sizes of meteoroids and orbital debris. (Klinkrad, 2006)

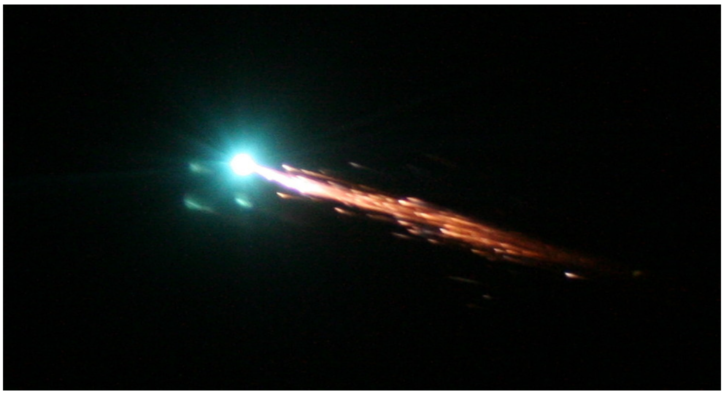


Figure 2.10: The Automated Transfer Vehicle (ATV) breaking and burning up in the Earth's atmosphere. (Klinkrad, 2009)

building up because the density of the atmosphere increases with decreasing height. The debris loses energy and reaches lower altitudes while speeding up, and is heated up due to the friction with the atmosphere until it breaks up and, if the object is not too dense, completely burns up.

However, massive and compact objects like space stations, large satellites and transport vehicles, or objects that contain stainless steel or titanium which have high melting temperatures, might not completely burn up and impact the surface of the Earth. Of space stations like Salyut, Skylab and Mir with masses of 40 up to 120 tonnes which have re-entered in the past, 20-40% can survive and reach the ground.

Due to the risks involved, satellites that might not completely burn up can be steered to re-enter above unhabited areas. However, most orbital debris can not be controlled. Large uncontrollable objects that might not completely burn up can be tracked, see Section 2.2, so that it is accurately known where surviving fragments might impact the Earth's surface, and the risk can be calculated.

Re-entry simulations can be done beforehand to see whether fragments will survive and where they will re-enter. Some surviving fragments are light enough to reach the ground at such low velocity that they are not dangerous.

In Figure 2.10 the Automated Transfer Vehicle (ATV) breaking and burning up in the Earth's atmosphere can be seen. In Figure 2.11 the remainders of a titanium motor casing of a Delta 2 rocket weighing 70 kg is depicted, which crashed into Saudi Arabia, while its tanks dropped down in Texas, U.S. The latter might be from a first stage though, which has never really gone into orbit.

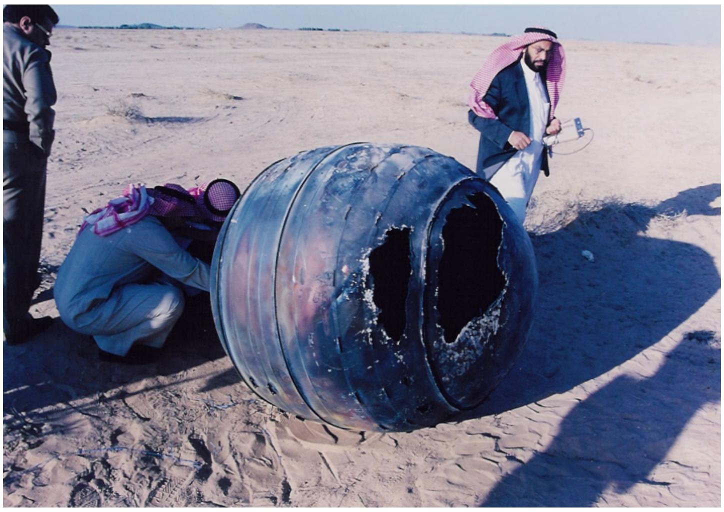


Figure 2.11: The remains of a decayed titanium motor casing of a Delta 2 rocket which landed in Saudi Arabia. (Stansbery, 2009)

For orbits with a low altitude or low perigee (high eccentricity) this decay process can happen within a few years, but for high enough (near) circular orbits decay could take hundreds of years. This process can be sped up by giving a satellite a so-called End-Of-Life (EOL) boost, see Section 2.3, to decay directly or within some years. For the latter option the position of decay can not be controlled accurately, however.

2.1.5 Negative Consequences

Orbital debris and meteoroids have some negative consequences for operating spacecraft, and large re-entering debris could pose a threat to life and property on Earth. The chance that a hazard would occur due to re-entering debris is very small however.

Debris and meteoroids smaller than 1 cm pose a serious threat to critical hardware of functioning spacecraft though, if no shielding would be applied, see Subsection 2.3.1. Small orbital debris and meteoroids have such high abundances that they impact spacecraft regularly, see Subsection 2.2.3. Particles with sizes in the order of μm merely degrade the surface of a spacecraft, chipping away very small pieces like a sandblasting process. Particles with sizes in the order of mm could disable a subsystem or do some local sensor damage already. Objects with sizes in the order of cm could disable an entire spacecraft and even create an explosion. If a collision were to occur with debris larger than 10 cm, the entire spacecraft would disintegrate, creating a large amount of newly formed small and larger debris, like with the Iridium/Cosmos collision in February 2009.

These consequences are so severe due to the relative velocities involved. Objects in LEO move at speeds of about 8 km/s with respect to the Earth. Satellites and debris can reach relative velocities up to 15 km/s and for meteoroids this could even be more than 72 km/s! With such high relative velocities the kinetic energy released on impact is also very high, even for very small masses (Wertz, 2001).

However, spacecraft can be protected with bulletproof-vest-like layers against objects up to 1 mm, and high-valued missions can be shielded against objects up to 1 cm. Furthermore, objects larger than 5-10 cm are tracked by the SSN and impact avoidance can be used in these cases.

This still leaves a gap though, for debris with sizes which could be catastrophic to a mission but which can not be tracked or protected against.

2.2 Observing, Measuring and Tracking

Tracking and observing orbital debris can be done by ground-based optical telescopes and radars, space-based telescopes, and analysis of impacted spacecraft surfaces returned from space. Like said before, more than 14000 orbital debris objects larger than 5-10 cm in LEO and 30-100 cm in GEO are currently regularly tracked and cataloged by SSN, and can be traced back to their original launch event and thus owner.

Debris down to 30 cm can be observed in GEO by optical telescopes. Objects as small as 2 mm can be observed in LEO using radar.

However, smaller objects can not be tracked because they cannot be measured with large enough accuracy and their orbits are too unstable. For debris smaller than 1 cm down to a few μm one can analyse measurements obtained by in-situ impact detectors, or inspect surfaces of spacecraft that are returned to Earth and have been impacted by such small debris.

Also, hypervelocity impact studies are being performed on Earth to simulate what will happen if debris hits a spacecraft surface.

Dozens of optical telescopes and radars positioned at different longitudes and latitudes around the world are used to observe orbital debris in different parts of the celestial sky, and are mostly operated by organisations in the USA, Russia, Japan and Europe.

Next to the SSN, also the Russian Space Surveillance System and the (experimental) French GRAVES surveillance system keep an orbital debris catalog, with a number of respectively 6000 and 2500 objects versus the then observed 9000 objects of the SSN (Klinkrad, 2006).

However, not all telescopes and radar that scan for orbital debris are connected to these catalogs. Many of them scan known objects with better accuracy, smaller objects that are not cataloged, or provide statistical data of small-sized objects.

2.2.1 Optical Telescopes

The telescopes that observe orbital debris make use of one of the two large wavelength regions of the electromagnetic spectrum in which the Earth's atmosphere is transparent, namely the optical window with wavelengths from 0.3 μm to 2 μm. Even within this optical window the observed light coming from the Sun shows some secondary absorption bands after passing through the atmosphere, so telescopes are mostly located at high altitudes where the air density has dropped significantly.

The telescopes operate at 'astronomical night' conditions, such that there is the least amount of light pollution while the observed object is still illuminated by the Sun. The reflected sunlight on the spacecraft is then collected by the telescope aperture, after which it is passed on to a Charge-Coupled Device (CCD) through some lenses and/or mirrors.

Angular information of the object can be obtained with reference to the stars observed or through the telescope's mechanical axes whose positions are calibrated with the Earth. By using multiple CCD read-outs of the same object at different epochs, the orbit can be determined. Also the size of the object can be obtained, as well as its light curve (light intensity versus time) and spectrum.

These observations can help in identifying the object for a catalog.

Optical telescopes are mainly used for observations of orbital debris in MEO and GEO, since they perform better at higher altitude than radar based observations. This is due to the fact that the sensitivity of optical telescopes is proportional with $1/r^2$, with r being the distance to the target, while for radar this factor is $1/r^4$ due to active illumination, see Subsection 2.2.2. Objects down to 30 cm in size can be observed in GEO.

The U.S. National Aeronautics and Space Administration (NASA) Orbital Debris Observatory (NODO, Figure 2.12), located at Cloudcroft in the USA, uses a Liquid Mirror Telescope (LMT, Figure 2.13) to generate the largest optical debris dataset to date.

An example of two observed orbital debris objects in GEO (marked in yellow and red) can be seen in Figure 2.14. The objects observed are dots while the background stars appear as stripes, this is because the moving objects are followed by the telescope while the stars are relatively standing still.

Furthermore, there have been some space-based telescopes that observed

Figure 2.12: The NASA Orbital Debris Observatory (NODO), located at Cloudcroft NM, USA. (Stansbery, 2009)

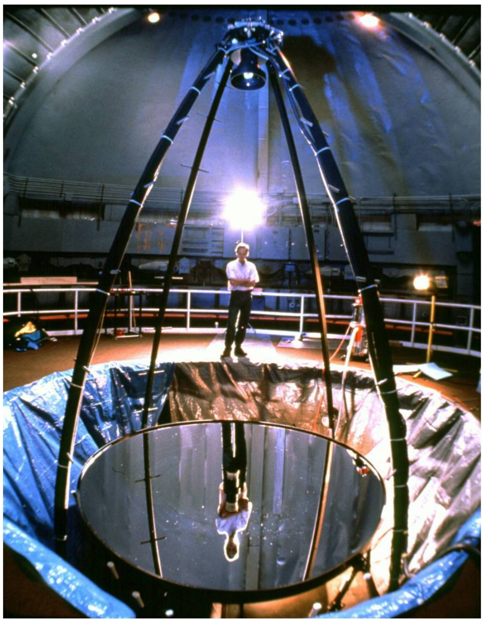


Figure 2.13: The Liquid Mirror Telescope (LMT) of NODO. (Stansbery, 2009)

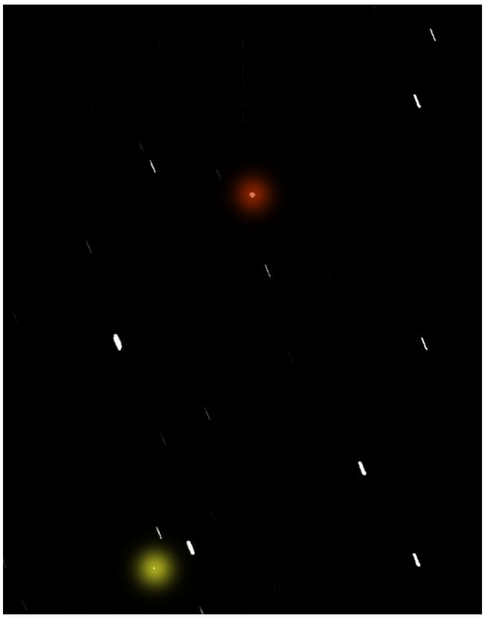


Figure 2.14: Two orbital debris objects in GEO (marked in yellow and red) observed by an optical telescope. (Klinkrad, 2009)

orbital debris: for the Infra-Red Astronomical Satellite (IRAS) as a by-product of its primary objective to scan the celestial sphere, and for the Mid-Course Space Experiment (MSX) satellite as its second primary mission, using the Space-Based Visible (SBV) instrument (Leloux, 2010b). The latter could observe objects in LEO and GEO with a significantly smaller size than is possible for Earth-based telescopes.

Very recently, on the 25th of September 2010, the first 'Pathfinder' satellite of the SBSS was launched, with its sole purpose of identifying, observing and tracking orbital debris. It will increase the capability of the SSN, in terms of the amount of orbital debris observed, by a factor two.

2.2.2 Radar

Radars that observe orbital debris make use of the radio window of the Earth's atmosphere, with wavelengths from 1 cm to 100 cm. They can detect objects of cm size in LEO and down to 2 cm at an altitude of 1000 km.

An object can be tracked by first transmitting a radar pulse which is then reflected by the object and received back at the radar antenna. By measuring the travel time of the signal and its Doppler shift, the distance to the target as well as its time derivative can be determined.

Furthermore, by measuring the direction of the highest gain of the received signal, the direction (azimuth and elevation) of the target can be obtained. Also, the radar echo (history) and its polarization can be used to deduce information on the target's cross-section, attitude and shape (Klinkrad, 2006).

Due to the fact that a radar first has to illuminate an object and afterwards receive this signal, its performance drops with the previously mentioned factor of $1/r^4$. Consequently, they can only be used for observations in LEO, with current technology standards.

However, performance can be boosted by using a second receiver antenna at another location: the so-called bi-static mode. In this way the receiving antenna aperture is effectively increased leading to a larger receiver gain. Combined with the fact that the data can now be correlated and the average of two measurements can be taken, this leads to more accurate results.

An example of such a bi-static radar system is the FGAN (ForschungsGesellschaft für Angewandte Naturwissenschaften, in English: Research Establishment for Applied Sciences) Tracking and Imaging Radar (TIRA) located at Wachtberg in Germany, which can be seen in Figure 2.15. During an experiment this system even led to detection statistics of debris as small as 0.9 mm.

In Figure 2.16 an artist's impression of a cutout of the main transmitting and receiving antenna of the TIRA system can be seen.

The source with the largest abundance of data for orbital debris in the size range of 1-30 cm is the Massachusetts Institute of Technology (MIT) Haystack Observatory which is located near Boston in the USA, of which the Haystack Long-Range Imaging Radar (LRIR) and Haystack Auxiliary Radar (HAX) can be seen in Figure 2.17.

2.2.3 Surface Examinations and In-Situ Sensors

By examining impact craters on surfaces of spacecraft hardware that have been in orbit and are retrieved to Earth, orbital debris and meteoroids smaller than 1 mm can be analysed as well. The size, velocity and direction of the debris and meteoroids can be deduced by measuring the impact hole size and shape.

The distribution can be obtained statistically from multiple surface examinations in different orbits. The origin of the impactor can be determined through analysis of the chemical composition of the residues inside the impact craters.

Due to the fact that this hardware has to be re-entered into the atmosphere, it is very costly. So far this was only possible with the Space Shuttle, and thus the retrieved hardware can not exceed orbits higher than 600 km or inclinations outside its reach.

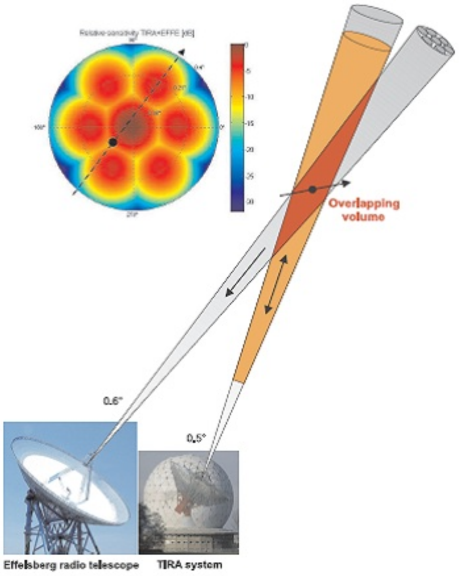


Figure 2.15: The bi-static FGAN Tracking and Imaging Radar (TIRA) located in Wachtberg, Germany. (Klinkrad, 2006)

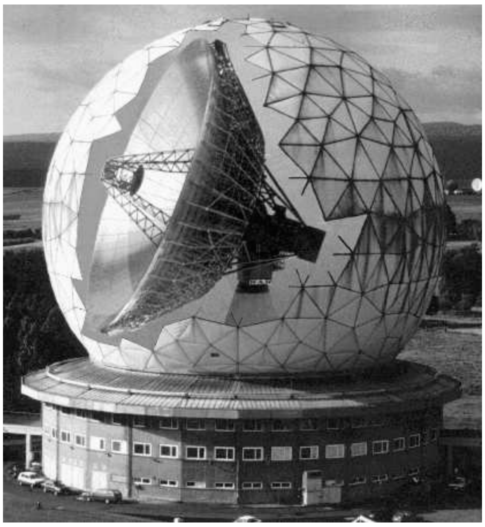


Figure 2.16: Artist's impression cutout of the main transmitting and receiving antenna of the TIRA system. (Klinkrad, 2006)

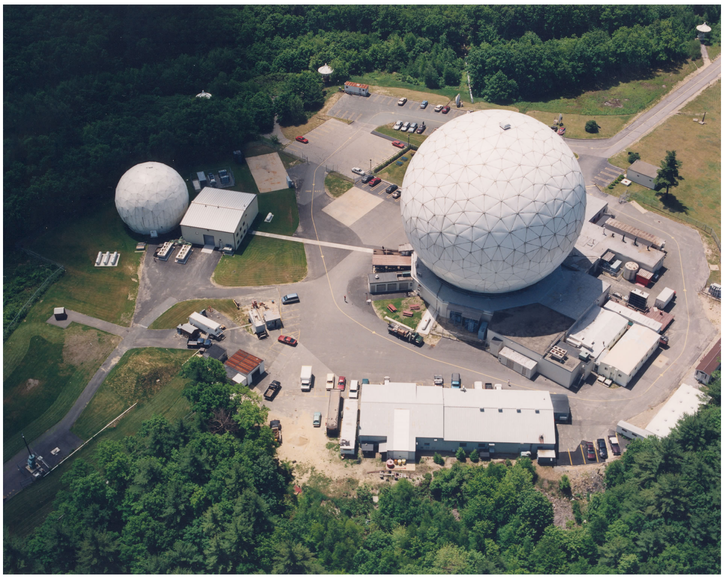


Figure 2.17: The Haystack Long-Range Imaging Radar (LRIR) and Haystack Auxiliary Radar (HAX) located near Boston, USA. (Stansbery, 2009)

A good example of examined surfaces is the retrieved Long Duration Exposure Facility (LDEF) in 1990, which had a total exposed surface of 130 m^2 and was in orbit for 5.7 years. On the LDEF 30,000 impacts could be seen with the naked eye of which 20,000 were documented, 5000 were larger than 0.5 mm and 1000 were chemically analysed. The LDEF can be seen in orbit in Figure 2.18 and a retrieved panel thereof in Figure 2.19.

Other examples include windows and other surfaces of the Space Shuttles, see Figure 2.21, retrieved satellites and parts thereof, see Figure 2.20, solar panels of the International Space Station (ISS), Mir and Hubble Space Telescope (HST), see Figure 2.22 and surface experiments on Mir. The chemical analysis of impact residues of the solar panels of the Hubble Space Telescope resulted in 69% meteoroids, 10% man-made debris, while 21% could not be analysed. Of this 10% man-made debris, 98% was due to solid rocket motor firing particles and 2% due to paint flakes.

For regions out of the space shuttle's reach, active in-situ impact sensor surfaces can measure the amount, velocity and mass of impactors directly and send their data back to Earth.

Examples of such sensors are ESA's Geostationary Orbit Impact Detector (GORID), see Figure 2.23, which was attached to a Russian satellite launched into GEO in 1996, which detected 2.4 impacts per day in a duration of 5 years, while in 2001 the Debris in-Orbit Evaluator (DEBIE) was

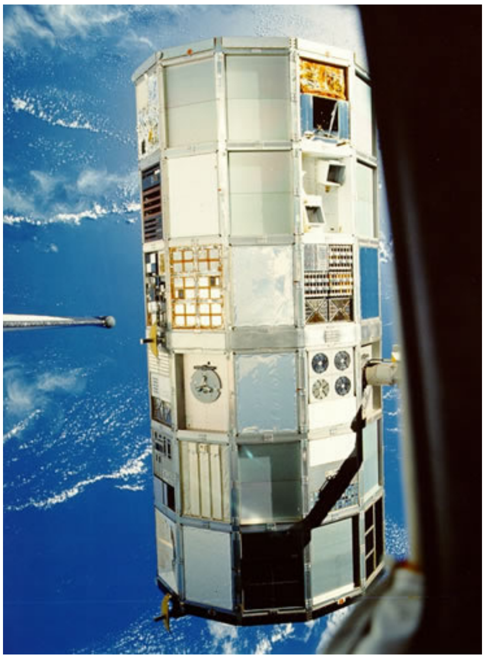


Figure 2.18: The Long Duration Exposure Facility (LDEF) in orbit after deployment from a Space Shuttle, clearly showing its surface experiments. (Stansbery, 2009)

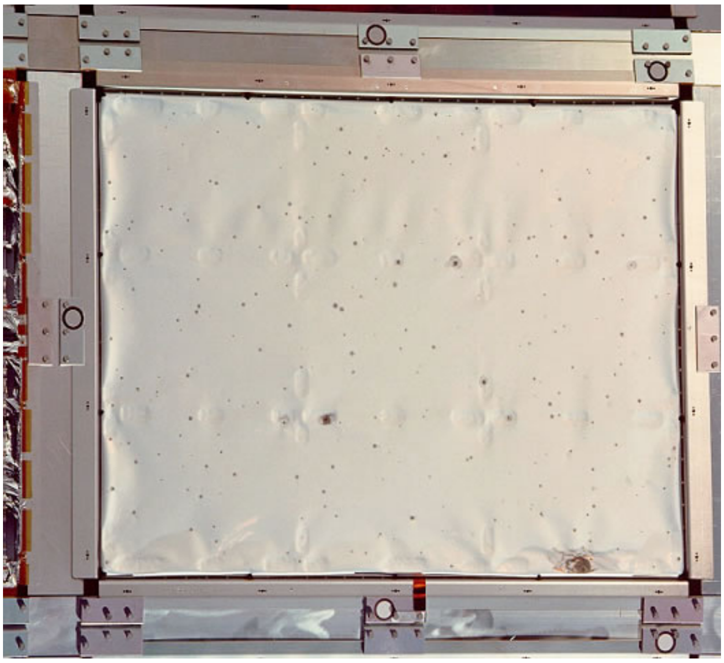


Figure 2.19: A retrieved surface panel of the LDEF that has been impacted by small debris and meteoroids. (Stansbery, 2009)

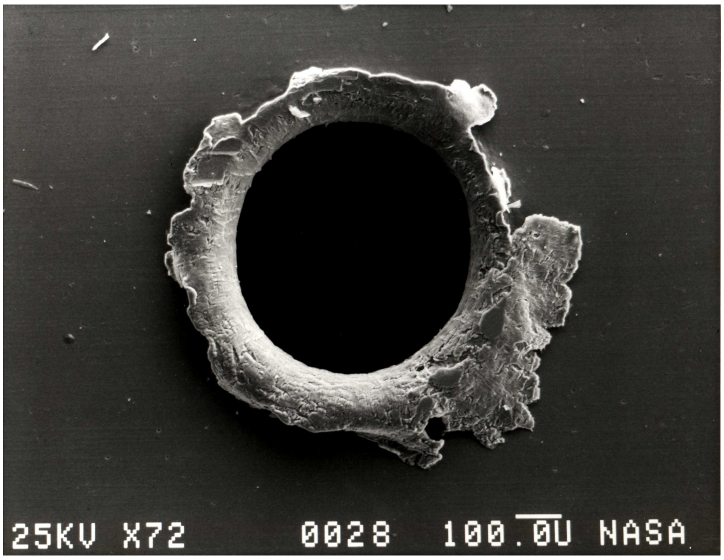


Figure 2.20: Impact crater on the Solar Maximum Mission (SMM) satellite. (Stansbery, 2009)

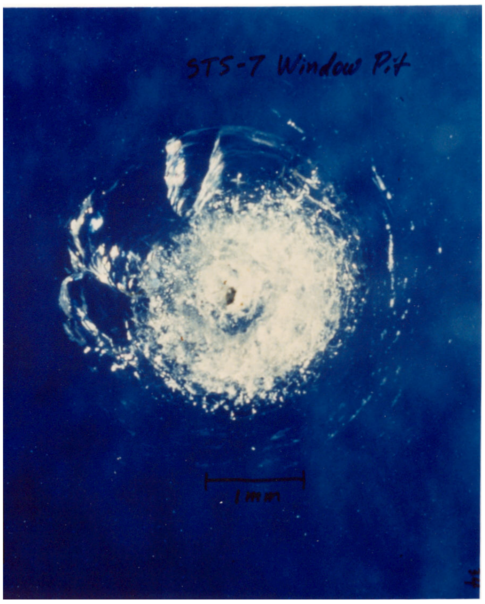


Figure 2.21: Impact crater on a Space Shuttle window. (Stansbery, 2009)

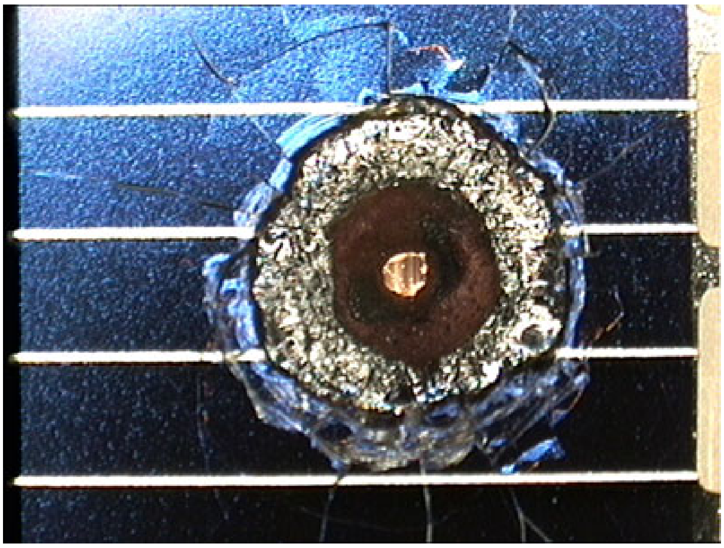


Figure 2.22: Micrometeoroid impact crater on a solar array of the Hubble Space Telescope with a diameter of approximately 4 mm caused by a particle of about 0.5 mm. (Klinkrad, 2009)

attached to ESA's PROBA-1 (Project for On-Board Autonomy) satellite. Currently DEBIE II is still measuring impacts on the Columbus science module docked to the ISS, see Figures 2.24 and 2.25.

2.2.4 Hypervelocity Impact Studies

Small-sized orbital debris and meteoroid impacts into satellites or their shields (see Section 2.3) can be simulated on Earth by so-called hypervelocity impacts.

Hypervelocity is defined as the speed of sound within the solid material which is impacted (Klinkrad, 2009). So when a projectile impacts a certain material with a relative speed higher than this speed of sound, it is called a hypervelocity impact, which is reached at about 4-5 km/s for metal and 1-2 km/s for glass.

Orbital debris mostly reaches relative speeds of 0-15 km/s and meteoroids 5-30 km/s (Klinkrad, 2006), although meteoroids could even go up to 72 km/sec.

During hypervelocity impacts, the strength of the material is very low compared to the inertial stresses. As a consequence, the material under hypervelocity impact behaves like a fluid instead of a solid. A shock-wave is created that travels through the material and is reflected by the surfaces. The waves created can interfere and strengthen each other until the deformation leads to higher local stress levels than the material can withstand.

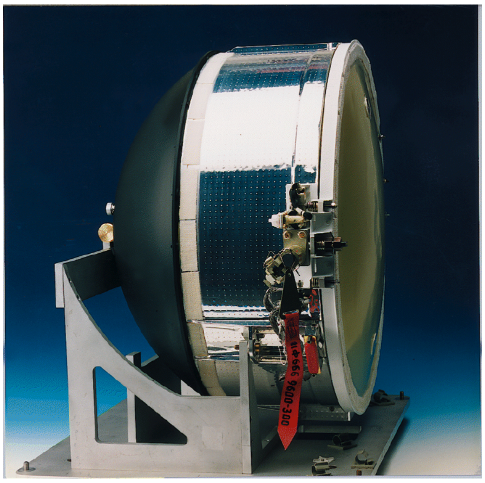


Figure 2.23: Geostationary Orbit Impact Detector (GORID) which was attached to a Russian GEO satellite. (Klinkrad, 2009)

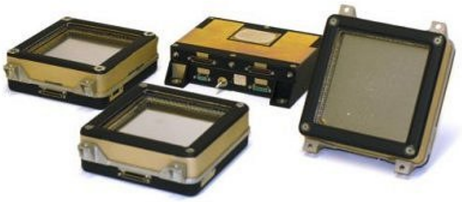


Figure 2.24: Debie II impact panels. (Stansbery, 2009)

The effect of an impact depends on the material of the impactor and target, the velocity and incident angle of the impactor, and the mass and shape of the impactor. With increasing speeds, the impact results in elastic to plastic deformation, craters, projectile break-up and melting, and even mass ejections resulting in craters 2-5 times the diameter of the projectile. If for instance the thickness of the target is decreased, the effects change from craters, via cracks, to spall detachment, and finally to clear hole penetration, see Figure 2.26.

Two-stage light-gas guns are typically used to accelerate projectiles to such high speeds, because they can handle different shapes of projectiles and the test conditions can be controlled accurately. They work by igniting gun powder in a tube to force a piston to compress a light gas like helium or hydrogen through the converging end of the tube, where the pressure builds up until a calibrated barrel is teared open and the projectile is accelerated by the increasing speed of the gas into a vacuum chamber (Christiansen, 2006).

A schematic of such a gun can be seen in Figure 2.27, and one which is used by NASA's hypervelocity impact technology facility for impact studies can be seen in Figure 2.28.

Using these or similar types of guns, ranging from powder guns to electrostatic guns, projectiles with masses of 1200 gram down to 0.05 gram can be accelerated from 1 km/s to 18 km/s respectively (Klinkrad, 2006).

The impact into a certain material sheet is then captured with high-speed cameras. The velocity can also be measured by lasers or light detectors, and the integrity of the projectile is checked using flash x-rays. High-speed camera pictures of such an impact event can be seen in Figure 2.29.

Next to real testing, also numerical simulation of hypervelocity impacts can be done to study the effects. These simulations can be compared and calibrated with real test data, which increases their performance in terms of for instance accuracy and reliability.

2.3 Solutions to the Problems of Orbital Debris

A number of solutions to the threat of orbital debris and meteoroid impacts and collisions are already in use, while others are under investigation.

This section will first deal with the protection of satellites from orbital debris through shielding and collision avoidance. Then it will continue with (growth) mitigation solutions and strategies including self- and external removal. Finally it will conclude with a section on international cooperation through guidelines proposed by space agencies and other institutes.

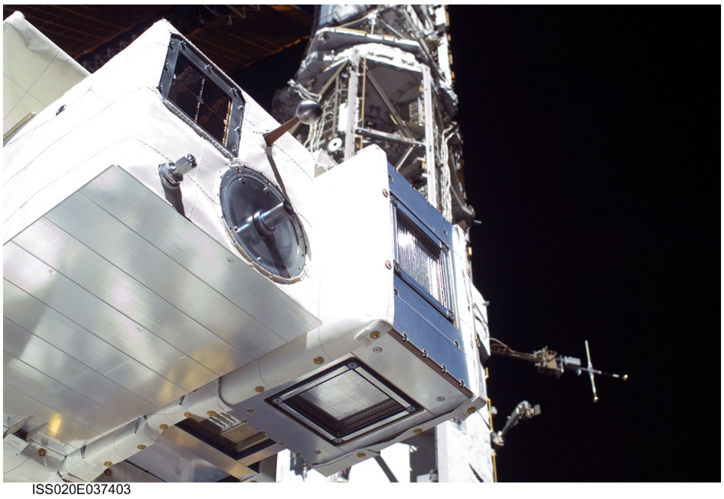


Figure 2.25: Debie II impact panel on the EUTeF on the Columbus module of the ISS. (Klinkrad, 2009)

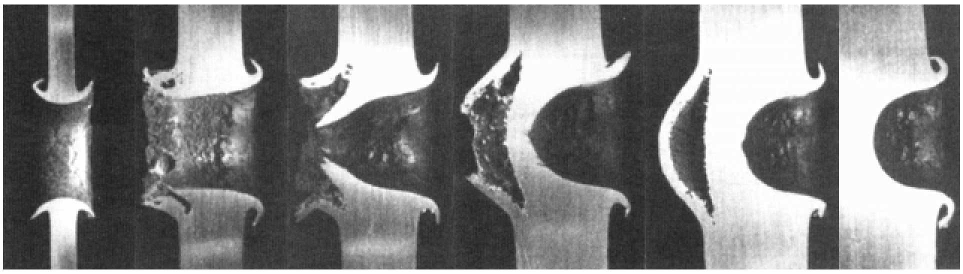


Figure 2.26: Pictures of varying results after an impact takes place on sheets with different thicknesses. (Klinkrad, 2006)

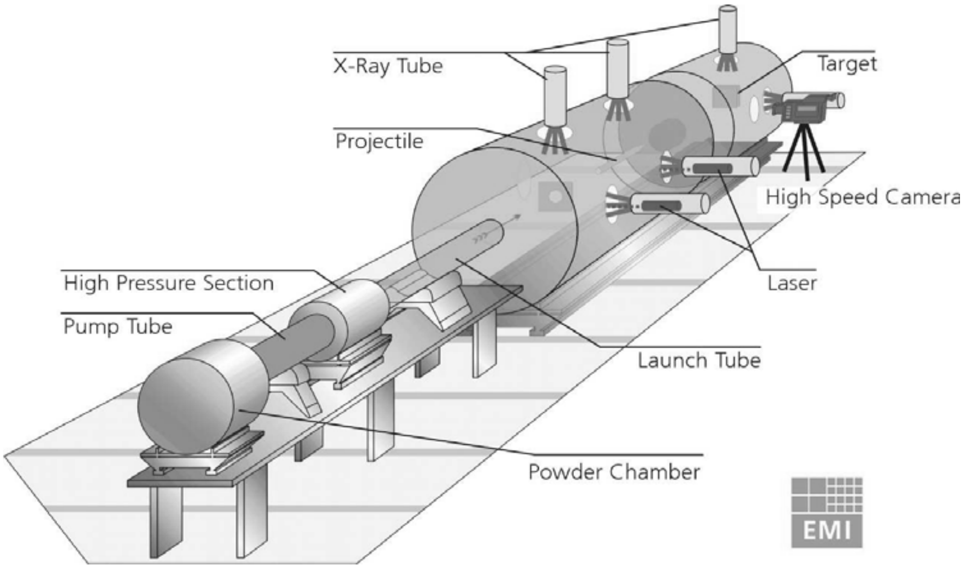


Figure 2.27: Schematic of a two-stage light gas gun used for hypervelocity impact analysis. (Klinkrad, 2006)

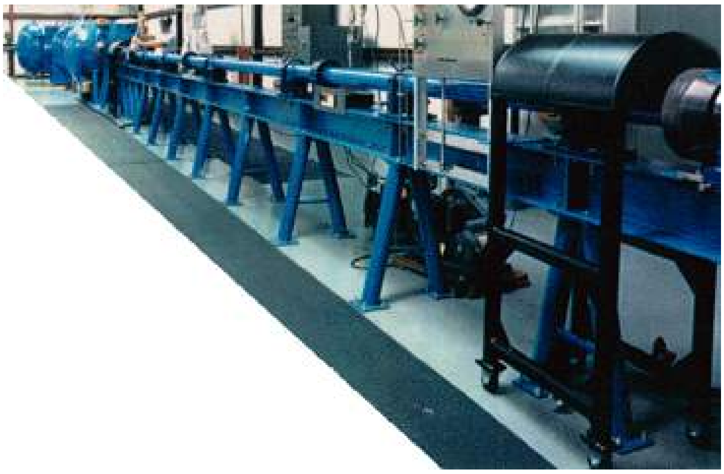


Figure 2.28: A two-stage light gas gun in use by the NASA hypervelocity impact technology facility. (Christiansen, 2006)

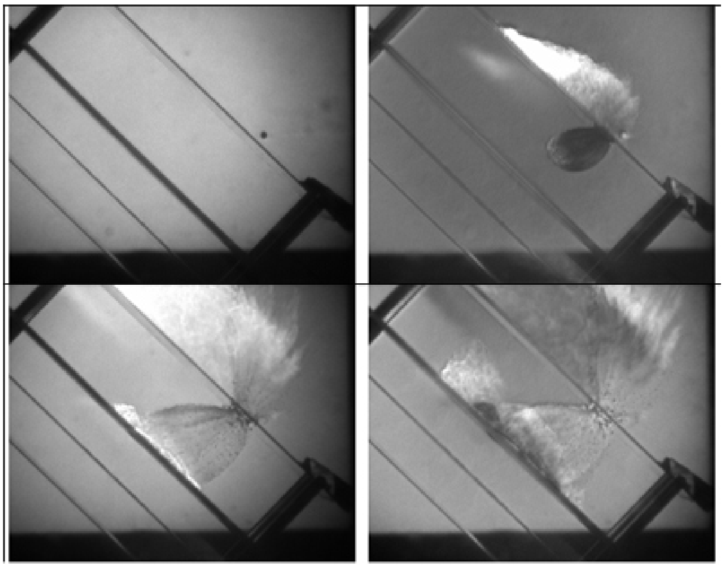


Figure 2.29: High-speed camera pictures taken of a hypervelocity impact on a Whipple shield. (Klinkrad, 2009)

2.3.1 Protection

To protect spacecraft from impacts of and collisions with orbital debris and meteoroids, use can be made of passive protection techniques such as shielding and active protection solutions like collision avoidance, which are both elaborated upon further below.

Shielding

Passive protection of spacecraft from impacts is usually done by shielding. Critical hardware can either be shielded by less critical structures or by making use of the so-called Whipple shield.

When a spacecraft is shielded by just increasing the thickness of the outer wall, the structural mass will increase too much compared to the total satellite because a single wall will have to be relatively thick to protect the spacecraft from critical impacts. This is why Fred Whipple in the 1940's invented a shield consisting of an outer thin bumper wall which disintegrates the impactor into (liquid) fragments after which an inner wall with a certain spacing from this bumper withstands the created cloud of debris and plasma containing both bumper and impactor material.

In this way, the impactors' momentum is distributed over a large area of the second wall, which can now be made substantially thinner compared to the case with just one wall. This leads to significant mass reductions.

An example of a hypervelocity impact on such a Whipple shield captured by a high-speed camera can be seen in Figure 2.29.

Nowadays stuffed Whipple shields with aluminum, Nextal and Kevlar bumper layers are being used. Multiple layers in-between slow down the debris cloud even more, leading to a higher levels of protection with less mass.

Other passive solutions include adding redundancy to a spacecraft design, by integrating multiple copies of critical or less reliable components.

Collision Avoidance

Collision with debris might be avoided by passive methods such as creating a spacecraft with a small cross-sectional area, or by changing its attitude in such a way that it has a smaller cross-section with respect to the expected debris impact direction. Since the average relative impact velocity is significantly higher at the front side of the spacecraft in along-track direction, this leads to smaller average relative impact velocities.

Another solution could be to design the orbit of the satellite in such a way that it does not cross highly populated orbital debris regions, but this is mostly not possible.

By using orbital data of debris cataloged by the SSN, conjunction analysis can be performed. Since both the orbits of the satellite and the orbital debris are known with a certain accuracy, orbit propagation of these orbits can lead to collision probabilities.

Other variables that are used in this analysis are the object size, orbit geometry and the smallest predicted distance between the two objects. This conjunction analysis is further investigated in the remainder of this thesis.

When the collision probability is higher than a certain threshold, the orbits can be measured by radars with high accuracy leading to better collision probability results. If the threat still is too large, active collision avoidance can be performed by thrusting the spacecraft into a slightly different orbit with less probability of collision.

Since this requires extra propellant, a large amount of coordination with other institutions, and also has an impact on the mission, it is very costly. Therefore it has only been used for high-valued (manned) spacecraft such as the ISS, Space Shuttle, large LEO constellations, and expensive scientific Earth observation satellites.

Active collision avoidance is however limited, due to the fact that only 5% of the tracked catalog objects is maneuverable and a very large amount of catastrophic debris larger than 1 cm is not routinely tracked.

2.3.2 Growth Mitigation

To minimize the growth of the orbital debris population, a number of measures have been proposed and implemented. Orbital debris mitigation measures include the passivation of spent upper stages and satellites and EOL disposal, which are explained below. Debris released during normal operations can be minimized, and intentional destruction of spacecraft could be forbidden. Also some external removal solutions have been proposed.

Other measures discussed in the previous subsection on protection, like collision avoidance and shielding, serve in principle to protect the spacecraft, but they also help in mitigating the growth of orbital debris.

Due to the fact that explosions are the source of a large part of the orbital debris population, measures can be taken to minimize these events. Causes of these explosions like residual fuel in spacecraft and upper stages, can be burned up or released into space. Pressurized vessels can be vented and batteries can be discharged.

EOL disposal can be applied to both LEO and GEO missions.

LEO satellites can be propelled into a decaying orbit which will make them burn up in the atmosphere considerably faster than if nothing would be done. However, this does require extra propellant, and thus extra mass, increasing the costs. Orbits with higher altitudes require larger amounts of propellant for the same effect and thus this option is only practical for low enough orbits.

A disadvantage here is that the spacecraft is then still able to perform some of its primary function, since it must still be controllable to be able to de-orbit, and thus some hardware and thrusting capabilities are still functioning, which are now wasted.

GEO satellites are nowadays required to be kicked into a so-called graveyard orbit, some hundreds of km above the geo-stationary ring, so that they do not interfere with new GEO missions. This can also be applied to high LEO satellites, moving them away from the populated LEO region.

However, the GEO graveyard orbit still poses some risk to satellites in the geostationary ring, especially in the long run, when the objects in the graveyard orbits may orbit back into the geo-stationary ring again.

Newly proposed solutions include rolling out an electrodynamic tether on a cable at EOL of LEO missions, or attaching a sail or balloon, which can reduce the energy of the spacecraft and make it decay into the atmosphere faster.

Some solutions to external removal of orbital debris have also been studied. De-orbiting orbital debris with ground-based lasers (Kelso, 2010b), for instance, but this is not yet tested due to international agreements which forbid the use of high-powered lasers in space. This solution might also not be efficient enough.

Another solution might be to actively capture orbital debris using spacecraft, but this is regarded too costly.

2.3.3 International Cooperation

In order to coordinate all activities related to the issues of orbital debris between space-fairing nations, the Inter-Agency Space Debris Coordination Committee (IADC) has been set up, in which 11 of the largest space agencies in the world participate. The purposes of this committee are to exchange research results on orbital debris, review current research, stimulate future research, and to determine debris mitigation solutions. They contain a steering group and working groups on measurements, modelling, protection and mitigation, and organize annual meeting to discuss their findings (IADC, 2010).

NASA was the first space agency that set up orbital debris mitigation guidelines in 1995 which transformed into standard practices two years later. Later other nations and agencies around the world designed their own rules, and in 2002 the IADC had completed an agreement with all of its members on guidelines to mitigate the growth of orbital debris.

Afterwards, in 2007 the Scientific and Technical Subcommittee (STSC) of the United Nations (UN) Committee on the Peaceful Uses of Outer Space (COPUOS) also finished similar mitigation guidelines which were approved by the UN in 2008. These guidelines generally include all mitigation solutions mentioned previously (UNOOSA, 2010).

Today, debris mitigation standards are being developed by the International Organization for Standardization (ISO). These standards focus more on how the guidelines can be implemented by other entities. These ISO standards are not mandatory, but space agencies and nations have also implemented their own set of standards for orbital debris mitigation, which are binding in their own nations (Klinkrad, 2009).

Furthermore, organisations that launch objects into space can be held legally responsible by the United Nations Outer Space Treaty for damages caused by these objects, either on ground or in orbit.

However, this treaty was only signed by 27 nations (excluding most of the space-fairing nations) and ratified by 98 (UNOOSA, 2010). Ratified means that the country is legally bound by the treaty under international law, signing means the treaties principles are supported and the country is in the process of ratifying it in full effect.

Chapter 3

Astrodynamics

Astrodynamics is an application of the classical theories of celestial mechanics, which in turn may be considered part of the broad field of astronomy. Celestial mechanics, and thus astrodynamics, is based upon four laws: Newton's three *laws of motions* and Newton's *law of gravitation.*

The fundamental astrodynamical theories that were used in the development of the SGP4 and SDP4 TLE propagators, the conjunction analysis methods and the collision probability methods, are described in this chapter. The theory described here is largely modified and summarised from K.F. Wakker (2007) and Wakker (2007), unless indicated otherwise.

Included is the general theory of all perturbing forces, which act on a satellite in orbit about the Earth, that were implemented in SGP4. The SGP4 and SDP4 propagators will be explained in detail in the next chapter.

3.1 History

Johannes Kepler (1571-1630) published two laws in Kepler (1609b) and a third in Kepler (1609a) on the orbits of the planets about the Sun, which were deduced from observations of planetary motions taken by Tycho Brahe and others over long periods of time. In today's language they read:

First law The orbit of a planet is an ellipse, of which the Sun is located in one of the foci.

Second law The radius vector of the planet sweeps out equal areas in equal intervals of time.

Third law The ratio between the square of the period and the cube of the major axis of an elliptical orbit is equal for all planets.

Isaac Newton (1643-1727) formulated three laws of motion (Newton, 1687), which in today's language state:

First law Every particle continues in its state of rest or uniform motion in a straight line relative to an inertial reference frame, unless it is compelled to change that state by forces acting upon it.

Second law The time rate of change of linear momentum of a particle relative to an inertial reference frame is proportional to the resultant of all forces acting upon that particle and is collinear with and in the direction of the resultant force.

Third law If two particles exert forces on each other, these forces are equal in magnitude and opposite in direction (action = reaction).

of which the first was already thought of by Galileo Galilei (1564-1642) and the second by Christiaan Huygens (1629-1642).

With the first law some fundamental concepts still used in physics today are defined: force, particle (or point mass), time, uniform motion and inertial reference frame.

Satellites and spacecraft are very small with respect to the Earth and other celestial bodies, and the planets and their moons are in turn small compared to the Sun. In astrodynamics these bodies are often treated as point masses, since only translational motion is treated where objects most of the times have large distances with respect to each other, and no rotational motion.

Kepler's laws will be derived in subsequent sections by making use of Newton's laws which are stated below. However, in reality Kepler deduced these laws from observations *before* Newton published his laws.

The definition of an inertial reference frame can be derived from the first law to be: 'An inertial reference frame is a reference frame with respect to which a particle remains at rest or in uniform motion if no resultant force acts upon that particle.' The first law thus defines an inertial reference frame in which the second law is valid. The second law can be written as

$$\mathbf{F} = \frac{d}{dt}(m\mathbf{V}) \tag{3.1}$$

in which F is the force, m is the mass of the particle and V its velocity.

It can be proven that an entire class of inertial reference frames is known that perform a uniform rectilinear translational motion with respect to the first one, see Figure 3.1. The above equation can be derived to be invariant in different inertial reference frames only if the mass m is constant

$$\mathbf{F} = m\frac{d\mathbf{V}}{dt} \tag{3.2}$$

In his book Newton also formulated the law of universal gravitation,

$$F = G\frac{m_1 m_2}{r^2} \tag{3.3}$$

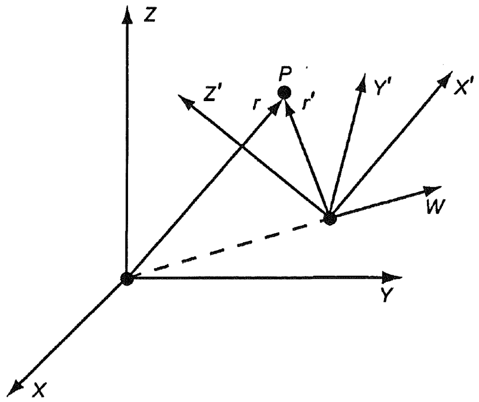


Figure 3.1: Inertial reference frame XYZ and reference frame X'Y'Z' that moves with a constant velocity relative to the XYZ frame. (K.F. Wakker, 2007)

which states that two particles of mass m_1 and m_2 respectively, attract each other with a force F directed along the straight line intersecting both particles through the instantaneous distance r. The force is directly proportional to the product of the masses of both particles and inversely related to the distance squared.

G is the Universal Gravitational Constant which has a value ranging in between 6.672-6.676·10^{-11} $\mathrm{m}^3\mathrm{kg}^{-1}\mathrm{s}^{-2}$ according to different literature (Montenbruck & Gill, 2001), (Vallado, 2007), (Wertz, 2001), see also Figure 3.2.

The equation can also be written in vector form,

$$\mathbf{F}_{12} = -G\frac{m_1 m_2}{|r_{12}^3|}\mathbf{r}_{12} = -G\frac{m_1 m_2}{|r_{12}^2|}\hat{r}_{12} \tag{3.4}$$

where $\mathbf{F}_{12}$ is the force that acts on object 2 due to object 1, $|r_{12}| = |r_2 - r_1|$ is the distance between objects 1 and 2, and $\hat{r}_{12}$ is the unit vector from object 1 to 2. In this vector form, the unit vector is directed from object 1 to 2, and so a minus sign applies here.

The force acting upon particle m_2 can be envisioned to be caused by a gravitational vector field generated by m_1. The force per unit of mass of m_2 at the location of m_2 defined by the position vector $\mathbf{r}_2$ from m_1 to m_2 is called the field strength $\mathbf{g_2 23}$ of the gravitational field generated by m_1, and losing the subscripts, this can be written as

$$\mathbf{g} = -G\frac{m_1}{r^3}\mathbf{r} \tag{3.5}$$

so that we can write

$$\mathbf{F} = m\mathbf{g} \tag{3.6}$$

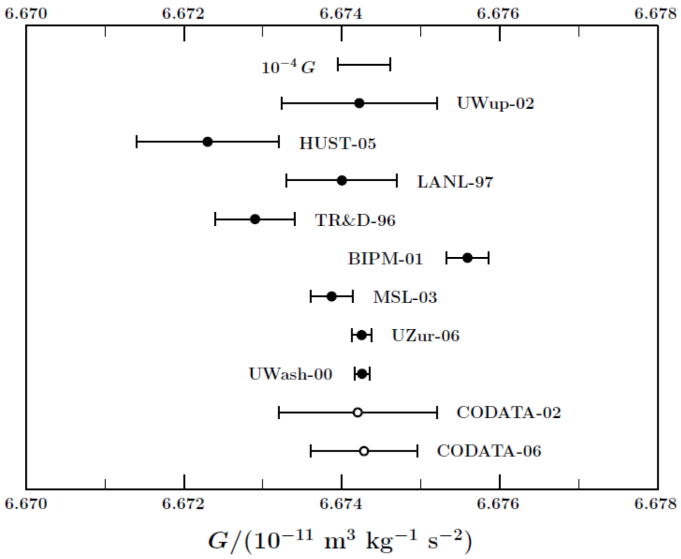


Figure 3.2: Values of the Newtonian constant of gravitation G with their 1-σ-bars of standard uncertainty. (Mohr et al., 2008)

The gravitational potential U of a particle m at an arbitrary distance r can now be expressed as

$$U = -G\frac{m_1}{r} \tag{3.7}$$

so that we can write

$$\mathbf{g} = -\nabla U \tag{3.8}$$

for a conservative gravitational field.

In astronomy, phenomena exist that are not precisely described by Newton's theories, and that is why Albert Einstein (1879-1955) set up the theory of general relativity in Einstein (1915). Even though the theory was proved by experiments, these modifications to Newton's work result in negligibly small deviations for short time spans, and can safely be ignored. However, the modifications have to be incorporated in very precise measurements involving the use of atomic clocks in for instance the navigation systems GPS (Global Positioning System) and Galileo.

3.2 Many-body Problem

Consider a system composed of n bodies, which may be considered point masses, in which each mass m_i has coordinates x_i, y_i, z_i with respect to an inertial reference system, see Figure 3.3. The position of body j with respect

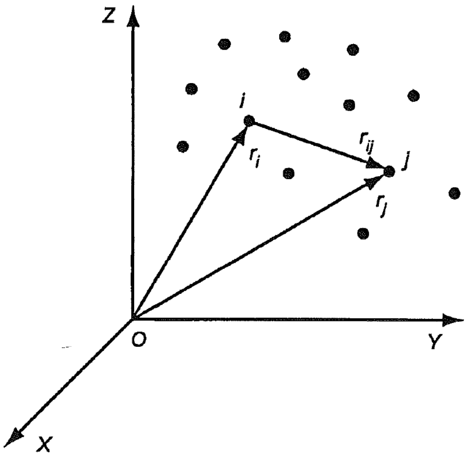


Figure 3.3: The position of n point masses with respect to an inertial reference frame XYZ. (K.F. Wakker, 2007)

to body i can be expressed as

$$\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i \tag{3.9}$$

where the magnitude of vector $\mathbf{r}_{ij}$ is

$$r_{ij} = \left[(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2\right]^{1/2} \tag{3.10}$$

When it is assumed that outside this system no other bodies are present, no external forces act on the system and within the system only gravitational forces occur, then the motion of body i with respect to the inertial reference frame can be written as

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \sum_j^{*} G \frac{m_i m_j}{r_{ij}^3} \mathbf{r}_{ij} \tag{3.11}$$

in which the notation * indicates that a summation is taken from $j = 1$ to $j = n$, excluding $j = i$. The motion is thus mainly governed by bodies that have a large mass and are close to body i.

The above equation of motion can also be written as three scalar second-order differential equations, and thus $3n$ second-order differential equations can be written for n bodies. This set of equations can not be solved analytically for $n > 3$ and thus can only be solved by numerical integration techniques to find the motion of the bodies.

Some general characteristic called the ten integrals of motion can be derived however, which are derived in K.F. Wakker (2007).

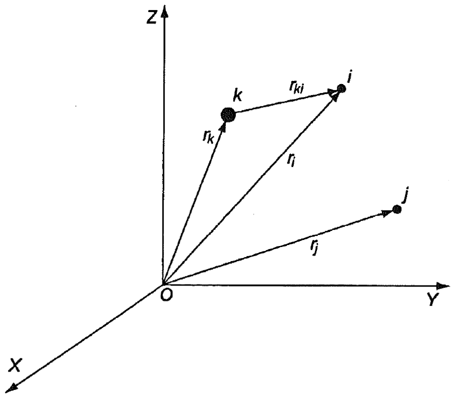


Figure 3.4: The position of bodies i, j and k with respect to an inertial reference frame XYZ. (K.F. Wakker, 2007)

3.3 Relative Motion in the Many-body Problem

The equations of motion stated above were derived with respect to an inertial reference system. However, for most problems it is more advantageous to set up the equations with respect to another body. For instance, for a satellite orbiting the Earth, it is wise to set up the equations of motion of the satellite with respect to the centre of mass of the Earth, as this simplifies the equations.

The equations of motion for body i with respect to one of the n bodies, referred to as body k, will now be set up. The motion of bodies i and k with respect to a non-rotating reference frame XYZ with its origin at the centre of mass of the n-body system (inertial reference frame), see Figure 3.4, can be written as (using Equation 3.11)

$$m_i \ddot{\mathbf{r}}_i = \sum_{j \neq i} G \frac{m_i m_j}{r_{ij}^3} \mathbf{r}_{ij} \tag{3.12}$$

$$m_k \ddot{\mathbf{r}}_k = \sum_{j \neq k} G \frac{m_k m_j}{r_{kj}^3} \mathbf{r}_{kj} \tag{3.13}$$

Expanding the summation with * notation, dividing by m_i and m_k respectively, and then subtracting the second equation from the first, and furthermore fixing the centre of a non-rotating reference frame to body k (and thus losing subscript k) results in

$$\ddot{\mathbf{r}}_i = -G \frac{m_i + m_k}{r_i^3} \mathbf{r}_i + G \sum_j^{*} m_j \left(\frac{\mathbf{r}_j - \mathbf{r}_i}{r_{ij}^3} - \frac{\mathbf{r}_j}{r_j^3} \right) \tag{3.14}$$

in which the * denotes a summation from $j = 1$ to $j = n$ excluding $j = i$ and $j = k$. This equation now describes the motion of body i with respect to a non-rotating reference frame fixed at body k under the influence of all gravitational forces between k, i and the $n-2$ number of bodies j.

A modified potential can now be set up as

$$\hat{U}_i = -G\frac{m_k + m_i}{r_i} \tag{3.15}$$

and a scalar perturbing function R_i as

$$R_i = -G\sum_j^* m_j \left(\frac{1}{r_{ij}} - \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{r_j^3}\right) \tag{3.16}$$

for which it can be shown that

$$\ddot{\mathbf{r}}_i = -\nabla_i(\hat{U}_i + R_i) \tag{3.17}$$

holds true. Body i can be envisioned to move in a force field set up by the (primary and perturbing) potential $\hat{U}_i + R_i$, which is non-central and non-conservative.

Considering only one perturbing body (like the Moon or Sun), Equation 3.14 can be used to set up the motion of a satellite with respect to a non-rotating reference frame fixed at the centre of the Earth

$$\ddot{\mathbf{r}}_s = -G\frac{m_E}{r_s^3}\mathbf{r}_s + Gm_d\left(\frac{\mathbf{r}_{sd}}{r_{sd}^3} - \frac{\mathbf{r}_d}{r_d^3}\right) \tag{3.18}$$

in which the subscripts E, s and d denote the Earth, satellite and perturbing body respectively and furthermore the mass of the satellite is neglected with respect to the mass of the Earth.

The magnitudes a_m and a_d of the main and perturbing accelerations can now be derived to be

$$a_m = G\frac{m_E}{r_s^2} \tag{3.19}$$

$$a_d = Gm_d\frac{r_s}{r_d^3}\sqrt{1 + 3\cos^2\alpha} \tag{3.20}$$

in which α is the angle between $\mathbf{r}_d$ and $\mathbf{r}_s$, see Figure 3.5. For the second equation is was assumed that the satellite is much closer to the Earth than to the Moon and a series expansion was used neglecting terms of order three and higher.

The (maximum) relative perturbing acceleration of a the satellite can now be set up as

$$\frac{a_d}{a_m} = \frac{m_d}{m_E}\frac{r_s^3}{r_d^3}\sqrt{1 + 3\cos^2\alpha} \tag{3.21}$$

$$\left(\frac{a_d}{a_m}\right)_{max} = 2\frac{m_d}{m_E}\left(\frac{r_s}{r_d}\right)^3 \tag{3.22}$$

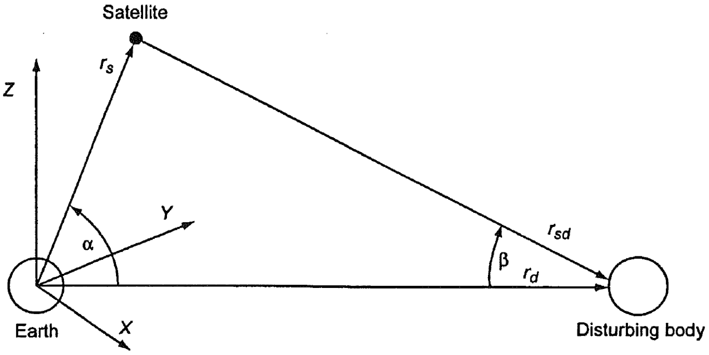


Figure 3.5: The relative positions of the Earth, satellite and a disturbing body. (K.F. Wakker, 2007)

For a satellite in a circular geostationary orbit about the Earth (r =42240 km) the maximum relative perturbing acceleration caused by other celestial bodies can be seen in descending order in Table 3.1, where it has been assumed that all planets move in circular orbits about the Sun.

Table 3.1: Maximum relative perturbing acceleration of a geostationary satellite caused by gravitational attraction of other celestial bodies. (K.F. Wakker, 2007)

Perturbing body	m_d/m_E	r_d/r_s	$(a_d/a_m)_{max}$
Moon	0.0123	9.1	$3.3 \cdot 10^{-5}$
Sun	$3.33 \cdot 10^5$	$3.48 \cdot 10^3$	$1.6 \cdot 10^{-5}$
Venus	0.815	$9.03 \cdot 10^2$	$2.2 \cdot 10^{-9}$
Jupiter	317.9	$1.39 \cdot 10^4$	$2.4 \cdot 10^{-10}$
Mars	0.107	$1.20 \cdot 10^3$	$1.0 \cdot 10^{-10}$
Mercury	0.056	$1.83 \cdot 10^3$	$1.8 \cdot 10^{-11}$
Saturn	95.2	$2.83 \cdot 10^4$	$8.4 \cdot 10^{-12}$
Uranus	14.6	$6.11 \cdot 10^4$	$1.3 \cdot 10^{-13}$
Neptune	17.2	$1.02 \cdot 10^5$	$3.3 \cdot 10^{-14}$
Pluto	0.11	$1.01 \cdot 10^5$	$2.1 \cdot 10^{-16}$
α Proxima Centauri	$3.7 \cdot 10^5$	$9.68 \cdot 10^8$	$8.0 \cdot 10^{-22}$
Large Magellanic Cloud	$6.7 \cdot 10^{15}$	$3.9 \cdot 10^{13}$	$2.3 \cdot 10^{-25}$

For satellite orbits about the Earth with high altitudes it can thus be stated that for first-order perturbing acceleration analysis due to celestial bodies only the influence of the Moon and the Sun needs to be taken into account, as the influence of Venus (3^{rd} on the list) is a factor 10^4 smaller.

3.4 Two-body Problem

If the gravitational attraction between bodies j and i is neglected, Equation 3.14 reduces to

$$\ddot{\mathbf{r}}_i = -G\frac{m_k + m_i}{r_i^3}\mathbf{r}_i \tag{3.23}$$

which now describes an approximation of the motion of body i with respect to a non-rotating reference frame attached to k. In the case of a satellite orbiting the Earth, this would mean that only the gravitational attraction between the Earth (body k) and the satellite (body i) is taken into account. It is shown in Table 3.1 this is valid for a first-order approximation of the motion of the satellite.

If the subscript i is discarded and m_i is ignored with respect to m_k, which can safely be done for a satellite orbiting the Earth, the above equation can be written as

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \tag{3.24}$$

with the gravitational parameter μ defined as

$$\mu = Gm_k \tag{3.25}$$

which is 398600.4418 $\text{km}^3\text{s}^{-2} \pm 0.0008$ for the Earth and with these boundaries agrees with different literature (Wertz, 2001) (K.F. Wakker, 2007) (Montenbruck & Gill, 2001) (Vallado, 2007).

Conservation Laws

Two conservation laws can now be derived. Taking the scalar product of Equation 3.24 and $d\mathbf{r}/dt$ and then integrating yields

$$\frac{1}{2}V^2 - \frac{\mu}{r} = constant = E_k + E_p = E \tag{3.26}$$

in which the first term E_k is the kinetic energy and the second term E_p is the potential energy, both per unit mass of body i, which together form the total energy E. The potential energy can be derived with the gravitational potential as well, using Equation 3.15

$$\hat{U} = -\frac{\mu}{r} \tag{3.27}$$

Taking the vector product of Equation 3.24 and $\mathbf{r}$ and then integrating yields

$$\mathbf{r} \times \mathbf{V} = constant = \mathbf{H} \tag{3.28}$$

which states that the angular momentum of the motion of body i per unit of mass remains constant.

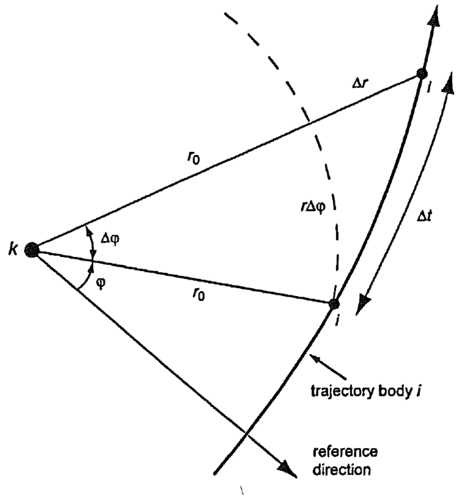


Figure 3.6: Plane of motion of body i about k coinciding with plane XY of a non-rotating reference frame. (K.F. Wakker, 2007)

If the XY plane of a non-inertial reference frame with its origin at body k coincides with the plane of motion, see Figure 3.6, the above equation can be used to write

$$r^2\dot{\phi} = constant = H \tag{3.29}$$

in which H is the magnitude of the angular momentum vector per unit of mass. The angular velocity is thus larger for smaller r.

The area of a surface element A defined by the vector $\mathbf{r}_0$ and $\mathbf{r}_0\Delta r$ defined at time t_0 and $t_0 + \Delta t$ respectively, see Figure 3.6, can be derived to be

$$A = \frac{1}{2}H\Delta t \tag{3.30}$$

by taking the limit $\Delta t \rightarrow 0$, making use of Equation 3.28 and integrating. This equation shows that body i sweeps out equal area segments in its orbital plane during equal time intervals Δt.

With two different mathematical derivations, it can now be shown that the solution for r is given by

$$r = \frac{H^2/\mu}{1 + c\cos(\phi - \omega)} \tag{3.31}$$

in which c and ω are integration constants.

Orbit Shape

In mathematics, a conic section is defined as the curve obtained by intersecting a cone with a plane. This curve is described by the geometrical

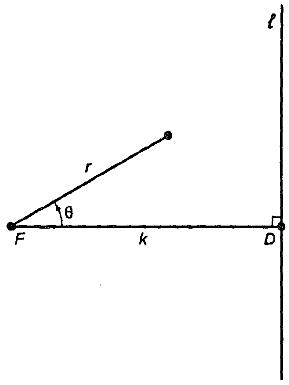


Figure 3.7: Geometrical definition of a conic section. (K.F. Wakker, 2007)

collection of all points P for which the ratio of the distance to a fixed point F and the distance to a fixed line l is constant, see Figure 3.7.

The fixed point F is called the focus, the fixed line l is the directrix and the above-mentioned ratio of distances is the eccentricity e of the conic section. The radius r can be derived as

$$r = \frac{p}{1 + e\cos\theta} \tag{3.32}$$

with

$$p = ek \tag{3.33}$$

The following types of conic section curves can now be distinguished for different values of the eccentricity

$$e = 0 \rightarrow \textit{circle} \tag{3.34}$$

$$0 < e < 1 \rightarrow \textit{ellipse} \tag{3.35}$$

$$e = 1 \rightarrow \textit{parabola} \tag{3.36}$$

$$e > 1 \rightarrow \textit{hyperbola} \tag{3.37}$$

For the hyperbola another curve on the right-hand side of line l is defined by inserting a minus sign in front of p and e in Equation 3.32.

If Equation 3.32 is compared with Equation 3.31, it can be seen that body i moves in a conic section about body k, with $p = H^2/\mu$, $e = c$, $\theta = \phi - \omega$ and $H = r^2\dot{\theta} = r^2\dot{\phi}$. The hyperbola curve on the right side of line l has no physical meaning, since the gravitational force is attractive and not repulsive.

The orbits described by conic sections are called Keplerian orbits.

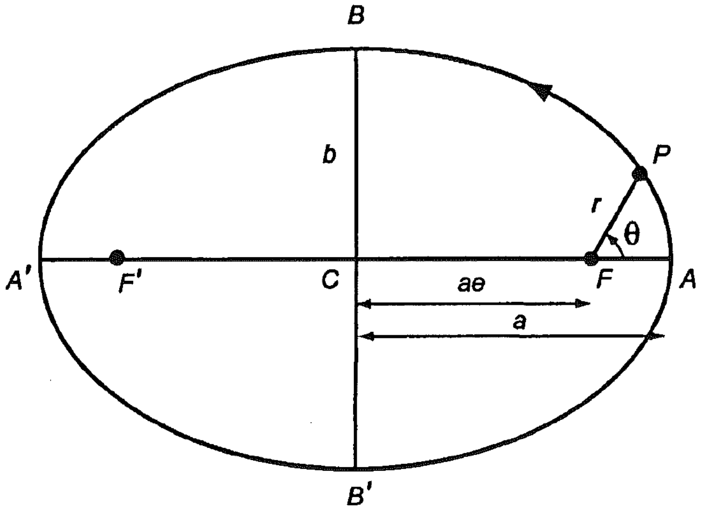


Figure 3.8: Geometry of an elliptical orbit. (K.F. Wakker, 2007)

3.5 Elliptical Orbits

Only orbital debris which is in orbit about the Earth is included in the catalog of the SSN, so only elliptical (and circular) orbits are treated here, as parabolic and hyperbolic orbits cause objects to move away from the Earth.

Geometry

In Figure 3.8 the geometry of an elliptical orbit can be seen along with some definitions of parameters. Lines AA' and BB' are two axes of symmetry of length 2a and 2b respectively which are called the major and minor axis of the ellipse.

Using the figure and Equation 3.32 it can be derived that

$$p = a(1 - e^2) \tag{3.38}$$

and inserting this in Equation 3.32 yields

$$r = \frac{a(1 - e^2)}{1 + e\cos\theta} \tag{3.39}$$

in which θ is called the true anomaly.

The radius at apocenter and pericentre can now be obtained

$$\begin{aligned} r_p &= r_{\theta=0} = a(1 - e) \\ r_a &= r_{\theta=\pi} = a(1 + e) \end{aligned}$$

from which a relation for a and e can be derived

$$a = \frac{r_a + r_p}{2} \tag{3.40}$$

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{3.41}$$

The distance between the centre of the ellipse and the focus F can be written as

$$CF = a - r_p = ae \tag{3.42}$$

while it can be derived that the crossing point of the orbit with the minor axis occurs when

$$\theta = \pm \arccos(-e) \tag{3.43}$$

Furthermore b can be written as

$$b = a\sqrt{1 - e^2} \tag{3.44}$$

$$\cos\theta = -e \tag{3.45}$$

while distance FB can be derived as

$$FB = a \tag{3.46}$$

The eccentricity can now be written as

$$e = \sin(FBC) \tag{3.47}$$

Some other characteristics of an ellipse:

- The ellipse has a second focus F', for which holds: $CF' = CF$.
- If a circle with center C and radius a is drawn around an ellipse, then for each line through a point P on the ellipse and perpendicular to a, see Figure 3.9, holds

$$\frac{PG}{P'G} = \frac{b}{a} = \sqrt{1 - e^2} \tag{3.48}$$

- The area enclosed by an ellipse is equal to πab

Furthermore it can now be derived that

$$E = E_k + E_p = constant = \frac{V^2}{2} - \frac{\mu}{r} < 0 \tag{3.49}$$

It can be shown that the larger the orbital energy, the larger the semi-major axis of the orbit.

Orbital Velocity and Period

A relation for the velocity in terms of the radius, called the Vis-Viva integral, can be derived using the above equations for the ellipse

$$V^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right) \tag{3.50}$$

Some conclusions can now be drawn about the variation of the velocity in an elliptical orbit:

- The velocity reaches a minimum value when r is at its maximum at the apocenter
  $$V^2_{min} = V^2_a = V^2_{c_a}(1-e) \tag{3.51}$$
  in which $V^2_{c_a}$ is the local circular velocity at apocenter.
- It reaches a maximum at pericentre
  $$V^2_{max} = V^2_p = V^2_{c_p}(1+e) \tag{3.52}$$
- The ratio between the maximum and minimum velocity can be written as
  $$\frac{V_p}{V_a} = \frac{1+e}{1-e} = \frac{r_a}{r_p} \tag{3.53}$$
- The velocity in an elliptical orbit is equal in magnitude (not in direction) to the local circular velocity for $r = a$. From the geometry of the ellipse it follows that the satellite is then located on the minor axis.
- The radial velocity is zero at pericentre and apocenter. It reaches a maximum value at $\theta = 90°$ and $\theta = 270°$.
- The normal velocity is always positive and reaches a maximum value at pericentre and minimum value at apocenter.
- The flight path angle is zero at pericentre and apocenter and reaches an extreme value at $\cos\theta = -e$, thus where the minor axis of the ellipse intersects with the orbit.

The period of an elliptical orbit is given by

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{3.54}$$

which can be used to write the mean angular motion n as

$$n = \frac{2\pi}{T} = \sqrt{\frac{\mu}{a^3}} \tag{3.55}$$

Circular Orbits

A circular orbit is a special case of an elliptical orbit in which the velocity, called the circular velocity V_c, is constant. It is given by

$$V_c = \sqrt{\frac{\mu}{r}} \tag{3.56}$$

Now the orbital period of the circular orbit can be written as

$$T_c = \frac{2\pi r}{V_c} = 2\pi\sqrt{\frac{r^3}{\mu}} \tag{3.57}$$

Kepler's Third Law and Equation

Using the period of an elliptical orbit, Kepler's third law can now be derived by using

$$\frac{a^3}{T^2} = \frac{\mu}{4\pi^2} = \frac{Gm_k}{4\pi^2}\left(1 + \frac{m_i}{m_k}\right) \tag{3.58}$$

which is a more accurate form of Kepler's law. But if m_i is assumed to be negligible with respect to m_k, one arrives at Kepler's third law in its original form

$$\frac{a^3}{T^2} = \frac{Gm_k}{4\pi^2} = constant \tag{3.59}$$

To find an expression for the relation between position in the orbit and time, previous analysis can be used to write dt as

$$dt = \frac{r^2}{\sqrt{\mu p}} d\theta \tag{3.60}$$

$$= \frac{p^2}{\sqrt{\mu p}} \frac{d\theta}{(1 + e\cos\theta)^2} \tag{3.61}$$

and integrated as

$$t = \sqrt{\frac{p^3}{\mu}} \int_0^{\theta_e} \frac{d\theta}{(1 + e\cos\theta)^2} \tag{3.62}$$

The evaluation of the integral for an elliptical orbit yields

$$t - \tau = \sqrt{\frac{a^3}{\mu}} \left[2\arctan\left(\sqrt{\frac{1-e}{1+e}}\tan\frac{\theta}{2}\right) - e\sqrt{1-e^2}\frac{\sin\theta}{1 + e\cos\theta}\right] \tag{3.63}$$

where τ is the time of passage of the pericentre ($\theta = 0$). However, this solution is not really suitable for analytic nor numerical analysis.

A simpler expression can be obtained using Figure 3.9, in which a circle is drawn enclosing the ellipse. A line perpendicular to the major axis through

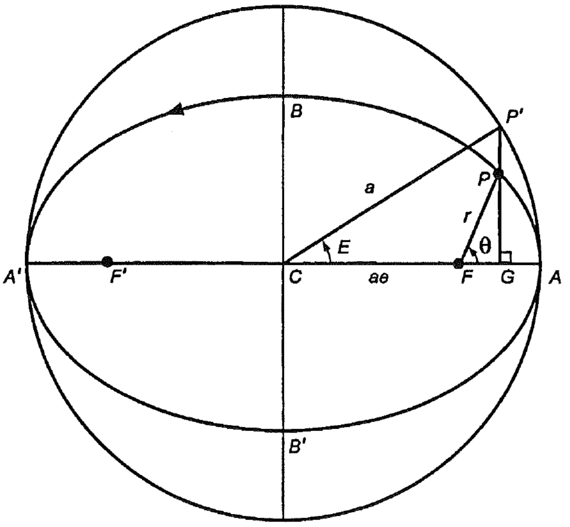


Figure 3.9: Drawing a circle enclosing the ellipse, the eccentric anomaly E is defined. (K.F. Wakker, 2007)

body i (point P) intersects the circle in P'. The angle ACP' is called the eccentric anomaly E of the satellite.

From Figure 3.9 the following can be obtained

$$r\cos\theta = a\cos E - ae \tag{3.64}$$

and using the previously described characteristic of a circle around an ellipse it follows

$$r\sin\theta = a\sqrt{1-e^2}\sin E \tag{3.65}$$

and thus for an ellipse it holds

$$r = a(1 - e\cos E) \tag{3.66}$$

Using trigonometric relations the following holds for θ

$$\tan\frac{\theta}{2} = \sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2} \tag{3.67}$$

And using previous relations the following expression is derived

$$\begin{aligned} E - e\sin E &= \sqrt{\frac{\mu}{a^3}}(t-\tau) && (3.68)\\ &= n(t-\tau) && (3.69)\\ &= M && (3.70) \end{aligned}$$

in which τ is an integration constant and M is called the mean anomaly. τ is found to be the time of last pericentre passage when filling in $E = 0$.

The above equation is known as Kepler's equation and can be used to determine E at a certain time t, after which θ can be obtained from Equation 3.67. Going from θ to t is not really a problem, but obtaining θ from t involves solving Kepler's equation for E which can not be solved analytically.

Therefore the following function is set up using Kepler's equation

$$f(E, M) = E - e \sin E - M \tag{3.71}$$

which can be shown to always have a value E for each M when the function is set to zero.

The Newton-Rhapson method can be applied to solve the equation numerically as

$$E_{k+1} = E_k - \frac{f(E_k, M)}{\frac{d}{dE}\left\{f(E, M)\right\}|_{E=E_k}} = E_k - \frac{E_k - e \sin E_k - M}{1 - e \cos E_k} \tag{3.72}$$

and because $e < 1$ the following iterative method can be used to converge to E

$$E_{k+1} = M + e \sin E_k \tag{3.73}$$

in which $E_0 = M$ can be used as a starting value.

Orbital Elements

Summarising from the above, the constants a, e and ω determine respectively the size, shape and orientation of the ellipse in the orbital plane, while τ is required to determine the position of body i in its orbit at a specified moment in time.

The nodal line is defined as the line intersection between the reference plane YX and the orbital plane. The argument of perigee ω defines the orientation of the conic section, it is the angle between the nodal line and the radius vector from the origin to the pericentre.

The argument of latitude is defined as

$$u = \theta + \omega \tag{3.74}$$

However, to define an elliptical orbit in three-dimensional space two more parameters are needed.

The first one is the inclination i which is defined as the angle between the orbital plane and the reference plane XY, see Figure 3.10.

Depending on whether the reference plane is the ecliptic or the equatorial plane, the second parameter needed is called the longitude of the ascending node or right ascension of the ascending node, indicated by Ω, where rortation goes from South to North. It is the angle between the nodal line, and the X-direction, usually the vernal equinox. Ω ranges from 0 tot 360 degrees and i from 0 to 180 degrees.

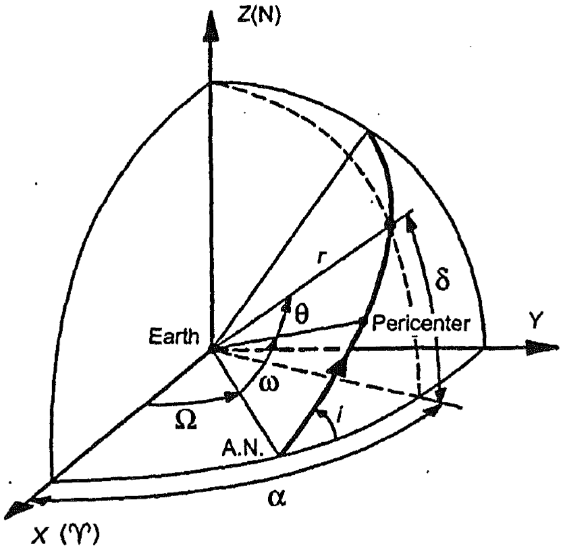


Figure 3.10: Definition of the 3-D orbital elements. (K.F. Wakker, 2007)

The six parameters $a, e, i, \Omega, \omega, \tau$ are called the classical or Keplerian orbital elements and combined with time t they define an elliptical orbit in three-dimensional space and the position of body i within this orbit. Instead of τ and t also the true anomaly θ (at a given epoch) can be used as the sixth orbital element.

However, the state of the satellite can also be fully described by its position and speed using rectangular coordinates that define position and velocity, $x, y, z, \dot{x}, \dot{y}, \dot{z}$, relative to a reference frame.

So it is not a coincidence that the amount of independent orbital elements is six, since the position and velocity are determined by three second-order differential equations. A solution to this set yields six integration constants which are chosen in such a way that they have a physical meaning as Keplerian orbital elements.

Transformation from Rectangular Coordinates to Orbital Elements

The following conversion from rectangular coordinates, defining position $\mathbf{r}$ and velocity $\mathbf{V}$, to orbital elements is taken from Wertz (2001). Different conversion strategies can also be found in K.F. Wakker (2007), Montenbruck & Gill (2001) and Vallado (2007). This conversion scheme can be used to transform the output of the SGP4 propagator, which is a state-vector in rectangular coordinates, back into orbital elements again, as a reference.

The position and velocity vector are given by their scalar components as

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{3.75}$$

$$\mathbf{V} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \tag{3.76}$$

and the magnitude of these vectors yields their scalar form as

$$r = |\mathbf{r}| = \sqrt{x^2 + y^2 + z^2} \tag{3.77}$$

$$V = |\mathbf{V}| = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{3.78}$$

The orbit angular momentum vector **h** can now be determined from

$$\mathbf{h} = \mathbf{r} \times \mathbf{V} \tag{3.79}$$

The vector to the ascending node **N** is given by

$$\mathbf{N} = \hat{\mathbf{z}} \times \hat{\mathbf{h}} = \hat{\mathbf{z}} \times \frac{\mathbf{h}}{|\mathbf{h}|} \tag{3.80}$$

in which $\hat{\mathbf{z}}$ is the unit vector perpendicular to the equatorial plane of the Earth.

If a reference frame with its Z-direction normal to the Earth equatorial plane is chosen it follows

$$\hat{\mathbf{z}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.81}$$

The unit vector to the ascending node can now be obtained from

$$\hat{\mathbf{N}} = \frac{\mathbf{N}}{|\mathbf{N}|} \tag{3.82}$$

The eccentricity vector **e** which lies in the direction of pericentre can be determined from

$$\mathbf{e} = \frac{\mathbf{V} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{r} \tag{3.83}$$

The semi-major axis a can now be determined from

$$a = \left(\frac{2}{r} - \frac{V^2}{\mu}\right)^{-1} \tag{3.84}$$

while the eccentricity of the orbit follows as

$$e = |\mathbf{e}| \tag{3.85}$$

Next, the inclination i is given by

$$\cos i = \frac{h_z}{|\mathbf{h}|} \tag{3.86}$$

in which i is defined from 0 to 180 degrees.

The right ascension of the ascending node Ω follows as

$$\tan \Omega = \frac{N_y}{N_x} \tag{3.87}$$

which needs an atan2 function that solves for a value between 0 and 360 degrees.

The argument of perigee ω can be obtained from

$$\cos \omega = \hat{\mathbf{e}} \cdot \hat{\mathbf{N}} = \frac{\mathbf{e}}{e} \cdot \hat{\mathbf{N}} \tag{3.88}$$

and will be between 0 and 180 degrees if $(\hat{\mathbf{N}} \times \mathbf{e}) \cdot \mathbf{h} > 0$ and between 180 and 360 degrees when this quantity is less than zero.

The true anomaly θ can now be determined from

$$\begin{aligned} \cos \theta &= \hat{\mathbf{r}} \cdot \hat{\mathbf{e}} && (3.89) \\ &= \frac{\mathbf{r}}{r} \cdot \frac{\mathbf{e}}{e} && (3.90) \end{aligned}$$

and will be between 0 and 180 degrees if $(\mathbf{e} \times \mathbf{r}) \cdot \mathbf{h} > 0$ and between 180 and 360 degrees when this quantity is less than zero.

Then, using relations set up in previous sections, the time of last pericentre passage τ, argument of perigee u, semi-latus rectum p, mean angular motion n, total energy E, the eccentric anomaly E, mean anomaly M and orbital period P can be computed.

3.6 Perturbing Forces and Perturbed Satellite Orbits

It was shown that the orbit of a body i (satellite) can only describe a perfect Keplerian orbit (conic section) if the primary body k (Earth) is assumed to be a point mass or if the mass distribution of body k is radially symmetrical, and if no other bodies are present and no other forces than the gravitational one act on the system. In reality this is of course never the case, and depending on the accuracy required, one has to account for different perturbing forces.

In the case that the primary body is the Earth, perturbations exist due to the Earth's irregular gravity field, atmospheric drag, attraction by other celestial bodies, solar radiation pressure and electromagnetic forces. The latter two are generally very small compared to the others and are therefore not taken into account in the SGP4 propagator and thus will also not be treated here.

An overview of the relative order of magnitude of all perturbing accelerations due to these forces with respect to the distance from the centre of the Earth can be seen in Figure 3.11, the content of which will be explained in subsequent sections. The perturbing forces are small compared to the total gravitational force and thus only cause small deviations in the Keplerian orbits.

However, resonance effects can cause small forces to accumulate significantly over a large number of orbital revolutions.

The perturbing accelerations from different sources will be treated in the subsequent chapters. An elaborate discussion on the effect of these accelerations on the orbital elements is given in Wakker (2007).

Earth's Gravity Field Forces

The general external gravitational potential of a body with arbitrary mass density distribution can be set up using mathematical spherical harmonics theory

$$U = -\frac{\mu}{r}\Bigg[1 - \sum_{n=2}^{\infty} J_n \left(\frac{R}{r}\right)^n P_n(\sin\phi) \\ - \sum_{n=2}^{\infty}\sum_{m=1}^{n} J_{n,m} \left(\frac{R}{r}\right)^n P_{n,m}(sin\phi)\left\{\cos m(llambda - \lambda_{n,m})\right\}\Bigg] \tag{3.91}$$

in which r,ϕ and λ are the spherical coordinates of the point at which the potential U is evaluated, with respect to the geocentric rotating reference frame when the Earth is considered as the body for which the potential is set up; r is the distance between the point under consideration and the centre of the Earth and must be larger than the maximum radius of the Earth, ϕ is the geocentric latitude and λ is the geographical longitude. Furthermore, R is the mean equatorial radius of the Earth, J_n, $J_{n,m}$ and $\lambda_{n,m}$ are gravity field coefficients (constants). Finally, $P_n(x)$ and $P_{n,m}(x)$ are respectively Legendre polynomials and associated Legendre functions of the first kind, in which $x = \sin\phi$ and the functions are evaluated as

$$P_{n,m}(x) = (1 - x^2)^{m/2}\frac{d^m P_n(x)}{dx^m} \tag{3.92}$$

$$P_n(x) = \frac{1}{(-2)^n n!}\frac{d^n}{dx^n}(1 - x^2)^n \tag{3.93}$$

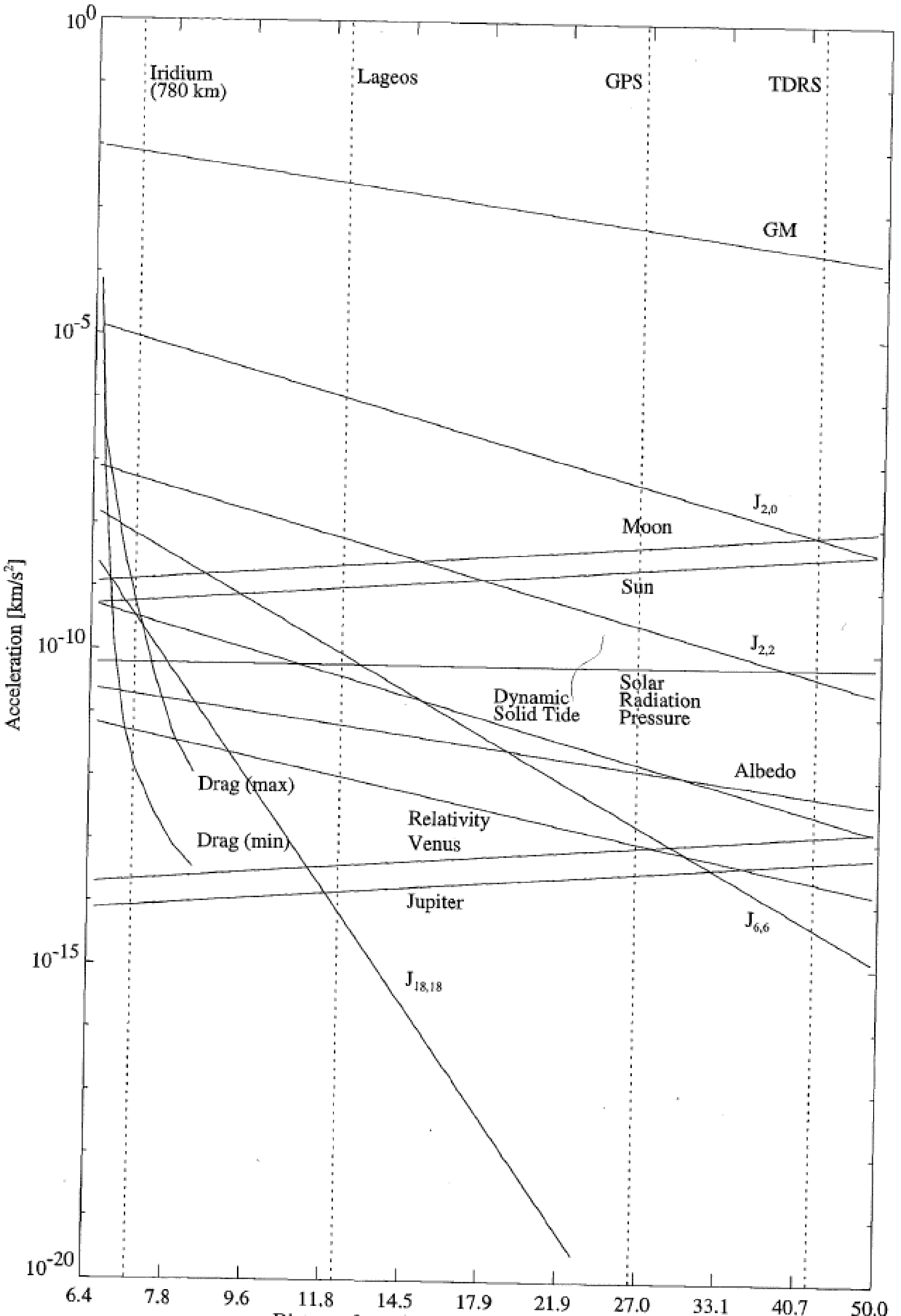


Figure 3.11: Order of magnitude of various perturbing accelerations of a satellite with respect to the distance from the centre of the Earth. (Montenbruck & Gill, 2001)

The first term inside the brackets in Equation 3.91 defines the perfect gravitational field for a point mass or a body with a radially symmetrical mass density distribution, and then a series of perturbing terms is added which try to simulate the real mass distribution of the Earth as well as possible.

The first series account for deviations in the mass density distribution in North-South direction and are called zonal harmonics, while the second series named tesseral ($m \neq n$) and sectorial ($n = m$) harmonics represent mass distribution in both North-South and East-West direction.

J_2, J_3 and J_4 are the only gravity field coefficients used in the SGP4 theory, and their values are, according to the WGS-72 model (used in SGP4)

$$J_2 = 1082.616 \cdot 10^{-6} \tag{3.94}$$

$$J_3 = -2.53881 \cdot 10^{-6} \tag{3.95}$$

$$J_4 = -1.65597 \cdot 10^{-6} \tag{3.96}$$

and since these coefficients only account for zonal harmonics in the first series in equation 3.91 no corresponding value for $\lambda_{n,m}$ is needed.

When comparing all gravity field coefficients of the Earth, J_2 is by far the largest, being a factor 10^3 larger than all other coefficients. For first-order perturbing analysis it is thus safe if only J_2, and in this case combined with J_3 and J_4, is accounted for (Klinkrad, 2006).

If one wants to know the magnitude of only the perturbing acceleration, this can be obtained from

$$\mathbf{f} = -\nabla\left(U + \frac{\mu}{r}\right) \tag{3.97}$$

and when only J_2, J_3 or J_4 is taken into account this is evaluated as

$$\mathbf{f_2} = -\nabla\left[\frac{\mu}{r}J_2\left(\frac{R}{r}\right)^2 P_2(\sin\phi)\right] \tag{3.98}$$

$$\mathbf{f_3} = -\nabla\left[\frac{\mu}{r}J_3\left(\frac{R}{r}\right)^2 P_3(\sin\phi)\right] \tag{3.99}$$

$$\mathbf{f_4} = -\nabla\left[\frac{\mu}{r}J_4\left(\frac{R}{r}\right)^2 P_4(\sin\phi)\right] \tag{3.100}$$

with the Legendre polynomials evaluated as

$$P_2(\sin\phi) = \frac{1}{2}(3\sin^2\phi - 1) \tag{3.101}$$

$$P_3(\sin\phi) = \frac{1}{2}(5\sin^3\phi - 3\sin\phi) \tag{3.102}$$

$$P_4(\sin\phi) = \frac{1}{8}(35\sin^4\phi - 30\sin^2\phi + 3) \tag{3.103}$$

Substitution of this relation for J_2 into the equation for f_2 now yields

$$\mathbf{f_2} = -\nabla \left[\frac{1}{2}\mu J_2 \frac{R^2}{r^3}(3\sin^2\phi - 1) \right] \tag{3.104}$$

and with $\sin\phi = z/r$

$$\mathbf{f_2} = -\nabla \left[\frac{1}{2}\mu J_2 \frac{R^2}{r^3}(3\frac{z^2}{r^2} - 1) \right] \tag{3.105}$$

Now the perturbative acceleration can be written in its rectangular components as

$$f_{2,x} = -\frac{3}{2}\mu J_2 \frac{R^2}{r^5} x(1 - 5\frac{z^2}{r^2}) \tag{3.106}$$
$$f_{2,y} = -\frac{3}{2}\mu J_2 \frac{R^2}{r^5} y(1 - 5\frac{z^2}{r^2}) \tag{3.107}$$
$$f_{2,z} = -\frac{3}{2}\mu J_2 \frac{R^2}{r^5} z \left(3 - 5\frac{z^2}{r^2}\right) \tag{3.108}$$

and taking the relevant partial derivatives we arrive at the spherical components

$$f_{2,r} = \frac{3}{2}\mu J_2 \frac{R^2}{r^4}(3\sin^2\phi - 1) \tag{3.109}$$
$$f_{2,\phi} = -\frac{3}{2}\mu J_2 \frac{R^2}{r^4} \sin 2\phi \tag{3.110}$$
$$f_{2,\lambda} = 0 \tag{3.111}$$

Using these relations it can be shown that $f_r = 0$ for $\phi = -35.26°, 35.26°$ and $f_\phi = 0$ for $\phi = -90°, 0, 90°$. The effect of J_2 can thus be seen to be caused by an extra band of mass around the Earth's equator and a subtraction of mass around the poles, applied to the radially symmetrical case. This corresponds to the physical ellipsoidial shape of the Earth.

Furthermore, the maximum values are

$$|f_{2,r}|_{max} = 3\mu J_2 \frac{R^2}{r^4} \qquad \text{for} \qquad \phi = -90°, 90° \tag{3.112}$$
$$|f_{2,\phi}|_{max} = \frac{3}{2}\mu J_2 \frac{R^2}{r^4} \qquad \text{for} \qquad \phi = -45°, 45° \tag{3.113}$$

and for a satellite at an altitude of 250 km these maximum values are about 0.3% of the local gravitational acceleration. Although not shown here, a similar derivation can be done for J_3 and J_4 as well.

Because the orbital plane of a satellite to first order is fixed in space while the Earth rotates about its axis, most deviations average out over periods of multiple days.

However, for satellites with an orbital period synchronised to the rotation of the Earth and for geostationary satellites, the same part of the gravity field is orbited repeatedly for longer periods of time and as a results resonance phenomena occur which cause the satellite to drift in a particular direction.

The SDP4 theory accounts for resonant geopotential perturbations for 12- and 24-hour orbits (Klinkrad, 2006).

Aerodynamic Forces

From aerodynamics theory the acceleration of the satellite due to atmospheric drag may be written as

$$\mathbf{f_D} = -C_D \frac{1}{2} \rho \frac{S}{M} \left| \mathbf{V} \right| \mathbf{V} \tag{3.114}$$

where ρ is the atmospheric density, $\mathbf{V}$ is the satellite's velocity relative to the Earth's atmosphere, C_D is its aerodynamic drag coefficient related to its cross-sectional area S perpendicular to the velocity vector, and M is the mass of the satellite. Due to physical reality the acceleration vector is opposite to the velocity vector and thus actually causes a deceleration of the satellite, which is integrated using a minus sign.

However, data on C_D and ρ in high-altitude regions of the atmosphere, in which satellites are orbiting, are not very accurate. This is due to the fact that the local thin atmosphere has a density that changes continuously over time, and is mainly dependent on the activity of the Sun.

In practise, the density of the atmosphere is determined with tracking data of multiple satellites and accelerometer data as well. The semi-major axis and eccentricity of a satellite's orbit decrease due to atmospheric drag and this decrease can then be used to determine the term $C_D\rho$.

When this is known, a drag coefficient versus altitude profile, which is based on a standard atmospheric model, is used to determine the atmospheric density.

In SGP4 a power-law altitude profile of air density is used (Klinkrad, 2006), see Section 4.3.

Attraction by Other Celestial Bodies

It was derived previously that the perturbing potential due to the gravitational field of bodies other than the Earth can be written as

$$R = -G \sum_{j \neq k,i} m_j \left(\frac{1}{r_{ij}} - \frac{x_i x_j + y_i y_j + z_i z_j}{r_j^3} \right) \tag{3.115}$$

where r_i and r_j are the position vectors of the satellite (body i) and the perturbing body j with respect to the origin of the reference frame centered at the Earth (body k).

The acceleration due to a perturbing body j can now be determined to be

$$\mathbf{f} = \nabla \left[\mu_j \left(\frac{1}{r_{ij}} - \frac{x_i x_j + y_i y_j + z_i z_j}{r_j^3} \right) \right] \tag{3.116}$$

in which $\mu_j = G m_j$. And with

$$r_{ij}^2 = (x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2 \tag{3.117}$$

the rectangular components of the perturbing acceleration can now be written as

$$f_x = \mu_j \left(\frac{x_j - x_i}{r_{ij}^3} - \frac{x_j}{r_j^3} \right) \tag{3.118}$$

$$f_y = \mu_j \left(\frac{y_j - y_i}{r_{ij}^3} - \frac{y_j}{r_j^3} \right) \tag{3.119}$$

$$f_z = \mu_j \left(\frac{z_j - z_i}{r_{ij}^3} - \frac{z_j}{r_j^3} \right) \tag{3.120}$$

The maximum ratio between the magnitude of the acceleration due to the perturbing body with respect to the main acceleration cause by the Earth has been derived in a previous section as

$$\left(\frac{f_d}{f_E} \right)_{max} = 2 \frac{m_d}{m_E} \left(\frac{r_i}{r_d} \right)^3 \tag{3.121}$$

where they were also evaluated numerically already for a LEO satellite.

This ratio has a value of about $6.5 \cdot 10^{-8}$ for an altitude of 400 km, and around $1.6 \cdot 10^{-5}$ for a geostationary satellite. Comparing these values it can thus be concluded that the effect of other celestial bodies starts to become more significant for higher orbits.

That is the reason why the SDP4 propagator was set up, which is used on orbits with a period of 225 minutes and larger. It includes perturbations due to the gravitational attraction of the Sun and the Moon as well as resonance effects of Earth's gravity.

Radiation Pressure

The acceleration of a satellite due to the solar radiation pressure force can be described by

$$f = C_R \frac{WS^*}{Mc} \tag{3.122}$$

where W is the power density of the incoming radiation in W/m^2, S^* is the effective cross-sectional area that points in the direcion of the radiation, c is the speed of light, M is the mass of the satellite and C_R is the reflectivity coefficient of the satellite.

In Earth orbit, direct sunlight with a power density of W_S =1360 W/m^2 as well as Earth albedo and infrared radiation emitted by the Earth need to be taken into account. At an altitude of 200 km the later two have values of W_{alb} =400 W/m^2 and W_{IR} =300 W/m^2 respectively, which will rapidly decrease with increasing altitude.

This force is generally very small compared to the Earth's gravitational attraction, but may still lead to significant perturbations for large-sized satellites with high reflectivity and small mass over a long period of time (e.g. Echo-1).

The SDP4 theory has a non-conservative term which can accommodate for solar radiation pressure effects (Klinkrad, 2006).

Special and General Perturbations Methods

The equation of motion including perturbations for a satellite with respect to a non-rotating geocentric equatorial reference frame can be written as

$$\frac{d^2\mathbf{r}}{dt^2} + \frac{\mu}{r^3}\mathbf{r} = -\nabla R + \mathbf{f} \tag{3.123}$$

in which the perturbing or disturbing potential R includes all perturbing accelerations that can be written as a potential function, and $\mathbf{f}$ describes all perturbing accelerations that can not be written as the gradient of a scalar function.

This equation can in general not be solved analytically, so numerical integration techniques or approximate analytical methods are used. The former are called special perturbations methods and the latter general perturbations methods. For the SGP4 (note: Simplified General Perturbations) theory the latter is used.

Special perturbations methods yield just one trajectory for one particular satellite using its unique initial conditions.

General perturbations methods however use analytical methods in which the perturbing accelerations are expanded into series and integrated analytically term by term. A limited number of terms is used to decrease complexity and computation time, while still being reasonably accurate.

The solutions of the differential equations are now obtained in the form of analytical expressions, and describe the change of the orbit as a function of time for a particular perturbing force.

These latter methods thus yield solutions which are generally applicable to all satellites for all initial conditions. In this way, the SGP4 theory can thus be used to propagate all objects in the TLE catalog to any point in time.

It is noted however that propagating further in time also increases uncertainty of the solution, but this also holds for special perturbations techniques.

3.7 Elementary Analysis of Orbit Perturbations

Using the equations above, elementary analysis can be performed yielding for instance

$$\frac{da}{dt} = \frac{2a^2}{\mu}\mathbf{V} \cdot \mathbf{f} \tag{3.124}$$

$$\frac{d\mathbf{H}}{dt} = \mathbf{r} \times \mathbf{f} \tag{3.125}$$

which shows that the semi-major axis and angular momentum of a satellite's orbit change in time due to any perturbating force, except when $\mathbf{V}$ and $\mathbf{f}$ are perpendicular, yielding the dot-product zero.

Furthermore, it can be derived that for small orbit perturbations the changes in the components of the angular momentum ΔH can be approximated by, see Figure 3.12

$$\Delta H_x = H_0 \sin i_0 \Delta\Omega \tag{3.126}$$

$$\Delta H_y = -H_0 \cos i_0 \Delta i - \Delta H \sin i_0 \tag{3.127}$$

$$\Delta H_z = -H_0 \sin i_0 \Delta i + \Delta H \cos i_0 \tag{3.128}$$

which yield changes in the orbit inclination Δi and right ascension of the ascending node $\Delta\Omega$ with respect to their initial values at time t_0 as

$$\Delta i = -\frac{\Delta H_y \cos i_0 + \Delta H_z \sin i_0}{H_0} \tag{3.129}$$

$$\Delta\Omega = \frac{\Delta H_x}{H_0 \sin i_0} \tag{3.130}$$

$$\Delta H = -\Delta H_y \sin i_0 + \Delta H_z \cos i_0 \tag{3.131}$$

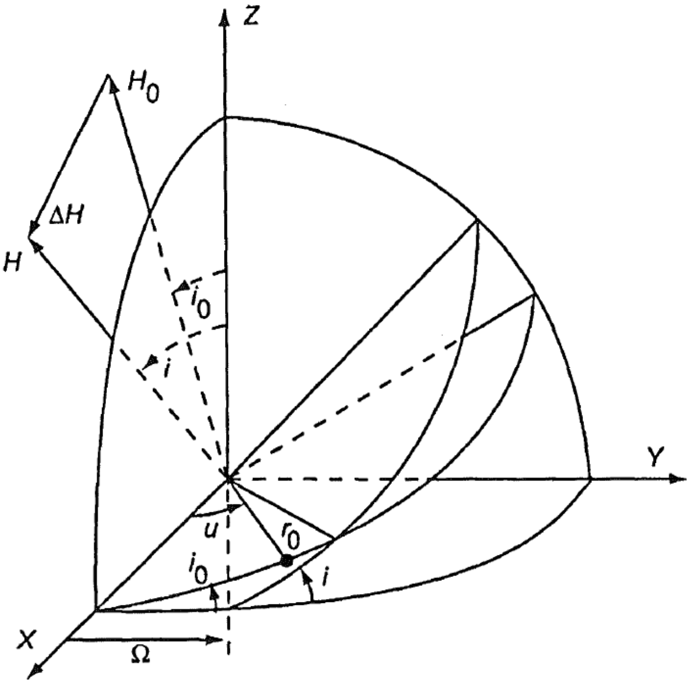


Figure 3.12: Initial and perturbed orbital plane and orbital angular momentum vector definition. (Wakker, 2007)

in which

$$H_0 = \sqrt{\mu r_0} = n_0 r_0^2 \tag{3.132}$$

where n_0 is the unperturbed mean motion.

These expressions can now be used to yield changes in the orbital elements due to certain perturbing accelerations for a defined period of time.

3.8 Variation of Orbital Elements Method

Without perturbations the position and velocity of a satellite can be converted to the orbital elements which stay constant. For perturbed orbits, one can still carry out this conversion to yield the fictitious instantaneous Keplerian orbit.

A perturbed orbit can be thought of to have ever-changing orbital elements which describe a Keplerian orbit that just touches the true orbit. The changing elements are called the osculating orbital elements, describing an osculating Keplerian orbit for that moment in time.

This theory can be used to write the time derivatives of all Keplerian elements of an orbit in terms of the partial derivative of the perturbing potential R with respect to these elements. These equations are called the Lagrange's Planetary Equations and can for instance be found in Wakker (2007) and Wertz (2001).

When the perturbing function is known as a function of x, y, z then the partial derivatives of the function with respect to the orbital elements can be found as well.

These equations also have a different form for low-eccentricity orbits to cope with singularities.

Furthermore, they can be written as functions of the accelerations as well, which has been done in Gauss' form of the planetary equations.

If the method of the variation of orbital elements is applied to the gravitational potential of the Earth, one can find accurate equations of the changes of a particular orbital element with respect to a certain time increase. When this is done, the resulting equations will show secular, short-period and long-period effects, as can be seen in Figure 3.13.

The secular effects are due to linear terms in the resulting equations and continuously increase or decrease the orbital elements with respect to time.

The short-period effects have a periodic time in the order of the orbital period of the satellite, due to the changing local gravitational attraction during one orbit.

The long-period orbits can have very different periods. For instance, for LEO orbits the long-period effects on the argument of perigee due to the J_2 term can have a period on the order of months or more, depending on the inclination of the orbit.

The amplitudes of the different effects can have very different values and ratios as well, depending on the orbital element and perturbing terms.

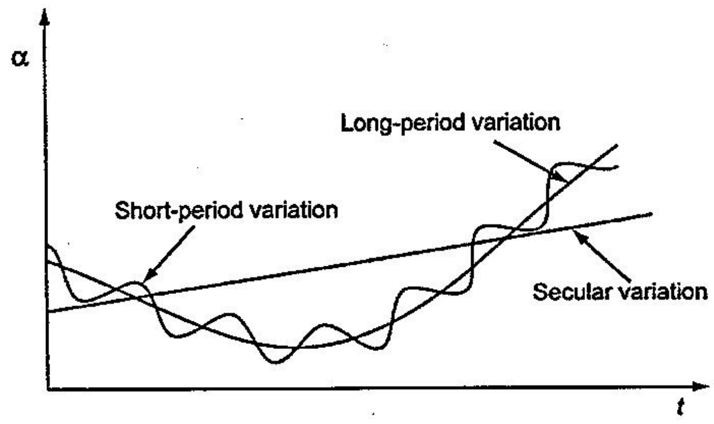


Figure 3.13: Secular, long-period and short-period variations of an orbital element α. (Wakker, 2007)

Chapter 4

Space Surveillance Network Data

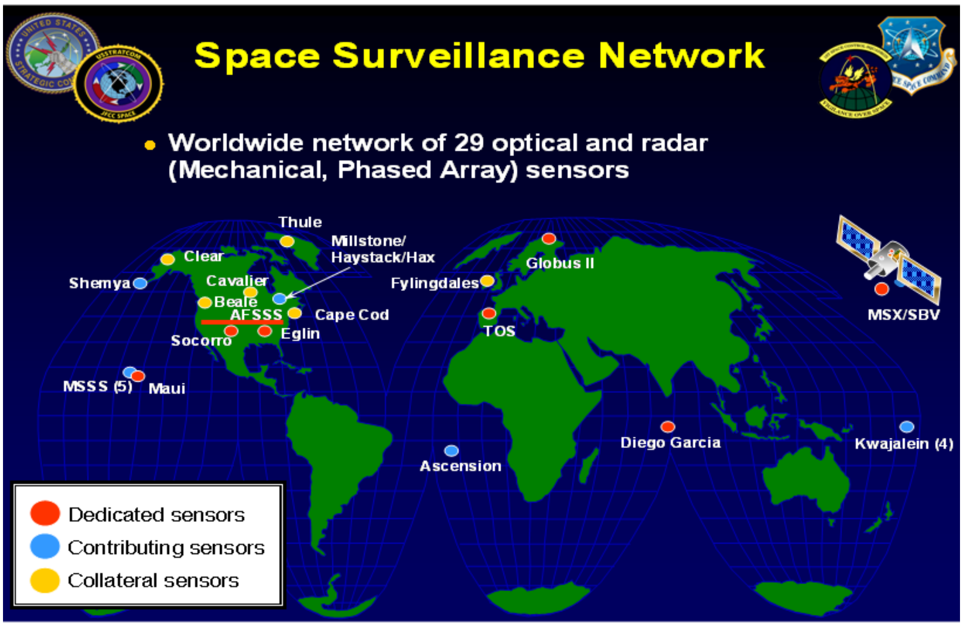
The TLE data determined by the SSN can be used to propagate objects with the SGP4 and SDP4 theory, which both will be treated in this chapter. The TLEs contain ‘mean’ Keplerian elements produced by removing the periodic perturbations in a particular way. These perturbations are then added again by the analytical SGP4/SDP4 theory, to result in the approximated ephemerides.

First an overview will be given on the history of the SSN and how to retrieve publicly available TLE data. Then the TLE data format and its interpretation along with an example will be given, followed by a discussion of the propagation of a TLE, including the history of the SGP4 and SDP4 propagators as well as their current implementation and the revised SGP4 version. The chapter will conclude with a description of the reference frame and the accuracy of TLEs and their propagation.

4.1 History and TLE Retrieval

Since the launch of the very first satellite Sputnik 1 in 1957, the United States Department of Defense (DoD) has been tracking and cataloging objects (satellites and debris) in space by making use of the Space Surveillance Network (SSN), which is currently composed of 29 tracking systems around the world, see Figure 4.1. Also called the SPACETRACK program, it is part of the mission of the United States Strategic Command (USSTRATCOM) to predict where these objects are and when they will decay and re-enter the Earth’s atmosphere (USSTRATCOM, 2008).

This was primarily done to avoid hazards, prevent the triggering of false alarms in the missile-attack warning systems, and to know where satellites of foreign countries are, but is now also done to keep track of all orbital debris, propagate their orbital paths to predict and avoid collisions, and detect



Space Surveillance Network

- Worldwide network of 29 optical and radar (Mechanical, Phased Array) sensors

Figure 4.1: Distribution of the Space Surveillance Network (SSN) tracking systems. (USSTRATCOM, 2008)

new mad-made objects in space.

The North American Aerospace Defense Command (NORAD) has been maintaining a catalog of these trackable objects by making use of the so-called Two-Line Element set (TLE) data format. This data, which can be used to propagate the tracked object's orbit in time by making use of the currently applied Simplified General Perturbations Satellite Orbit Model 4 (SGP4), has been made publicly available through the SPACETRACK website (USSTRATCOM, 2004) since 2004 but can also be retrieved from the CelesTrak website (Kelso, 2010a).

The latter websites offers a program called the TLERetriever which can automatically download pre-selected TLE data, which has already been checked for errors. It does require a valid SPACETRACK account though.

Currently almost 20,000 objects are being tracked of which the updated TLE data of more than 14,000 can be downloaded today by making use of the TLERetriever. A selection can be made as to which satellite data needs to be downloaded in which format, and can then be processed to generate a corresponding text document which lists all available and updated TLEs underneath each other.

For a screen shot of the program and an example of such a text document containing the example which is used in the next section, see Figure 4.2.

New element sets are generated by NORAD on an as-needed basis rather than according to an established timetable. How often these updates occur depends upon a number of factors such as the orbit type or maneuvering capability of the satellite.

For example, a satellite in LEO, such as the US Space Shuttle, would have its element sets updated several times a day because of the somewhat unpredictable results of atmospheric drag as it varies its attitude and the maneuvering being performed. A satellite in a low-drag orbit which does not maneuver, such as LAGEOS II (LAser GEOdynamics Satellite), might only need updates once or twice a week.

Objects such as rocket bodies, defunct payloads, or other orbital debris, will not be updated as frequently either, unless there is a prediction of a close approach with an operational payload. Special-interest objects, such as a large object re-entering the Earth's atmosphere, normally get special treatment.

Normally, all new objects are added to the master list as soon as the object is cataloged by NORAD. After 30 days have elapsed, elements are only maintained if someone requests that they be. Otherwise, they are removed.

At present, the master list contains a broad range of satellite element sets used by a large number of people around the world. It contains elements for various communications, navigation, weather, and other scientific satellites. Over the years, it has grown to include almost 15,000 satellite element sets.

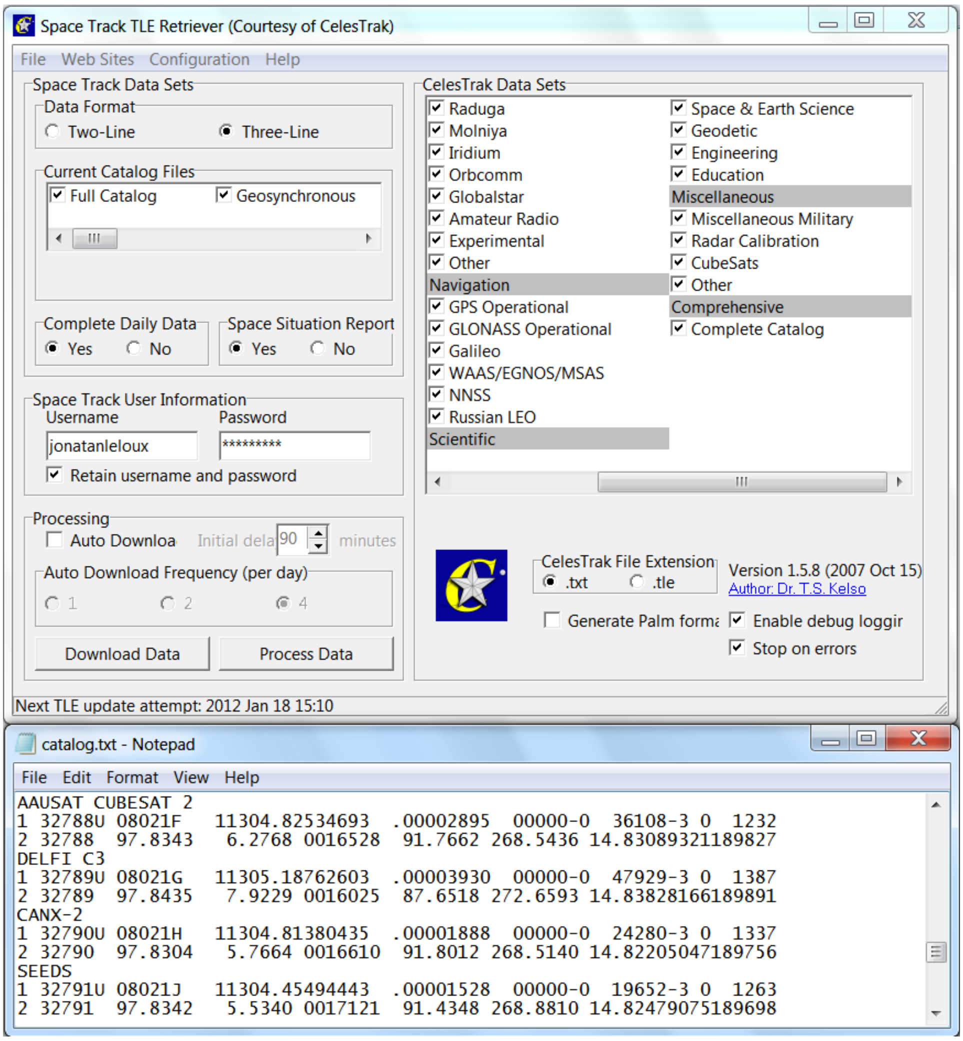
The master list was originally intended as a single source of distribution back when this was originally done via the Celestial Bulletin Board System (BBS) and later the Usenet newsgroups.

However, many users are interested in only a particular category of satellites for their applications, such as amateur radio satellites, and do not want a large list. As such, the TLERetriever sorts out the satellite element sets into separate categorical lists to make it easier to find the elements of interest.

There is also a form on the SPACETRACK and CelesTrak websites to request historical two-line element sets. Archives run from January 1980 to the present and contain tens of millions of TLEs. Those historical element sets which are requested frequently are kept online in the historical archives.

4.2 TLE Data Format

The publicly available TLE data always has the same format. A sheet of the original format was distributed in the SPACETRACK Report No. 3 (Hoots & Roehrich, 1980) and can be seen in Appendix A. The following description of the TLE has largely been obtained and modified from (Kelso,



Space Track TLE Retriever (Courtesy of CelesTrak)
File Web Sites Configuration Help
Download Data
Process Data
Author: Dr. T.S. Kelso
catalog.txt - Notepad
File Edit Format View Help

Space Track Data Sets

Data Format: ○ Two-Line ● Three-Line

Current Catalog Files: ☑ Full Catalog ☑ Geosynchronous

Complete Daily Data: ● Yes ○ No

Space Situation Report: ● Yes ○ No

Space Track User Information

Username: jonatanleloux

Password: *********

☑ Retain username and password

Processing

□ Auto Downloa Initial dela 90 minutes

Auto Download Frequency (per day): ○ 1 ○ 2 ● 4

CelesTrak Data Sets

☑ Raduga
☑ Molniya
☑ Iridium
☑ Orbcomm
☑ Globalstar
☑ Amateur Radio
☑ Experimental
☑ Other

Navigation

☑ GPS Operational
☑ GLONASS Operational
☑ Galileo
☑ WAAS/EGNOS/MSAS
☑ NNSS
☑ Russian LEO

Scientific

☑ Space & Earth Science
☑ Geodetic
☑ Engineering
☑ Education

Miscellaneous

☑ Miscellaneous Military
☑ Radar Calibration
☑ CubeSats
☑ Other

Comprehensive

☑ Complete Catalog

CelesTrak File Extension: ● .txt ○ .tle

Version 1.5.8 (2007 Oct 15)

□ Generate Palm forma ☑ Enable debug loggir

☑ Stop on errors

Next TLE update attempt: 2012 Jan 18 15:10

```
AAUSAT CUBESAT 2
1 32788U 08021F   11304.82534693  .00002895  00000-0  36108-3 0  1232
2 32788  97.8343   6.2768 0016528  91.7662 268.5436 14.83089321189827
DELFI C3
1 32789U 08021G   11305.18762603  .00003930  00000-0  47929-3 0  1387
2 32789  97.8435   7.9229 0016025  87.6518 272.6593 14.83828166189891
CANX-2
1 32790U 08021H   11304.81380435  .00001888  00000-0  24280-3 0  1337
2 32790  97.8304   5.7664 0016610  91.8012 268.5140 14.82205047189756
SEEDS
1 32791U 08021J   11304.45494443  .00001528  00000-0  19652-3 0  1263
2 32791  97.8342   5.5340 0017121  91.4348 268.8810 14.82479075189698
```

Figure 4.2: A screen shot of the TLERetriever and part of a produced text file containing all currently available TLEs.

2010a) and (USSTRATCOM, 2004).

The TLE format is composed of one 24-character and two 69-character lines of data containing the orbital elements of an object, among other data. The only valid characters in a TLE are the numbers 0-9, the capital alphabetical letters A-Z, the period (.), the space ( ), and the plus (+) and minus (-) signs.

However, not all valid characters can be used at any position (column) in the TLE. The following data format shows the allowed characters in a certain column:

```
AAAAAAAAAAAAAAAAAAAAAAAA
1 NNNNNC NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNNN
```

in which columns with a space or period can have no other character. Columns with an 'N' can have any number 0-9 or, in some cases, a space. Columns with an 'A' can have any character A-Z, any number 0-9, or a space. The column with a 'C' can only have a character representing the classification of the TLE, normally either a 'U' for unclassified data or an 'S' for secret data. Of course, only unclassified data are publicly available. Columns with a '+' can have either a plus sign, a minus sign, or a space and columns with a '-' can have either a plus or minus sign (if the rest of the field is not blank).

Further restrictions are placed upon the values in each column as the individual fields of data are defined.

Here follows a TLE example for the recently launched satellite CryoSat-2:

```
CRYOSAT 2
1 36508U 10013A   10130.46204870  .00000003  00000-0  00000+0 0   356
2 36508  92.0231 294.5305 0013120 162.0732 198.1059 14.52323328  4625
```

The following description defines each of the individual fields for lines 0-2. The format of the field number used here is defined as the line number in front of the dot and the field number of that line after the dot. The description also lists the columns that are used by those fields, as well as an example of such a field taken from the CryoSat-2 TLE example stated above.

In appendix A, Table A.1 a quick overview with a definition of the fields, corresponding columns and example is given.

Field 0.1 | Columns 01-24 | Example: CRYOSAT 2 | This field indicates the common name for the object based on the information in NORAD's Satellite Catalog (SatCat). This is usually just the normal name of the satellite or rocket body (indicated with 'R/B'), or the name of the satellite which this fragment of debris (indicated with 'DEB') belonged to.

Field 1.1 | Column 01 | Example: 1 | The first column of line 1 and 2 represents the TLE line number. The value '1' should be present in Field 1.1 to indicate that this is line #1 of the TLE. If this is not the case, the TLE is invalid. It thus also indicates the format of the rest of the line, which can be used in a computer code verifying the TLE.

Field 1.2 | Columns 03-07 | Example: 36508 | Here the Object Identification Number is stated of the object the TLE data is for. This is the Object Catalog Number generated by NORAD, which is a unique identifier for each Earth-orbiting artificial object cataloged.

The most recent TLE holds a number of '36591' here, so it is already running past one-third of its limit. This could be problematic in the near future since smaller and thus more debris can be tracked with passing time. Object numbers less than 10000 are always aligned to the right, and padded with zeros or spaces to the left.

Field 1.3 | Column 08 | Example: U | This character indicates the security classification of the TLE data. All publicly available data will have a 'U' in this field to indicate Unclassified data. Classified (Secret) TLE data would have a 'S' here, but of course, no such data should be available to public users.

Fields 1.4 | Columns 10-11 | Example: 10 | The Fields 1.4 - 1.6 together define the International Designator of the object. These three fields can be left blank, but all must be present if any is. This identifier is an additional unique designation assigned by the World Data Center (WDC) for Satellite Information in accordance with international treaty of 1975: Convention on Registration of Objects Launched into Outer Space. The WDC works together with NORAD and NASA's National Space Science Data Center (NSSDC) in maintaining this registry.

Although there have been some changes in format since it was first used back in the late 1950s, this first field of this International Designator now indicates the year of the launch. It gives the last two digits. One should add 1900 to the value of this field if it is >= 57 and add 2000 for values <= 56, to get the launch year.

This logic has been applied to solve the 'millennium bug', since changing the TLE format to a 4-digit year would require too many software changes around the world, or otherwise lead to wrong results. The reasoning for this logic is that since the first satellite launch was performed in 1957, there will be no 2-digit year values < 57 in TLEs for launches performed in the 1900's. Before the year 2057, a TLE format change will be required though.

Field 1.5 | Columns 12-14 | Example: 013 | This is the second part of the International Designator. Here the launch number of the year indicated in Field 1.4 is stated, the number is incremented with every new launch. A maximum of 999 launches is thus supported by the TLE format. However, the maximum number to date in any one year was 129 launches in 1984.

Field 1.6 | Column 15-17 | Example: A | This is the third and final part of the International Designator. It indicates the piece of the launch for each object connected to that launch, and is 'numbered' alphabetically. Piece 'A' is usually the payload. The letters 'I' and 'O' are not used and thus a total of 13,824 pieces is supported by the format. The largest amount of debris originating from a single satellite is due to the recent Chinese anti-satellite test, the last cataloged piece there has the International Designator 99025DXM. This format thus also holds a reasonable limit. The field can be either right or left justified, but the latter is preferred.

There are some significant differences between NORAD's Catalog Number and the International Designator. NORAD assigns a catalog number based upon when the object was first observed, whereas the International Designator is always tied to the original launch. For example, the 81st launch of 1968 carried four payloads into orbit: OV2-5, ERS 21 and 28, and LES 6. Together with the Titan 3C Transtage rocket body, these objects were assigned International Designators 68081A through E and Catalog Numbers 03428 through 03432. But afterwards, NORAD cataloged additional pieces associated with this launch of which some are still in orbit as Catalog Numbers 25000, 25001 and 30000, among many others, which now have the International Designators 68081G through H. Piece F and others not listed anymore are assumed to have been re-entered. The cataloging of smaller debris, which we may be unable to correlate with the original launch, presents potential problems for the International Designator.

Fields 1.7 | Column 19-20 | Example: 10 | This field and the next define the reference time for the TLE and are together referred to as the epoch. Field 1.7 is the two-digit year, for which the same format reasoning is applied as for Field 1.4.

Field 1.8 | Columns 21-32 | Example: 130.46204870 | This is the second part of the epoch, indicating the day of the year of Field 1.7. The Coordinated Universal Time (UTC) is used here. The epoch defines the time to which all of the time-varying fields in this TLE are referenced.

An epoch of 98001.00000000 corresponds to 0000 (Universal Time) UT on 1998 January 01. In other words: midnight between 1997 December 31 and 1998 January 01. An epoch of 98000.00000000 would correspond to the beginning of 1997 December 31. Note that the epoch day starts at UT midnight (not noon) and that all epochs are measured in mean solar time units rather than sidereal time units.

Fields 1.9 | Column 34-43 | Example: .00000003 | Field 1.9 repre-sents the first derivative of the mean motion divided by two, in units of revolutions per day^2.

Field 1.10 | Columns 45-52 | Example: 00000-0 | This field represents the second derivative of the mean motion divided by six, in units of revolutions per day^3. Together with Field 1.9, a second-order picture of how the mean motion is changing with time is given. However, these two fields are not used by the SGP4/SDP4 orbital models which are currently in use, but only by the previously used simpler SGP model. Therefore, they serve no real purpose anymore. The different SGP/SDP orbit propagation models are explained later.

Field 1.10 has a somewhat different format than the other fields. In particular, a modified exponential notation is used with an implied leading decimal point. This convention is inherited from the Philco 2000 - Model 212 and the TAC assembly language used on the original 496L system, where all such numbers range from 0 to less than 1. The first six columns of the field represent the mantissa and the last two represent the exponent. For example, the value -12345-6 corresponds to $-0.12345{\cdot}10^{-6}$. Each of these two fields can be blank, corresponding to a value of zero.

Field 1.11 | Columns 54-61 | Example: 00000+0 | This field repre-sents a parameter called B* (BSTAR), which is an SGP4-type drag coefficient. In aerodynamic theory, every object has a ballistic coefficient, B, that is the product of its coefficient of drag, C_D, and its cross-sectional area, A, divided by its mass, m:

$$B = C_D A/m \tag{4.1}$$

The ballistic coefficient represents how susceptible an object is to drag; the higher the number, the more susceptible. $B*$ is an adjusted value of B using the reference value of atmospheric density, ρ_0.

$$B* = B\rho_0/2 \tag{4.2}$$

$B*$ has units of inverted Earth radii, R_E^{-1}.

The format used is the same as for Field 1.10, which is the modified exponential notation with an implied leading decimal point.

Field 1.12 | Column 63 | Example: 0 | Here the ephemeris type (i.e. orbital model) used to generate the data is indicated. Spacetrack Report No. 3 suggests the following assignments: 1=SGP, 2=SGP4, 3=SDP4, 4=SGP8, 5=SDP8. However, this value is used for internal analysis only. All distributed TLEs have a value of zero and are generated using the SGP4/SDP4 orbital model, as appropriate.

Field 1.13 | Column 65-68 | Example: 35 | This field represents the TLE number. Normally, this number is incremented each time a new TLE is generated. In practice, however, this does not always happen. When operations switch between the primary and backup Space Control Centers, sometimes the TLE numbers get out of sync, with some numbers being reused and others skipped. Unfortunately, this makes it difficult to tell if all the TLEs for a particular object are present.

Field 1.14 | Column 69 | Example: 6 | The last column on the line represents a modulo-10 checksum of the data on that line. To calculate the checksum, simply add the values of all the numbers on each line. Ignore all letters, spaces, periods, and plus signs. And assign a value of 1 to all minus signs. The checksum is the last digit of that sum. Although this is a very simple error-checking procedure, it should catch 90 percent of all errors. However, many errors can still sneak through. To eliminate these, all data posted on (Kelso, 2010a) not only pass the checksum test, but must also pass both format and range-checking tests.

Field 2.1 | Column 01 | Example: 2 | The value '2' in column 1 should be present here to indicate that this is line #2 of the TLE. If this is not the case the TLE is invalid.

Field 2.2 | Columns 03-07 | Example: 36508 | This field should be the same as Field 1.2. If this is not the case, the TLE is invalid.

Field 2.3 | Columns 09-16 | Example: 92.0231 | Fields 2.3 - 2.8 are the mean elements calculated using the SGP4/SDP4 orbital model, which are explained later. The current field represents the inclination, and can range from 0 to 180 degrees.

Field 2.4 | Columns 18-25 | Example: 294.5305 | The right Ascension of Ascending Node is indicated here, and can range from 0 up to 360 degrees.

Field 2.5 | Columns 27-33 | Example: 0013120 | Here the eccentricity is stated. It is a unitless value with an assumed leading decimal point. For example, a value of 1234567 corresponds to an eccentricity of 0.1234567.

Field 2.6 | Columns 35-42 | Example: 162.0732 | This is the Argument of Perigee, which can range from 0 to 360 degrees.

Field 2.7 | Columns 44-51 | Example: 198.1059 | In this field the Mean Anomaly is listed, and can also range from 0 to 360 degrees.

Field 2.8 | Columns 53-63 | Example: 14.52323328 | This field indicates the Mean Motion, and is measured in revolutions per day.

Field 2.9 | Columns 64-68 | Example: 462 | This is the revolution number. Since there are several conventions for determining revolution numbers, this field also bears some clarification. In NORAD's convention, a revolution begins when the satellite is at the ascending node of its orbit and a revolution is the period between successive ascending nodes. The period from launch to the first ascending node is considered to be revolution 0, and revolution 1 begins when the first ascending node is reached.

Field 2.10 | Column 69 | Example: 5 | The last column here is the modulo-10 checksum of Line 2. For more information see the explanation for Field 1.14, as it is calculated in the same way.

For all fields, to any number smaller than the maximum field size leading spaces or leading zeros can be added. For instance, an epoch can be represented as either 98001.12345678 or 98 1.12345678 or an inclination can be represented as 28.1234 or 028.1234.

Mostly, leading zeros are used for fields 1.5 and 1.8 and leading spaces for all other fields, but both are valid.

4.3 TLE Orbit Propagation

Here follows a general discussion on orbital propagation, and the TLE SGP4/SDP4 implementation thereof, including its history and the latest revised version.

4.3.1 Orbital Propagation

To propagate the orbit of an object means that one predicts (calculates) where the object is going to be in the future, by making use of the starting conditions (state) of that object at a particular time. The state of that object can for instance be described by its three-dimensional position and velocity, or by its six independent orbital elements, which can be written in a state vector.

If one now knows all the forces that act upon the object, the state vector can be determined at a point forward in time, according to Newton's laws. The main force will be the gravitational attraction of the Earth if the object is near the Earth, which is assumed to be spherical and homogeneous here. Next to this main force, perturbing forces are present, such as gravitational perturbations due to the fact that the Earth is not spherical and homogeneous, the gravitational attraction of the Sun, Moon, or even other planets, a drag force due to Earth's atmosphere, or the force due to radiation pressure of the Sun.

These perturbations can be secular, long-periodic or short-periodic. In Figure 3.13 the variations on an orbital element α due to these different perturbations were shown.

Secular means that the perturbing effect is constant in time, which results in a linear behaviour of one or more orbital elements. Long-periodic perturbations can be averaged out over longer periods, but this will result in errors at a certain instant, since the amplitude can be rather large. Finally, short-periodic effects can also result in errors when averaged out.

If only the main gravitational attraction of the assumed spherical homogeneous Earth would be taken into account, the object would orbit the Earth according to Kepler's laws. That is to say, the semi-major axis, eccentricity, inclination and argument of pericentre would all remain constant. For increasing precision however one should include subsequently secular, long-periodic and short-periodic perturbations too.

The time step taken in the above calculation of a point forward in time should be small however, if one wants to obtain a high enough accuracy. An object in orbit about the Earth has such a high speed that with a time step of 1 second, it will move over a distance of almost 8 km!

Thus, if one wants to compute the state vector of the object a large time step ahead and still maintain a high enough accuracy, the larger time step should be divided into a large amount of small time steps, and the state vector should be calculated at each successive small time step ahead. Propagating the orbit in this way involves a large amount of calculations, and the technique described here uses numerical integration.

Orbital propagation can however also be done analytically. In this way, one can calculate the state vector of an object a large time step ahead by just inserting the future time value into the analytical equations.

However, such a model of analytical equations has a certain accuracy for a corresponding domain. If one wants to obtain a higher accuracy, a more complex analytical model should be used, thus also increasing the computational effort. There is thus always a trade-off between the accuracy of the resulting state vector and the time that is needed for a computer to calculate it. The previous discussion on adding different types of perturbations for higher accuracy also holds true here.

Since applying the numerical integration technique to one object already requires a large computational effort, using it for the entire satellite catalog, while still maintaining high enough accuracy, is simply not possible. At least, in the past it was not.

This is why in the 1950's and 1960's analytical methods were developed, implemented and improved, which did not require too much computational effort while still resulting in a reasonable accuracy. These models developed during the starting period of space flight are more or less still in use today, and their history is described in (Hoots et al., 2004), and will be briefly described in the next subsection.

4.3.2 History of Analytical Orbital Propagation Models used in the SSN

The first few models that were implemented at the U.S. National Space Surveillance Control Center (NSSCC) used satellite observations from 150 different sites using different measuring techniques and were sent to the NSSCC by a variety of communication types. The NSSCC then updated the orbital data of the objects, calculated new ephemeris after which three products were generated that were sent back to the sites for future observation.

The technical details of this ephemeris model can be read in (Hoots et al., 2004), and it was first reported by Wahl in 1959. In the following discussion, historic references that are used can be viewed in Hoots et al. (2004) as well.

At the same time, the U.S. Navy developed a system which almost automatically detected and cataloged satellites using the Naval Space Surveillance System (NAVSPASUR), also known as the Fence, which became operational in 1960. The Fence is a continuous-wave multistatic radar interferometer composed of three transmitters and six receivers along a great-circle arc of the Earth.

Almost all orbits of satellites near the Earth which are in its field of view could be determined automatically without using any information beforehand. The first computers used to process the data took 15 minutes for one satellite's orbit, but this was already reduced to 1 minute in 1961 by making use of new programming techniques and a new computer.

One of the first implemented theoretical solutions for the motion of a near-Earth satellite was developed by Brouwer under project SPACETRACK in

1959 which included the zonal harmonics $J_2 - J_5$. At the same time, Kozai published a similar model. These two methods still serve as the basis of the models currently in use by the SSN. In 1963 the singularity of eccentricity and inclination in the model of Brouwer was removed by Lyddane using a change of variables.

The latter analytical satellite prediction model was implemented by NAVSPASUR and is known as Position and Partials as functions of Time (PPT), and included all long-periodic effects. It was changed and implemented in 1964 by Richard H. Smith using ideas from King-Hele with some theory from Kozai, resulting in a model that included atmospheric drag and was now numerically closer to the original model of Kozai than that of Brouwer. Currently the upgraded model PPT3 is in use.

Also in 1964, the Simplified General Perturbations (SGP) model became the primary orbital prediction model at the Space Detection and Tracking System (SPADATS) Center, which was the relocated follow-up of the NSSCC. This SGP model was also based on the Brouwer and Kozai solutions, and transformed by Arsenault et al. into parameters without singularity.

Only long- and short-period terms in position without eccentricity as a factor were included from the Brouwer model. Atmospheric drag was included in almost the same way as Smith implemented it. This SGP model was now also used at most of the tracking systems in the world.

Then, for the first time in 1965, the model of Brouwer was upgraded with an efficient and accurate solution including atmospheric effects using an analytical density model rather than the empirical (i.e. making use of tabular experimental data) one in use. This development was done by Lane, and it subsequently was further improved by Lane and Cranford in 1969.

However, the size of the catalog in 1969 had grown too big for this model to be handled by the computers used at that time. So, a simplified version including only the secular effects of drag was made and integrated in 1970, named SGP4. In 1979, Lane and Hoots documented the SGP4 derivation of Lane and Cranford. In that same year SGP4 became the only model used for satellite catalog updating.

In 1977 Hujsak integrated parts of a solution including perturbations by the Sun and the Moon as well a geopotential resonance effects, developed by Bowman 1967, into the SGP4 model. The resulting SDP4 model is used for so-called deep-space satellites, which here means orbits about the Earth with a period larger than 225 minutes, see next subsection. The SGP4 and SDP4 models are still in use today in the SSN.

The SPACETRACK Report No. 3 (Hoots & Roehrich, 1980) lists the formulas used by the SGP, SGP4, SDP4, SGP8 and SDP8 models, as well as their implementation in FORTRAN code, and some test cases.

The SGP8 model was developed in 2008 by Hoots and is comparable to the SGP4 models since it is based on the same theory of Lane and Cranford for atmospheric drag and gravitational potential. However, the differential equations are integrated in a different manner, resulting in a different form of the solution.

The SDP8 model is again the deep-space variant of the SGP8 version, and uses the same added perturbing effects that were used in the SDP4 version. The SGP8/SDP8 models are more accurate for high drag conditions, and are believed (by the public scientific community) to be used when an object is re-entering.

However, it is not known by the public scientific community when and if the SGP8/SDP8 models are used and the re-entering phase is of short duration.

4.3.3 The SGP4/SDP4 Model

It is believed by the public scientific community that the SSN still more or less uses the SGP4/SDP4 models to generate today's TLEs, and only these models should be used to propagate the TLEs corresponding orbits in time to produce accurate results, as explained hereafter.

An object's orbit is measured by the SSN including periodic variations. Then a TLE is generated from this measured orbit with the SGP4 or SDP4 model, which removes these periodic variations resulting in an element set with mean values.

So, in order to reconstruct the original orbit with periodic variations, one has to reconstruct this orbit with exactly the same models that were used in generating it! So, even if one uses other more accurate propagating methods on TLEs, or even a numerical integrator, this would still result in less accurate results.

The end result of the analytical propagation to a certain time in the future or past of the epoch of the TLE is a three-dimensional position (x,y,z) and velocity $(\dot{x}, \dot{y}, \dot{z})$ in the Earth-Centred Inertial (ECI) reference frame which is True Equator and Mean Equinox (TEME). This reference frame is explained in the next subsection.

The SGP4 model is used for objects in near-Earth orbit with a period of less than 225 minutes. The SDP4 model is used for objects in orbit about the Earth with a period of 225 minutes or larger. Its model is the same as the SGP4 model, but with a part added for the deep-space perturbations and resonances.

In (Hoots & Roehrich, 1980) the deep-space portion was unfortunately only given in FORTRAN code without derivation due to its size.

The entire set of equations used in these models with some explanations of the variables and included perturbations can be seen in Appendix B, which was copied from the Appendices of (Hoots et al., 2004), in which the deep-space model was elaborated upon for the first time.

For the SGP4 model one should use Appendix B in the copy, ignoring the subsections for the deep-space perturbations. For the SDP4 model one should use Appendix B in the copy, and the subsections for the deep-space perturbations are then described in Appendix A of the copy, while section A.F can be ignored, which is only used in the PPT3 model.

Summarising, the SGP4 model includes Secular Effects of Atmospheric Drag and Earth Zonal Harmonics, Long-Period Periodic Effects of Earth Gravity and Short-Period Periodic Effects of Earth Gravity. While the SDP4 model also includes Secular and Long-Period Periodic Effects of Lunar and Solar Gravity and Resonance Effects of Earth Gravity.

During the beginning of the 1990's, the 1990 standalone SGP4 code from project SpaceTrack was sent to the NASA Goddard Space Flight Center (GSFC) to study on orbit propagation models for the SeaWiFS (Sea-viewing WIde Field-of-view Sensor) Mission (Vallado et al., 2006). The unrestricted code, here called the 'GSFC version', was then released publicly on the internet and to organizations around the world involved in the SeaWiFS mission during 1996-1997, and confirmed changes which were already discovered by a large amount of researchers who were working independently.

4.3.4 Revised SGP4 Version

After the SGP4/SDP4 models used by NORAD were published to the public for the first time in the SPACETRACK Report NO.3 in combination with the first public availability of TLEs on the Internet, many users (companies, institutions, experts, amateurs and others) could now accurately propagate the orbit of the objects in the catalog themselves. However, these end users soon found errors with rare cases and corrected the implementation of the code, and made it more efficient, which resulted in many different versions in use by the public.

Since changes to the model used by DoD are still not published publicly by the DoD itself as of today, the code used by the public to propagate the TLEs is different from the code used to generate the TLEs.

To this end, a paper was published (Vallado et al., 2006) for which all of these changes were researched and verified. This research resulted in a new publicly available SGP4 code (with SDP4 integrated) which is designed to be highly compatible with the recent code in use by the DoD.

Issues with the original SGP4 version pointed out in (Vallado et al., 2006) were related to the data format, coordinate system, time system, and Greenwich Hour Angle (GHA). A few of the major implemented changes were the restructuring and merging of the SDP4/SGP4 code as well as some programming and mathematical changes, to increase the efficiency, accuracy and clearness of the code.

Furthermore, the theory for the lunar and solar perturbations and the iterating solution to Kepler's equation were updated for better accuracy and to solve certain jumps and errors in resulting graphs.

While the previously mentioned changes were verified to be implemented in the current code in use by the DoD, others were assumed to be implemented, as they seemed obvious. Assumed changes include updates regarding error checking, physical constants, negative inclination orbits, integrator problems and additional Kepler's equation robustness. Many of the found and solved errors only apply to exceptional cases.

Propagated ephemerides differences between the revised C++ version of Vallado et al. (2006) and the GSFC version can be seen in Figure 4.3. The top plot shows significant differences which are due to a bug in the GFSC version. This bug was related to secular integrator problems for geosynchronous and semi-synchronous orbits from 720 minutes prior to epoch backwards, resulting from resonance with the orbital period.

When only propagations forward in time are considered, see bottom plot, the differences are negligible. This verifies that the revised version is very close to the GSFC version, and thus the proper changes were made, while new found errors were also fixed like for instance this resonance bug.

Comparing the revised C++ version to the original SpaceTrack Report no. 3 (STR#3) version of the code, see the top plot of Figure 4.4, rather large differences can be seen for the entire catalog. But when changing the format of all the variables of the original STR#3 code to 'double precision', which was one of the coding changes to the new revised version, significant reduction of the differences is already obtained for a large amount of objects, see the bottom plot of the figure.

Many operations have already switched to numerical processing for higher accuracy, however, the analytical method still has benefits in certain applications, and certainly when a large number of satellites has to be processed like with the catalog.

These applications include rapid assessment of conjunction events, satellite visibility for ground stations and generation of communication schedules. Other applications are the tracking of antennas with limited CPU (Central Processing Unit, computer processor) power, or initial orbit design with

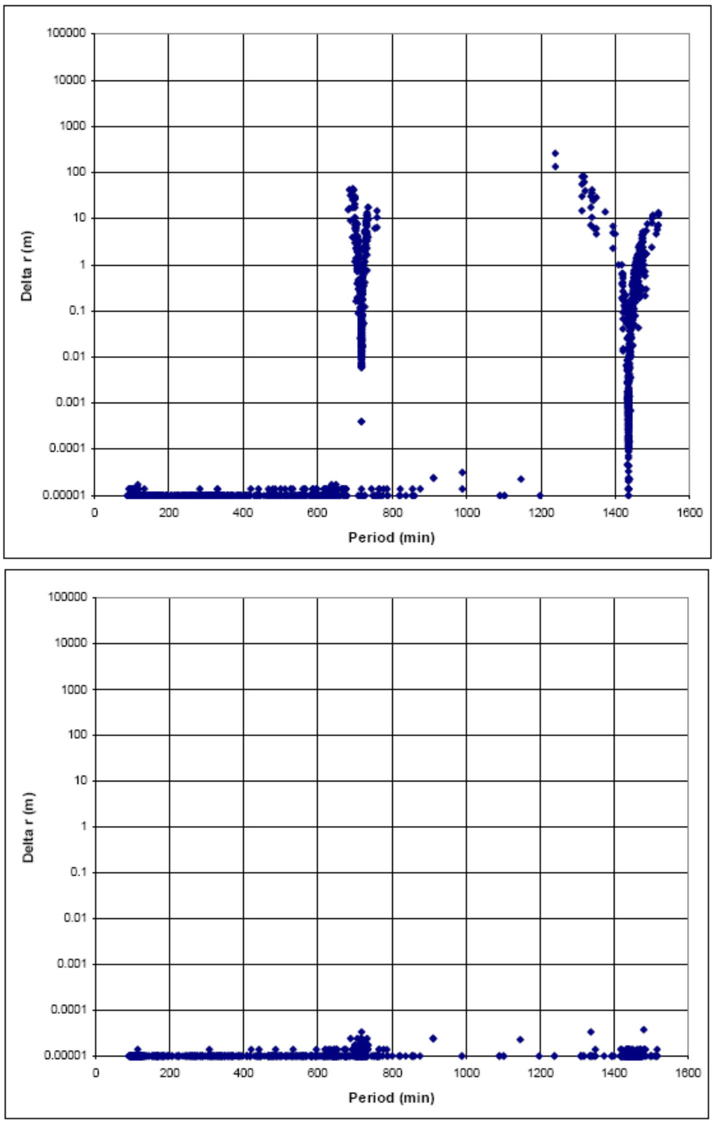


Figure 4.3: For ephemerides generated for the entire catalog for 2 days negative and positive from epoch, the maximum difference between the revised C++ version and the GSFC version of the SGP4 code is shown in the top plot. The bottom plot shows the same comparison but only for propagations positive from epoch. (Vallado et al., 2006)

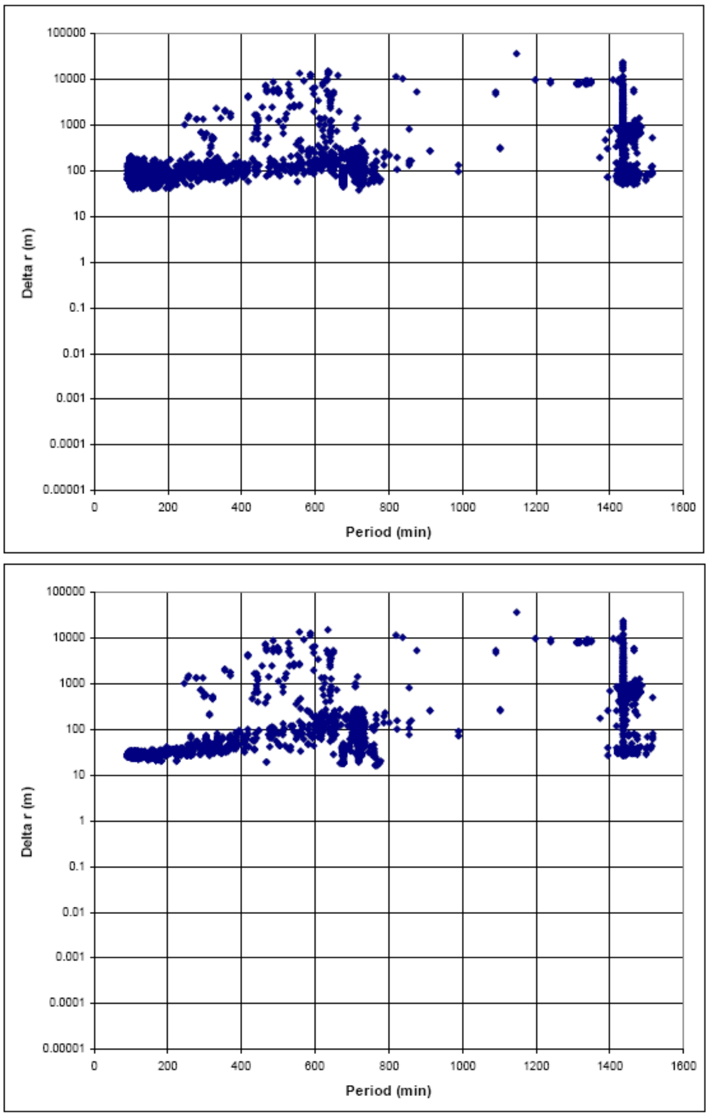


Figure 4.4: For ephemerides generated for the entire catalog for 2 days negative and positive from epoch, the maximum difference between the revised C++ version and the original STR#3 version of the SGP4 code is shown in the top plot. The bottom plot shows the same comparison but with all variables in the STR#3 version using the 'double precision' coding format. (Vallado et al., 2006)

low-precision requirements.

4.4 Reference Frame of Propagated TLE Data

The TLE ephemeris, when propagated with the SGP4 model, result in state vectors that are defined in the ECI reference frame which is True Equator, Mean Equinox (TEME) of epoch. The origin of the coordinate system is centred at the centre of mass of the Earth.

Here, inertial means that the frame is not co-rotating with the Earth but 'fixed' in space with respect to the distant stars, which are relatively fixed. If an Earth-fixed, and thus rotating, frame were chosen, the equations of motion for an object in orbit about the Earth would be much more complex.

The ECI frame is not truly inertial in this way since it is for instance still translating with the Earth in orbit about the Sun. However, in a two-body problem like a satellite orbiting the Earth this does not produce errors. Only when a third body is introduced, like the perturbing effects of the Moon or Sun, the accelerations of the ECI frame should be taken into account as well (Wikipedia, 2010a).

The ECI frame is defined as a Cartesian coordinate system, which means all of 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's rotational axis pointing North, and the Y-axis completes the orthogonal system.

The Vernal Equinox is an imaginary point is space defined as follows. If the intersectional line of the Earth's equatorial plane and the plane of the Earth's orbit about the Sun (the ecliptic) is taken, the Vernal equinox lies in the direction of that line pointing to the Sun around March 21st. A schematic of the definition of the ECI coordinate system can be seen in Figure 4.5.

The Z-axis is thus aligned with the true instantaneous North pole. However, due to gravitational perturbations the Earth's rotational axis changes direction, and thus the Earth's equatorial plane changes as well.

The two types of motion of the Earth's rotational axis are called precession and nutation, see Figure 4.6. Here, the rotational axis is indicated with the green R, precession is the large rotational periodic motion indicated by the blue P, and nutation, indicated by the red N, is the smaller periodic motion on top of the precession.

In the ECI-TEME frame, precession is accounted for, but not nutation (Kelso, 2010a). The angle between the Earth's rotational axis and the ecliptic, which is called the obliquity, is thus changing with time. This does not affect the

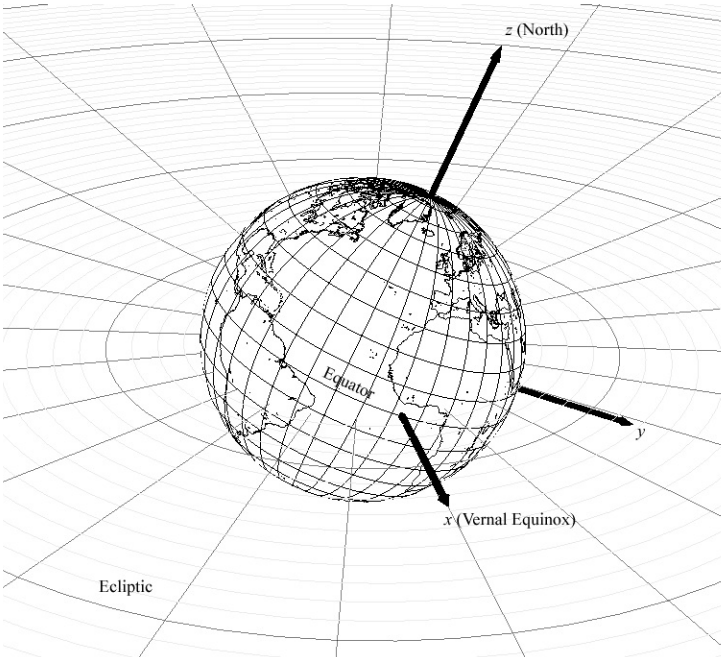


Figure 4.5: The Earth-Centered Inertial (ECI) Coordinate System. Please note that the y-axis does not coincide with the ecliptic, while the x-axis does. (Kelso, 2010a)

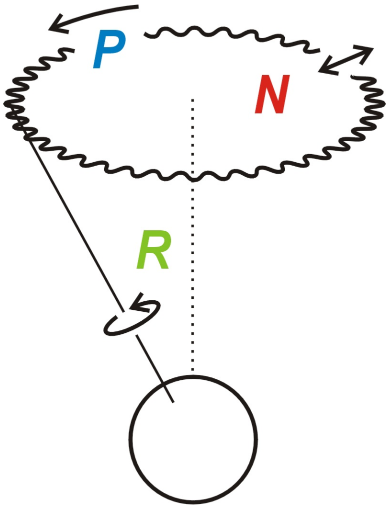


Figure 4.6: The two types of motion of the Earth's rotational axis (green R), called precession (blue P) and nutation (red N). (Wikipedia, 2010b)

orthogonality of the ECI frame though, since it is defined with respect to the Earth itself.

The Earth's orbital plane also changes over time. Thus, the vernal equinox, the equatorial plane of the Earth, and the ecliptic plane also vary with time and can be determined for a particular epoch.

When the short-term periodic oscillations, like nutation, are averaged out, they are called 'mean' as opposed to 'true'.

The choice for this ECI-TEME frame is logical, since the observations of the SSN made with sensors fixed to the Earth's surface are referenced to the Earth's true equator, while the position of the Earth in inertial space, relative to the vernal equinox, is not exact.

4.5 Accuracy of TLE Data and Propagation

The accuracy of a particular TLE depends on many variables. Each tracking sensor has its own measuring technique and accuracy.

The range to the object plays a part, as well as the amount of data (variables) collected on a particular object, the type of orbit and its space environment condition. The amount of data collected on an object and its accuracy are in turn dependent on the objects' size, shape, reflectivity and contact duration, among other things.

NORAD has tried to work with methods incorporating the accuracy into the TLEs, but they did not succeed, simply because there are too many variables per TLE.

4.5.1 TLE Consistency

The consistency of TLEs can be checked easily however. If the difference between two states resulting from TLEs with successive reference epochs is taken, after one has been propagated towards the epoch of the other TLE, the magnitude of this difference can be taken as a measure of its consistency.

If this is done for an object with a large number of TLEs, the general accuracy of the data can be obtained, when it is assumed that the data was generated statistically unbiased, and the orbit determination and TLE generation were error-free.

With this method successive TLEs might be correlated due to theory errors though. Using multiple TLEs of a certain object, the validity of a certain TLE with respect to time can also be deduced.

In (Legendre et al., 2006) the TLE consistency accuracy determination approach was used to set up a function based on a mixture of gaussian laws to be able to predict TLE errors, with good results. Following this study, (Legendre et al., 2008) made this function also dependent on propagation time with encouraging results, but further study needs to be done to improve and verify these results.

4.5.2 Accuracy Determination Methods

There are some other ways to test the accuracy of TLEs though.

Comparison with tracking data

One is to directly compare the propagated TLE data of operational payloads with the orbital data from the operator or owner of these satellites. However, this method can only be used for a limited number of satellites and can thus not cover all orbit regions.

Operational satellites are routinely tracked, and often with higher accuracy than the results of their TLEs. Examples of high-precision publicly available orbital data are the GPS and GLONASS (GLObal'naya NAvigatsionnaya Sputnikovaya Sistema, which is Russian for: GLObal NAvigation Satellite System) constellations, as well as the Intelsat and Meteosat satellites, and geodetic satellites.

This data can also be converted into TLEs, and if the same model is used to propagate them, the results will be of a much higher accuracy than normal TLEs. These high-precision TLEs can also be obtained from (Kelso, 2010a), and were also compared with GPS and GLONASS precise ephemerides accurate to the centimetre level.

For the GPS case, the NORAD TLEs showed an average error of 7.54 km with a maximum of 32.45 km, while the high-precision TLEs gave an average error of 0.87 km with a maximum of 2.37 km, all in any one direction. For the GLONASS case, the NORAD TLEs showed an average error of 3.30 km with a maximum of 9.39 km, while the high-precision TLEs gave an average error of 0.20 km with a maximum of 0.54 km.

Comparisons of TLE data with high-precision orbital data of operational satellites have been made for Intelsat as well as GPS (Kelso, 2010a) satellites. Orbital elements for all geosynchronous INTELSAT satellites derived from a private dual-ranging system, with a maximum position accuracy of 250 m when propagated over 14 days, were compared with TLE data propagated between TLE updates for a one-year time span (Chan & Navarro, 2001).

The Root Mean Square (RMS) of the differences ranged from 3-10 km for radial, 25-60 km for tangential and 5-15 km for normal direction. This resulted in RMS position and velocity differences of 30-60 km and 2-4 m/s respectively.

Furthermore, no systematic temporal variations were discovered over the one-year time span, indicating consistent accuracy of the TLE data. Systematic biases were discovered which were related to the geopotential acceleration indicating differences in the modelling of the terrestrial geopotential harmonics.

In (Kelso, 2007), publicly available post-processed high-precision ephemerides, with an accuracy of less than 25 cm, were used for a comparison with propagated TLE data for 22 GPS satellites.

In-track errors were dominant, with a maximum difference of 10 km at TLE epoch, going up to 50 km when propagated over 15 days forward or backward. The radial errors had a maximum one-sigma variation of 2-4 km for 0-15 days of propagation respectively. The cross-track errors showed a maximum of 2 km one-sigma variation both at TLE epoch and after 15 days propagation.

Furthermore, biases found in the TLE errors could lead to overestimation of the propagated error. These biases could be estimated and removed to increase TLE accuracy. Also, TLE errors for different satellites with similar orbits can be significantly different.

Multiple TLEs for one GPS satellite and with a maximum epoch difference of 15 days were used for a consistency analysis described above, and gave accuracy results with a good match to the previous comparison with respect to propagation time, direction and magnitude. The downside is that the error at TLE epoch can not be determined with this analysis and is thus set to zero.

Pseudo-Observations

Another way to evaluate the accuracy of TLEs currently in use by the European Space Agency (ESA) (Flohrer et al., 2008) is to compare the states derived directly from the TLE data to states resulting from orbit determination based on pseudo-observations. These pseudo-observations are derived from the TLEs by simulating inverted tracking station models.

Then an orbit determination process takes place and subsequently numerically propagation yields the object states. The differences (residuals) between the states generated directly from the TLEs and the states generated with the latter numerical propagation can be used to determine standard deviations in radial (U), along-track (V) and out-of-plane (W) direction. These deviations can then be used to generate covariance look-up tables for orbital debris conjunction analysis. This method can be used for the entire TLE catalog.

A schematic of the process of this TLE accuracy assessment in use by ESA can be seen in Figure 4.7. The orbit determination and propagation block

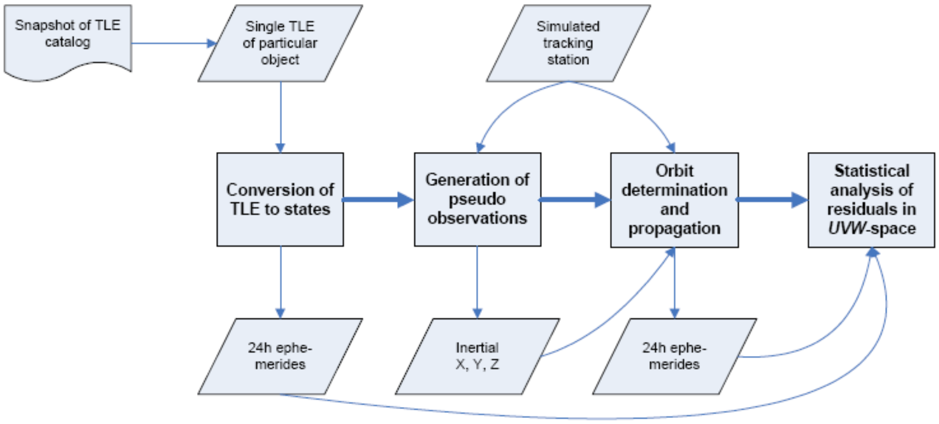

Snapshot of TLE catalog
Single TLE of particular object
Simulated tracking station
Conversion of TLE to states
Generation of pseudo observations
Orbit determination and propagation
Statistical analysis of residuals in UVW-space
24h ephe-merides
Inertial X, Y, Z
24h ephe-merides

Figure 4.7: A schematic of the process used by ESA for TLE accuracy assessment. (Flohrer et al., 2008)

is done by ESA's Orbit Determination with Improved Normal Equations (ODIN) software. The covariance tables resulting from this analysis are used in ESA's Collision Risk Assessment Software (CRASS).

The precision of the orbit determination and propagation from the pseudo-observations can be checked by comparison with high-precision orbital data of certain operational satellites, obtained from combined radar-, laser- and Doppler-tracking. By comparing the ODIN software with high-precision orbital data, ESA determined that the uncertainties of TLEs could be assessed down to the order of about 20 m, if 24 hour arcs around the TLE epoch are used.

The estimated uncertainties of all objects in the TLE catalog of January 1st 2008 can be seen in Figure 4.8, in the form of standard deviations σ for the U, V, and W directions. The figure can be subdivided into several orbit classes from LEO to GEO which are defined in Table 4.1, and for which the results can be seen in Figure 4.9.

The average uncertainties per orbital regime can be seen in Table 4.2. A look-up table of averaged results from this analysis as a function of eccentricity, inclination, and perigee altitude can be seen in Table 4.3. The ESA results are displayed here since they show similar results to the other studies and are the only full catalog results.

In a similar study (Dong & Chang-yin, 2010) on a part of the catalog, TLE errors for LEO objects ($h < 5000$ km) were estimated to range from 0.3-0.8 km, for MEO ($10,000 < h < 25.000$) from 0.5-1.5 km, for GEO from 1.5-2.5 km, and for HEO ($0.1 < e < 0.8$) from 1-8 km increasing with eccentricity. When propagating the LEO, MEO, GEO and HEO objects for respectively

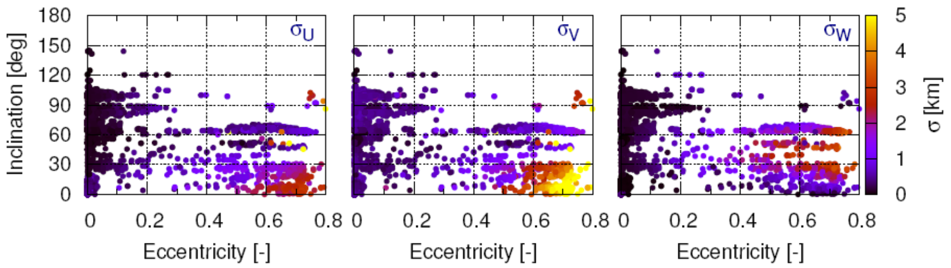
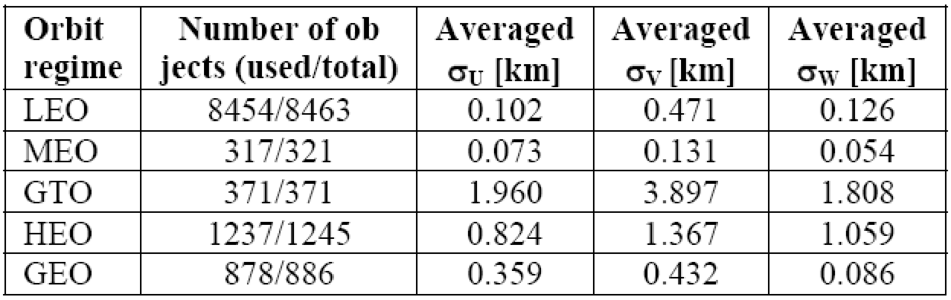


Figure 4.8: Estimated 1-σ-uncertainties (color coded) of all objects in the TLE catalog of January 1st 2008 in U, V and W direction as a function of eccentricity and inclination. (Flohrer et al., 2008)

Table 4.1: Orbit class definition in terms of perigee altitude h_p and apogee altitude h_a. (Flohrer et al., 2008)

Acronym	Description	Definition
LEO	Low-Earth Orbit	$h_a < 2000 + R_\oplus$
MEO	Medium-Earth Orbit	$h_p > 2000 + R_\oplus$ & $h_a < 40164$
GEO	Geostationary Orbit	$h_p > 40164$ & $h_a < 44164$
GTO	Geostationary Transfer Orbit	$h_p < 2000 + R_\oplus$ & $h_a > 40164$
HEO	Highly Elliptical Orbit	All other orbits

Table 4.2: Average 1-σ-uncertainties in U, V and W direction per orbital regime for the entire TLE catalog of January 1st 2008. (Flohrer et al., 2008)



Orbit regime	Number of ob jects (used/total)	Averaged σ_U [km]	Averaged σ_V [km]	Averaged σ_W [km]
LEO	8454/8463	0.102	0.471	0.126
MEO	317/321	0.073	0.131	0.054
GTO	371/371	1.960	3.897	1.808
HEO	1237/1245	0.824	1.367	1.059
GEO	878/886	0.359	0.432	0.086

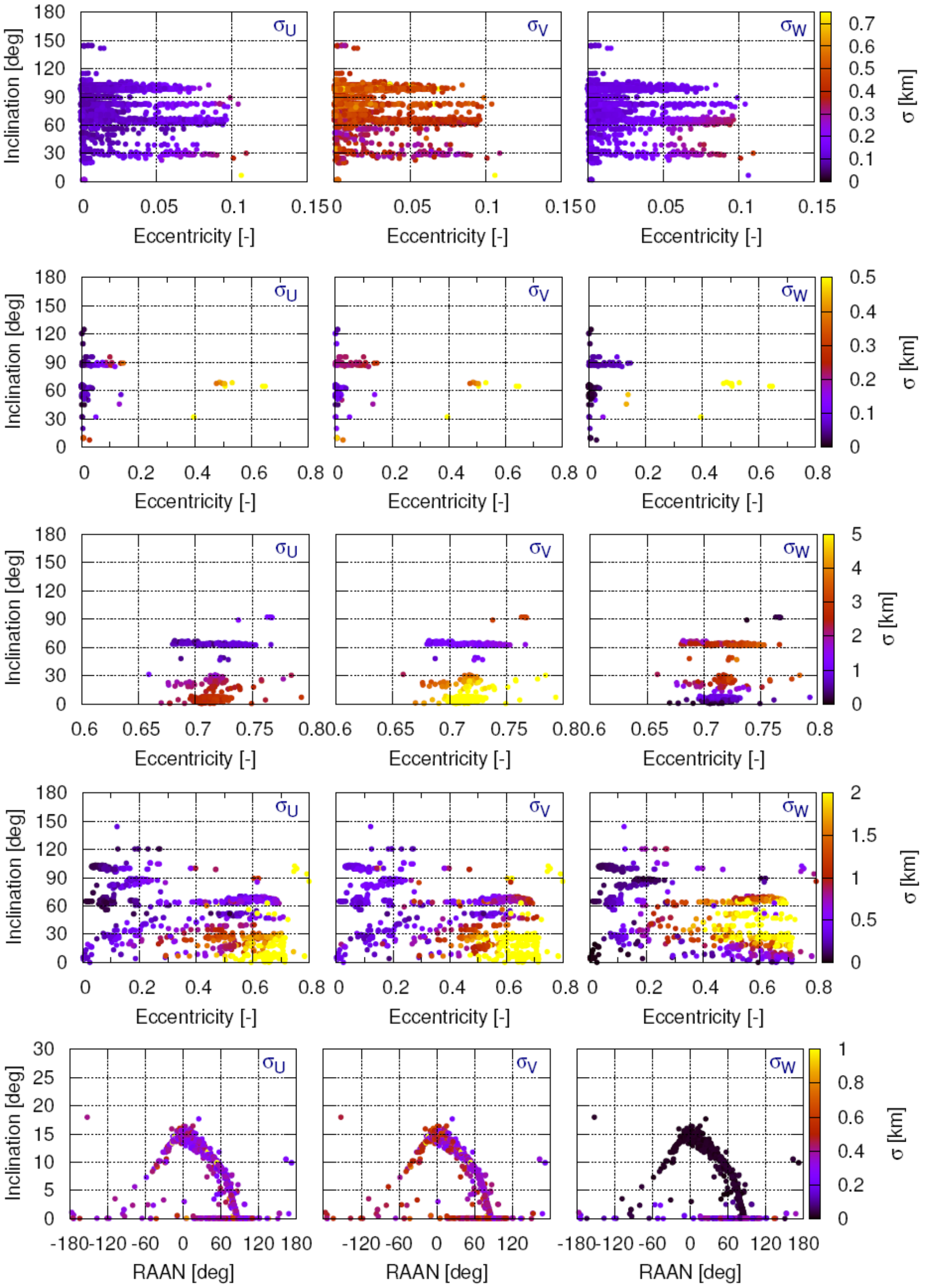
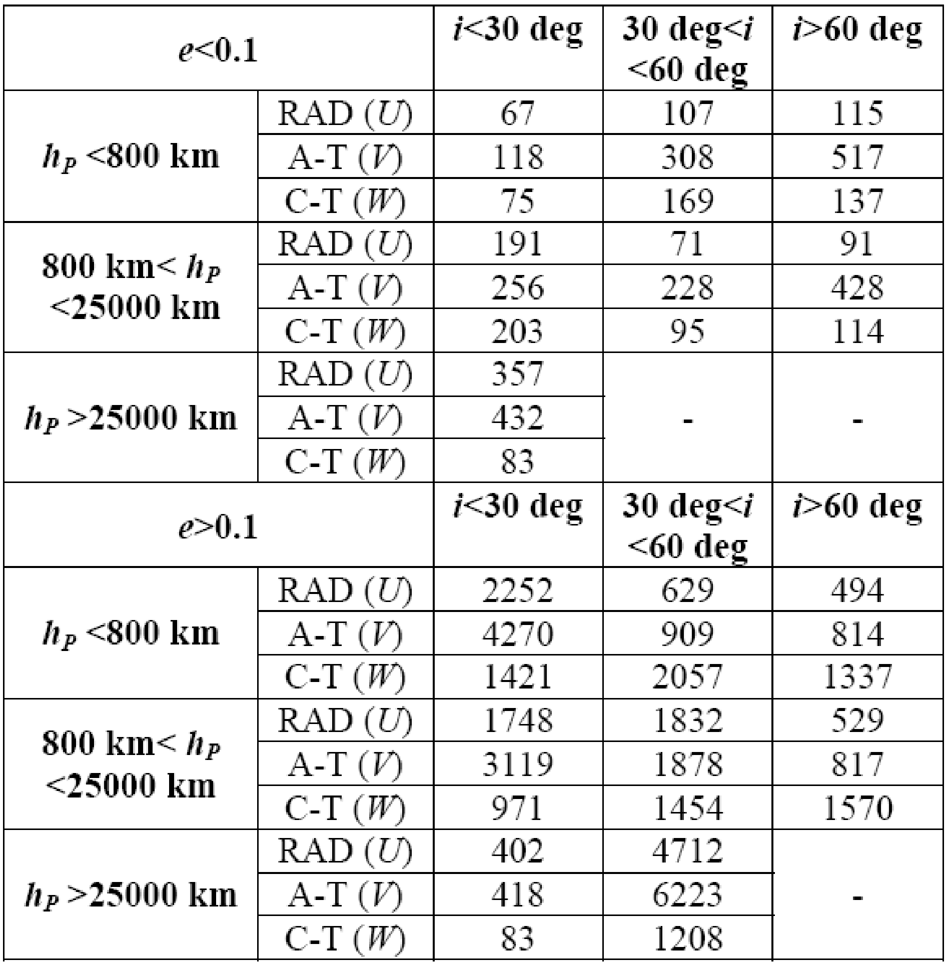


Figure 4.9: Estimated 1-σ-uncertainties (color coded) of all objects in the TLE catalog of January 1st 2008 in U, V and W direction as a function of eccentricity and inclination, per orbital regime. Descending down from top to bottom the Figures correspond to LEO, MEO, GTO, HEO and GEO respectively. (Flohrer et al., 2008)

Table 4.3: Look-up table for average 1-σ-uncertainties in meters in U, V and W direction of all objects in the TLE catalog of January 1st 2008 as a function of inclination, eccentricity and perigee altitude h_p. (Flohrer et al., 2008)



***e*<0.1**		***i*<30 deg**	**30 deg<*i* <60 deg**	***i*>60 deg**
hh_P **<800 km**	RAD (*U*)	67	107	115
	A-T (*V*)	118	308	517
	C-T (*W*)	75	169	137
800 km< h_P **<25000 km**	RAD (*U*)	191	71	91
	A-T (*V*)	256	228	428
	C-T (*W*)	203	95	114
h_P **>25000 km**	RAD (*U*)	357	-	-
	A-T (*V*)	432		
	C-T (*W*)	83		
***e*>0.1**		***i*<30 deg**	**30 deg<*i* <60 deg**	***i*>60 deg**
h_P **<800 km**	RAD (*U*)	2252	629	494
	A-T (*V*)	4270	909	814
	C-T (*W*)	1421	2057	1337
800 km< h_P **<25000 km**	RAD (*U*)	1748	1832	529
	A-T (*V*)	3119	1878	817
	C-T (*W*)	971	1454	1570
h_P **>25000 km**	RAD (*U*)	402	4712	-
	A-T (*V*)	418	6223	
	C-T (*W*)	83	1208	

3, 30, 15 and 1 days, using the latest release of the SGP4 model, the position errors stayed below 40 km, increasing with decreasing altitude.

These TLE accuracy results are about the same as the ESA results. The MEO propagation results are of the same order as in (Kelso, 2007). The GEO propagation results seem to be better than in (Chan & Navarro, 2001), but this might be due to the new and better SGP4 model used.

(Wang et al., 2009) shows similar TLE accuracy results for LEO objects. Here it is stated that propagation errors for very low altitude orbits decrease with eccentricity due to more atmospheric drag within one orbital revolution.

Future methods

A possible future accuracy analysis method might be to reproduce the raw observations for a certain TLE, if one knows the entire TLE production process. The measurement errors and sensor biases may then be estimated to yield TLE accuracy as well.

An improved orbit prediction method using TLEs, which is based on the method and conclusions of (Flohrer et al., 2008), was investigated in Levit & Marshall (2010). The new method yields far more accurate results than the normal SGP4 propagation method. However, since the method and or code is not published in detail it can unfortunately not be used here.

Chapter 5

Orbital Debris Conjunction Analysis

When two objects move within a certain predefined small distance with respect to one another, this is called a conjunction. For object collision probability analysis of the entire catalog, one first has to determine whether two objects are in conjunction. For instance, to determine if a manoeuvre is needed (which is beyond the scope of this research) by a certain high-valued mission in order to mitigate the risk of collision, the probability of collision has to be determined, and is checked to exceed certain threshold values.

This probability analysis, which was treated shortly in Leloux (2010a), can be done after pairs of objects have been found that are in conjunction. This way, when applied to the entire SSN catalog, one does not have to calculate the probability of collision of all combinations of pairs of objects possible. This would take far too many calculations and thus CPU processing time since probability analysis is quite more extensive than conjunction analysis alone.

Furthermore, even conjunction analysis itself can become quite burdensome when applied to the so called all-vs-all catalog test case. In this analysis the possible conjunction of each object of the catalog with respect to every other object has to be determined. Since evaluating the minimum distance of all object pairs of the catalog takes up too much computational effort, a series of filtering techniques or sieves can be used to exclude certain pairs of objects from the catalog which can be proven to never move within a certain distance of each other for the time interval that the conjunctions are determined.

Different techniques have been designed, tested, compared and implemented into existing software by both government and private companies since the 1980s. Their methods and results will be treated in Sections 5.1 and 5.2 respectively.

Due to ever increasing capabilities of the SSN to observe and track orbital debris, in combination with the increasing population, the number of objects in the debris catalog keeps on increasing. A new Space Based Surveillance System has been launched which is predicted to increase the observation capability, in terms of the amount of orbital debris detected, by a factor 2 (Leloux, 2010b).

Furthermore, due to recent intended and unintended collisions like the Iridium/Cosmos collision and the Chinese anti-satellite test, the number of trackable objects also drastically increased, and these events might of course also occur in the near future. The catalog has increased by 60% from 2007-2010, of which 80% was due to the aforementioned two collision events (Kelso, 2010b).

Faster and more efficient ways to detect conjunctions are thus needed in the future to be able to keep up with this catalog growth.

One could argue that only operating spacecraft need to be checked against the orbital debris catalog. However, all-on-all analysis provides the capability to find possible collisions of debris with each other pre-hand, so that possible resulting debris clouds can be followed as fast as possible. Furthermore, the total number of conjunctions in combination with the probability of these conjunction provides a statistical basis for the future debris growth trends and possible Kessler syndrome prediction.

This so-called Kessler syndrome is a kind of 'snowball-effect' of orbital debris, similar to how for instance the rings of Saturn were created. When the amount of orbital debris grows larger and larger, and the collision rate grows beyond a certain threshold, the amount of debris could grow exponentially and this grow would be self-sustaining, possibly dooming the future of safe spaceflight (Donald J. Kessler and Burton G. Cour-Palais, 1978).

In the future, debris collisions could also be mitigated by using laser beams, from the surface of the Earth or from a spacecraft in LEO, that could de-orbit small objects (up to about 10 cm) to decrease the collision probability. Increasing the laser power, orbital debris could be removed in this way by changing the orbit so that it reenters the Earth's atmosphere, or escapes from Earth's gravity field (Wolfgang O. Schall, 1991).

5.1 Overview of Catalog Filtering Methods

The distance of closest approach between two objects can be determined by simply taking the difference of their position vectors against time for a certain period of interest and then numerically stepping forward in time to find the absolute minimum. Analysing all possible pairs in the current catalog like this consumes too much processing power however.

Therefore, faster analytical filters have been set-up to discard objects pairs of which the chaser object does not orbit within a certain reference sphere or ellipsoid around the target object. These analytical methods are also useful since analytical orbit determination was used in setting up the TLE data SGP4 propagator as well.

5.1.1 Hoots

Hoots et al. (1984) designed one of the first analytical methods to determine a close approach between two objects. A series of geometrical prefilters is used to discard any pair of objects which will never be within a distance D of each other.

It is first assumed that the objects follow two-body Keplerian orbits with no perturbations. The effect of perturbations is small and will not change the results if D is set large enough. After close-approach pairs have been distinguished that pass all filters, the perturbations can simply be added by modifying the analytical equations.

Perigee-Apogee

The first, simplest and most used prefilter is the perigee-apogee test. Let q denote the larger of the two perigees of the objects and let Q denote the smaller of the two apogees. If

$$q - Q > D \tag{5.1}$$

then the pair does not need to be considered further. All objects pairs that pass this prefilter move on to the next.

A more in-depth description including a sketch of the situation will be given in Section 8.2, where the implementation is also discussed.

Geometrical

The second prefilter considers the relative geometry of the two ellipses in space. There will be two values d_1 and d_2 of closest approach between the two elliptical paths, see Figure 5.1. So if the smallest of these values is larger than D, the pair is filtered out. An analytical solution to this problem is given in (Hoots et al., 1984).

For near-coplanar objects, that is, for objects which orbital planes have a relative angle close to zero, the solution, which is based on the relative inclination between the two planes, becomes indeterminate, so an exceptional rule is set up.

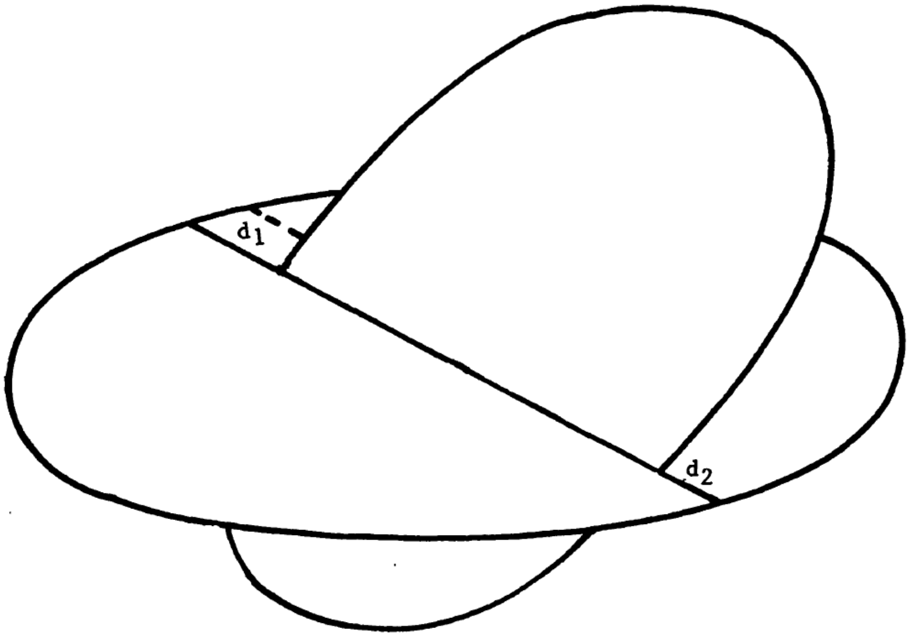


Figure 5.1: Definition of two values d_1 and d_2 of closest approach between two elliptical paths. (Hoots et al., 1984)

Time

Even though the elliptical paths lie within a distance D of one another, this does not necessarily mean that the objects are at this region of close approach in their respective orbits at the same time. A next prefilter can thus be set up based on the time that the objects cross the line of intersection of the two orbital planes.

For non-coplanar object pairs, angular windows around these crossing points are set up which can be converted to time in the period of interest. The windows are then checked for overlaps, and if they do, its midpoint will be used in an iterative process to determine the point of closest approach. This is done by using Newton's method. The period in time in which the TLEs was smaller than D can be determined by this method as well.

For the coplanar cases, the derivative of the TLEs between the two objects is taken and with numerical and interpolation techniques a point of closest approach is determined. Since this process is only needed for a few cases, it will not take much extra processing power. The point of closest approach determined here is also used as a starting point for Newton's method to determine a more accurate point.

As a last step, all prefilters are analytically modified to incorporate perturbations caused by the Earth's oblateness and atmospheric drag.

5.1.2 Khutorovsky

Khutorovsky et al. (1993) proposed a different procedure to search for approaching pairs, which is composed of two parts: preliminary selection and refinement. It was stated that all pairs of objects j and k should be identified that satisfy the condition

$$(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2 < c^2 \tag{5.2}$$

where x, y and z are the elements of the position vector of an object and c is a threshold satisfying two contradictory conditions.

If this condition is tested at intervals $[t_i, t_i + h]$ with time step h, c should be large enough so that no close approach within the interval is missed. But c should be small enough for the filter to be efficient, causing it to reject a large amount of pairs. This procedure consumes too much operations however.

If x, y, z are converted to integer values and binary scales are used for efficient computer implementation, CPU time for this preliminary selection is reduced by a factor 40 for c =400 km and h = 50 sec on a 64-bit processor (Khutorovsky et al., 1993).

In the refinement part, a more accurate moment at which the TLEs between the pairs is minimum within the interval h is determined. The same procedure is now applied three times to the interval $[t_i, t_i + h]$ at which the closest approach in the previous filter was found to be. Every time a smaller time step $h_3 < h_2 < h_1 < h$ and smaller threshold $c_3 < c_2 < c_1 < c$ is used. The object pairs that pass all filters proceed to probability calculations.

5.1.3 Klinkrad

Klinkrad (1993) primarily used the method of Hoots et al. (1984), which was implemented at ESA/ESOC (European Space Operations Centre), to set up a conjunction detection process, which was tested on the accurately known orbits of ERS-1 (European Remote Sensing satellite) and EURECA (EUropean REtrievable CArrier). Furthermore, is was tested on the orbit of STS-48 (Space Transportation System) Discovery, and compared to historic conjunction data of NASA.

A conjunction is here defined as a fly-by of a catalog object at a closest distance inside the reference ellipsoid $d_u \times d_v \times d_w$ of dimensions 10 × 25 × 10 km centred at the target, in which u, v and w stand for the radial, along-track and out-of-plane direction respectively.

The process started off with an epoch filter discarding all TLEs which had an update time of with for instance more than 30 days difference with respect to the reference epoch, since these would lead to unacceptable propagation errors. Secondly, an altitude filter is applied based on the apogee-perigee filter previously mentioned, including altitude decay.

Then the geometrical filter of Hoots et al. (1984) described above is applied, followed by the orbit phase filter. Both of these filters have a collision reference sphere around the target with $r_{sph} = d_v$, since this is the largest dimension of the ellipsoid.

At last, an ellipsoid pass check is performed, discarding all objects that orbit within distance r_{sph} but are outside the reference ellipsoid $d_u \times d_v \times d_w$, which can be determined as follows. If the fly-by distance is defined as

$$\Delta \underline{r} = \underline{r}_c - \underline{r}_t \tag{5.3}$$

in which c defines the chaser and t the target, a conjunction event is accepted by the last filter for $k < 1$ with

$$k^2 = \frac{\Delta r_u^2}{d_u^2} + \frac{\Delta r_v^2}{d_v^2} + \frac{\Delta r_w^2}{d_w^2} \tag{5.4}$$

since the closest distance is then inside the reference ellipsoid.

The time of closest approach t_{con} is determined by iteratively solving for a zero transition of the range-rate between target and chaser

$$\dot{\rho}(t_{con}) = 0 = (\dot{\underline{r}}_c(t_{con}) - \dot{\underline{r}}_t(t_{con})) \cdot \Delta \underline{r}(t_{con}) \tag{5.5}$$

The relative velocity of the chaser with respect to the centre of the moving target at the conjunction event is defined as

$$\Delta \underline{V}(t_{con}) = \Delta \dot{\underline{r}}(t_{con}) = \dot{\underline{r}}_c(t_{con}) - \dot{\underline{r}}_t(t_{con}) \tag{5.6}$$

which would be the collision velocity in case of a direct hit. The azimuth and elevation angles of the relative velocity can be derived as above.

Later on, Klinkrad (1997) used this method, to determine conjunction events of ERS-1 and ERS-2 with TLE data for the period of one year. Here the last two filters were combined however, checking for the reference ellipsoid pass (instead of the reference sphere) within the orbit phase filter.

The same method is used in Sanchez Ortiz et al. (2001) in an implementation for GMV (Grupo Mecanica Vuelo, Spanish for: Flight Mechanics Group) on collision risk assessment for ERS-1 and ERS-2.

5.1.4 Healy

Healy (1995) devised an efficient program to detect close conjunction which can be run on parallel computers, and can thus also be used to run on multiple processors or processor cores. In this program a catalog at a certain time instance can very efficiently be run against itself, one other satellite, or another catalog.

By making use of a Taylor series extrapolation on the motion of an object, the time of closest approach can be found within a certain prescribed time range $[t_0 - T/2, t_0 + T/2]$ in which the state vector is determined by any propagator at time $t_0 + kT$ with k taken to be any integer. Truncating the Taylor series at the linear term, the time of closest approach can be determined to lowest order to be

$$t_{closest} = t_0 - \frac{\mathbf{x}_0 \cdot \dot{\mathbf{x}}_0}{\dot{\mathbf{x}}_0 \cdot \dot{\mathbf{x}}_0} \tag{5.7}$$

where $\mathbf{x}_0$ is the relative state-vector between two two objects, which follow from the propagator. Now the distance at the time of closest approach is also easily determined.

Furthermore, finding a conjunction within some specified critical distance c is done by shifting arrays of position vectors of one catalog at a particular time t_0 with respect to another and computing the distances between the vectors.

This process is then optimised by first sorting the position vectors of the catalog in ascending order by one element, for instance z. When the catalogs are now run against each other the process can stop when a distance is found to be larger than c, since all further computed distances will be larger than c because of the ascending order.

The above method can be applied to all catalog time instances $t_0 + k * T$ and a fast technique combined with the Taylor series can be used to find the minimum distance. The comparison process can be split up in smaller arrays which will be run on parallel computers.

With this method large catalogs can be run against each other efficiently by making use of smart programming methods on parallel computers.

5.1.5 Alarcon-Rodriguez

GMV developed a software tool for ESA/ESOC, which predicts conjunction events of spacecraft with objects in the US Satellite Catalog (Alarcon-Rodriguez et al., 2004). It will be part of a larger collision risk assessment tool (CRASS), which will be treated in more detail later, which determines the collision risk and results in collision warnings based on the probability of collision for a time duration of one week.

For the conjunction detection software, GMV developed a ‘smart sieve’ method (Alarcon-Rodriguez et al., 2002) which comprises a series of sieves (filters) based on fundamental flight principles. These sieves have no constraints or singularities, the method uses less computing time than previous methods, and has better results (more conjunctions detected).

Furthermore, the method includes acceleration effects in determining the conjunction volume as well, which had not been done by previous implementations.

The exact same method has also been implemented in GMV’s focusSUITE in the software tool *closeap* (Mate et al., 2009). Here the catalog is first filtered for old TLE data sets that generate inaccurate propagation and for decayed objects that do not pose a threat any more.

The same process is also described in Klinkrad et al. (2005), Klinkrad (2006) and Klinkrad (2007) as an implementation for collision avoidance of ESA satellites, next to a description on orbit prediction uncertainties and collision risk assessment, respectively.

Theory

In the sieve method, the state vectors of all objects involved are computed at all defined time steps in the interval to be analysed. If the distance between two objects is higher than the threshold distance R_{th}, the pair is discarded. The threshold distance is defined such that the distance between the objects can not attain smaller values than the critical distance R_{cr} until the next step.

Adding the maximum radius of both objects together to form a critical volume with radius R_{cr}, the objects can never touch each other if the chaser stays outside of the volume centred at the target. The relative velocity between two objects in an elliptical orbit about the Earth can never reach two times the escape velocity V_{esc}.

So during a time step duration Δt, the chaser can not enter the critical volume and get out of the threshold volume before the next check is performed at the next time step, if the threshold radius R_{th} is defined by

$$R_{th} = R_{cr} + d_{esc}/2 = R_{cr} + V_{esc}\Delta t \tag{5.8}$$

in which d_{esc} is the distance travelled during Δt with the maximum relative velocity of $2V_{esc}$, see Figure 5.2.

For a chaser object that does enter the threshold volume of a target object at a certain time step, the minimum distance r_{min} between the pair of objects during the time step interval is estimated (excluding acceleration term) by

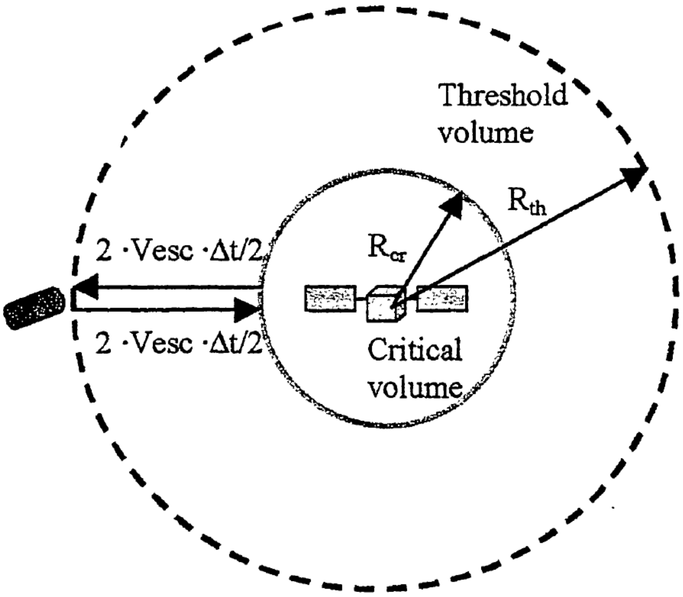


Figure 5.2: Definition of the critical volume and radius R_{cr}, and the threshold volume and radius R_{th} defined by the maximum relative velocity (v_{esc}) of two objects. (Alarcon-Rodriguez et al., 2002)

Healy (1995) to be

$$r_{min} = \sqrt{r_0^2 - \left(\bar{r}_0 \cdot \frac{\bar{v}_{rel}}{v_{rel}}\right)^2} \tag{5.9}$$

in which the subscript 0 denotes the time step at which the state vectors were determined, see Figure 5.3.

The relative acceleration can never exceed twice the acceleration at sea level due to Earth's gravity field g_0, because the objects always orbit above sea level. Using the equation of motion, the maximum deviation d_{acc} from a straight constant velocity path during the time interval Δt can be determined as

$$d_{acc} = \frac{1}{2}(2g_0)\Delta t^2 \tag{5.10}$$

Thus the critical volume can be expanded using the radius of the acceleration safety volume R_{acc}, defined as

$$R_{acc} = R_{cr} + d_{acc} = R_{cr} + g_0\Delta t^2 \tag{5.11}$$

see Figure 5.4.

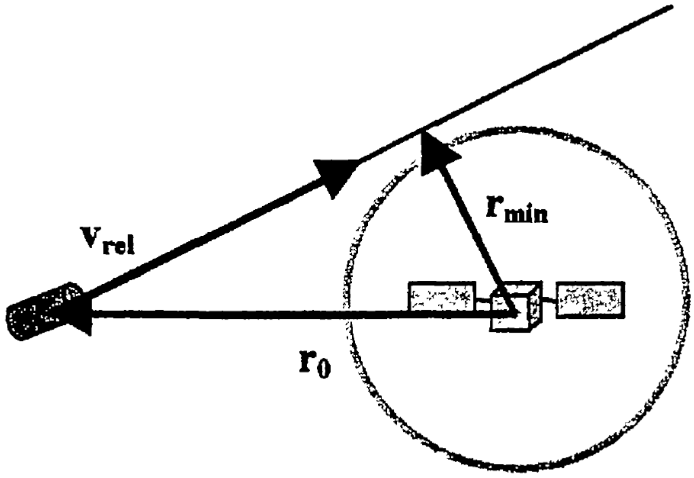


Figure 5.3: Definition of the minimum distance vector $\mathbf{r}_{min}$ defined by the relative velocity vector $\mathbf{v}_{rel}$ and relative position vector $\mathbf{r}_0$ at t_0. (Alarcon-Rodriguez et al., 2002)

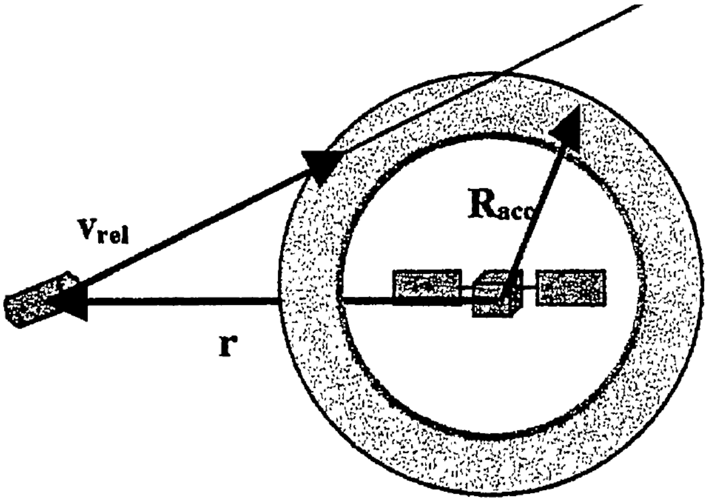


Figure 5.4: Definition of the acceleration safety volume and radius R_{acc} which is defined by the critical volume radius R_{cr} and the maximum relative acceleration($2g_0$). (Alarcon-Rodriguez et al., 2002)

Application

The theory described above is used in a series of sieves, described below, to filter out object pairs that do no reach a predefined distance.

Apogee-perigee filter The method starts with the apogee-perigee filter of Hoots et al. (1984) described above.

X, Y, Z sieves It is followed by X, Y and Z sieves which discard pairs of which one component of their relative position vector **r** is larger than the threshold radius R_{th}

$$r_i > R_{th} \Rightarrow r > R_{th} \qquad \text{for } i = x, y, z \tag{5.12}$$

Step skipping If the distance between two objects is very large and it is known that the chaser will not enter the threshold volume at a number of following steps, this number of future steps N_{skip} can be skipped for analysis for this pair. A safe amount of steps to be skipped according to the reference is

$$N_{skip} = \text{int}\left(\frac{r - R_{th}}{2}\right) \tag{5.13}$$

However, this is not believed to be true. Simply dividing the distance between the two objects minus the threshold radius divided by two would yield just a distance and not a dimensionless number of steps to be skipped. If the conservative maximum relative velocity of twice the escape velocity is taken into account the equation would be derived as

$$N_{skip} = \text{int}\left(\frac{r - R_{th}}{2v_{esc}\Delta t}\right) \tag{5.14}$$

In this way, the aforementioned distance is divided by the maximum relative velocity times one time step, yielding the number of steps which can safely be skipped. Rounding this value off to one integer below, the chaser object will never be able to penetrate the threshold volume, not even if their relative speed would be twice the escape velocity.

r^2 sieve Object pairs that pass the X, Y, Z sieve can still have a distance larger than r, which is checked in the r^2 sieve

$$r^2 = r_x^2 + r_y^2 + r_z^2 > R_{th}^2 \tag{5.15}$$

Minimum distance sieve The minimum distance sieve discards object pairs for which $r_{min}^2 > R_{acc}^2$ where the acceleration effects are included in R_{acc} and r_{min} is calculated with the method of Healy (1995) described above.

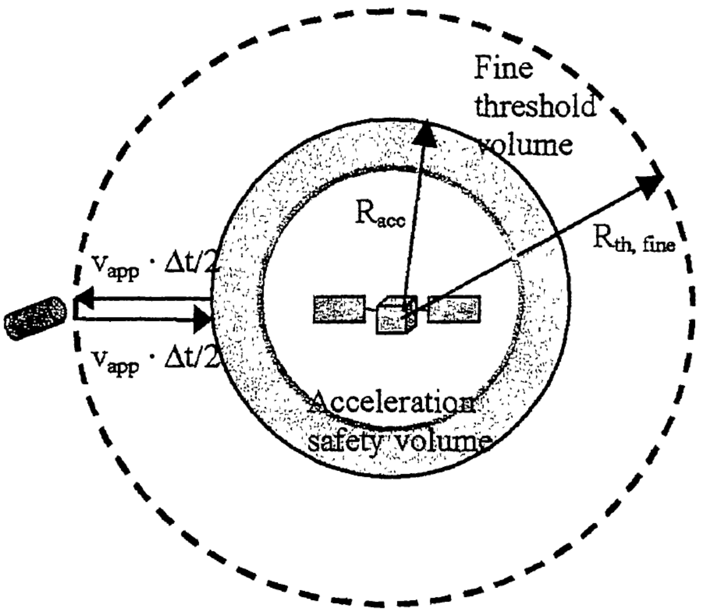


Figure 5.5: Definition of the fine threshold volume and radius $R_{th,fine}$ defined by the actual approach velocity v_{app} and acceleration safety volume radius R_{acc}. (Alarcon-Rodriguez et al., 2002)

Fine r^2 sieve Then, a fine r^2 sieve is applied. A chaser object might achieve a distance with respect to the target smaller than R_{th} but still not enter the critical volume during the following time step considering the actual relative velocity.

The sieve is similar to the r^2 sieve, but uses the actual relative velocity v_{app} instead of $2v_{esc}$, and R_{acc} (Equation 5.11) instead of R_{th} (Equation 5.8). It discards object pairs for which $r^2 > R^2_{th,fine}$, where $R_{th,fine}$ is given by

$$R_{th,fine} = R_{acc} + \frac{1}{2}v_{app}\Delta t = R_{acc} + \frac{1}{2}\left|\bar{r}_0 \cdot \frac{\bar{v}_{rel}}{v_{rel}}\right|\Delta t \tag{5.16}$$

according to the original reference (see Figure 5.5).

However, on examination, the equation does not seem to be correct, and it is believed that the correct version should be

$$R_{th,fine} = R_{acc} + \frac{1}{2}v_{app}\Delta t = R_{acc} + \frac{1}{2}\left|\bar{v}_0 \cdot \frac{\bar{r}_0}{r_0}\right|\Delta t \tag{5.17}$$

which was later confirmed by Klinkrad (2006), wfere a summary of the theory described in Alarcon-Rodriguez et al. (2002) is given.

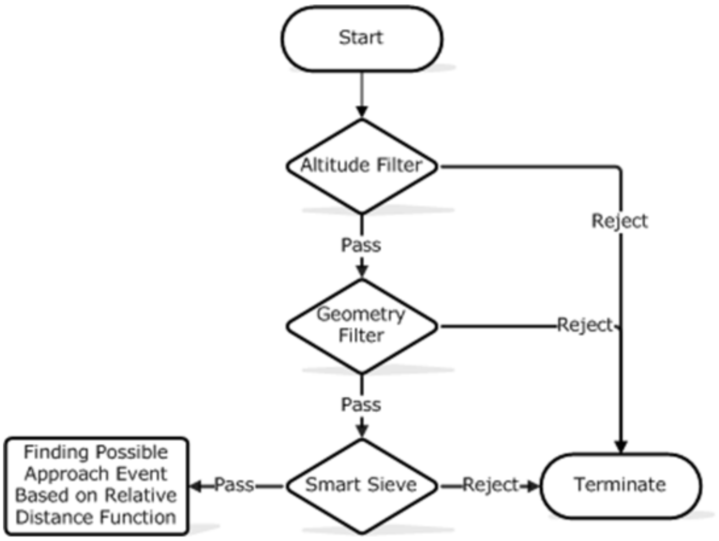


Figure 5.6: Diagram of the process of the new conjunction analysis method by (Ting & Hai, 2008)

Fine conjunction detection At last, a fine conjunction detection is performed which numerically finds null values for v_{app}^2 to accurately find the point of closest approach.

5.1.6 Ting & Hai

Ting & Hai (2008) investigated a new method to determine close approaches between satellites, both for the conjunction of a single satellite against the complete catalog or the catalog against itself. Improved versions of the apogee-perigee, geometrical and time filter from the method of Hoots et al. (1984) were used, along with adjusted sieves of Alarcon-Rodriguez et al. (2002).

Finally, the method of Alfano (1994) is used to determine the final minimum distance and time of closest approach, which is called the relative distance function method. The whole process is illustrated in Figure 5.6.

The basic apogee-perigee filter is adjusted by adding the maximum effect of orbital decay and perturbations; a fixed value of 10 km is used here. The geometrical and time filter are transformed to an interval analysis of the arguments of latitude of pairs of objects around the time of closest approach, including perturbations, and furthermore using a step skipping process.

The threshold and fine threshold distance in the smart sieve are adjusted, as well as the step skipping process. The r^2, minimum distance sieves and fine conjunction detection are not used.

The method finishes with an accurate distance determination of the con-

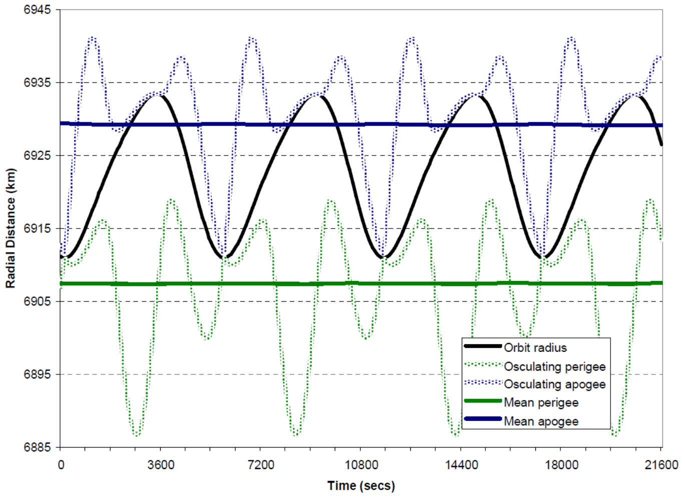


Figure 5.7: Comparison of the mean and osculating values of perigee and apogee of object #26281 in the analysis interval. (Woodburn et al., 2009)

junction events, based on a relative distance function which was republished in Vallado (2007) (the original publication could not be found), which was improved in Alfano (1994).

5.1.7 Woodburn

The three classical filters set up in Hoots et al. (1984) were investigated and improved in Woodburn et al. (2009) as well. In the classical method the apogee and perigee were sampled at the beginning of the analysis interval based on mean orbital elements. The orbital elements were determined at the midpoint of the analysis interval, and using approximate rates of those elements the value at the start point was determined.

In Figure 5.7, the real radius of a LEO object is shown next to the osculating and mean apogee and perigee values for an interval under consideration. It can be seen that the osculating values at the start of the interval can yield rather large errors. Even when using the mean value for the apogee or perigee, would provide a better approximation than using the osculating value, for most cases.

However, for HEO objects, the osculating and mean values for apogee and perigee fluctuate significantly with respect to the mean values and thus taking a sample at the start of the interval does not lead to good values that can be safely used in the filter.

The proposed method now takes samples at both the start and end of the analysis interval and uses a pad which is added to the detection threshold to account for periodic effects during the interval. A minimum padding distance is required which covers the periodic effects but still also minimises the amount of pairs that pass through the filter. However, the pad needs to be set large enough to cover all test cases.

To determine the pad size, the maximum error was determined between the actual largest and smallest radii for all objects and what would be the result when the osculating or mean values were taken at the start and end time of the interval. The maximum error is now used to determine a pad size of 30 km. However, the filter may still fail using this value in the rare occasion that both samples at start and end time are taken at a deep through in the variation for apogee and perigee in the HEO case.

Another proposed method considers to determine the actual value of the apogee and perigee of the entire interval so that only a pad size of 1 km can be used to increase the efficiency of the filter. This method however does increase the total computational effort significantly, since for TLEs the orbits needs to be propagated over the entire interval.

Furthermore, the classical orbit path filter was also investigated in the same way and it was determined that simply adding a pad, even while taking samples at the start, mid and end points of the interval, was not enough to solve the problem of false elimination of pairs in the filter. Improving the filter requires even more samples and a technique to find the minimum orbit-to-orbit distance over the analysis interval.

However, these improvements require complex algorithms to find trends in the minimum distance and these add significantly to the total computational loads and thus may not be beneficial. Furthermore, when using this improved filter, some conjunctions were missed due to false rejection of certain pairs.

Finally, the time filter was investigated and it was determined that the orbit crossing intervals computed in the classical filter do not comply with the actual time period near the node of these orbits, and the error will furthermore increase with time. A functional flow diagram of the improved code was set up which was not described in detail.

The period between the first two nodes of each object were determined using osculating elements. This period is now used to determine the next node crossings which is subsequently checked as well. A system can be set up in which the check might skip some periods while still being accurate enough. It was determined that using the time filter decreased the total computation time significantly and could safely be applied.

In Coppola et al. (2009) this improved method of Woodburn et al. (2009) was used to run an all-vs-all conjunction analysis using parallel processors and machines. The total amount of data was split up in smaller groups which were then allocated to different processors. An analysis was done using varying source data, time intervals, number of processors, machines, group sizes, number of groups and tasks, and combinations of the first and last filter.

5.2 Results of Filtering Methods

The results of the previously described filtering methods have been extracted from their respective documents and are listed in Table 5.1, which is discussed below.

5.2.1 Catalog Date

For each method the date of the catalog which has been used for the analysis is listed, along with the amount of objects n in that catalog. Some methods were run on one or two satellites versus the catalog, while others analysed the entire catalog versus itself. This has consequences for the amount of object pairs p analysed with respect to a certain catalog size.

Then, the conjunction distance D defining when a close approach is counted as a conjunction is listed for the different methods, along with the time interval T for which the analysis was performed. Next to that, the time step Δt used in the time interval T is given for some methods. Finally, the number of conjunctions c detected by the method is given, using the previously described input. For fields that are empty, no data was given in the corresponding document.

The catalog date in Hoots et al. (1984) is not given, but the catalog date must be older than the document received date. In Khutorovsky et al. (1993) an average value is given for catalogs starting from the stated catalog date. The analysis by Klinkrad (1997) is done over a time interval of more than 14 months using multiple catalogs.

5.2.2 Number of Objects

The number of objects n in the analysed catalog is sometimes not exactly given in the document, but rather a rounded off value is stated like a 'population of over 8000 objects' in Alarcon-Rodriguez et al. (2002). This is likewise indicated in the table.

Furthermore, the amount of object pairs p analysed for an all-vs-all catalog analysis (not all results in Table 5.1 are for an all-vs-all analysis) is determined with

$$p = \frac{n^2 - n}{2} \tag{5.18}$$

Table 5.1: Comparison of the different conjunction analysis methods; their input parameters and number of conjunctions detected.

method [**author**]	**catalog date** [**y.m.d**]	**n** [#]	**p** [#]	D [**km**]	T [**d**]	Δt [**s**]	**c** [#]
Hoots et al. (1984)	< 1983.12.23	4003	4003	100	1		84
Khutorovsky et al. (1993)	> 1992.06	±7000	±24496500	30	1	50	2000
Healy (1995)	1993.01.17	±7000	±24496500	2	1		140
Klinkrad (1997)	1995.12.04-1997.02.17	8210	16420	7.5	440		4596
Sanchez Ortiz et al. (2001)	1998.06.15	8119	8119	7.5	14		3
Alarcon-Rodriguez et al. (2002)	2002.05.15	>8000	>31996000	25	1	180	10546
Ting & Hai (2008)	2007.08.03	±12000	±71994000	25	1	425	116198
Woodburn et al. (2009)	2009.02.11	11970	71634465	5	1		5338

since a satellite can not have a conjunction with itself (subtract n from number of combinations n^2), and every pair only has to be analysed once (divide by two). After this analysis was done, the equation was verified by Coppola et al. (2009), a reference which was found afterwards.

5.2.3 Conjunction Distance

The conjunction distance D is mostly given for a reference sphere, but for the method of Klinkrad (1997) a reference ellipsoid is used. In the latter case the average of the three dimensions of the ellipsoid was taken, for comparison with the reference spheres of the other methods.

The data of Woodburn et al. (2009) was complemented with data of Coppola et al. (2009) which partly overlapped. It was assumed that the same results applied, since the research was done on the exact same catalog and by some of the same people in the same time period and presented at the same conference.

Three methods were run for varying conjunction distance D, which resulted in a different number of conjunctions detected, see Table 5.2. The amount of conjunctions in Khutorovsky et al. (1993) where the resulting average of multiple runs (time intervals), and so can attain decimal values.

To be able to compare different methods with respect to each other, a relation between the different input variables might be found. If trendlines added to the above results, it is found that a polynomial of the second degree has a perfect fit for all three cases. In setting up this trendline, the top results of the first two methods are not taken into account, since they deviate both in value as well as in position in the documents.

An attempt to compare the detection performance of different methods was made by assuming that the number of conjunctions detected is proportional to the amount of object pairs and the time interval analysed and quadratically related to the conjunction diameter D (ignoring the linear and constant terms in the polynomial):

$$c \propto pTD^2 \tag{5.19}$$

So, when calculating the ratio $\frac{c}{pTD^2}$ for different values of D for the same method, a constant value should result. However, this was barely the case for (part of) two of the three methods. The equation above can thus not safely be applied to compare different methods with respect to each other. If this is done anyway, the results differ too much and the expected increase in performance with newer methods is not obtained.

When the entire polynomial was used in calculating the ratio for each method, a constant ratio did result. But since the polynomial is different for each method, it cannot safely be applied to compare all methods with

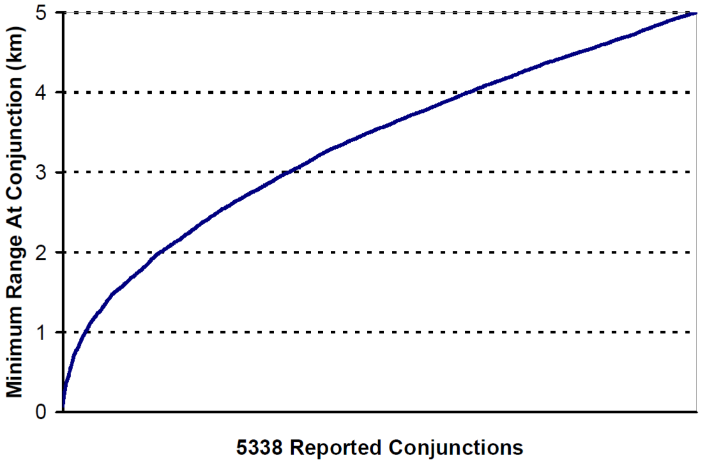


Figure 5.8: Distribution of minimum range at conjunction for all 5338 reported conjunctions in (Coppola et al., 2009).

respect to each other. It can thus be stated that the amount of conjunctions detected c is dependent on other variables as well, which were not taken into account.

Furthermore, the abundance and domain of the results prove not to be large enough to safely perform this kind of statistical analysis. Some of the input values might not be accurate enough as well, and the methods differ too much from each other.

Since the tried method did not have fruitful results, the plots and ratio values are not shown here in detail here.

In Coppola et al. (2009), the minimum range of the conjunction was plotted versus the reported number of conjunctions, see Figure 5.8. After analysis of the inverse of this graph, it was found to show a perfect second order polynomial function (e.g. $c \propto D^2$), supporting the previous discussion. This reference was found after the previously mentioned analysis had been done.

5.2.4 Time Interval

The analysed time interval T was not the same for the different methods, although multiple methods used a time interval of one day. The method of Woodburn et al. (2009) was also applied to a time span of 5 days, which yielded 26933 conjunctions, so about a factor 5 more. The proportionality with time interval assumed in Equation 5.19 was thus reasonable.

5.2.5 Time Step

The time step Δt used in the analysed time interval was optimised with respect to the CPU running time for three methods, but for most methods it was not given or used. The optimisation for the time step of Alarcon-Rodriguez et al. (2002) and Ting & Hai (2008) can be seen in Table 5.3.

A smaller time step than the optimum value yields a longer running time because the analysis has to be done at more points in the interval T. With a larger time step however, more conjunctions pass the first sieves, which are more efficient, and thus more conjunctions have to be analysed by the last computationally intensive fine detection or relative distance function.

5.2.6 Filter Efficiency

The performance of the successive (pre)filters and sieves used in four methods can be seen in Table 5.4, for which data sometimes was extracted from the text in certain documents. It lists the percentage of rejected object pairs with respect to the amount of pairs analysed by a certain filter, and the percentage that passed the filter with respect to the total amount of pairs analysed at the beginning of the analysis.

In this way, the relative efficiency of the filters can be seen, and if this is compared to the relative running time needed for the filter, an optimum order might be determined. Similar filters in different methods have been given the same name for better comparison.

5.2.7 Computation Time

In Alarcon-Rodriguez et al. (2002) and Ting & Hai (2008), a comparison was made with older methods on the same machine, in order to assess the performance of their newly defined method. This can be seen in Table 5.5, in which the CPU running time and number of conjunctions detected is listed for different methods.

For each of the researches, the input catalog, conjunction distance, threshold distance and time interval were set to the same values in the different methods analysed, in order to make a proper comparison. Nothing is said about the time step however, but one might assume that for each method an optimal value was used.

Alarcon-Rodriguez et al. (2002) used a 500 MHz Pentium III and Ting & Hai (2008) used a 1.66 GHz Centrino Duo.

The results of Alarcon-Rodriguez et al. (2002) shown in this table indicate that previous methods had a longer running time while being able to detect less conjunctions. Especially the filtering method scores very poorly, while Hoots et al. (1984) states that all 84 conjunctions were found by this same filtering method compared to a truth model defined by a brute force

Table 5.2: Number of conjunctions less than distance D, for three different methods.

Khutorovsky et al. (1993)		Healy (1995)		Klinkrad (1997)	
D [km]	c	D [km]	c	D [km]	c
30.0	2000	2.00	140	7.500	4596
3.0	375	1.50	135	3.750	1167
2.0	166	1.00	57	2.500	513
1.0	41.6	0.75	33	1.500	189
0.5	10.7	0.50	18	0.750	44
0.3	4.1	0.40	12	0.500	18
0.2	2.7	0.25	6	0.375	8
0.1	0.65	0.15	2	0.300	6

Table 5.3: CPU running time versus time step Δt.

Alarcon-Rodriguez et al. (2002)		Ting & Hai (2008)	
Δt [s]	CPU time [m:s]	Δt [s]	CPU time [m:s]
60	19:40	300	11:18
90	14:58	400	10:43
120	12:50	425	10:20
180	11:07	450	10:29
240	12:40	500	10:50
300	16:24		

Table 5.4: Performance of the successive filters/sieves for different methods.

	Hoots et al. (1984)	
prefilter	rejected/analysed [%]	passed/total [%]
p-a	23.5	76.5
geometrical	36.2	48.8
time	98.4	0.42
processing	80.7	0.08
	Klinkrad (1997)	
filter	rejected/analysed [%]	passed/total [%]
epoch	4.7	95.3
p-a	69.7	28.9
geometrical	2.2	28.23
time	94.4	1.58
	Sanchez Ortiz et al. (2001)	
filter	rejected/analysed [%]	passed/total [%]
epoch	13.5	86.5
p-a	69.5	26.4
geometrical	98.0	0.52
time	92.9	0.04
	Alarcon-Rodriguez et al. (2002)	
sieve	rejected/analysed [%]	passed/total [%]
p-a	59.4	40.6
X	74.3	10.4
Y	72.9	2.8
Z	50.8	1.4
r^2	34.5	0.92
min. dist.	90.4	0.88
fine r^2	41.9	0.51
fine det.	99.0	0.005

Table 5.5: CPU time and number of conjunctions c detected for different methods.

Alarcon-Rodriguez et al. (2002)		
method	CPU time [m:s]	c
filtering	30:34	10[1]
sieve	20:53	9406
p-a + sieve	13:28	9363
smart sieve (ss)	11:07	10546
Ting & Hai (2008)		
method	CPU time [m:s]	c
rel. dist. func.	181:44	116198
ss + rel. dist. func.	18:10	116198
proposed in ref.	10:20	116198

stepping technique. This means that either the brute force stepping technique can not find all conjunctions, which seems unlikely, or the method was not applied very well by Alarcon-Rodriguez et al. (2002), or both.

Ting & Hai (2008) again improved previous versions, and shows that all conjunctions are detected, by comparing it to other techniques which result in the same number of conjunctions while using more CPU running time. However, this could also mean that the results are biased because all methods conclude with the relative distance function method. This only proves that the sieves used prior to this function do not miss any conjunctions *not also* missed by the function, but this function might not necessarily be able to find all conjunctions. A different reference method is thus missing here.

Furthermore, the method of Hoots et al. (1984) was said to have a run-time savings factor of 95 with respect to their brute force method.

The method of Woodburn et al. (2009) also gave computation time results, both when using TLE data and interpolated ephemerides data, of which the latter was determined by propagating the TLE data. Different filter combinations were set up and tested on objects with different orbit regions.

The results showed that in the all-on-all test case the combinations of multiple filters decreased the total running time by a maximum factor of 42 with respect to no filters used at all, when TLE data was used. In this case however some conjunctions were missed.

When the orbit path filter was excluded a maximum gain of 18 could be achieved. In this combination the apogee-perigee filter calculated the precise values for the extremes. Although this added to the calculation time

[1] This result seems to be contradictory to the results of Hoots et al. (1984), see text for details.

up front, this was outweight by the increased number of rejected pairs.

In Coppola et al. (2009) it was also shown that applying these two filters decreased computation time by a factor 17. Furthermore, it was shown that the computation time can also be decreased by a smart choice of the number of groups and tasks allocated to different processors or machines, and by using binary data instead of ASCII (American Standard Code for Information Interchange) files.

Increasing the time interval by a factor 5 increased computation time by a factor of about 4. This latter in-proportionality was to be expected since some preprocessing needs to be done for both cases.

Finally, it was stated that the ephemeris memory footprint increases very rapidly when increasing the catalog size and time interval. When the memory footprint exceeds 3 GB a 64-bit operating system is required since more memory storage is not possible on 32-bit machines.

Chapter 6

TLE Catalog Analysis

In order to analyse a TLE catalog, or use its data to propagate one or more objects, some steps need to be taken.

The data has to be downloaded (Section 6.1), and read from file into memory and converted to programming variable types (Section 6.2). Integrity checks can be performed to make sure the data is valid. Then, some variables can be calculated which are not written explicitly in the TLE, but can be derived from the data within a TLE (Section 6.3).

Once this has been done, the variables can be written as output to a text file, which can be read in with Matlab code to perform statistical analysis and make figures (Section 6.4).

All this (except for the Matlab part) is accomplished with c++ code which has been written for Tudat as well. The structure of this code along with some other useful functions is explained in Section 6.6.

6.1 Downloading TLE Data

A catalog dataset can be downloaded by making use of the TLERetriever program of CelesTrak (Kelso, 2010a) (Chapter 4.1), which processes data received from SpaceTrack (USSTRATCOM, 2004).

With this program one can download the most recent entire catalog (which is updated twice per day), or a subset of the catalog, i.e. only weather or navigation satellites, or satellites of one constellation, see also Figure 4.2.

Furthermore, one can choose to download complete daily data, which contains a set of all TLEs of that day. When one has datasets of multiple days, one can process them in order to create the most complete catalog of that moment, containing the single most recent TLE of every object it can find. So even if the most recent TLE of an object was for instance from three days ago, it would now be included in the catalog.

The data is saved in a text file in the format which was explained in Chapter 4. A user can choose between 2-line or 3-line data as well, the latter containing an extra line with the common name of the object. A valid SpaceTrack account is needed though, which has to be validated and activated by USSTRATCOM after registration.

6.2 TLE Reader

Once the data is downloaded and processed, it needs to be read into the memory of a computer using programming code, if one wants to do analysis with it. The exact details of the code will be given in Section 6.6, but a general outline of all the functionality will be given first.

Since the TLE data has a specific format, i.e. with implied leading points, exponents and numbers stacked together, the data needs to be read in and converted to the appropriate variable types (integers, doubles, strings) by the code.

Once this is done, some sanity checks can be performed, to see if all TLEs are valid. For instance, each TLE needs to be made up out of two or three lines, both lines need to start with the '1' or '2' integer, the checksum of both lines has to be correct, the object ID's of both lines need to be the same, the data public users download always has the 'U' character for unclassified data, and the orbital model is always '0' nowadays.

If the integrity of a TLE does not appear to be correct, it is deleted from the list and the program continues with the correct TLEs.

6.3 TLE Calculations

Once the integrity of the TLE data is checked, and the variables obtained from the TLE are stored in memory using their corresponding variable types, some calculations can be performed. Variables which are not explicitly stated in the TLE, but can be derived from the data in the TLE, can then be stored as well.

For instance, from the mean motion n of an object which is stated in its TLE in rev/day, converting to rad/s, one can determine the semi-major axis a using the following equation

$$n = \sqrt{\frac{\mu}{a^3}} \tag{6.1}$$

in which μ, the gravitational parameter of the Earth is defined in Chapter 3a,d its value can be found in the now following table.

Table 6.1: Fundamental WGS-72 and WGS-84 constants. (Vallado et al., 2006)

Constant	WGS-72 Value	WGS-84 Value
μ	398,600.8 km^3/s^2	398,600.4418 km^3/s^2
$R_\oplus$	6378.135 km	6378.137 km
J_2	0.001 082 616	0.001 082 629 989 05
J_3	-0.000 002 538 81	-0.000 002 532 153 06
J_4	-0.000 001 655 97	-0.000 001 610 987 61

For TLEs and SGP4, the WGS-72 (World Geodetic System) constants were originally used (Hoots & Roehrich, 1980), although it might be that internally USSTRATCOM has switched to the WGS-84 constants, but this is unfortunately not known to the public. The WGS-72 set is used here and in subsequent analysis, as this was recommended in Vallado et al. (2006).

See Table 6.1 for an overview of the constants in both sets.

Since one also knows the eccentricity e from the TLE, the radius at perigee r_p and radius at apogee r_a can be determined with the following equations

$$r_p = a(1 - e) \tag{6.2}$$

$$r_a = a(1 + e) \tag{6.3}$$

see also Chapter 3.

Since the TLE can only state the revolution number up to 99999, which loops back to 0, an attempt was made to recover the total number of revolutions by making use of the objects' launch year and mean motion. Simply multiplying the orbital lifetime in days of a satellite with the average of its mean motion n (now in rev/day) over this lifetime will result in the number of orbits.

The launch year of an object is given in the TLE, so the lifetime can be determined with an uncertainty of ±1 year. To determine the average mean motion however, one would need all TLEs of all objects, which unfortunately are not all available.

An approximation is made by using the current mean motion. When one uses this value in the calculations, an approximate number of total revolutions results. When one compares this value to the TLE revolution number of the object, which is exact (although missing a multiple of 100k), one can add the appropriate multiple of 100,000 revolutions to get the exact total number of revolutions.

Finally, all variables that were read in and calculated, can be written as output to a text file in tabular form, for subsequent analysis.

6.4 TLE Statistical Analysis

The tabular output data can now be read into Matlab for statistical analysis.

Histograms of some interesting variables in terms of number of objects were made, see Figures 6.1 - 6.10, for complete and processed catalog files of the 10th of January 2011 and the 10th of April 2011. The former catalog is indicated in blue and the latter in red, so that a nice comparison can be made.

The increment of the variable on the horizontal axis can be seen in the titles of the figures, as well as the total number of objects in that catalog.

Some variables of the TLE objects are plotted versus each other in Figures 6.11 - 6.15. For these plots only data from the catalog of 2011.04.10 was used, as they did not differ that much from the same graphs of the catalog of three months older.

6.4.1 TLE Object Histograms

In Figure 6.1 a histogram of the Epoch Day can be seen, which shows that most TLEs are very recent.

The bulk of objects has a maximum age of 2-3 days, while the maximum age of a certain TLE is less than 30 days here. This is a result of the processing done by SpaceTrack or CelesTrak, which kicks out outdated TLEs older than 30 days. Negative values indicate the previous year (2010) here, so that the graph is more clearly distributed on one small axis.

Figure 6.2 shows the inclination i versus number of objects, which contains a small peak above 0 degrees representing the geostationary satellites, with their ground tracks following the equator. The next peak is due to the navigation and Molniya orbits, of which the latter have an inclination of about 63.4 degrees.

Two other peaks are of the debris clouds of the Cosmos 2251 and Iridium 31 satellites, with inclinations of 74 and 86 degrees (Kelso, 2010c) respectively. The largest peak can be seen at the end, around 98 degrees inclination, around which sun-synchronous orbits are situated.

Furthermore, there are a large number of semi-polar orbits around 82 degrees, while some other smaller peaks are due to the latitudes of launch sites with certain azimuth launch constraints.

The right ascension of the ascending node Ω in Figure 6.3 shows uniformly distributed scattered results, which was to be expected, although one peak seems to be shifted over 30 degrees in 3 months. The cause for this effect is unclear.

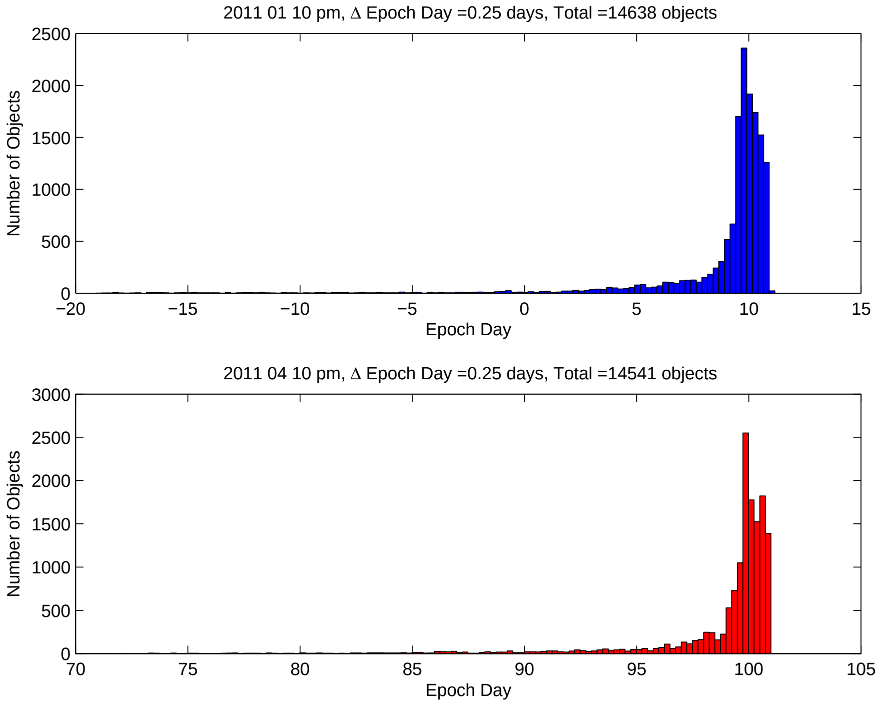


Figure 6.1: Histogram of the Epoch Day.

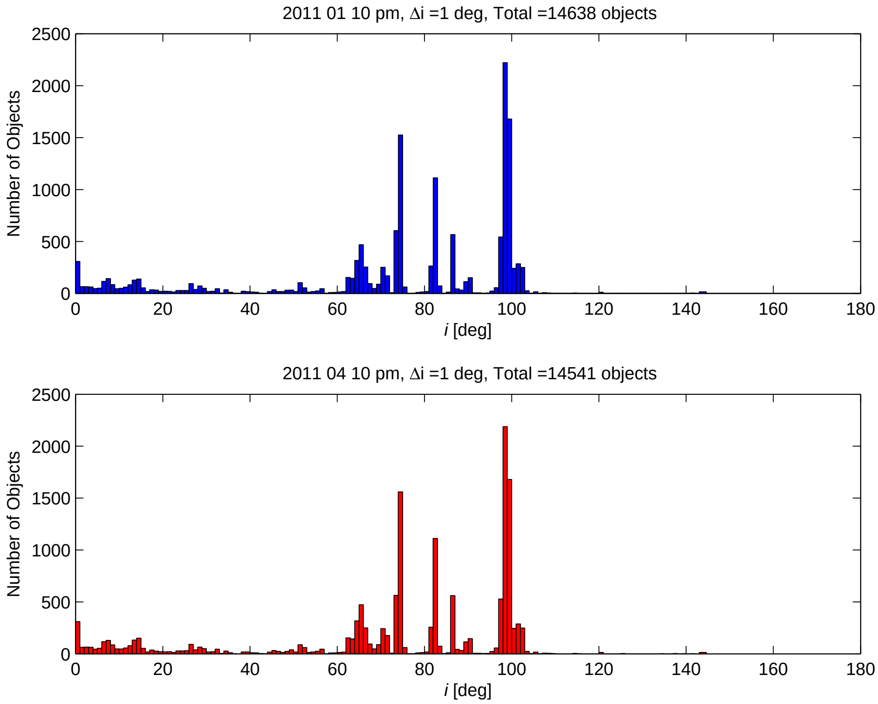


Figure 6.2: Histogram of the inclination i.

One could argue that all sun-synchronous orbits have about the same Ω at a particular time. However, after three months Ω would then have to be changed by $360/4 = 90°$, which is not the case.

The eccentricity e can be seen in Figure 6.4, in which most objects are concentrated near the zero value, as most orbits are (near-) circular. This distribution can be seen more clearly in Figure 6.5, which is zoomed in to the 0-0.1 region.

Furthermore, a peak is seen in the 0.65-0.75 region, which can be seen more clearly in Figure 6.6. This region includes GTO and Molniya objects, both with eccentricities around 0.72.

The argument of perigee ω and mean anomaly M are distributed evenly in Figures 6.7 and 6.8, which was to be expected, as the objects are scattered around the Earth.

The mean anomaly does seem to have some peaks near 0/360 degrees, indicating that more objects would be closer to the their perigee. This does not seem logical however, since one would expect more objects near the apogee, as the objects move more slowly there.

But, since most orbits are circular, the peak might also occur due to the fact that the perigee is not always well defined for a circular orbit. A certain reference value might be taken for near-circular objects for which this is the case.

In the histogram of the mean motion n some peaks can be seen as well.

The peak at 1 rev/day represents the GEO objects, while the peak around 2 rev/day is due to the navigation and communication constellations of satellites like GPS and GLONASS, and due to the GTO objects. The peaks around 12.5, 13.8 and 14.5 rev/day are LEO objects corresponding to preferred orbital heights for science, weather and remote sensing satellite, respectively.

In Figure 6.10 the total number of revolutions can be seen.

The amount of objects decreases with increasing revolutions, since newer objects are more abundant, due to the decay of objects into the Earth's atmosphere. The two peaks are due to the debris clouds of the Chinese ASAT (Anti-SATellite weapon) test and collision.

The TLE variables not shown here were analysed as well, but their histograms were not interesting enough to show in this report. Some of them will be discussed shortly now.

For the TLE line number it was a nice extra check to see if all lines 1 and 2 of the catalog carried the integer 1 and 2 respectively, which they did.

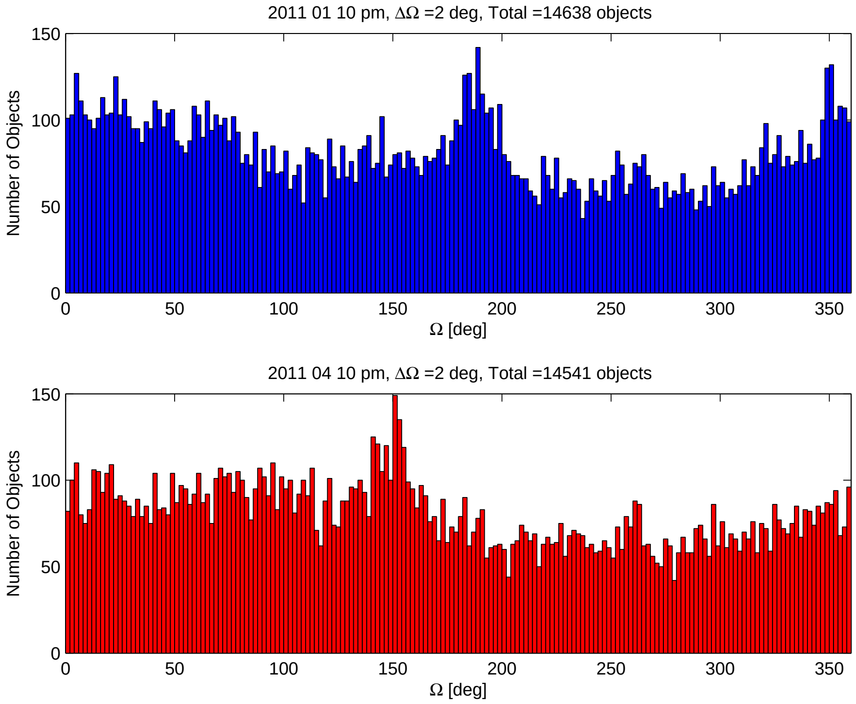


Figure 6.3: Histogram of the right ascension of the ascending node Ω.

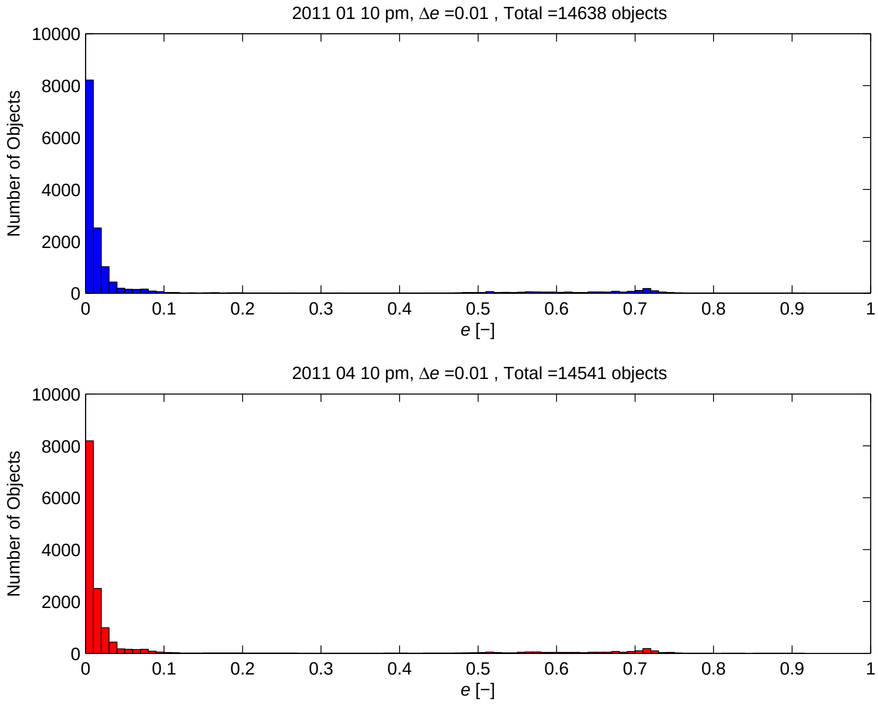
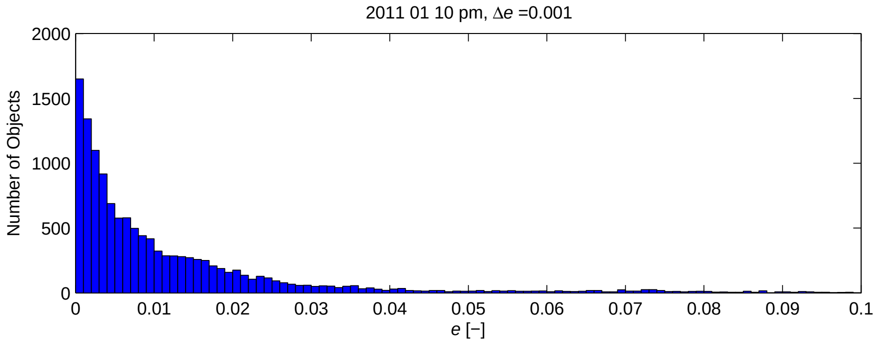


Figure 6.4: Histogram of the eccentricity e.



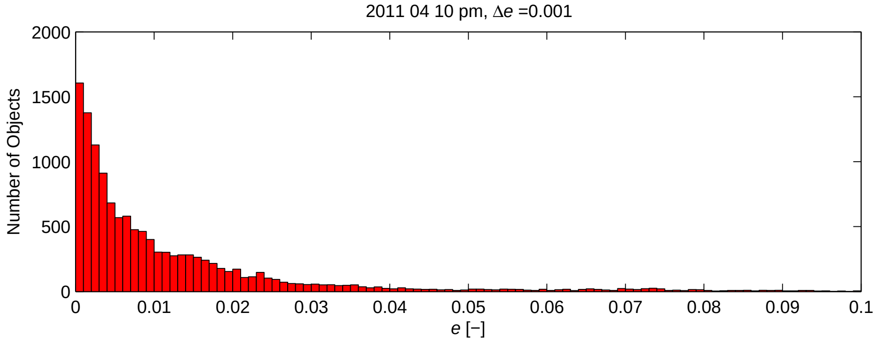
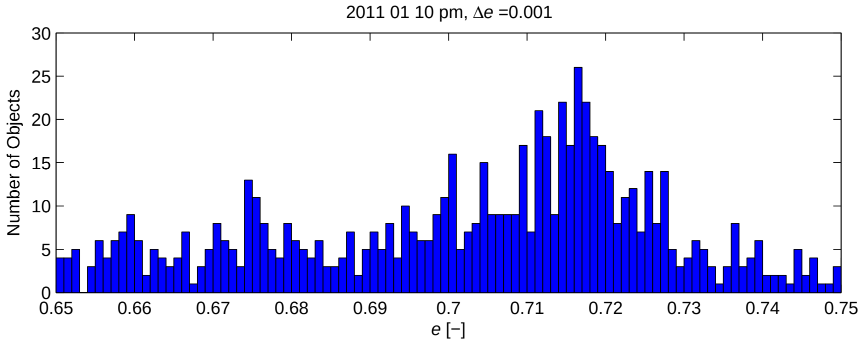


Figure 6.5: Zoomed in version of Figure 6.4 from 0-0.1.



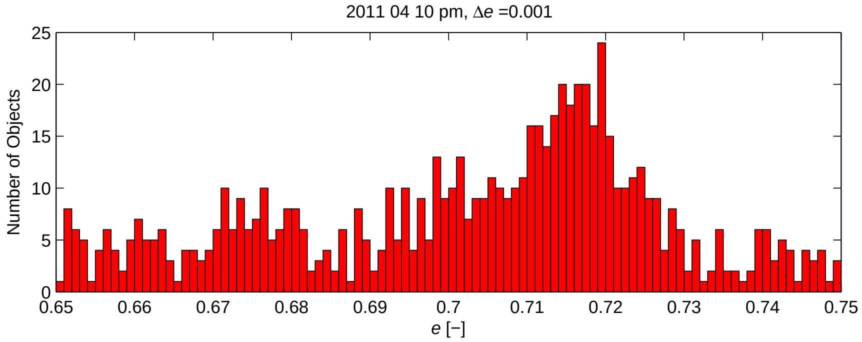


Figure 6.6: Zoomed in version of Figure 6.4 from 0.65-0.75.

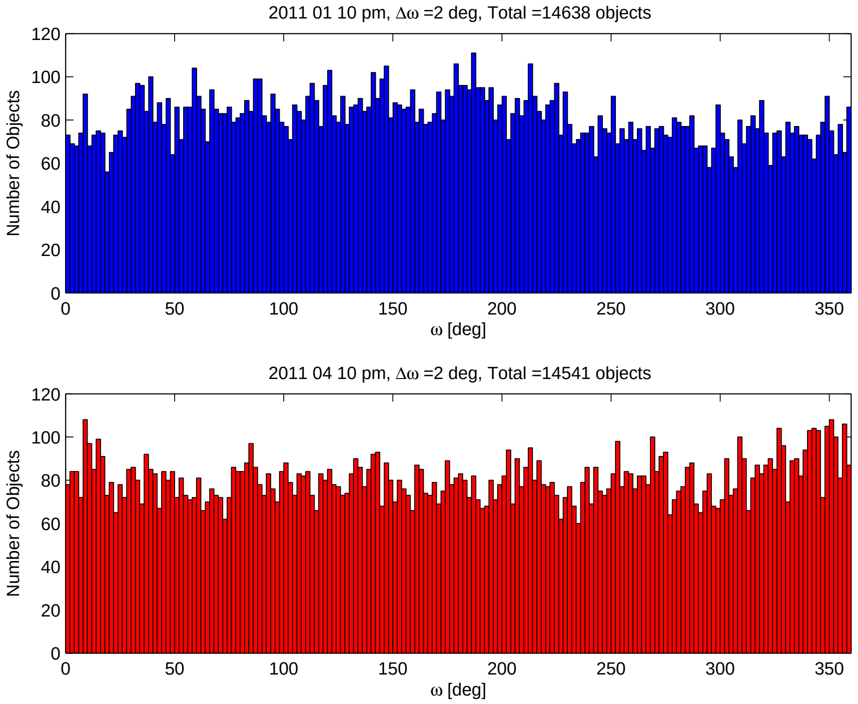


Figure 6.7: Histogram of the argument of perigee ω.

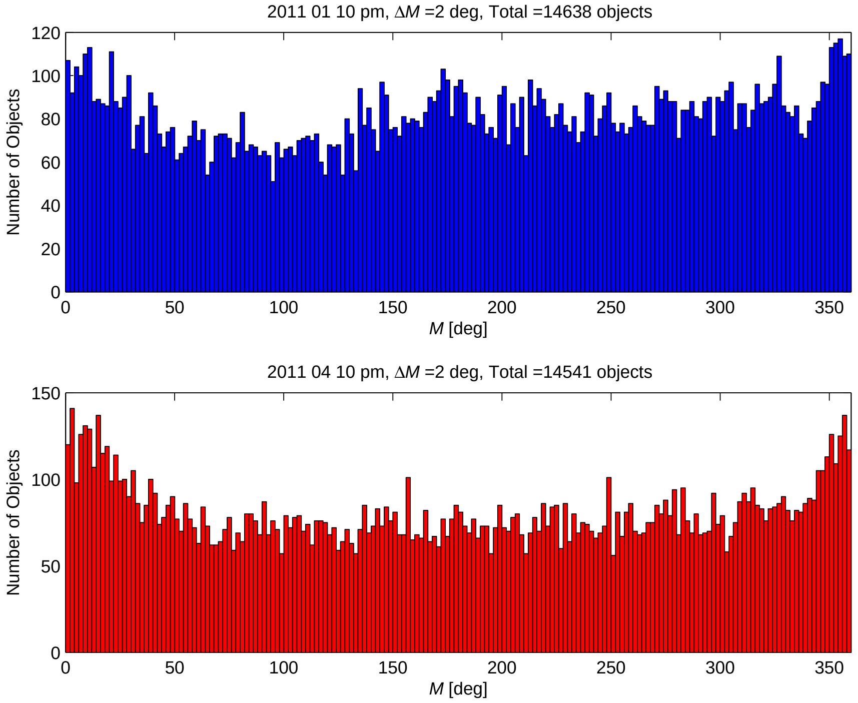


Figure 6.8: Histogram of the mean anomaly M.

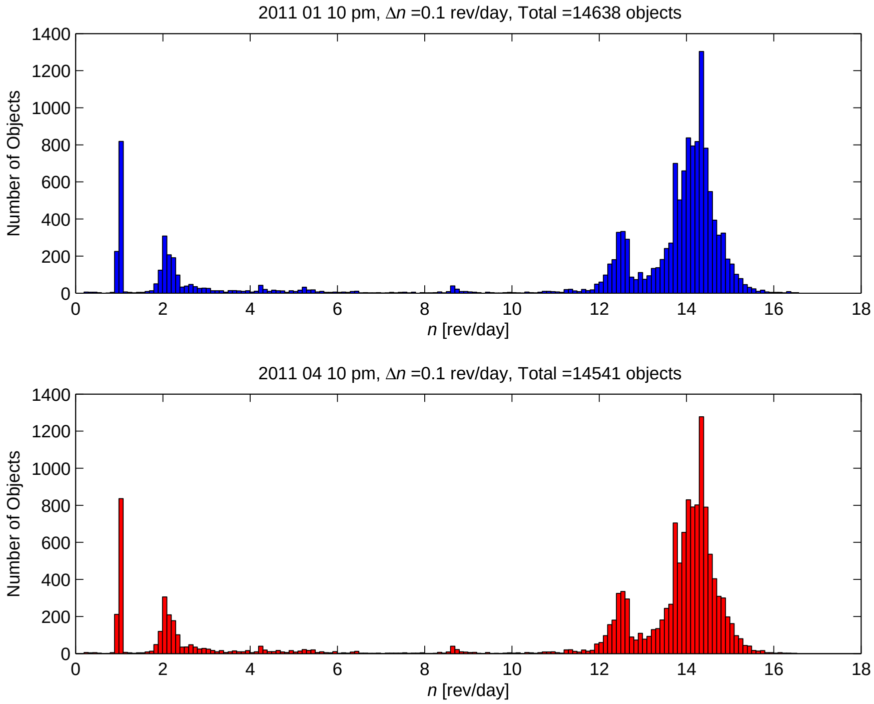


Figure 6.9: Histogram of the mean motion n.

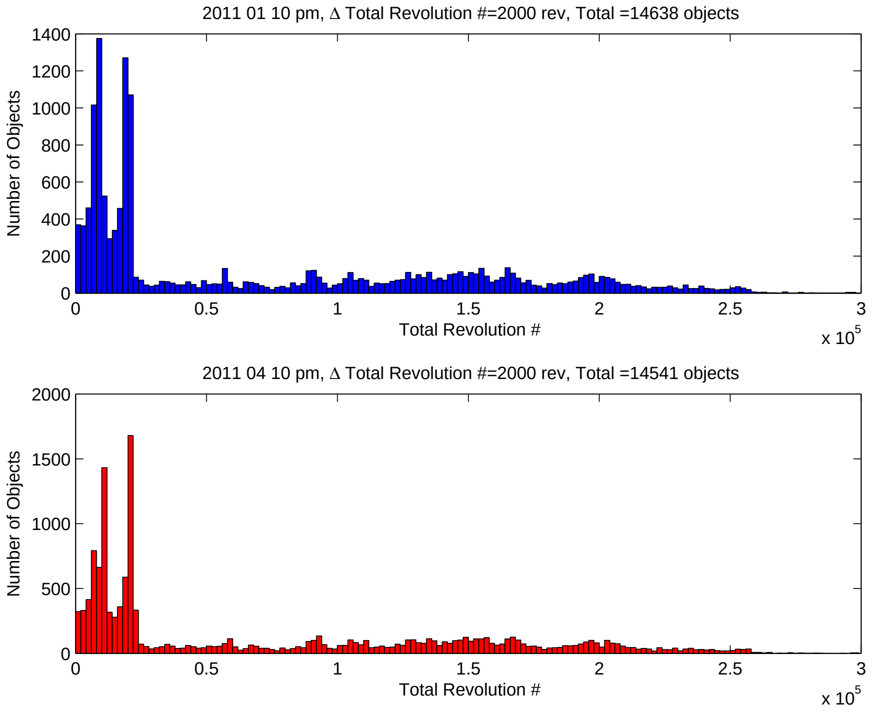


Figure 6.10: Histogram of the total revolution number.

The same story holds for the orbital model and classification which were always ‘0’ and ‘U’ respectively.

The modulo-10 checksum showed a more or less constant distribution over the numbers 0 to 9, which is to be expected after little less than 15k objects are taken into account.

The object ID number gave scattered results, although the number of objects seems to increase somewhat linearly with time. This is to be expected since older missions (smaller object ID numbers) have had more time to burn up into the atmosphere and thus disappear from the catalog.

Some large peaks and troughs can be seen in this particular histogram as well, maybe because some in-orbit breakup events caused a large number of new objects to be discovered in serial order, thus creating a peak or a trough when they all have burned up in the atmosphere.

The histogram for the launch year shows two very large peaks in 1993 and 1999, the launch year of TLE names ‘COSMOS 2251’ and ‘FENGYUN 1C’ respectively. These two satellites were the source of very large clouds of debris, as explained earlier.

Some smaller peaks in the 1960s to 1980s might indicate some very productive years in the cold war. The same debris cloud peaks can be seen in the launch number graphs.

For increasing TLE number, the number of objects decreases linearly, since more new objects with less TLEs remain, as older objects with more TLEs had more time to disappear into the atmosphere.

Unfortunately, it was not possible to determine the total TLE number for an object from the looped (99999 → 1) version, like done for the revolution number. To do this, one would need to spit through the entire database of catalogs for every object since its creation, and count the number of TLEs, which is impossible without access to this entire database.

6.4.2 TLE Object Variable Plots

The inclination i is plotted versus the eccentricity e in Figure 6.11, in which each dot represents one object, and the different orbital regions are coloured.

The GEO region in the lower left corner can be clearly seen to have very low inclination and eccentricity values, which was to be expected.

The LEO objects hold low eccentricity values (if their eccentricity would be larger, they would decay to circular orbits), and have various inclinations. The groups at certain inclination bands were explained earlier with the inclination histogram.

Furthermore, the GTO region can be clearly seen to have the same ec-

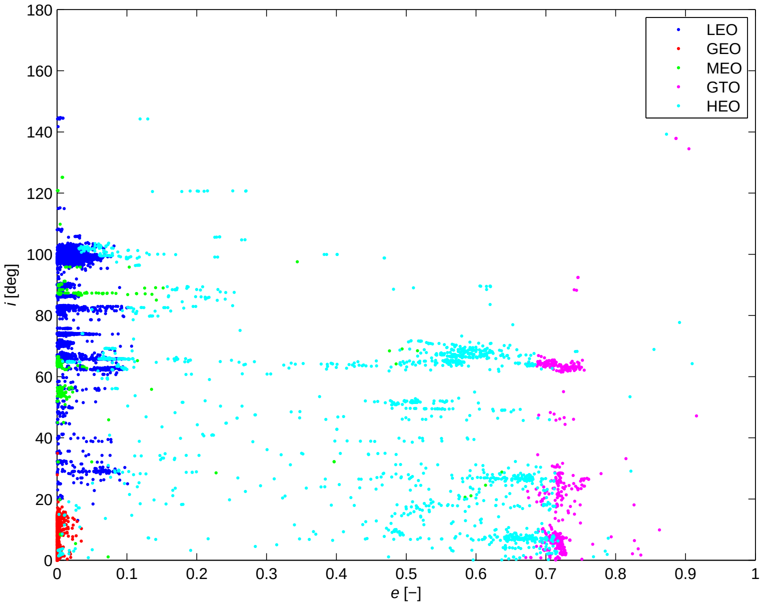


Figure 6.11: Scatter plot of inclination i versus eccentricity e, for the catalog of 2011.04.10. Each dot represents one object, coloured per orbital region.

centricity, of which the group with the higher inclinations represents the Molniya orbits.

In Figure 6.12 one can see the eccentricity e versus the semi-major axis a. The graph has an upper limit, which can be explained to be the limit of eccentricity that an object can have for a certain semi-major axis, or else it would decay into the atmosphere.

The LEO, GTO and GEO objects can be clearly seen in their semi-major-axis bands, with corresponding eccentricities.

Figure 6.13 shows the inclination i versus semi-major axis a for all objects, showing some similar results as explained previously. Interesting to note are the groups of MEO objects within a narrow band of semi-major axis and inclination, belonging to the constellations.

The apogee radius r_a is plotted versus the perigee radius r_p in Figure 6.14, which has been zoomed in excluding a few 'wanderers' to present the bulk more clearly in Figure 6.15. These wanderers can be seen to have very large apogee values. Which satellites this are exactly, will be shown later.

In Figure 6.15, the black lines represent the borders between orbital regions, defined in Table 4.1. The objects belonging to a certain orbital region

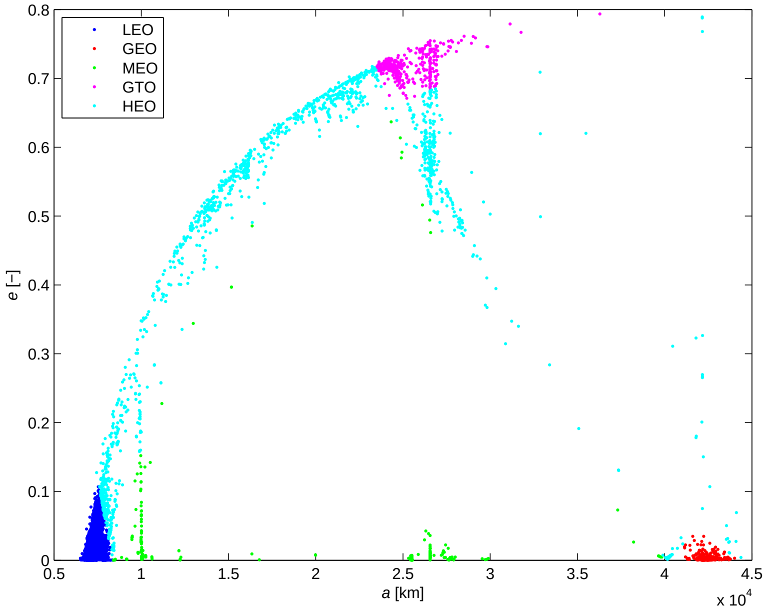


Figure 6.12: Scatter plot of eccentricity e versus semi-major axis a, for the catalog of 2011.04.10. Each dot represents one object, coloured per orbital region.

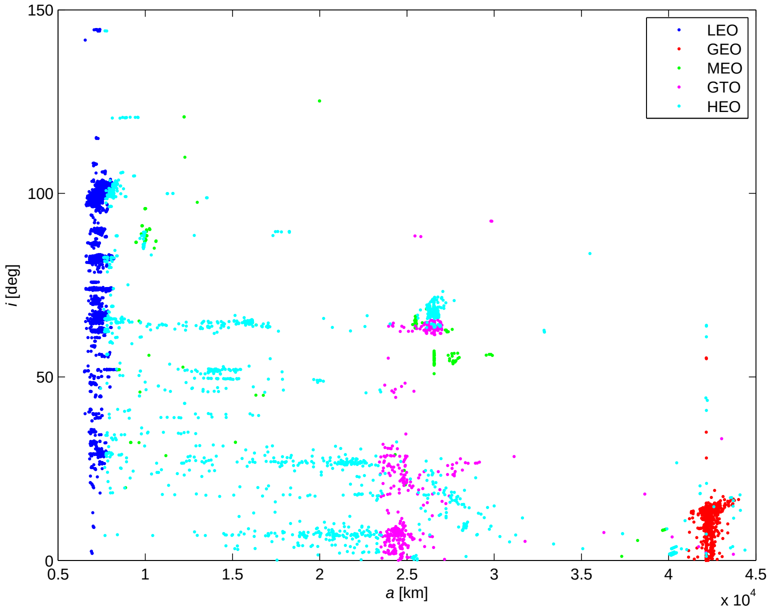


Figure 6.13: Scatter plot of inclination i versus semi-major axis a, for the catalog of 2011.04.10. Each dot represents one object, coloured per orbital region.

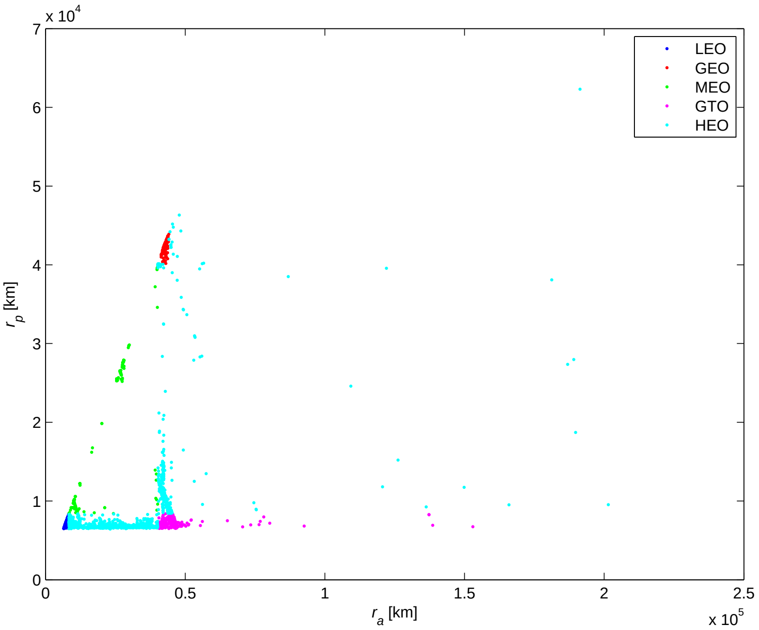


Figure 6.14: Scatter plot of apogee radius r_a versus perigee radius r_p, for the catalog of 2011.04.10, each dot represents one object, colored per orbital region.

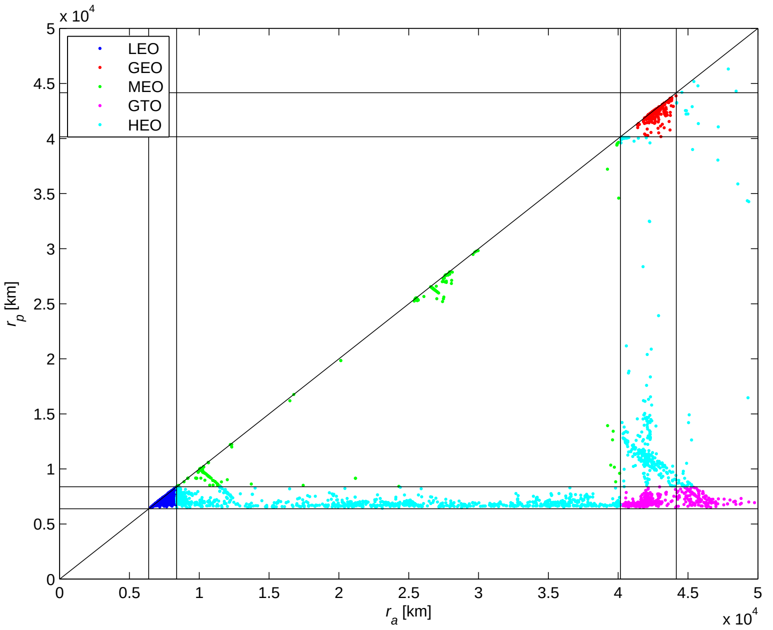


Figure 6.15: Zoomed in version of Figure 6.14, with added orbital region boundary lines.

can be clearly seen to be fitted in between these lines. An upper border is defined here as well, representing the line above which the apogee of an object would turn into a perigee, positioning itself underneath the line again.

For the MEO objects, there are some short lines of objects visible, which could be more clearly seen when zooming in further. These lines represent constellations of satellites, with a constant perigee over apogee ratio, such that they have an equal orbital period.

6.4.3 Fragment and Source Counts

A function was implemented which counts the current amount of fragmented objects in the catalog belonging to one original payload. It does this by looking at both the objects' name, which is equal for different parts of debris belonging to one original payload, as well as the combination of the objects' launch year and number. This way, it can be determined if multiple objects originate from the same launched payload, since some objects have the same name while not originating from the same payload.

The resulting list is sorted in descending order, of which the top-10 can be seen in Table 6.2, in which the results for the two different analysed catalogs have been combined for comparison.

It can be seen that the amount of debris objects for some object sources decreased in these three months. This mostly seems to be the case for the ASAT test and collision event treated earlier (top-3). Since these events happened recently and the clouds are relatively large, the objects of these clouds are more likely to decay faster into the Earth's atmosphere.

Most of the amounts of the other sources seem to stay fairly constant. Another reason could be that the SSN made less observations in the latter period, or stopped observing some objects.

A similar function was implemented which counts the amount of objects with the tag 'R/B' or 'DEB' in their name, corresponding to rocket body or (fragmented) debris objects. Table 6.3 lists the results for the two catalogs.

It can be seen that the major part of the catalog is filled with debris objects, while there are a large amount of rocket bodies as well. The latter actually increased, while mainly debris objects disappeared from the catalog.

The other objects in the list are mainly (non-)active payloads.

There are some other tags as well, like '(YO)' for yo-yo de-spin devices and '(metal obj)' for general metal objects.

However, the abundance of these tag name categories compared to their marginal appearance made the implementation of all these tags too tedious. If one is interested in any of these sources as well, the function could easily be extended.

Table 6.2: Count in descending order of number of fragments per object name & launch year/number.

launch year	no.	TLE object name	2011.01.10 #	2011.04.10 #
1999	25	FENGYUN 1C DEB	2791	2761
1993	36	COSMOS 2251 DEB	1313	1307
1997	51	IRIDIUM 33 DEB	462	457
1981	53	COSMOS 1275 DEB	251	251
1970	25	THORAD AGENA D DEB	243	243
1975	52	DELTA 1 DEB	207	207
1999	57	CZ-4 DEB	186	184
1961	15	THOR ABLESTAR DEB	178	175
1973	86	DELTA 1 DEB	173	173
1992	93	SL-16 DEB	169	169

Table 6.3: Count of debris, rocket body or other objects.

	2011.01.10	2011.04.10
Total number of objects	14638	14541
Debris (DEB) objects	9807	9723
Rocket Bodies (R/B)	1083	1085
All other objects	3748	3733

Furthermore, the amount of objects within a certain orbital region was counted as well, see Table 6.4. As could already be seen from previous analysis, most objects are in the LEO region, followed by the HEO and GEO regions.

The amount of GEO objects increased, while all other regions decreased. For most regions the decrease could be explained by atmospheric decay, except for the MEO region. These MEO objects could have been switched to another region however.

To check the evolution of the catalog, a current version was downloaded and processed. It showed even less objects than before, proving the decreasing trend.

Table 6.4: Count of different orbital regime objects.

	2011.01.10	2011.04.10
Total number of objects	14638	14541
LEO objects	11358	11281
MEO objects	353	350
GEO objects	984	986
GTO objects	482	476
HEO objects	1461	1448
All other objects	0	0

6.5 Sorting

A function was implemented to sort the catalog with respect to a certain variable, of which an example for the apogee radius r_a, sorted descendingly, can be seen in Table 6.5, for the catalog of 2011.01.10.pm. The top-10 is listed along with some other relevant variables, like object name (as in their TLE), object ID, launch year (l.y.), apogee and perigee radius (r_a and r_p), semi-major axis a, eccentricity e and inclination i.

The record holder, 'PROGNOZ 6', is the only one out of 10 original Prognoz satellites (launched by Russia in the 1970s and 1980s) still in the catalog, and its mission was to study solar radiation and plasma, and the Earth's magnetosphere (Astronautica, 2012).

In the catalog of three months later, its perigee radius already decreased by over 400 kilometres, indicating that it might decay into the Earth atmosphere soon. Its apogee radius also went up by more than 400 km though, most probably due to the gravitational attraction of the Moon (and partly also the Sun), as the object is already more than halfway to the Moon when located in the apogee of its orbit!

In the catalog of three months later, the object 'ATLAS AGENA D R/B' already disappeared, which after inspection could be seen to have decayed on the 18th of January 2011 (Kelso, 2010a).

Table 6.5: Top-10 of objects with largest apogee radius, along with some other relevant variables.

Object Name	ID	l.y.	r_a [km]	r_p [km]	a [km]	e [-]	i [deg]
PROGNOZ 6	10370	1977	201048	9951	105499	0.91	61.03
GEOTAIL	22049	1992	191782	61455	126619	0.51	38.86
SL-6 R/B(2)	32705	1977	187669	20732	104200	0.80	52.82
ASTRON	13901	1983	187184	29847	108515	0.72	68.92
SL-12 R/B	20413	1983	185602	28839	107220	0.73	68.45
OPS 3662 (VELA 3)	836	1964	183873	35501	109687	0.68	75.51
INTEGRAL	27540	2002	165219	10217	87718	0.88	79.56
OGO 5	3138	1968	152383	7422	79903	0.91	45.94
ATLAS AGENA D R/B	3145	1968	152258	6598	79428	0.92	47.57
CXO	25867	1999	149111	12462	80786	0.85	68.28

6.6 TLE Code Implementation

The code which was written to perform the above analysis will now be explained, along with some functionalities which were not mentioned yet. The programming language used is C++, and most of the functions have been implemented in Tudat as well.

A flow diagram of the processes and data streams in the TLE input process, and the functions of the code, can be seen in Figure 6.16. The flow diagrams shown here, as well as subsequently in this report, are not standard (C++) coding diagrams, as the level of detail is not that deep.

These 'official' flow diagram formats exist to show the process of the actual code in detail, containing every for-loop and if-structure. The purpose here is moreover to give a general macro overview of how the code was setup at the highest level, showing the largest classes, functions and data structures only.

The process starts with the measurements made by the SSN, which are converted to TLE data and made available on SpaceTrack. This data is then downloaded and processed by the TLERetriever of CelesTrak, like explained before.

The TLE data is saved in text files and are then read in by Tudat C++ code, which starts with the file reader class. The text file is opened and its lines are stored as strings by the text file reader class. Lines can be skipped if needed, and lines with a specific starting character can be ignored as well.

The TLE text file reader class then converts the strings into the TLE variable types, according to the format description of Chapter 4. These variables are stored in the TLE data class, of which one object per TLE is made.

When all the data is converted, the variables can be analysed to check for file integrity. To see if the the TLEs are valid, the checks, which were described above, are performed, i.e. certain variables always have certain values.

When all TLEs are checked to be valid, and the invalid TLEs are deleted, some extra variables can be calculated with the TLE data, as explained above. These new variables are also stored in the TLE data object. In this calculation process, some WGS constants are used, which are defined in the WGS constants class, see also Table 6.1.

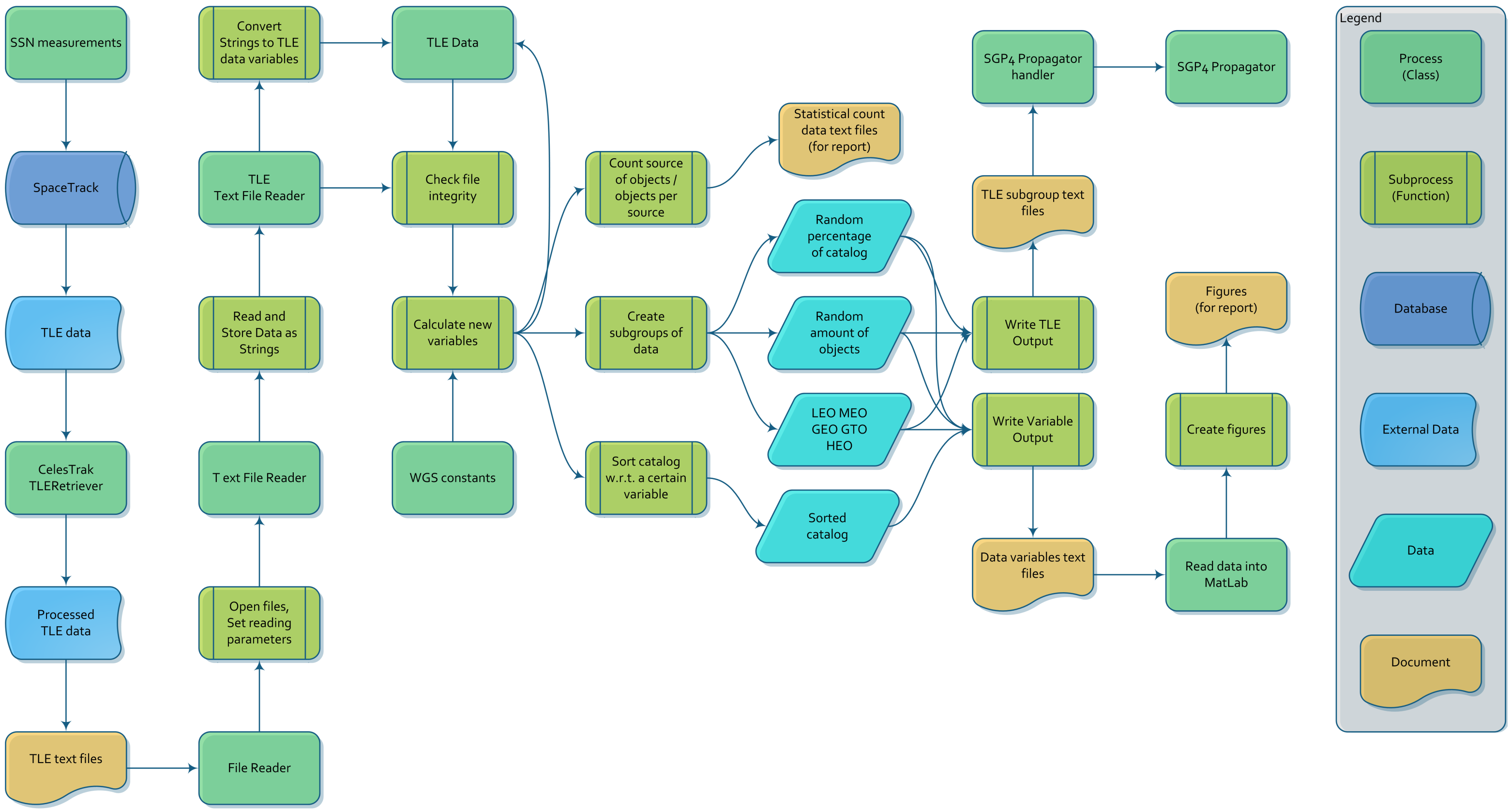


Figure 6.16: Flow diagram of the TLE input process. See legend for the explanation of the block types. The first 3 blocks are performed by external institutions, while the fourth and fifth are performed using an external program.

The TLE objects can now be divided into subgroups. A percentage of the catalog can be created, with randomly picked objects, so as to be a representative smaller catalog. This can also be done by specifying the amount of objects for the new randomly chosen subgroup.

Furthermore, the catalog can be subdivided into the different orbital regimes like defined in Chapter 4. This leads to new groups of objects which can later on be written to separate files again.

Other options include a sort function which can sort the entire catalog with respect to a certain variable in ascending/descending order. This sorted catalog can then also be written to a text file.

The number of objects originating from one launch and carrying the same name (so originating from the same initial body) can be counted and be written to a text file in ascending/descending order. The number of objects carrying the tag ‘R/B’ (Rocket Body) or ‘DEB’ (Debris) can also be counted and written to a text file as output.

There are two functions which can write the entire catalog or a subgroup to a text file, respectively. One writes the data as the original TLEs which can be used by the SGP4 propagator. The other writes certain specified converted variables in tabular form according to a specified order, which can for instance be read into Matlab to make statistical figures.

A copy of the code including an explanation of all functions can be downloaded from the repository of the Tudat project, see Tudat website (AS, 2011).

Chapter 7

SGP4 Propagator Implementation

Now that the TLE reader is completed, the information contained in the TLEs can be read and used by the SGP4 propagator.

There exist multiple versions of this propagator on the internet, most of which are made by individuals, but all stemming from the original code published in Hoots & Roehrich (1980). This code was described in a more clear way in Hoots et al. (2004), in which a history of the evolution of the analytical orbit modelling in the U.S. Space Surveillance System was treated.

An improved version of this original SGP4/SDP4 code was published in Vallado et al. (2006), which was now combinedly called SGP4. This improved version was not published by USSTRATCOM (which generates the TLEs), but was rather made through the combined efforts of the public scientific community, who tried their best to find possible improvements to the original code, and thought about changes that most probably would have been implemented by now. The results of these improvements were verified to be correct.

In Miura (2009) it was even concluded, that NASA JSC should implement this new revised version of the code, as this would lead to more accurate results by orders of magnitude over longer periods of time. The revised code was also compared to SATRAK, a 'black-box' program released to selected agencies, and led to similar positive results.

In conclusion, the revised SGP4 code described in Vallado et al. (2006) is currently believed to be the best available way to propagate the objects of the TLE catalog. This code is therefore used in this thesis.

7.1 SGP4 Propagator Code Implementation

The revised SGP4 code was released in a number of languages, of which one is claimed to be C++, but this is actually original C code. This C version is used to allow for a proper integration in Tudat (which is programmed in C++), and the rest of the code written for this thesis.

An SGP4 handler class is implemented which handles the input and output of the SGP4 propagator.

7.1.1 Tudat SGP4 Propagator Handler

A flow diagram of the data and processes of the SGP4 propagator and its code structure can be seen in Figure 7.1. The diagram starts where the TLE reader diagram, shown in Chapter 6, left off at the SGP4 propagator block.

The TLE data that was read by the TLE reader, might have been transformed into subgroups, depending on the application. Either these subgroup TLE data files, an entire TLE catalog file, or even a file with only containing one TLE can be used as an input for the SGP4 propagator. Along with the name and location of this input file some other input data are needed (which can be set in any order).

First of all, the operations mode needs to be chosen, which can be set by one of the following characters:

'a' = AFSPC is the best understanding of how the AFSPC version of the SGP4 code works, which was also used in verifying the revised version in Vallado et al. (2006).

'i' = improved is the revised SGP4 version of the code resulting in smoother behavior and improved results.

Then the type of run needs to be indicated, which can be picked by making use of one of the following characters:

'm' = manual is the most general manual running mode, for which the time interval can be set manually to three different modes which will be described later: minutes from epoch, year and day of year, and date and time. This mode is set as default.

'v' = verification is the verification run which uses a special modified TLE file that comes with the release of the code, and is used to verify if the SGP4 code is working. The resulting ephemerides data can be compared with the data which is listed in Vallado et al. (2006).

'c' = compare is a running mode which is described as comparing 1 year of full satellite catalog data. However, when testing this mode, the result

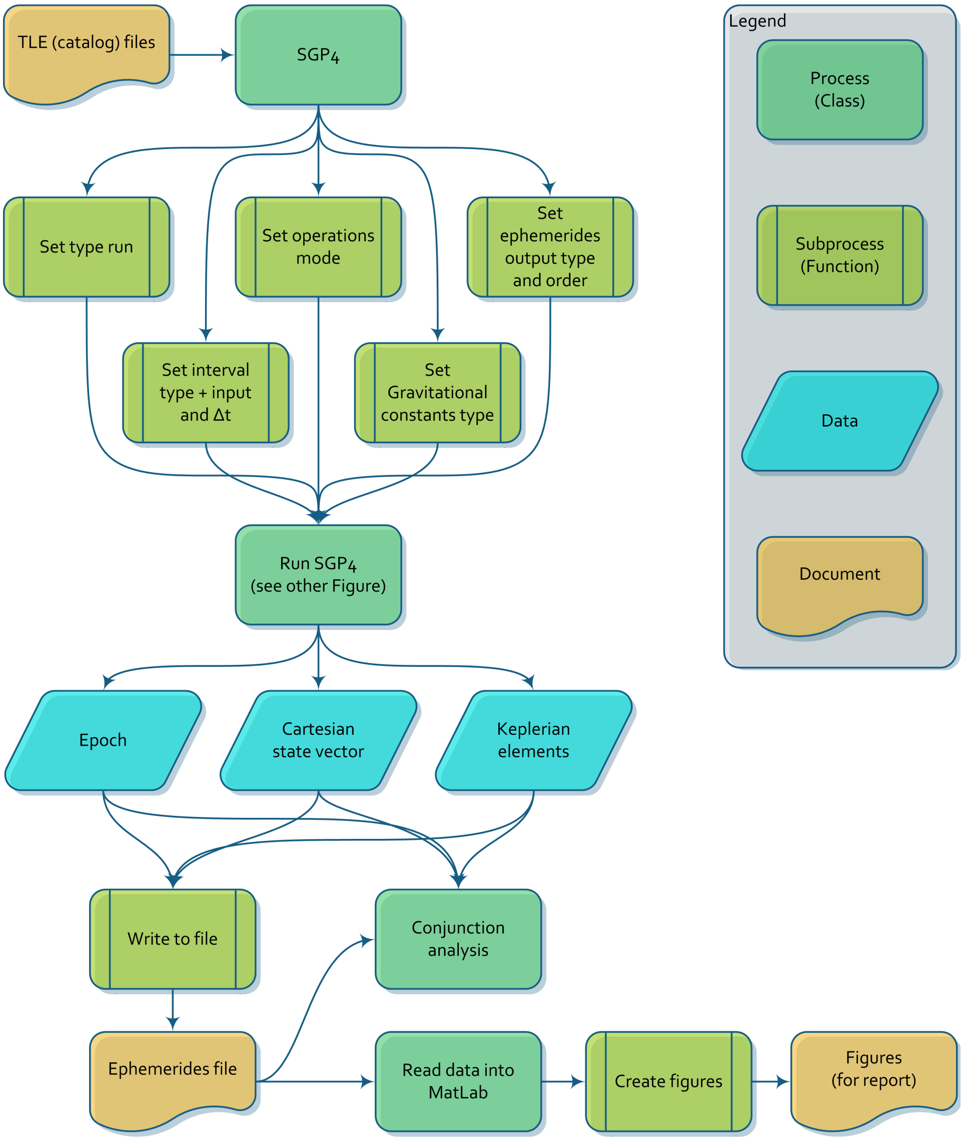


Figure 7.1: Flow diagram of the SGP4 input/output process. See legend for the explanation of the block types.

was that for each TLE used as input, ephemerides were generated for a time interval of -1440 to 1440 minutes from epoch with a time step of 10 minutes, which can thus just as easily be done with the manual mode.

When choosing manual input, one can choose the type of input and the analysis interval through one the following functions:

Minutes From Epoch sets the analysis interval with two doubles, one for the starting time and end time respectively, counted as minutes from the epoch of the TLE.

Date and Time sets the interval using two times five consecutive integers and one double, containing the starting and stopping year, month, day, hour, minute and second, respectively.

Year and Day sets the interval using a starting and stopping year integer and day of that year as a double.

Furthermore, the time step Δt in minutes can be set as well, using a double. One can also choose which set of constants is used, by setting one of the following:

wgs72old are the WGS72 constants listed in the original description of the SGP4/SDP4 code (Hoots & Roehrich, 1980). They are of low-precision however.

wgs72 are the WGS72 constants with higher precision, variables are derived from one another here. Since this set of values was used in the newer publication, this is the default set of constants used. Most external programs also make use of this set, so this will create less difficulties when any two different programs are mixed.

wgs84 sets the newer WGS84 constants to be used. It might be possible that USSTRATCOM uses these constants nowadays, but this has never been published. The accuracy of the results of the SGP4 propagator will not really be impacted, so these constants only need to be set when using external programs which make use of this set as well.

Finally, the type and order of ephemerides output combined with their epoch can be set, either through one of the following sets of variables, or all of the variables individually:

Epoch lists the epoch of each ephemeris either in minutes from epoch, Julian date or ‘date time’ format as described above.

Cartesian State Vector lists the Cartesian coordinates x, y, z and their time derivatives $\dot{x}, \dot{y}, \dot{z}$ at each time instance in the interval. They are defined in the ECI TEME reference frame described in Section 4.4.

Keplerian Elements Lists any of the following Keplerian elements (with their SGP4 variable name between brackets) of the ephemeris: semi-latus rectum (p), semi-major axis (a), eccentricity (ecc), inclination (incl), longitude of ascending node (omega), argument of perigee (argp), true anomaly (nu), mean anomaly (m), argument of latitude (arglat), true longitude (truelon) and longitude of periapsis (longper).

After all of the above described input settings have been set, the SGP4 propagator can be run. It will then produce an output file containing the set output variables like epoch, state-vector and/or orbital parameters. For some select functions (for example: propagating only one object to one time instance) this data will be stored in an array/vector in the computer's memory and will be used in the conjunction analysis process.

7.1.2 Tudat External SGP4 Propagator Code

A flow diagram containing the code structure of the external SGP4 propagator can be seen in Figure 7.2. The code consists of the following three files containing functions which are called from the SGP4 propagator handler, or each other:

SGP4Ext contains general mathematical functions, as well as a Kepler's equation solver, conversion from Cartesian to Kepler elements as well as date/time, Julian date and year / day of the year conversions.

SGP4IO comprises the reading, converting and saving of the variables in the TLE input file, as well as converting the different possible input intervals to Julian date.

SGP4Unit holds all functions originally listed in Hoots & Roehrich (1980), which are now improved. A description of these functions follows below.

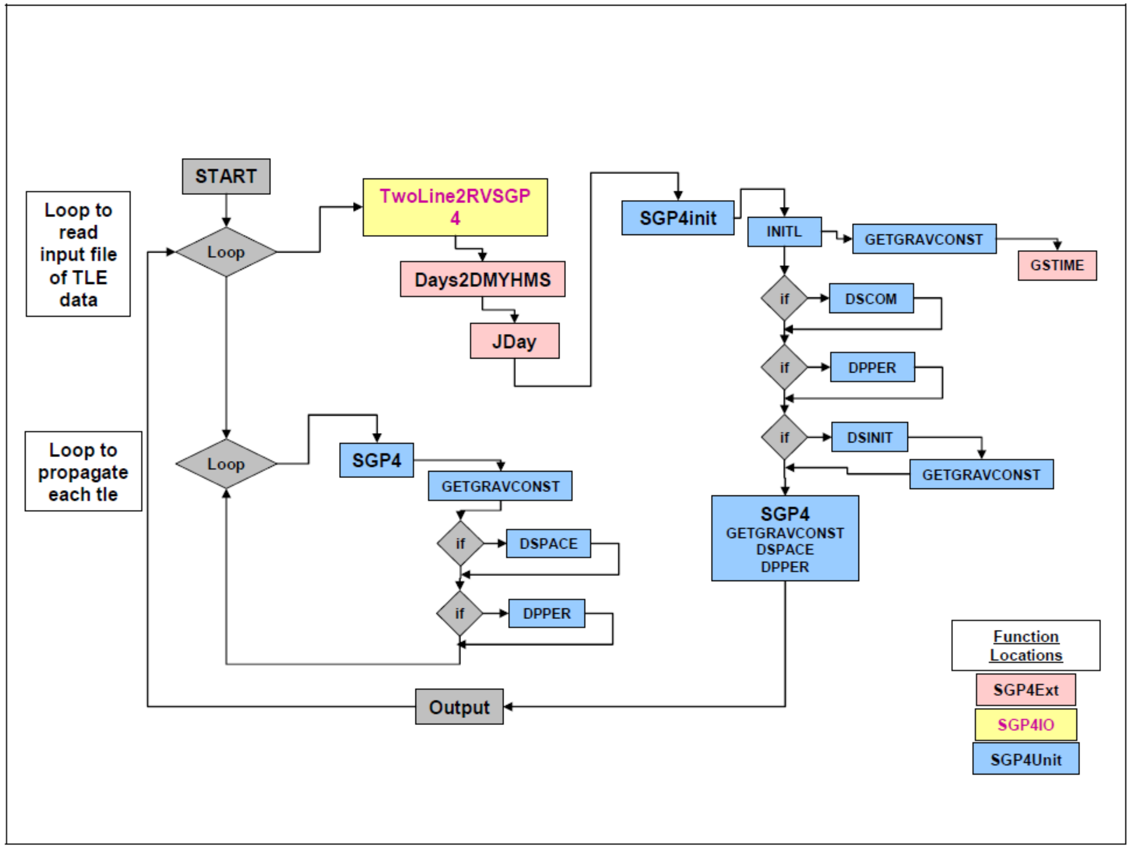


Figure 7.2: Flow diagram containing the code structure of the SGP4 propagator from Vallado et al. (2006).

The functions of the analytical SGP4 propagator described in Vallado et al. (2006) are:

dpper determines deep space long-period periodic contributions to the mean elements using a recurring function.

dscom provides deep space common items used by both the secular and periodic subroutines, and is a reorganisation of the original 'dpper' function.

dsinit calculates deep space contributions to the derivative of the mean motion, due to geopotential resonance.

dspace analyses deep space contributions to the mean elements for a perturbing third body, averaged over one revolution of the Sun and Moon. For Earth resonance effects, the effects have been averaged over a number of revolutions of the satellite, using the mean motion.

initl initializes the SGP4 propagator. All the initialization is done here instead of having multiple loops inside other routines.

sgp4init initializes variables for SGP4.

sgp4 is the sgp4 prediction model from AFSPC.

gstime finds the Greenwich sidereal time.

getgravconst gets the constants for the propagator as chosen by the user. The common usage is WGS72.

The code works as follows, following Figure 7.2.

When the code is started up in the outer loop, it reads 1 TLE, converts the input interval to Julian date, and initialises the variables for SGP4 and the propagator. Furthermore, it receives the constants used, and finds the Greenwich sidereal time for that epoch.

Then it checks if the object is defined as 'deep-space' (period larger than 225 minutes) or as 'near-Earth', and applies the deep space routines only if needed. The code then continues in the inner loop to determine all ephemerides for that object. When the entire interval has been analysed, the inner loop is completed and the outer loop is entered again, to start with the next TLE.

Since for each TLE the SGP4 propagator is now only initialised once, using the new inner and outer loops, it is 10% faster than the old way described in Hoots & Roehrich (1980) in which the initialisation was integrated in only one loop.

Applying the SGP4 propagator to the entire catalog file, resulted in so-called error codes 1 and 6 for some 20 objects, which were removed from subsequent analyses. Error number 1 occurs when the eccentricity somehow gets below 0.0 or above 1.0 during propagation, while error number 6 occurs when an object is flagged as decaying.

Furthermore, any not-flagged decaying objects negatively influencing the results of analyses done in this report were removed from the catalog as well.

7.2 SGP4 Propagator Verification

To verify the proper implementation of the code, and to verify the validity of the results of the SGP4 propagator, the results of three reference documents were reproduced. As a start, the verification run described before was carried out, for which the results are listed in Hoots & Roehrich (1980), and these were verified to be correct.

This document also contained seven figures with plotted results in terms of Keplerian or Cartesian elements of the SGP4 propagator. These figures were reproduced, of which two can be seen in Figures 7.4 and 7.6, with the originals in Figures 7.3 and 7.5, respectively.

Unfortunately, a working copy of the original code of Hoots & Roehrich (1980) is not at hand, so the results of the old propagator indicated here as 'STR#3', could not be reproduced. However, after verification of all figures, it is proven that the produced ephemerides are correct, both as Keplerian and as Cartesian elements.

Furthermore, at the end of Hoots & Roehrich (1980) one tabular example is given of each of the orbital models. The original tables of the SGP4 and SDP4 orbital models can be seen in Tables 7.1 and 7.3. The TLE which was used to produce these tables was listed as well, so the tables could be reproduced with the newly implemented code, as can be seen in Tables 7.2 and 7.2.

The results can be seen to match from five up to eight significant numbers. The difference is due to the improvements made in the newer version of the code.

Finally, in Woodburn et al. (2009), a graph of plotted results with its corresponding TLE can be found as well, see Figure 7.7. In this figure the radius of the object is plotted as well as its osculating and mean perigee and apogee.

The osculating perigee and apogee were taken from the Keplerian elements ephemerides resulting from the SGP4 propagator, while the radius was obtained by taking the square root of the squared Cartesian coordinates.

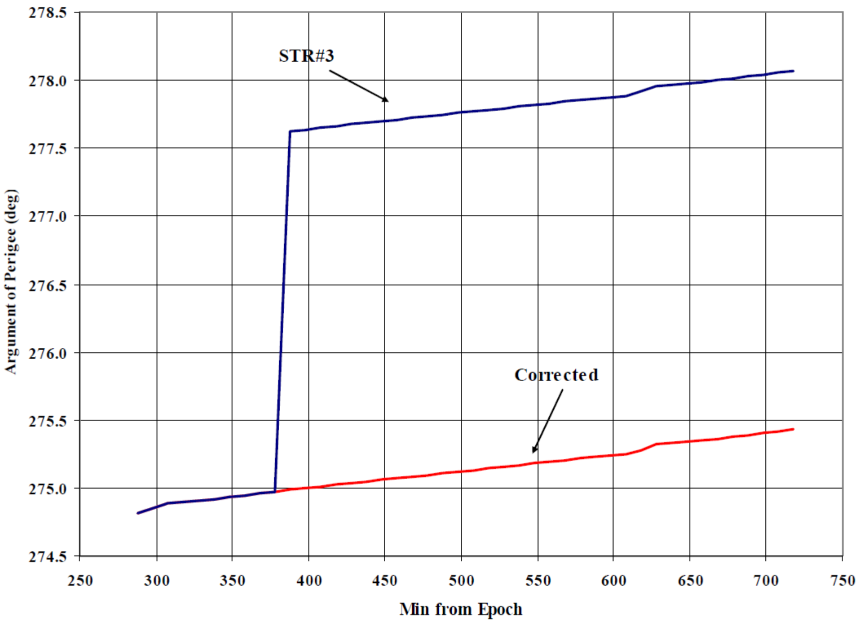


Figure 7.3: Original Figure 2 (top) from Vallado et al. (2006), object 23599.

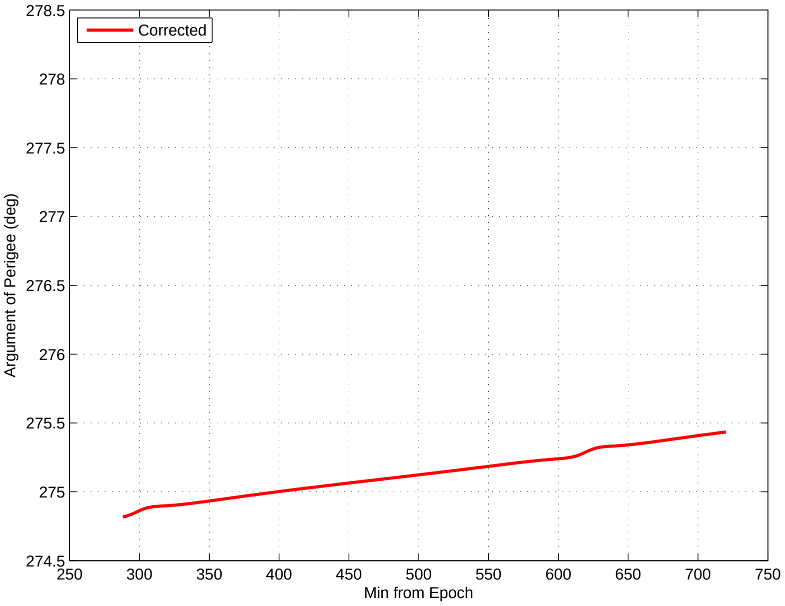


Figure 7.4: Recreated Figure 2 (top) from Vallado et al. (2006), object 23599.

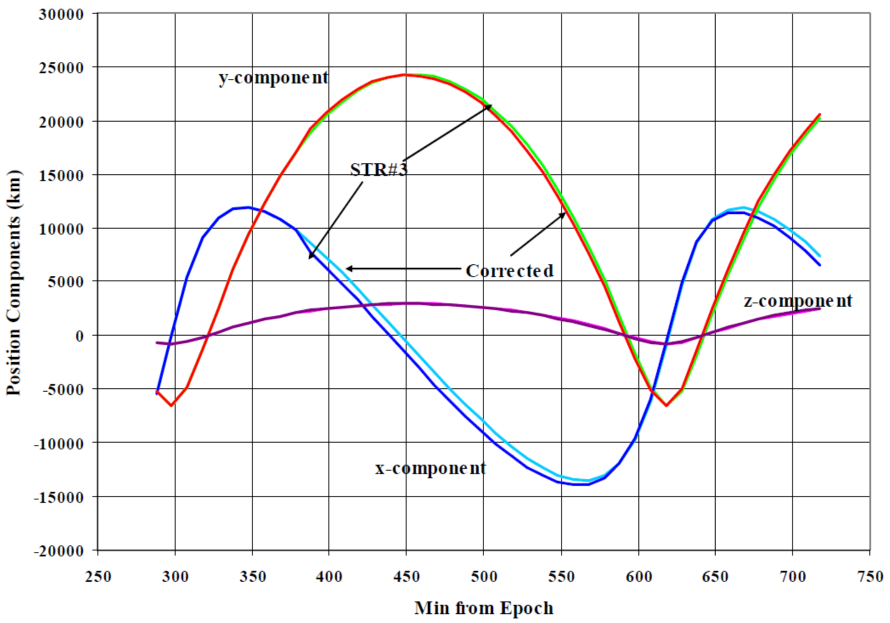


Figure 7.5: Original Figure 2 (bottom) from Vallado et al. (2006), object 23599.

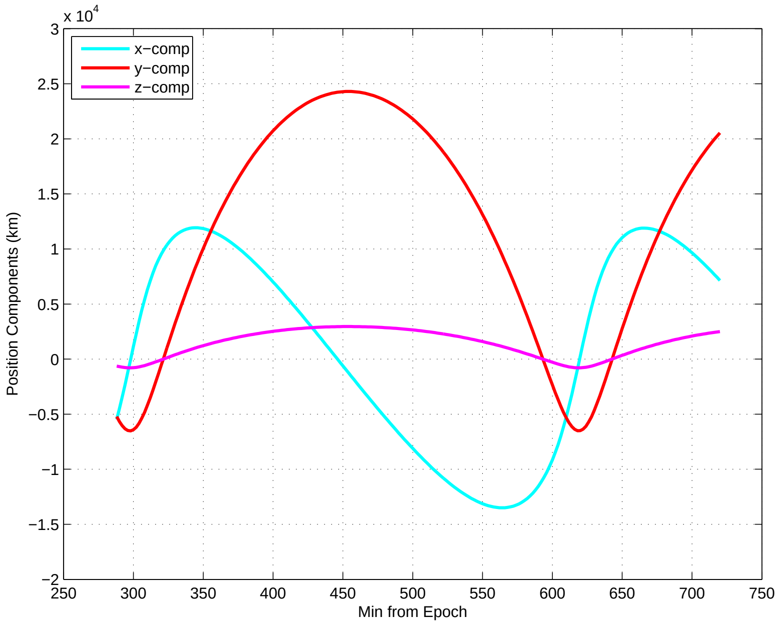
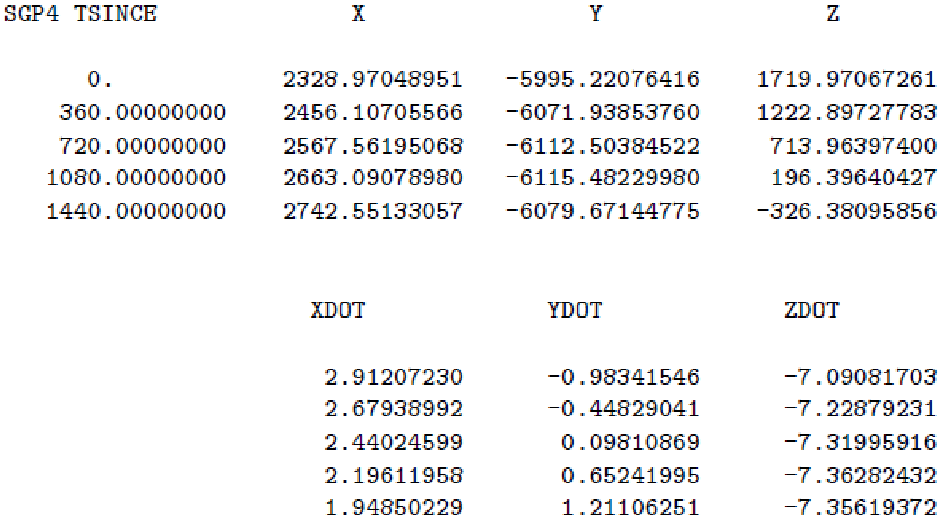


Figure 7.6: Recreated Figure 2 (bottom) from Vallado et al. (2006), object 23599.

Table 7.1: Original SGP4 table from Page 82 of Hoots & Roehrich (1980).

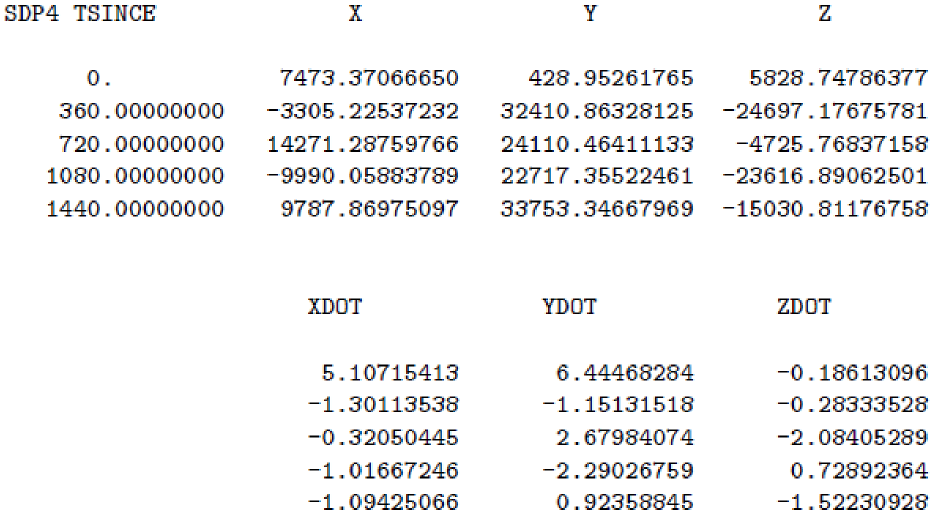


```
SGP4 TSINCE              X                Y                Z

      0.           2328.97048951   -5995.22076416    1719.97067261
    360.00000000   2456.10705566   -6071.93853760    1222.89727783
    720.00000000   2567.56195068   -6112.50384522     713.96397400
   1080.00000000   2663.09078980   -6115.48229980     196.39640427
   1440.00000000   2742.55133057   -6079.67144775    -326.38095856


                      XDOT             YDOT             ZDOT

                     2.91207230      -0.98341546      -7.09081703
                     2.67938992      -0.44829041      -7.22879231
                     2.44024599       0.09810869      -7.31995916
                     2.19611958       0.65241995      -7.36282432
                     1.94850229       1.21106251      -7.35619372
```

Table 7.2: Recreated SGP4 table from Page 82 of Hoots & Roehrich (1980).

SDP4 TSINCE	X	Y	Z
0.00000000	2328.96975262	-5995.22051338	1719.97297192
360.00000000	2456.10706533	-6071.93855503	1222.89768554
720.00000000	2567.56229695	-6112.50383922	713.96374435
1080.00000000	2663.08964352	-6115.48290885	196.40072866
1440.00000000	2742.55398832	-6079.67009123	-326.39012649
	XDOT	YDOT	ZDOT
	2.91207328	-0.98341796	-7.09081621
	2.67939004	-0.44829081	-7.22879216
	2.44024575	0.09810900	-7.31995926
	2.19612156	0.65241509	-7.36282415
	1.94849765	1.21107268	-7.35619313

Table 7.3: Original SDP4 table from Page 83 of Hoots & Roehrich (1980).



```
SDP4 TSINCE              X               Y                Z

     0.            7473.37066650    428.95261765    5828.74786377
   360.00000000   -3305.22537232  32410.86328125  -24697.17675781
   720.00000000   14271.28759766  24110.46411133   -4725.76837158
  1080.00000000   -9990.05883789  22717.35522461  -23616.89062501
  1440.00000000    9787.86975097  33753.34667969  -15030.81176758


                     XDOT            YDOT             ZDOT

                    5.10715413      6.44468284      -0.18613096
                   -1.30113538     -1.15131518      -0.28333528
                   -0.32050445      2.67984074      -2.08405289
                   -1.01667246     -2.29026759       0.72892364
                   -1.09425066      0.92358845      -1.52230928
```

Table 7.4: Recreated SDP4 table from Page 83 of Hoots & Roehrich (1980).

SDP4 TSINCE	X	Y	Z
0.00000000	7473.37102491	428.94748312	5828.74846783
360.00000000	-3305.22148694	32410.84323331	-24697.16974954
720.00000000	14271.29083858	24110.44309009	-4725.76320143
1080.00000000	-9990.05800009	22717.34212448	-23616.88515553
1440.00000000	9787.87836256	33753.32249667	-15030.79874625
	XDOT	YDOT	ZDOT
	5.10715539	6.44468031	-0.18613330
	-1.30113732	-1.15131560	-0.28333582
	-0.32050453	2.67984154	-2.08405436
	-1.01667439	-2.29026798	0.72892334
	-1.09425155	0.92358991	-1.52231101

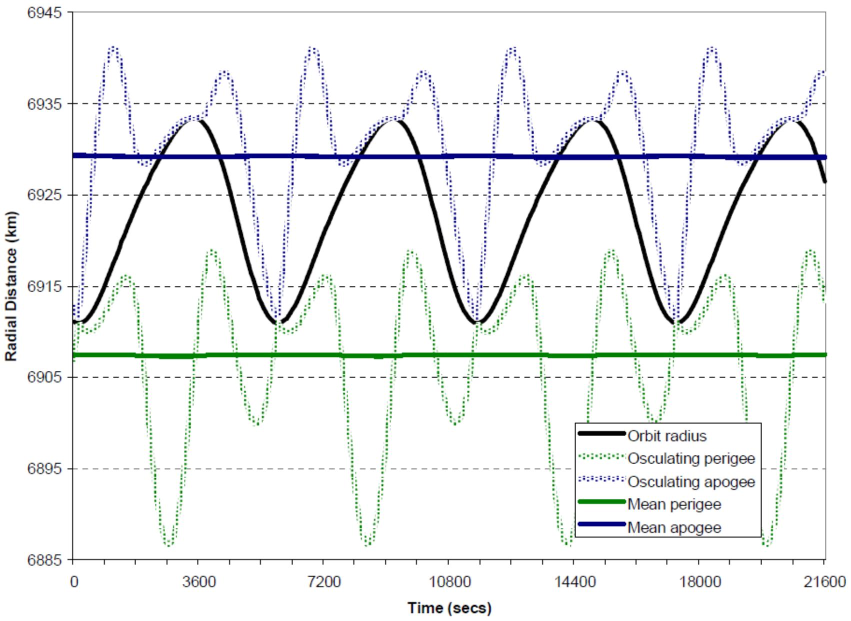


Figure 7.7: Original Figure 2 from Woodburn et al. (2009), object 26281.

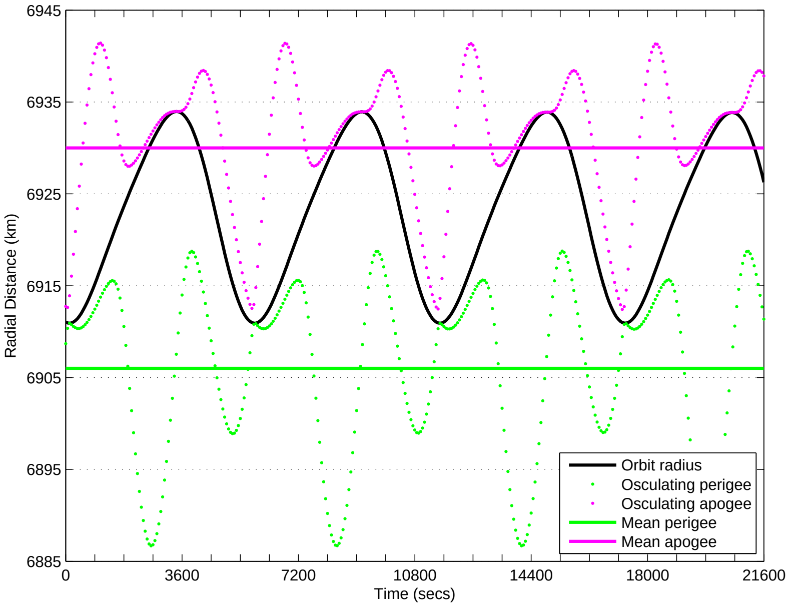


Figure 7.8: Recreated Figure 2 from Woodburn et al. (2009), object 26281.

Although it is not clear how the mean values were generated in the reference document, in the recreation the mean was taken as an average of the osculating elements over one orbital period. This seems to be close to the reference document, but not exactly (difference is approximately 1 km). The mean values directly generated from the TLE variables matched even less well.

7.3 Catalog Plots

When an entire catalog is taken as the TLE input source file, and it is propagated to only one moment in time, a 3D representation of the location of all objects in that catalog can be made, using the produced Cartesian components.

Exactly this was done for the catalog of 2011.05.05 am which contained 14,394 objects, and was propagated to 2011.05.05 12:00:00. See Figures 7.9 and 7.10 for a 3D plot of the location of these propagated objects with respect to the Earth, for the LEO region, viewed from the plane of the equator and the North pole axis respectively.

Furthermore, see Figures 7.11 and 7.12 for the same plot, but now for the GEO region, in which the geostationary ring of satellites is clearly visible. The LEO region can also here be seen to have the highest abundance of objects.

The same catalog was used again, but now each object was propagated to its TLE epoch. Propagation is still needed here, as the SGP4 propagator is used to determine the state vector of the objects at the epoch of the TLE. The resulting plots can be seen in figures 7.13 and 7.14, in which the Earth is drawn merely for scale, and not for orientation. The latter is because each TLE has a different Epoch, and consequently would each need a different orientation of the Earth in one plot.

The figure makes for a peculiar sight at first glance. It seems like almost all objects are located in the equator plane, while the measurements made by the SSN are done all around the world, and are averaged over a period of time to form the TLE.

It thus seems to be the case that the TLEs are produced in such a way that the position at the TLE epoch is defined as being on the equator, at the location of the node of the orbit. Why just a few objects are positioned away from the equator at their TLE epoch is unclear.

Two orbit starting points with constellations of satellites can clearly be seen in the MEO region, in Figure 7.14.

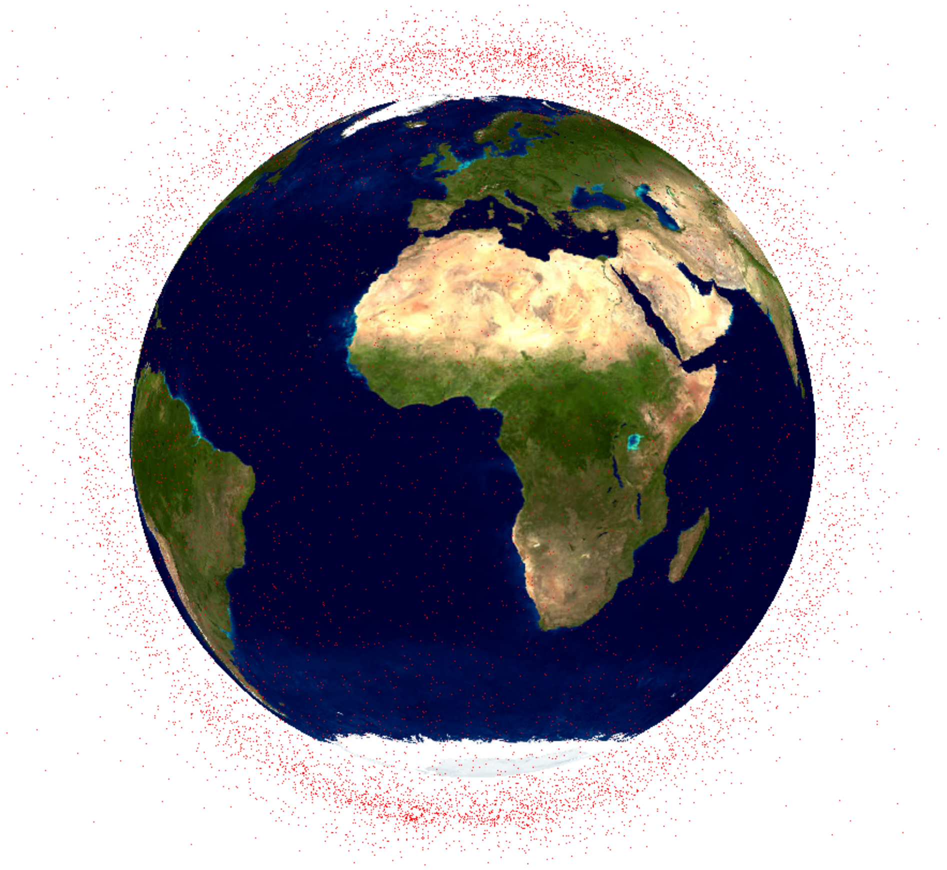


Figure 7.9: All 2011.05.05 am catalog objects propagated to 2011.05.05 12:00:00, LEO ring equator view.

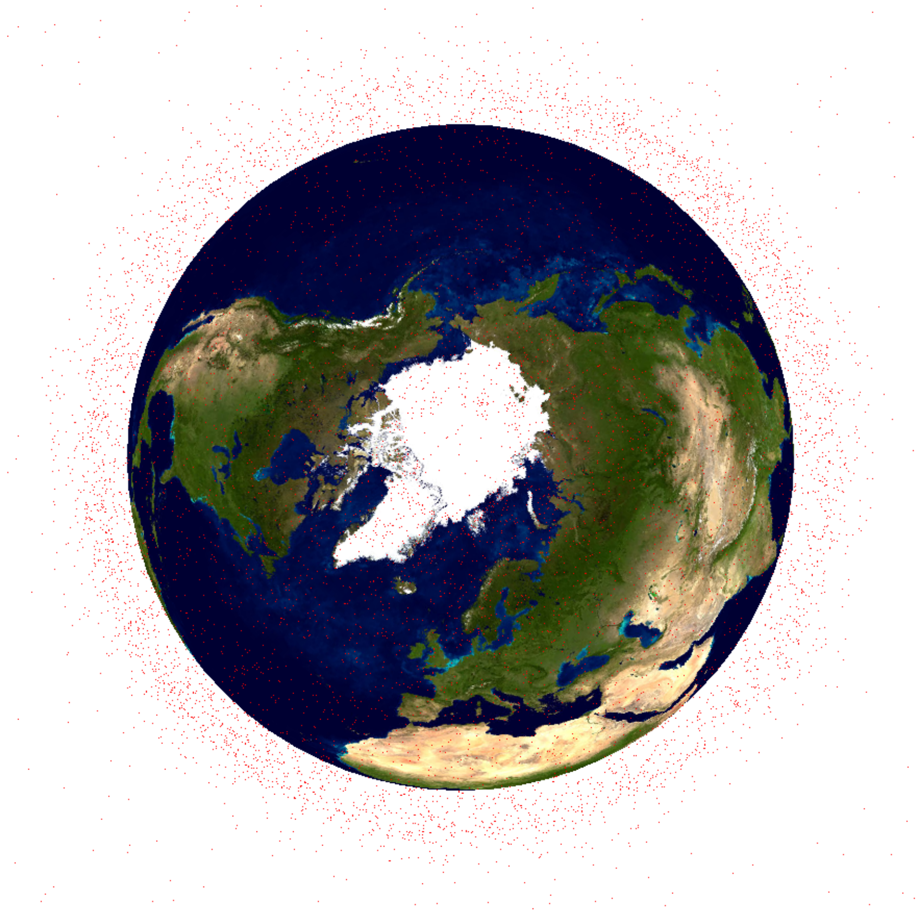


Figure 7.10: All 2011.05.05 am catalog objects propagated to 2011.05.05 12:00:00, LEO ring North pole view.

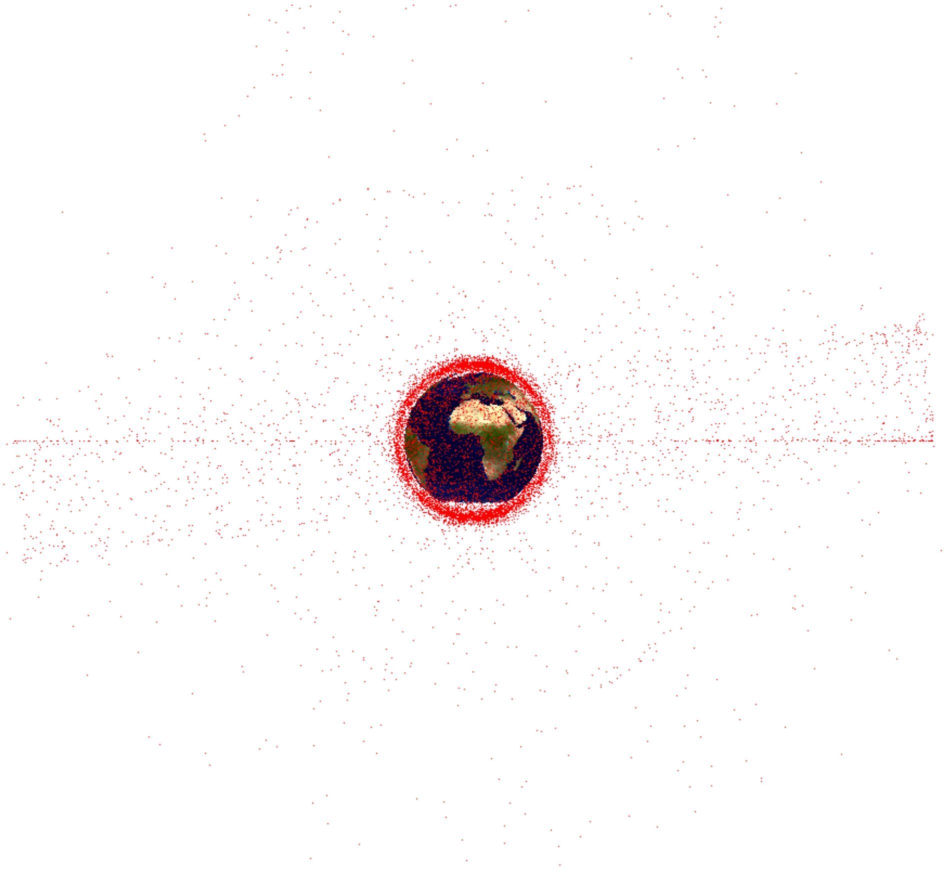


Figure 7.11: All 2011.05.05 am catalog objects propagated to 2011.05.05 12:00:00, GEO ring equator view.

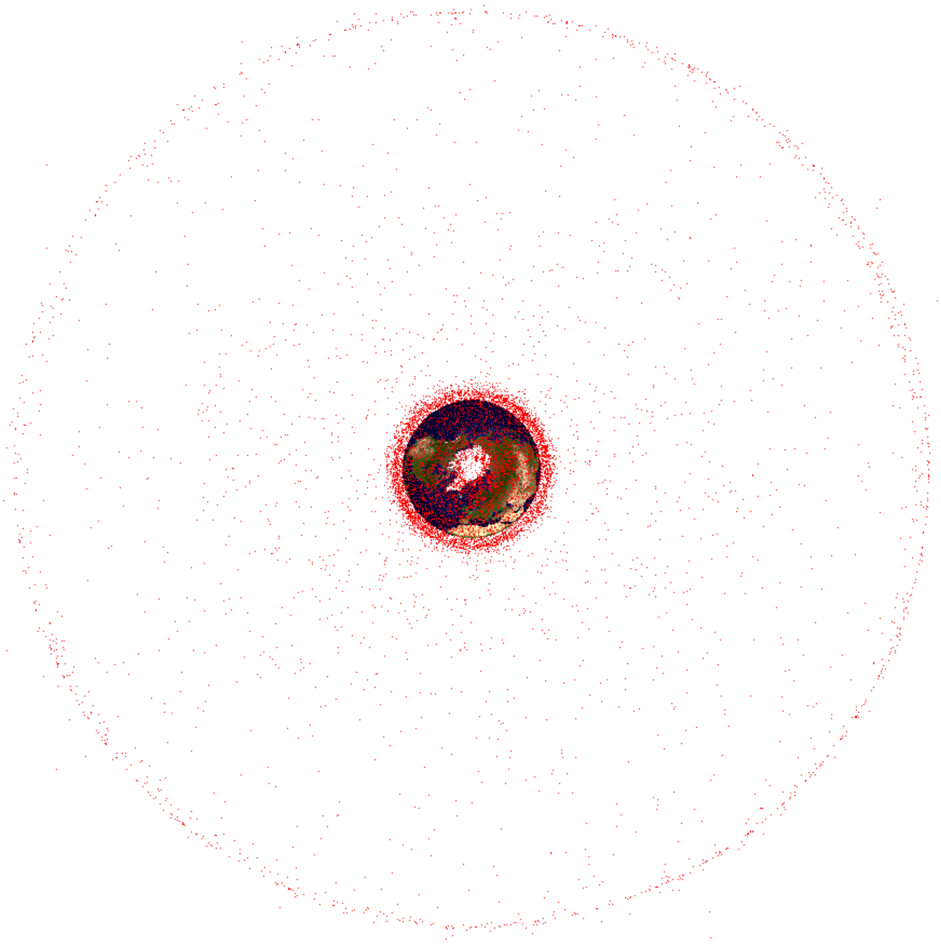


Figure 7.12: All 2011.05.05 am catalog objects propagated to 2011.05.05 12:00:00, GEO ring North pole view.

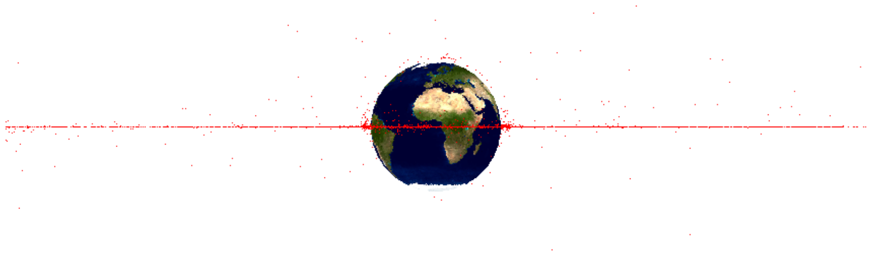


Figure 7.13: All 2011.05.05 am catalog objects propagated to their TLE epoch, GEO ring equator view.

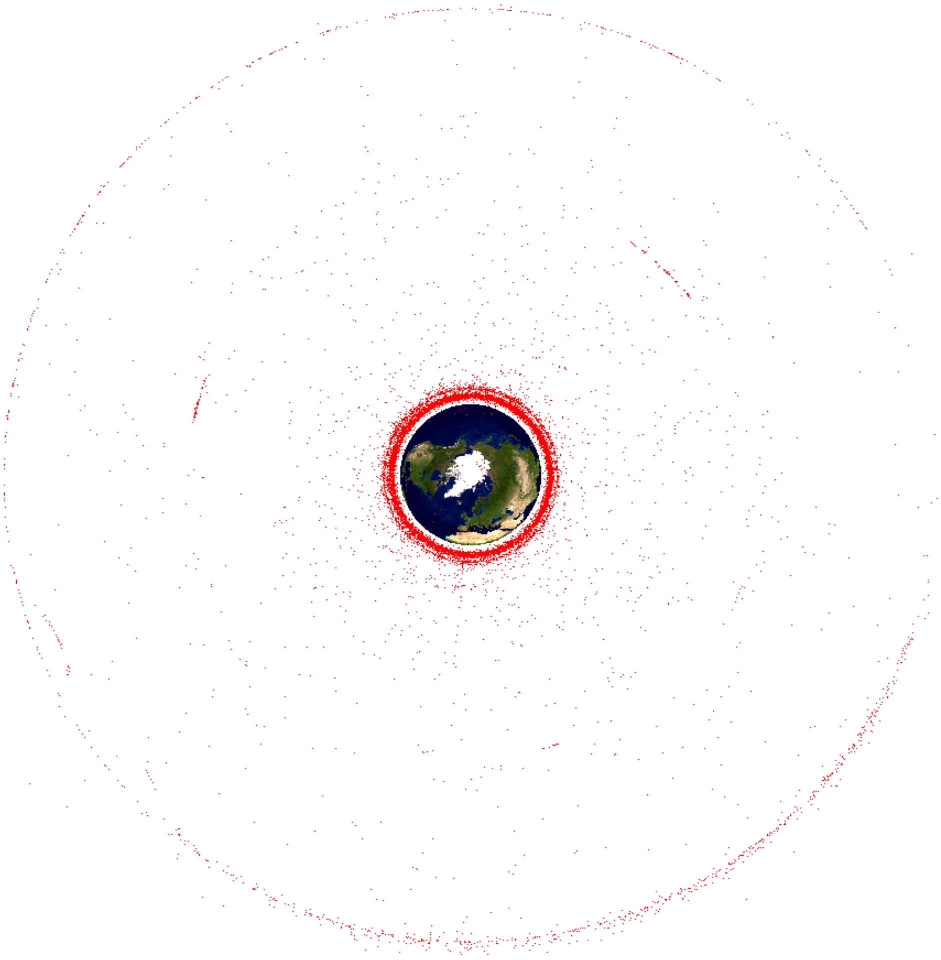


Figure 7.14: All 2011.05.05 am catalog objects propagated to their TLE epoch, GEO ring North pole view.

7.4 Cartesian Component Plots

To eventually be able to establish a basis for conjunction analysis, objects in the catalog need to be propagated for a certain interval of time T, at following time instances t_i, defined by a time step Δt, resulting in Cartesian coordinates at each of those time instances.

As a reference method, conjunction analysis will first be done by a so-called 'brute-force' method, which simply subtracts the coordinates of one object from another, resulting in the vector between the pair of objects, and thus their distance.

However, to be able to determine the minimum conjunction distance D in this interval accurately, the position of both objects in this time interval needs to be accurately determined as well. That is, the minima and maxima in the relative coordinates need to be well defined in order for the minimum of the conjunction distance to be reliably assessed.

To do this, the time step Δt needs to be small enough that these extrema are reasonably well described, so that not an entire extremum is missed in the process. If the latter is the case, then, when defining a certain conjunction distance threshold, a pair of objects might be missed in the process, while in reality their minimum distance is lower than this threshold.

The above is illustrated in Figures 7.15 - 7.19, in which an example is given of a propagated object from each of the previously defined orbital subgroups: LEO, MEO, GEO, GTO and HEO. One TLE was taken from each subgroup created by the TLE reader of Chapter 6 and their ephemerides were plotted for a period of 100 minutes. Since for the GEO and GTO objects one period and its extrema can not really be seen in these 100 minute interval, a subplot is made in the same figure containing one period.

The propagation was performed for both a time step of $\Delta t = 1$ and 5 min. The Cartesian coordinates x, y, z and their radius r are plotted with different colours for $\Delta t = 1$ min while the 5 min steps are plotted over these results with black crosses.

The extrema are due to the objects orbital period, which are the phenomena with the smallest period for the Cartesian components using the SGP4 propagator. That is, when one would plot these variables with any smaller time step Δt and time interval T, one would not see another disturbance with a smaller period on top of the shown sinusoid, which might result in different extrema.

It can be seen clearly that sometimes the extrema are not well defined with the black crosses, that is, relatively to the coloured dots. The accuracy of this extrema determination is dependent on a number of variables.

If the period of the sinusoid increases, the same extremum will be 'spread out' over a larger period of time and thus contain more data points, and thus the extremum will be better defined.

Furthermore, if the amplitude of the sinusoid is larger, the difference in the coordinate of one data point with respect to the following one, will be larger than for a sinusoid with a smaller amplitude, while having the same period.

Thus, objects with small orbital periods P and large fluctuations in their coordinates over one period, will lead to worse accuracy when determining the extrema with the same Δt. This is confirmed by visual inspection of the graphs, zooming in and out on the maxima and approximately determining the difference between one or two coloured dots and the 'real' extremum.

This difference turns out to be the largest for the LEO, MEO and HEO objects, which have the largest period-over-amplitude ratio. For a time step Δt of 1 minute, the maximum difference and thus also the approximate accuracy, is still on the order of 10 km. And as the accuracy of TLEs is comparable to this value, it would be wise to use smaller time steps than 1 minute.

However, this problem can not be circumvented, because, even if the time step is made infinitesimally small, the real conjunction distance might still be smaller than the calculated one. It could thus happen that a pair of objects would not be counted as conjunction, while in reality it just was.

The approach taken in the following conjunction analysis will be one which analyses a catalog with a smaller and smaller time step each time, so that the number of conjunctions detected will rise and converge to a maximum, while keeping the CPU time below a reasonable limit. This converged state of conjunctions will then be taken as a 'truth' reference for subsequent analyses.

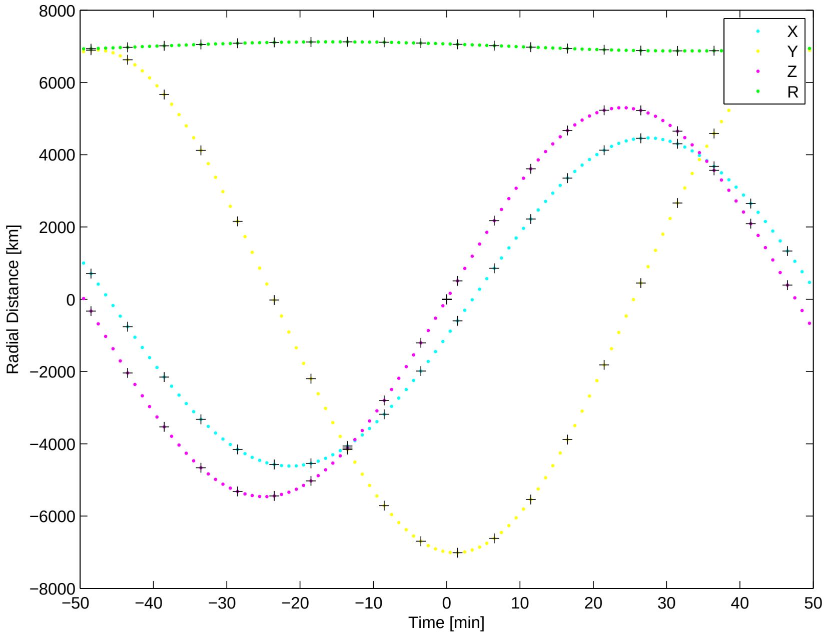


Figure 7.15: LEO object (ID #22), radius versus time from epoch.

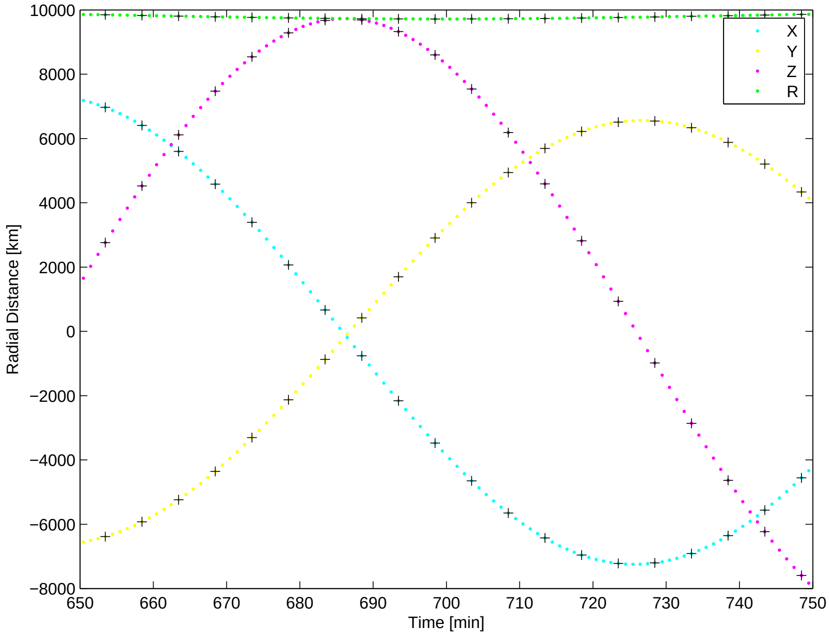


Figure 7.16: MEO object (ID #163), radius versus time from epoch.

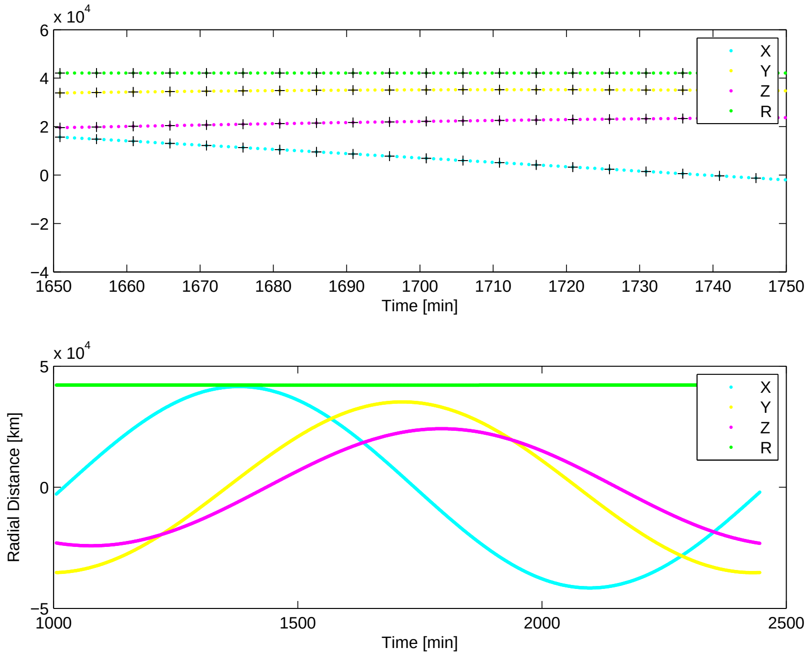


Figure 7.17: GEO object (ID #634), radius versus time from epoch.

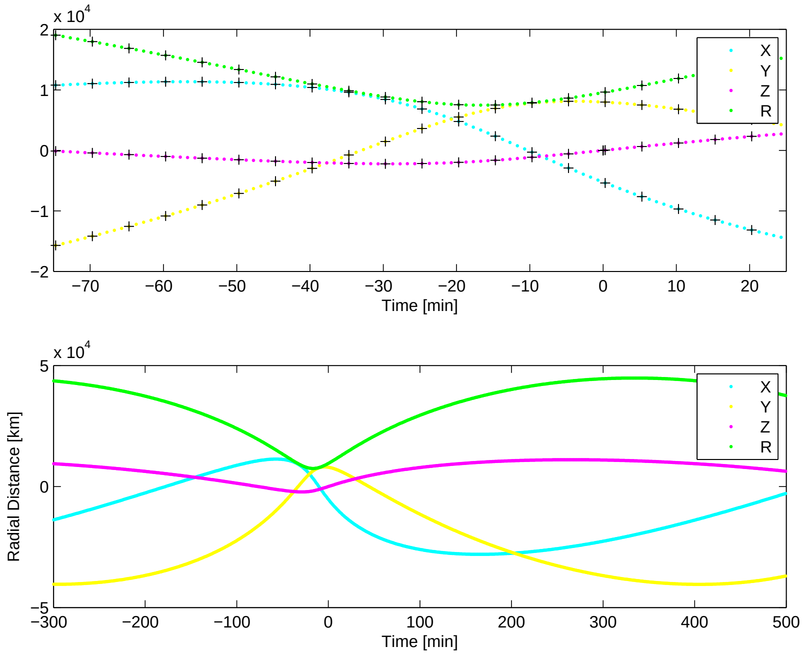


Figure 7.18: GTO object (ID #862), radius versus time from epoch.

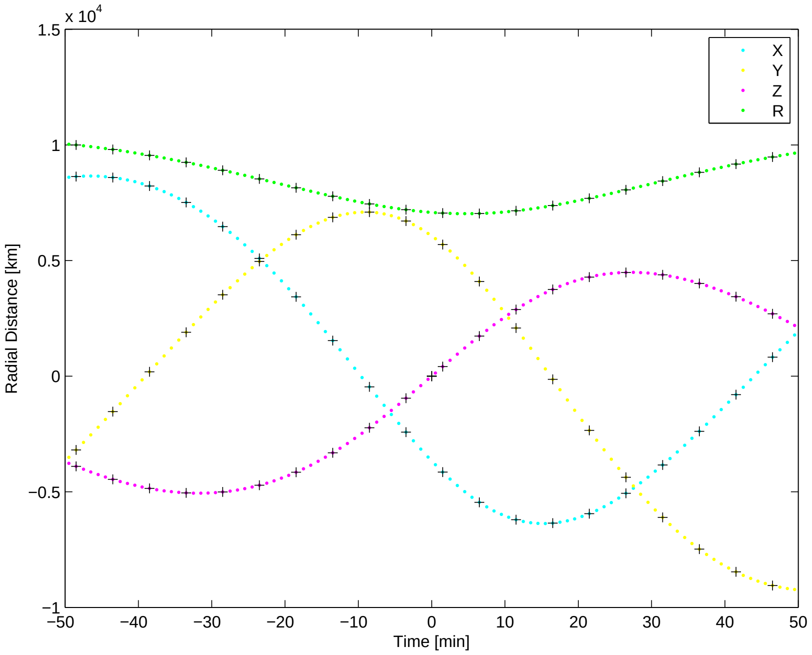


Figure 7.19: HEO object (ID #5), radius versus time from epoch.

Chapter 8

Orbital Debris Conjunction Analysis Implementation

Now that we are able to process TLE catalog files and propagate them with the analytical SGP4 propagator over a set time interval T, with a set time step Δt, and a set list of output variables, the resulting ephemerides file can be used for conjunction analysis.

The eventual objective is to find all conjunctions within a defined distance, for a time interval of 1 day, as efficient as possible. Since every half day a new catalog is placed online with new data (for each day, one is listed as 'am' and the other as 'pm'), this gives enough buffer. When a new catalog is available the conjunction analysis process can be started again with this new data, while having a buffer of half a day.

To be able to verify the results of the filters and sieves (which were discussed in Section 5.1), a benchmark is needed, which is called the "brute-force" method here. After the brute-force method has been verified to work correctly, filters and sieves will be applied to make the conjunction analysis process more efficient.

The difference between the terms '(pre)filter' and 'sieve' used here, lies in the fact that filters only have to be applied per object pair once, while sieves have to be applied per object pair *and* for each time instance, using the ephemerides. These different name conventions were merely copied from the respective references.

Note that in all analyses the ephemerides resulting from the SGP4 propagated TLEs are considered a 'truth' set here. The conjunction analysis process is optimised assuming the ephemerides are 'perfect' descriptions of the orbits.

Here now follows a description of the implementation and results of the brute-force method, in Section 8.1. Included is a description of the ephemerides generation, specialised for this conjunction analysis process.

Then, in Section 8.2 follows an analysis of the perigee-apogee filter (which was first described in Section 5.1), of which four different implementations are tested here.

Finally, a description of the implementation of the sieves and their results and performance is given in Section 8.3. Eight possible improvements to the original sieves were made as well, and to which the performance was compared. The implementation of the perigee-apogee filter is treated here as well.

8.1 Brute-Force Code Implementation

A flow diagram of the conjunction analysis code subjected to the brute-force method can be seen in Figure 8.1, in which the upper level processes (classes, functions) and data flows are shown. These processes are separately described in the following subsections.

The process starts with the ephemerides file generated by the SGP4 propagator, which contains the state of each object in the catalog at every time instance t_i in the time interval T, separated by a time step Δt, which is elaborated upon further in Subsection 8.1.1.

8.1.1 Ephemerides Generation

At first, the SGP4 ephemerides generation was done like before, that is, per object, all available output variables (date and time, minutes from epoch, Cartesian state vector, Keplerian elements) were listed per time step t_i underneath each other. Thus, in an output text file, first all ephemerides data for the entire time interval T of the first object was listed, underneath which all the other objects followed subsequently with their respective data.

However, as the time interval under consideration grew larger, for the conjunction analysis class to be able to handle large-sized ephemerides files, the ephemerides needed to be written per time instance t_i for all objects. Because in the original SGP4 way, if the conjunction analysis class were to analyse the conjunction at one particular time instance, it would have to scroll through the entire ephemerides file to collect the data of all objects for one t_i. Or the entire file would have to be read into memory first, which was not possible any more as the ephemerides files for longer time intervals grew larger than the available memory on the used machine. The previous SGP4 code was thus rewritten for this purpose.

Also, for a full-day analysis, using the old way, the ephemerides file would grow larger than the available storage space. The output data of each ephemeris line was consequently made as small and efficient as possible. This was done by listing, for each time instance in Julian Date (written only once), the Cartesian state vector for all objects, next to their ID. The Cartesian state vectors in the output were written in meters for the position and in

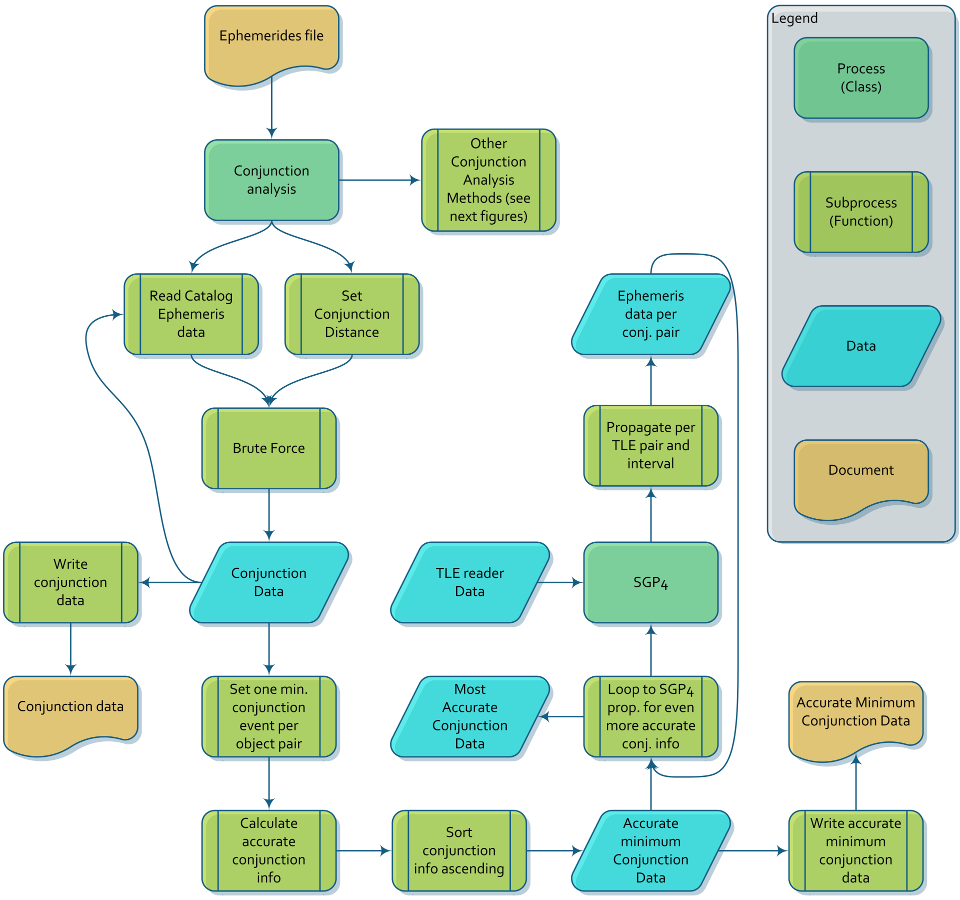

Ephemerides file
Conjunction analysis
Other Conjunction Analysis Methods (see next figures)
Read Catalog Ephemeris data
Set Conjunction Distance
Brute Force
Conjunction Data
Write conjunction data
Conjunction data
Set one min. conjunction event per object pair
Calculate accurate conjunction info
Sort conjunction info ascending
Accurate minimum Conjunction Data
Write accurate minimum conjunction data
Accurate Minimum Conjunction Data
TLE reader Data
SGP4
Propagate per TLE pair and interval
Ephemeris data per conj. pair
Loop to SGP4 prop. for even more accurate conj. info
Most Accurate Conjunction Data
Legend
Process (Class)
Subprocess (Function)
Data
Document

Figure 8.1: Flow diagram of the brute-force conjunction analysis process. See legend for the explanation of the block types.

meters per second for the velocity, such that it did not need a decimal point and could be read in as integers, while still maintaining sufficient accuracy. This resulted in an ephemerides file of 71.6 GB for a one-day analysis with a time step of 1 second, compared to about 200 GB before. Examples of the new output files will be given in subsequent sections.

8.1.2 Conjunction Analysis

The ephemerides data is then read in by a function in the conjunction analysis class, after which a certain conjunction analysis method can be called, in this case the brute-force method. At first, all ephemerides were read in and stored in memory, after which the conjunction analysis was performed. However, when analysing larger time intervals the ephemerides file began growing larger than the available memory.

Later on this was solved by reading in the ephemerides file (which was now written in a different way, as described above) per time instance, after which conjunction analysis is performed for this time instance only. When this is done, the relevant conjunction info is stored, and the method carries on to read in the ephemeris info of the next time instance, which is then analysed for conjunctions.

Conjunction Distance

Before the brute-force method is called, the conjunction distance D needs to be defined: only ephemerides sets of an object pair with a distance smaller than this set value will be registered as being a conjunction. Care must be taken when setting this variable with respect to the time step Δt, because, when it is set too low, some conjunctions might not be detected, and when it is set too high, too much conjunctions might be detected.

When two objects move with a relative speed of, say, 10 km/s directly towards each other, while the time step Δt of the ephemerides is set to 1 second, and the conjunction distance D is set to 5 km (or less), the conjunction event (or even collision event in this case) might be missed entirely. This is simply due to the fact that there might be no ephemerides of these objects defined during the time period in which the distance between the two objects is smaller than this 5 km.

Since no object in the catalog which is orbiting the Earth can ever exceed the escape velocity $V_{esc} = 11.18$ km/s of the Earth (calculated with the WGS-72 constants), the relative velocity of two objects can never exceed twice the escape velocity. In fact, the fastest flying object orbiting the Earth in the recent past was the Atlas Agena D rocket booster of Table 6.5, which reached a velocity at perigee of approximately $V_p = 10.76$ km/s.

If the time step Δt is 1 second, the maximum relative distance which can be covered by two such objects during one time step is thus 21.52 km, and has a maximum of 22.36 km for any object pair. Taking a small extra margin it would thus be wise to scale D with Δt as $D = 25\Delta t$, with D in km and Δt in seconds.

When this is done, an object pair which would be on a collision course would thus be detected at least twice. But, an object pair for which the secondary object merely touches the conjunction sphere centred around the primary object, might still not be detected. However, there exists a conjunction distance such that, when the distance of the object pair would at any point be smaller than this value, the conjunction would always be detected due to the ephemerides sampling defined by Δt.

This is shown more clearly in Figure 8.2, in which a primary object (p) and a secondary (s) object can be seen. If the primary object is defined as not moving, the secondary object moves with the relative speed of the object

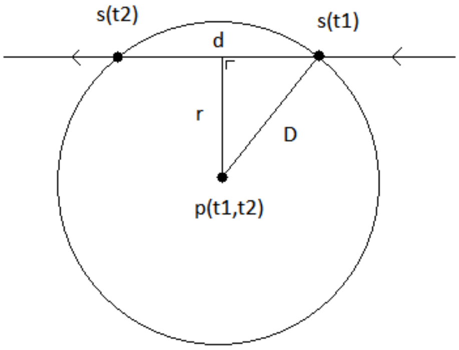


Figure 8.2: Schematic of the worst case scenario when a conjunction could just be missed.

pair. In a worst case scenario, there would exist ephemerides so that the secondary object would lie just on the edge of the conjunction sphere defined by D for two consecutive time instances t_1 and t_2, while the conjunction would just not be detected. Rectilinear motion is assumed here as the time step is small. Using Pythagoras' equation one can now determine r:

$$D^2 = \left(\frac{d}{2}\right)^2 + r^2 \tag{8.1}$$

For the worst case scenario the relative velocity would be two times V_{esc}, so the distance d travelled would be $2V_{esc}\Delta t$.

$$r = \sqrt{D^2 - v_{esc}^2 \Delta t^2} \tag{8.2}$$

With the conjunction distance defined as 25 km and the time step set at 1 second, the result would be r = 22.36 km. Thus, in the worst case scenario for this setup, an object pair which would have a distance of 22.36 km at closest approach, might still just not be detected by the brute-force method.

No matter how small Δt, and how large (or small) D, this problem can never be circumvented, since any secondary object touching the conjunction sphere centred around the primary object for a period smaller than the defined Δt might still not be detected (in fact, it *can* be done using a different method, as we will see later on). For further analysis, D will be determined as described above.

The results of the brute-force analysis will be used as a "truth state" for verification of the sieves and filters later on. If needed, these filters and sieves will make use of the same ephemerides data, resulting in no compatibility issues.

Brute-Force Method

The brute-force function then determines which object pairs at what time instances t_i had a relative position less than the set conjunction distance D. It does this by reading in the ephemerides of all objects in the catalog for one time instance t_i and analysing all possible objects pairs, thereby excluding combinations of an object with itself and double combinations (like object one combined with object two, and object two combined with object one).

The amount of pairs that need to be analysed in this way was given by Equation 5.18. This means that for the current public catalog size of 14,644 objects, 107,216,046 pairs need to be analysed *per time instance*! For a full-day analysis with a time step of 1 second this results in 9,263,573,590,446 pairs that need to be analysed.

For each pair the differences between all position vector components are taken, squared and added, to determine the distance squared. This value is then compared to the defined conjunction distance squared, and if it is smaller, the conjunction information will be stored, including the distance between the two objects, which is taken as the square root of the previously determined value:

$$\text{if} \quad r_{rel}^2 = (x_s - x_p)^2 + (y_s - y_p)^2 + (z_s - z_p)^2 < D^2 \tag{8.3}$$

$$\rightarrow r_{rel} = \sqrt{r_{rel}^2} \tag{8.4}$$

The resulting conjunction data contains, per conjunction, information on the two objects involved (by ID), the time of the conjunction, the distance between both objects at that moment, and the Cartesian state vectors of both objects. This data is written to a text file for future analysis or comparison.

The reason why the square root is not taken right away lies in the fact that it then has to be taken for every comparison, that is, for all 107,216,046 possible pairs per time instance. When comparing with the squared conjunction distance like above, the square root only needs to be taken for the found conjunctions that meet this requirement. Since taking the square root is a relatively time-consuming process for a computer, this small trick already resulted in reduction of CPU time by a factor two.

Post-Processing

However, during one *conjunction event*, the distance between two objects might be below the conjunction distance D for multiple following time instance t_i. The following function thus filters out these multiple defined conjunctions, and only stores the conjunction info at the time instance at which the determined distance is minimum, resulting in one determined conjunction per object pair.

Furthermore, after the step described above, the conjunction info is sorted ascendingly with respect to the distance between the two objects per found conjunction, resulting in a nice overview with the closest approaches on top. Then, the sorted minimum conjunction info is written to a text file.

Calculating Time and Distance at Closest Approach

The now found minimum distance between two objects for a conjunction event is not the *real* minimum distance that occurs during that conjunction event in “reality”, since we are stuck with a time grid defined by Δt for our analysis up until now. The time at which the real minimum distance for a conjunction lies somewhere around the time instance found, and can be determined with a Taylor series approximation, as described in (Healy, 1995).

The relative displacement vector $\mathbf{x}$ of the two objects can be approximated as

$$\mathbf{x} = \mathbf{x}_0 + (t - t_0)\dot{\mathbf{x}}_0 + \frac{1}{2}(t - t_0)^2\ddot{\mathbf{x}}_0 + \cdots \tag{8.5}$$

with which the point of closest approach can be found at the minimum value of $\mathbf{x} \cdot \mathbf{x}$. Truncating the previous equation at the linear term then results in

$$\mathbf{x} \cdot \mathbf{x} = \mathbf{x}_0 \cdot \mathbf{x}_0 + 2(t - t_0)\mathbf{x}_0 \cdot \dot{\mathbf{x}}_0 + (t - t_0)^2\dot{\mathbf{x}}_0 \cdot \dot{\mathbf{x}}_0 \tag{8.6}$$

The time of the extremum can be found by setting the derivative equal to zero,

$$\frac{d}{dt}(\mathbf{x} \cdot \mathbf{x}) = 2\mathbf{x}_0 \cdot \dot{\mathbf{x}}_0 + 2(t - t_0)\dot{\mathbf{x}}_0 \cdot \dot{\mathbf{x}}_0 = 0 \tag{8.7}$$

So, to lowest order, the time at closest approach occurs at

$$t_{closest} = t_0 - \frac{\mathbf{x}_0 \cdot \dot{\mathbf{x}}_0}{\dot{\mathbf{x}}_0 \cdot \dot{\mathbf{x}}_0} \tag{8.8}$$

with which the distance between both objects at this time can be determined using the initial values and Equation 8.5.

The time and relative displacement vector found earlier by the brute-force method are thus the initial values indicated by the subscript 0 in the previous analysis, with which the real time and distance of closest approach can be approximated. This approximation is only accurate for small time deviations, but when the time of closest approach lies within two time instances separated by a small time step Δt, the results are fairly accurate here, since it always lies within $\frac{1}{2}\Delta t$ from the initial value.

If the time of closest approach lies outside of the time interval T under analysis, it can deviate from the time at which the initial conditions are known by a much larger offset, however, and the results will in most cases not be accurate any more. These results will be ignored (while the results of the time grid will be kept), since they lead to conjunction events which have a time of closest approach outside of the time interval T under analysis in the first place!

Looping with SGP4

To obtain the most accurate time and distance possible at closest approach, one can feed the results of the Taylor series approximation back to the SGP4 propagator, and analyse the results. A relatively coarse time grid of say 10 time steps of 0.1 seconds around the approximated time of closest approach can be fed back into the SGP4 propagator, next to the TLEs of the objects under consideration, which are taken from the TLE reader.

A special function within the SGP4 propagator had to be written for this purpose, however, which can handle this type of input and feed the results right back into the conjunction analysis, without writing them to a file.

This newly generated ephemerides data for both objects in the conjunction pair can then be used to determine at which time instance the distance of the objects is minimum. When this is done, a new and more fine time grid around this point can then be used again to determine the conjunction more accurately. This whole process can thus be looped as many times as needed in order to find the conjunction info at a particular accuracy.

However, since this process is not really needed for the main purpose of this thesis, and costs too much time, it is not researched in-depth. The evaluation of the filters and sieves later on can already be done with just the conjunction information gained from the brute-force method combined with the Taylor series approximation.

Looping just one time can already be used for verification though, as will be shown later on.

8.1.3 Results

First some examples of the output text files of the code will be given, after which some results of all the runs that were made will be presented, followed by three verification methods applied to these results.

Example Output Files

The final setup of the ephemerides text file described above can be seen in Table 8.1, in which part of a file has been listed for two time instances in JD, including the first and last few objects of the catalog.

Table 8.1: Example of part of an ephemerides text file for two time instances. Per time instance t, the state vector of each object is listed. The first column lists the object ID, followed by the x, y and z coordinates, and then the components of the velocity $\dot{x}$, $\dot{y}$ and $\dot{z}$. The catalog file used is of 2011.10.20 am, and the time instances are 2011.10.21 00:00:00 and 1 second later.

t [JD]						
ID	x [m]	y [m]	z [m]	$\dot{x}$ [m/s]	$\dot{y}$ [m/s]	$\dot{z}$ [m/s]
JD= 2455855.500000000						
5	-8180105	-3527961	-4080313	557	-5584	1914
11	7007354	3007982	-652267	-3065	5436	4033
12	-1414288	8877550	-3626454	-5061	-1964	-2224
			⋮			
37842	2692036	6330543	2022886	-6921	2456	1388
37843	-13678754	12212717	8054	-1184	-1901	-2608
JD= 2455855.50001157						
5	-8179546	-3533545	-4078398	561	-5582	1916
11	7004286	3013417	-648234	-3071	5433	4033
12	-1419349	8875585	-3628678	-5061	-1967	-2222
			⋮			
37842	2685114	6332996	2024273	-6924	2450	1386
37843	-13679937	12210817	5445	-1183	-1901	-2608

In Table 8.2 an example of part of an output text file of the first step in the brute-force conjunction analysis can be seen. For each time instance, the object pairs that have a distance less than the set conjunction distance are listed. The state vectors of both objects are also stored for future analysis.

Then, when one conjunction event per object pair is set, the conjunctions are sorted ascendingly by distance, and the time and distance at time of closest approach (tca) are calculated as described above. The resulting output text file looks like the example in Table 8.3.

At first glance, a possible collision ($d_{ps} = d_{tca} = 0$) seems to be found! However, it turns out that these objects are called 'SOYUZ-TMA 02M' and 'PROGRESS-M 10M' respectively. Both are Russian spacecraft launched from Baikonur by a Soyuz rocket. The former, on 2011 June 7 at 20:12 UT, carrying a crew of three astronauts to the ISS. The latter, on 2011 April 27 at 13:05, resupplying the ISS with cargo. During the time interval T under analysis, both were still docked to the ISS, and had the same TLE parameters resulting in the same orbit and thus generating the distance of 0 meters.

Table 8.2: Example of part of a brute-force conjunction output text file for two time instances. Per time instance t_{con}, the ID's of both the primary (p) and secondary (s) object of the conjunction pair is given, followed by the distance between the objects and their state vectors. The catalog file used is of 2011.10.20 am, and the time instances are 2011.10.21 00:00:00 and 1 second later.

t_{con} [JD]	ID$_p$	ID$_s$	d_{ps} [m]	x_p [m]	$\cdots$	$\dot{z}_s$ [m/s]
2455855.50000000	12091	5446	18032	7005608		6944
2455855.50000000	22699	4737	16627	-6852330		7305
2455855.50000000	26638	25462	15703	22898305		-3
			⋮			
2455855.50000000	37479	37054	18816	799579		118
2455855.50000000	37633	25544	13554	3994459		-5460
2455855.50000000	37633	37396	0	3994459		-5455
2455855.50001157	12091	5446	19102	7003463		6942
			⋮			
2455855.50001157	37429	245	19987	-3259348		1688
2455855.50001157	37479	37054	18581	805538		125
			⋮			

Why the ISS itself did not result in such a close conjunction with the above two objects, is because the TLE for the 'ISS (ZARYA)' has an epoch of half a day later, thus resulting in a different orbit. The ISS turned out to have a minimum distance of 13470 meters from the above two objects in the time interval under analysis, according to the SGP4 theory, which is an indication for the accuracy of TLE propagation.

Another conjunction pair (not shown in the table) that keeps on showing up is formed by 'PRISMA (MANGO)' and 'PRISMA (TANGO)', two Swedish satellites which will test technologies and rendezvous and formation flying in space, hence the constant close encounters. Also 'TANDEM-X' and 'TERRASAR-X' form such a pair, which are two German radar mapping satellite that fly in close formation to obtain better accuracy.

Table 8.3: Example of part of an output text file after post-processing. Per object pair one conjunction event is listed with its time instance t_{con}, the ID's of both the primary (p) and secondary (s) object of the conjunction pair, followed by the distance between the objects, their state vectors (not shown here), and the calculated time of closest approach $t_{tca_{rel}}$ (relative to t_{con}), distance at time of closest approach t_{tca} and the time of closest approach $_{tca}$. The catalog file used is of 2011.10.20 am, and the time interval under analysis is a 1-day period from 2011.10.21 00:00:00 until 2011.10.22 the same time.

t_{con} [JD]	ID$_p$	ID$_s$	d_{ps} [m]	$\cdots$	$t_{tca_{rel}}$ [s]	d_{tca} [m]	t_{tca} [JD]
2455855.50000000	37633	37396	0		0.000	0	2455855.50000000
2455856.07379629	34032	33837	213		0.005	211	2455856.07379635
2455855.66113425	30906	30519	246		-0.005	235	2455855.66113420
2455855.54719907	34827	31617	323		0.018	176	2455855.54719928
				$\vdots$			
2455856.33766203	31438	31140	25000		0.003	25000	2455856.33766206
2455855.90634259	34522	21529	25000		-0.433	24431	2455855.90633758

Performance

Here follow the performance results of the brute-force method, but first an explanation of CPU time and a description of the system used, is given.

CPU time The performance of the code in terms of CPU time can be easily retrieved from the system at runtime by fetching the processor time at the start of the program, and at the end (or somewhere in between), and then subtracting the former from the latter. The result is then the number of clock ticks since the start of the program, and this can then be divided by the number of clocks per second of the respective machine.

However, if that respective CPU was busy with other calculations as well (other programs open or running), then the CPU power needed for these other calculations will be included in the resulting CPU time as well. The resulting time in seconds is then not quite the regular wall clock time, but the time the CPU needed to make all calculations during the time the code was executed.

This timing method is not perfect, but can work if the other processes are kept at a minimum, as we will see in the next paragraph.

System The system on which all code in this report was compiled and executed is a Hewlett-Packard HP EliteBook 8530w, with a Mobile Dual-Core Intel Core 2 Duo T9600 2800 MHz processor, and 4060 MB (DDR2-800 SDRAM) system memory. Of this system, one of the two cores was thus used for analysis, and, while the system has a 64-bit capable processor, the code was compiled and run as 32-bit. The editor/compiler used is QtCreator/MinGW. The CPU time shown for all analyses is rounded off to integer seconds, so as to account for the inaccuracy of this timing process.

Since a dual-core processor was used, the execution of the code is by default done by only one of the two cores, which always runs close to full power (50% of total CPU-power), while the other core is busy with all other processes such as the operating system and software that is running. Thus, the CPU time determined in the way described above will lie close to the CPU time which would be needed had the CPU devoted its full power to this process for the whole duration. Different runs can thus be compared in this way, with only a small margin of error.

Optimum time step After initially running the brute-force conjunction method including the post-processing steps (but without the loop) for the first time using different time step sizes Δt, for a time interval T of 60 seconds, the resulting CPU time needed is shown in Figure 8.3. It can be seen in this plot that for small time steps (5 and 6 seconds) the CPU time decreases with Δt while for large time steps (30 and 60 seconds), the CPU time increases with Δt. An optimum (minimum) time can thus be

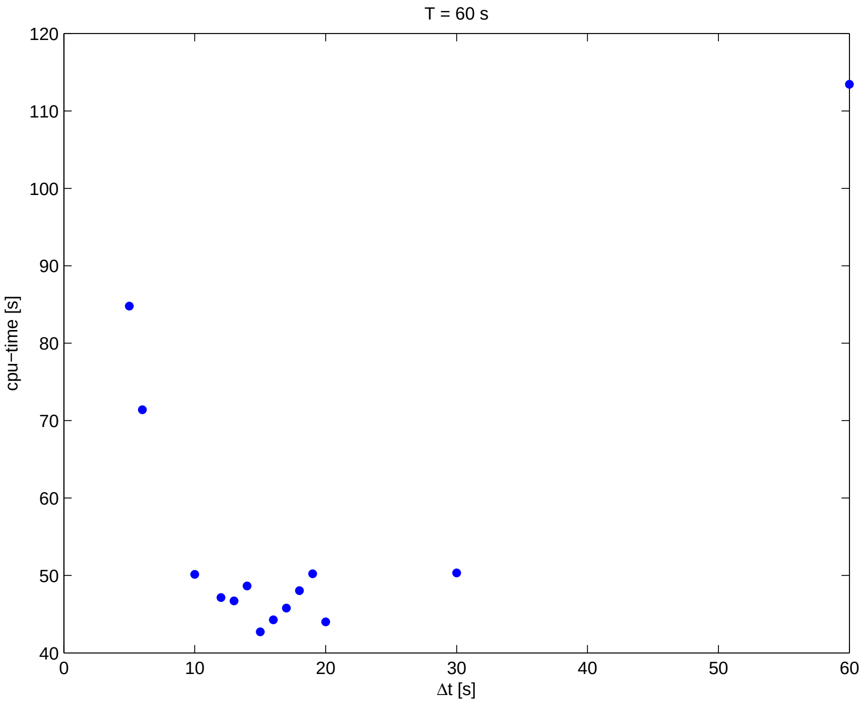


Figure 8.3: CPU time versus time-step Δt for the brute-force method, for a time interval T of 1 minute, applied to the catalog of 2011.06.22 am containing 14507 objects.

found somewhere in between, at $\Delta t = 15$ s, which was found by iterative processing.

The first and decreasing part of the plot can be explained by looking at the amount of ephemerides that have to be generated and analysed. For the conjunction analysis this even scales quadratically, since it has to analyse all possible pairs. For a smaller time step, more ephemerides are generated and thus more conjunctions need to be analysed, resulting in a higher CPU time.

The last and increasing part of the plot can be explained by looking at the number of conjunctions that are found. Since the conjunction distance is set at $D = 25\Delta t$, more and more conjunctions will be found for higher Δt, which need to be stored, written to a file and post-processed. Because of the higher conjunction distance which needs to be set in order to still catch a reasonable amount of the conjunctions (cf. Equation 8.2), the CPU time will rise because of the burden the large amount of conjunctions brings with it.

The middle saw-tooth like relation stems from the fact that for $\Delta t =$

Table 8.4: The performance of the brute-force method in terms of number of found conjunctions (one minimum conjunction per object pair), and CPU time needed for SGP4 propagation (SGP4), the brute-force method (brute) and post-processing (post), with different input for the time step Δt, time interval T and conjunction distance D. The catalog file used was of 2011.10.20 am, with all analyses starting at 2011.10.21 00:00:00.

				CPU time [s]			
Δt [s]	T [s]	D [km]	c [#]	SGP4	brute	post	total
1	1	25	41	5	14	0	19
0.1	1	2.5	2	7	74	0	81
0.1	1	25	41	7	74	0	81
15	60	375	57214	6	34	198	238
5	60	125	5723	8	93	5	106
5	60	375	60353	8	86	521	615
2	60	50	850	13	212	1	226
1	60	25	198	20	406	0	426
1	60	50	860	20	403	2	425
1	60	125	5816	20	422	22	465
0.5	60	25	202	35	809	0	844
0.5	60	50	861	35	840	3	878

$15 - 19$ s, the amount of time instances in the interval is 5 for all Δt, while for $\Delta t = 20$ s, this amount changes to 4, resulting in the drop. The slight increase in this saw tooth is because of the increase in the amount of conjunctions found due to the slightly increased Δt.

Amount of conjunctions To test the performance of the brute-force conjunction analysis system in terms of conjunctions found and CPU time needed, some runs were made with different input for Δt, T and D, of which the results are listed in Table 8.4. The number of conjunctions c listed here is after the post-processing, so one minimum conjunction per object pair is counted. The CPU time is listed separately for the SGP4 propagation (SGP4), the brute-force method (brute), and the post-processing (post).

To find out the influence of Δt on the number of conjunctions found, T and D need to be kept constant.

When comparing the number of found conjunctions for $T = 1$ s, $D = 25$, $\Delta t = 1$ s versus $\Delta t = 0.1$ s (both 41), one might think that all conjunctions might be found already with $\Delta t = 1$ s. But when making the same comparison for $T = 60$ s, $D = 25$, $\Delta t = 1$ s vs $\Delta t = 0.5$ s, the latter finds 4 conjunctions more (198 versus 202).

For larger values of Δt, this difference grows larger however, as can be seen for $D = 125$ km and $D = 375$ km, comparing the smaller Δt with the larger one. For the best results in terms of the number of found conjunctions, the time step Δt can thus best be set as small as possible, keeping $D = 25\Delta t$.

Furthermore, when looking at the CPU time, the time needed for SGP4 propagation can be ignored with respect to the time needed for the brute-force method, for larger time interval T. This is because the amount of pairs analysed by the brute-force method scales quadratically with the amount of objects propagated by SGP4.

Decreasing Δt for the same T, drastically increases CPU time for the brute-force method. However, increasing Δt and thus also D, increases the CPU time for post-processing. This is because, as the number of conjunctions found increases, the post-processing needs to make more and more comparisons, when filtering out the multiply defined conjunction pairs.

The CPU time needed for post-processing increases quadratically with the amount of conjunctions found, and thus Δt needs to be kept as small as possible, especially for larger time intervals like the full-day analysis.

Also, the file sizes grow for the conjunction info, which slows down future processes that need these files for comparison with the results of the filter and sieve processes. However, since smaller Δt increases the CPU time needed for the brute-force method again, a trade-off has to be made.

For a full-day analysis, the largest ephemerides file (71.6 GB) which can still be stored in the machine at hand is for $\Delta t = 1$ s. Since this value also leads to good performance in terms of number of conjunctions found and to reasonably low post-processing times (cf. Table 8.4), it was chosen to generate the "truth" data sets.

Results of new setup After the data for Table 8.4 was produced, the switch was made to the new setup for the SGP4 propagator and ephemerides file generation, with an accompanying new brute-force method code implementation, to be able to handle the very large ephemerides files. 'New' here is with the ephemerides ordered per time step, rather than per object, as explained before.

This resulted in a more efficient process leading to lower CPU times, as can be seen in Table 8.5, in which the results of larger time intervals T are shown, all for $\Delta t = 1$ s and $D = 25000$ m.

Looking at the second row of the table, the CPU time needed for SGP4 is reduced by more than a factor 2, compared to Table 8.4, for the same initial parameters (row 8). Furthermore, the CPU time for the brute-force method ('brute') in row 2 was also reduced by almost a factor 2.

Table 8.5: For $\Delta t = 1$ s and $D = 25000$ m, the amount of conjunctions (both multiple defined conjunction pairs (t.c.) as well as one minimum conjunction per object pair (m.c.)) found for different T, and their corresponding CPU times, are listed. The catalog file used was of 2011.10.20 am, with all analyses starting at 2011.10.21 00:00:00.

			CPU time [s]			
T [s]	# t.c.	# m.c.	SGP4	brute	post	total
1	75	41	1	8	0	9
60	2141	227	9	224	0	233
3600	118339	10296	502	13438	78	140118
86400	2848301	208911	12157	311832	96129	420118

In this table the total amount of conjunctions (so including multiple defined conjunction pairs) are listed as well as the conjunction with minimum distance (m.c.), defined once per object pair. Both amounts can be shown to more or less vary linearly with T, as would be expected. For the latter this is only the case for larger T, as a relatively high amount of duplicates are filtered out for smaller T.

Furthermore, the CPU time needed for the brute-force method and post-processing of the 1-day interval has really grown large (about 4 and 1 days of running, respectively), but fortunately still within reasonable limits.

Finally, using the resulting conjunction info file of the full-day analysis, the amount of conjunctions found could be listed versus the conjunction distance, as can be seen in Table 8.6.

The amount of found conjunctions for both types can be seen to grow exponentially, as could also be expected, since the conjunction volume grows cubically with D.

The total amount of conjunctions at first does not grow that fast, because it has many object pairs which are counted a relatively high amount of times (some have a conjunction at every time instance).

The CPU time listed here is just that of post-processing, as the input files used here was already containing all conjunctions with a distance smaller than 25 km. The CPU time can be seen to increase quadratically with respect to the amount of total conjunctions.

Verification

The results were verified in three ways, the first of which made use of a different source for an equation. A different expression for the minimum

Table 8.6: For $\Delta t = 1$ s and $T = 86400$ s, the amount of conjunctions (both multiple defined conjunction pairs (t.c.) as well as one minimum conjunction per object pair (m.c.)) found for different D, and their corresponding CPU times (only for post-processing here), are listed. The catalog file used was of 2011.10.20 am, with T running from 2011.10.21 00:00:00 until 2011.10.22 at the same time.

D [m]	# t.c.	# m.c.	CPU time [s]
1000	170013	69	221
2000	222991	506	430
5000	275149	5992	647
10000	389176	33426	1379
20000	1610676	136811	31107
25000	2848301	208911	96129

distance between two objects at time of closest approach is given in (Alarcon-Rodriguez et al., 2002) as

$$r_{min} = \sqrt{r_0^2 - \left(\bar{r}_0 \cdot \frac{\bar{v}_{rel}}{v_{rel}}\right)^2} \tag{8.9}$$

which is actually turns out to be the worked-out and scalar version which one gets when filling in Equation 8.8 in Equation 8.5.

This new scalar form of the equation gives exactly the same results as the previous vector equation, see Table 8.7 (r_{min} versus d_{tca}), and can be considered as a first (small) form of verification, as the equations were taken from different references.

Table 8.7: Verification of the determination of the minimum distance of an object pair. The time is here given in seconds past 2011.10.21 00:00:00, with d_{tca} determined by the Taylor series, r_{min} by the new scalar equation, and $d_{ps_{loop}}$ by looping with SGP4 using the newly determined time of closest approach t_{loop}.

t_{con} [s]	ID$_p$	ID$_s$	d_{ps} [m]	t_{tca} [s]	d_{tca} [m]	r_{min} [m]	t_{loop} [s]	$d_{ps_{loop}}$ [m]
22	31663	31346	2447	0.078	2446	2446	22.078	2446
9	19905	7019	5240	-0.257	4245	4245	8.743	4245
35	36198	32450	5481	0.013	5481	5481	35.013	5481
22	35797	20440	5841	-0.246	4539	4539	21.754	4539
58	32289	28858	5951	-0.412	5873	5873	57.588	5873

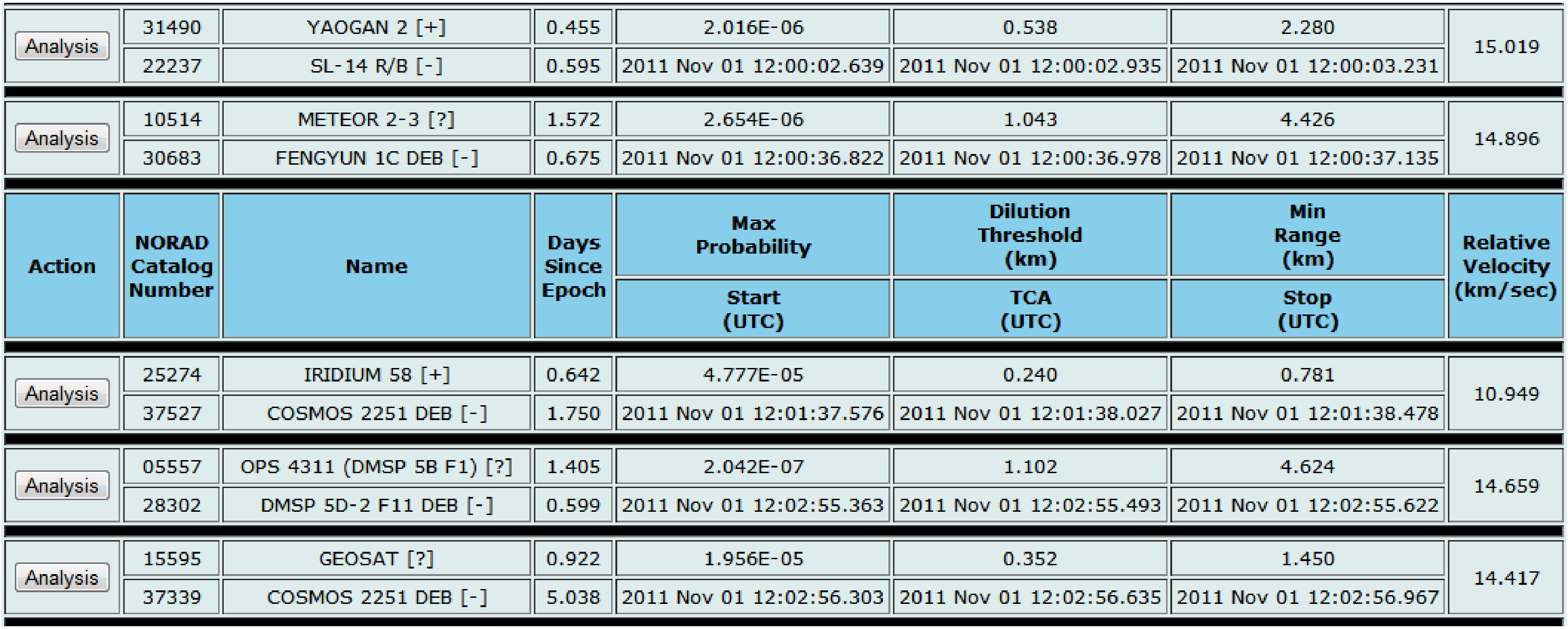
The second form of verification makes use of the SGP4 loop at the end of the brute-force process.

After determining the accurate time of closest approach and accompanying distance for an object pair with the Taylor series approximation, this data can be used as input for the SGP4 propagator again to directly determine the difference between the two objects at a time grid around that time, and thus not making use of the Taylor series approximation.

The results can be seen in Table 8.7, in which the distance of an object pair after the looped SGP4 determination, $d_{ps_{loop}}$, can be seen to match the results of the Taylor series approximation, d_{tca}. The results also show that this is in fact the real minimum distance at the right time, since the SGP4 loop used a fine time grid around the result of the Taylor series determination, and this was the minimum result of that grid.

The third and last (but definitely not least) verification method is carried out by comparing the results of the brute-force conjunction analysis method with the conjunction results that can be found in (Kelso, 2010a), using the search engine of the SOCRATES (Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space) system. The conjunction information for 5 conjunction pairs gained from that website is depicted in Figure 8.4.

A time interval around the time of closest approach (tca) of these conjunctions was now used as input for the SGP4 propagation, followed by the brute-force method with post-processing. The results of this conjunction analysis process are listed in Table 8.8, which can be seen to *exactly* match those of Figure 8.4! Two *independent* researches have hereby now thus established the exact same results.



Analysis	31490	YAOGAN 2 [+]	0.455	2.016E-06	0.538	2.280	15.019
	22237	SL-14 R/B [-]	0.595	2011 Nov 01 12:00:02.639	2011 Nov 01 12:00:02.935	2011 Nov 01 12:00:03.231	
Analysis	10514	METEOR 2-3 [?]	1.572	2.654E-06	1.043	4.426	14.896
	30683	FENGYUN 1C DEB [-]	0.675	2011 Nov 01 12:00:36.822	2011 Nov 01 12:00:36.978	2011 Nov 01 12:00:37.135	
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	25274	IRIDIUM 58 [+]	0.642	4.777E-05	0.240	0.781	10.949
	37527	COSMOS 2251 DEB [-]	1.750	2011 Nov 01 12:01:37.576	2011 Nov 01 12:01:38.027	2011 Nov 01 12:01:38.478	
Analysis	05557	OPS 4311 (DMSP 5B F1) [?]	1.405	2.042E-07	1.102	4.624	14.659
	28302	DMSP 5D-2 F11 DEB [-]	0.599	2011 Nov 01 12:02:55.363	2011 Nov 01 12:02:55.493	2011 Nov 01 12:02:55.622	
Analysis	15595	GEOSAT [?]	0.922	1.956E-05	0.352	1.450	14.417
	37339	COSMOS 2251 DEB [-]	5.038	2011 Nov 01 12:02:56.303	2011 Nov 01 12:02:56.635	2011 Nov 01 12:02:56.967	

Figure 8.4: Conjunction data of 5 objects pairs gained from the search engine of SOCRATES in (Kelso, 2010a).

Table 8.8: Conjunction results for 5 object pairs for verification with Figure 8.4. The time listed for t_{con} and t_{tca} are taken from 2011.11.1 12:00:00 onwards, t_{dfe} stands for time in days from the TLE epoch. The catalog used was that of 2011.11.01 am.

t_{con} [mm:ss]	ID$_p$	ID$_s$	t_{dfe_p} [days]	t_{dfe_s} [days]	d_{ps} [m]	$t_{tca_{rel}}$ [s]	d_{tca} [m]	t_{tca} [mm:ss]
00:03	31490	22237	0.455	0.595	2481	-0.065	2280	00:02.935
00:37	30683	10514	0.675	1.572	4438	-0.022	4426	00:36.978
01:38	37527	25274	1.750	0.642	835	0.027	781	01:38.027
02:55	28302	5557	0.599	1.405	8575	0.493	4624	02:55.493
02:57	37339	15595	5.038	0.922	5458	-0.365	1450	02:56.635

8.1.4 Results v2

After all of the above results were generated, two improvements in the way the code was run and timed lead to a necessary rerun of the brute-force method, generating new results.

Compiler Setting

It was found that the compiler used here, QtCreator, has a ‘release’ mode, next to the default ‘debug’ mode. This ‘release’ mode setting results in a much faster execution of the code.

This is due to the fact that in ‘debug’ mode, QtCreator will include additional debug symbols that one needs for debugging the application. So when the code is bug-free, and thus does not need the ‘debug’ mode anymore, it can be run using the much more efficient ‘release’ mode.

With the latter setting the CPU time was seen to decrease with a factor of 4-6, depending on the kind of code.

CPU Timer

Furthermore, a more accurate CPU timing process was implemented, by using the CPU timer from the Boost library (Dawes & Stewart, 2011), which can measure the CPU time of *only* the code that was executed. The resulting CPU time is the same as the wall clock time the code would take to run with the respective CPU at full power (100%).

So even if the program would run at say 75% of the CPU’s power, because it is also doing other things at the same time, this would not make a difference for the output. In this way different parts of code, can be compared easily in terms of CPU time performance.

Rerun of Brute-Force Method

Due to the above two new developments, it was decided to rerun the brute-force method to generate new results. Since a real CPU timer is now used, the brute-force can be compared to future analyses in a safer way. Furthermore, using the ‘release’ mode, the improvement of using a filter or sieve is much more clear.

Another solution to the problem would be to run future analyses also in the ‘debug’ mode, but the improvement of using one filter compared to a filter and a sieve for instance, would then not be that large, since most of the CPU time is used for the debugging process, which will be approximately equal for both runs.

This is due to the fact that, for instance, when using vectors in C++, the ‘debug’ mode will store a large amount of extra information per vector, to see if it evolves safely during the execution of the code, and to be able to state where in the code a bug was generated. Since the amount of vectors might be comparable for two methods, this is what causes the extra CPU time needed to be comparable. In 'release' mode this extra information is not needed, and thus large improvements between two methods can be observed.

Since the brute-force method now had to be run again anyway, some improvements were made to the method itself as well.

Conjunction event determination The theory explained with use of Figure 8.2, to determine the conjunction distance and the distance below which it was certain all conjunction would be found, is now reversed, to find *all* conjunctions within 25 km. This was done by setting r equal to 25 km, to determine the conjunction distance needed for the brute-force, resulting in $D = 27.386$ km for a time step of $\Delta t = 1$ s.

However, some conjunctions with a distance in between this new D and 25 km are now found at the time steps themselves, while still resulting in a minimum distance of above 25 km. These are subsequently filtered out during post-processing, resulting in a list of *all* minimum conjunction distances of below 25 km. This new list will thus theoretically be able to contain more conjunctions than the previously applied method.

Post-processing: One minimum conjunction Next to the above, the post-processing function to determine only one minimum conjunction per object pair is now rewritten as well. It will now be able to detect multiple minimum conjunctions events for one object pair, as two objects in fairly constant orbits might have a conjunction again when they have both made one extra revolution.

It will still filter out the multiply defined conjunction information though, affiliated with the same conjunction *event*. Next to this new implementation, it was also written much more efficiently, resulting in a significantly lower post-processing time. Furthermore, the final results are now sorted ascendingly with respect to the distance at closest approach.

An example of the top-10 of the final output file is given in Table 8.9.

Table 8.9: Example of the top-10 of a v2 output text file after post-processing. The catalog file used is of 2011.10.20 am, and the time interval under analysis is a 1-day period from 2011.10.21 00:00:00 until 2011.10.22 the same time.

$tcon$ [JD]	ID_p	ID_s	d_{ps} [m]	$\cdots$	t_{tca} [s]	x_{tca} [m]	t_{tca} [JD]
2455855.50000000	37396	37633	0		0.000	0	2455855.50000000
2455856.05105324	8004	17122	978		0.072	37	2455856.05105407
2455855.69494213	12962	30885	7212		0.490	41	2455855.69494779
2455856.05793981	33999	36304	1057		0.070	53	2455856.05794062
2455856.47300925	32342	34118	1610		0.185	78	2455856.47301139
2455856.20203703	4673	37061	6488		0.441	130	2455856.20204213
2455856.47914351	32232	37088	610		-0.095	132	2455856.47914241
2455856.44265046	4755	12652	4787		0.325	144	2455856.44265423
2455855.75068287	8159	23004	3149		0.224	152	2455855.75068546
2455855.95745370	20324	37415	877		-0.059	153	2455855.95745302

Table 8.10: Performance results of the new improved brute-force method.

			CPU time [s]			
T [s]	# t.c.	# m.c.	SGP4	brute	post	total
1	96	42	1	2	0	2
60	2751	232	8	51	0	59
3600	151240	10597	453	2988	1	3442
86400	3552394	255712	10793	71742	575	83109

Results The results of the new improved brute-force method can be seen in Table 8.10. Comparing these new results to Table 8.5, it is noted that the total number of conjunction is increased, due to the increasing D, and furthermore the number of minimum conjunctions is also increased, since now *all* conjunctions within 25 km are found, and also due to object pairs having multiple conjunction events. Furthermore, the CPU time for SGP4 propagation has only slightly decreased due to the fact that the SGP4 theory is written in C instead of C++, using arrays instead of vectors, and thus less debug information was used for the 'debug' mode.

However, the CPU time needed for the brute-force method is decreased with a factor of more than 4 due to the new 'release' mode. Furthermore, post-processing has gained a huge decrease in CPU time due to the combined effect of the 'release' mode and the more efficient rewritten code. And, when the post-processing was applied to the output of the brute-force method directly, after 50 seconds of reading in data, the new post-processing implementation only took 9 seconds to complete. This thus suggests that it would be more efficient to first write the data to a file, and then read it again to apply post-processing, but this was not implemented further.

Although part of the previous reasoning to choose a time-step of 1 second is now diminished by the faster post-processing times, the rest of the reasoning still applies. Thus the time step of 1 second was kept, as with larger time steps the Taylor approximation for the minimum conjunction distance would be less accurate.

Furthermore, using a larger time step will increase the needed conjunction distance D, and will thus also increase the amount of total conjunctions detected for which information has to be kept in memory during the process. The conjunction data written to a file is already around 500 MB for the current setup.

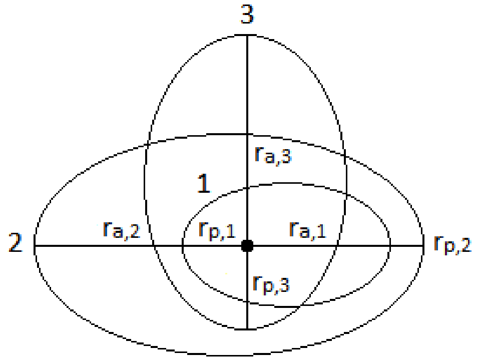


Figure 8.5: Schematic of different cases for the perigee-apogee filter, showing a filtered (#2) and an accepted (#3) candidate for a possible conjunction with a primary object (#1). Note: not to scale, the 'ovals' in the picture represent ellipses.

8.2 Perigee-Apogee Filter

Filters can be applied in a conjunction analysis process, to discard pairs of objects which are proven never to able to conjunct in the time interval under analysis. As applying these filters to one object pair can be very quick, compared to analysing a pair for the entire time interval with the brute-force method, the total conjunction analysis system can be made much more efficient, compared to the brute-force method.

The perigee-apogee filter (also called altitude filter), first described in Hoots et al. (1984), filters out object pairs for which the apogee of one of the two objects is smaller than the perigee of the other. Let q denote the larger of the two perigees of the objects and let Q denote the smaller of the two apogees. If

$$q - Q > D \tag{8.10}$$

then the pair does not need to be considered further. All object pairs that pass this prefilter move on to the next. This is illustrated in Figure 8.5, where a filtered (#2) and an accepted (#3) candidate for a possible conjunction with a primary object (#1) is shown. The combination of object #2 and #3 would also form a pair that would not be filtered by the perigee-apogee filter and thus continue to the next filter and/or sieves. Note: the orientation of the orbits does not make a difference.

The strength of this filter lies in the fact that if a pair of objects is filtered out, they need not be considered any more for the entire time interval under analysis. Furthermore, the perigee and apogee need only be determined once, per object, while the application of the filter then only consist of comparisons.

The objects will 'never' be able to have a conjunction, since they do not have an overlap in their altitude ranges. In fact, the difference must be at least equal to the conjunction distance D, for no conjunction to occur. However, since the apogee and perigee of both objects change over time, one has to account for this perturbation as well in the filter.

8.2.1 Hoots

In Hoots et al. (1984) the perturbations were taken into account, by calculating the apogee and perigee for both objects at the start of the time interval under analysis, based on the orbital elements at the middle of this time interval. This was done by using approximate rates of the mean motion and eccentricity, determined from the time derivative of the mean motion, and so accounting for perturbations due to the Earth's oblateness and drag. The perigee and apogee were calculated at the beginning of the time interval, since it was assumed that the perigee would remain constant, while the apogee would decrease with time. And thus, the filter would still remain valid.

It was not stated what kind of orbital elements to take at the middle of the time interval, but it is assumed that the mean elements were taken, as the osculating elements would lead to more erroneous results.

However, there are some problems with this theory. Although the time derivative can be obtained from a TLE, this average value will only be valid at the TLE epoch, and thus not necessarily during the time interval under analysis or at its midpoint. Furthermore, when looking at radius evolutions of multiple objects from the catalog, both the apogee and perigee can be seen to both increase and decrease with respect to time for different objects. This can for instance be seen in Figure 8.6, were the radius and osculating apogee and perigee of object #13901 are plotted versus time since epoch. The plot shows that the apogee is *increasing* and the perigee is *decreasing* in this case.

Furthermore, when using this theory, the mean orbital elements at the midpoint of the time interval need to be obtained from the ephemerides somehow and thus from the osculating orbital elements. An average could be taken over one or multiple orbital periods, but this would introduce even more uncertainty to the method. Furthermore, as not all perturbations of the SGP4 theory are taken into account here, this would introduce another error, especially for high-altitude orbits where then the third-body perturbations are not taken into account.

All of these error sources add up and therefore this method is not deemed reliable for the perigee-apogee filter.

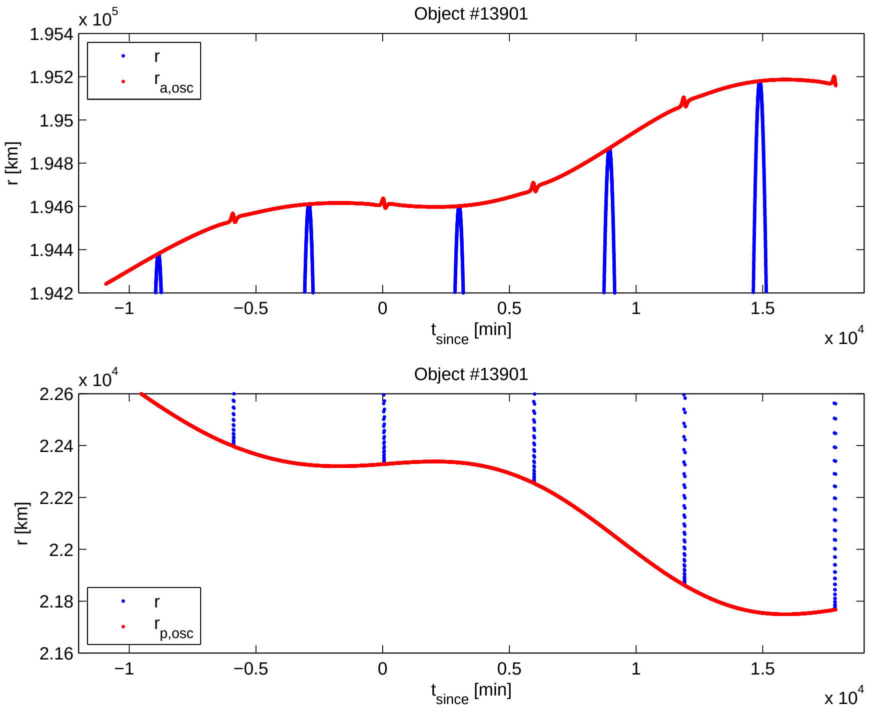


Figure 8.6: Radius and osculating perigee and apogee versus time since epoch.

8.2.2 Ting & Hai

In Ting & Hai (2008) the equation was changed to

$$q - Q > D + D_1 \tag{8.11}$$

where D_1 accounts for the sum of orbital decay and short-period altitude variations. It was stated there that calculations had shown that a typical value of D_1 is 10 km while it is also stated that for LEO orbits it is roughly 20 km, while not stating for which time period this holds. Later calculations in the publication are applied to a time interval of 1 day, so this is assumed as the time period for D_1 as well.

Furthermore, it is also unclear how the apogee and perigee were obtained, and which value was taken for D_1: 10, 20 or 30 km? Looking at Figure 8.6 it can be seen that both the apogee and perigee negatively influence the perigee-apogee filter with a change of more than 50 km per day, while this is not per se a worst-case scenario. D_1 would thus at least have to be 100 km in this case.

Four methods, of which three are new methods purposely designed for this MSc thesis study, were considered however, which will be described below.

For one of the three new method equations are used from a reference however.

8.2.3 Directly from TLE

The fastest way to obtain the apogee and perigee for an object would be to directly determine them from the TLE, using the eccentricity and mean motion. Since these are mean values at the TLE epoch, a certain error will be made with respect to the real apogee and perigee during the time interval under analysis. This error can be quantified however.

When first generating ephemerides for all objects for the time interval under analysis, using a very small time-step $\Delta t = 1$ s, and subsequently determining the minimum and maximum radius r_{min} and r_{max} for each object during this time interval, a 'truth' reference is set up. The error Δr_p and Δr_a made for each object is then simply obtained by subtracting the perigee from the minimum radius and the apogee from the maximum radius.

$$\Delta r_p = r_{min} - r_{p,mean} \tag{8.12}$$

$$\Delta r_a = r_{max} - r_{a,mean} \tag{8.13}$$

The maximum error that is made for the whole catalog with respect to both the perigee and apogee can now be determined, which can be used as a patch. Adding these patches to all obtained average values for perigee and apogee directly obtained from the TLE would lead to a safe application of the perigee-apogee filter. This can be equated as

$$(q - P_p) - (Q + P_a) > D \tag{8.14}$$

$$q - Q > D + P_p + P_a \tag{8.15}$$

where P_p and P_a are the (absolute) perigee and apogee patches respectively.

For the catalog of 2011.10.20.am, this maximum error negatively influencing the perigee-apogee filter would result in $\Delta r_{p,max} = -135.141$ km for the perigee patch (P_p is the absolute of this value in the equation above), and $\Delta r_{a,max} = 338.531$ km for the apogee patch. The resulting filter would lead to 29.56% of the total amount of pairs being discarded, thus making the filter relatively inefficient, as we will see later on.

Object #25811 generates the worst-case apogee error, which is shown in Figure 8.7, where the mean apogee from the TLE (constant), as well as the radius and osculating apogee are plotted versus time since TLE epoch.

The 'wobble' that can be seen in the apogee history, stems from the osculating semi-major axis and eccentricity, see Figure 8.8, which result from the SGP4 theory. The semi-major axis and eccentricity can be seen to amplify each other when generating the apogee, while they partly cancel each other out for the perigee, as for the latter $1 - e$ is used.

An improvement could be made to this method by not conservatively taking the patches constant and equal to the maximum error, but by trying to find a relation between the error made per object, and some other variable of the respective TLE.

In Figures 8.9 - 8.12, the error in perigee and apogee made per object is plotted versus a number of TLE variables. Here only the negative values for the perigee error and positive values for the apogee error are shown, as these are the ones negatively influencing the perigee-apogee filter. The mean motion n, eccentricity e and semi-major axis were chosen, as they might have the largest chance of showing some relation with error.

As can be seen from the first three of these plots, there exists no clear relation which falls closely on top of all the errors for any of the plots, both perigee and apogee, partly due to some large errors made for only a few objects. A relation could be set up, but then the patch would be relatively large with respect to most of the objects at hand. Furthermore, the larger errors made seem to be randomly distributed, so even if a relation could be set up for this catalog, it would certainly not hold for *any* catalog.

It might be that these larger errors are due to old TLEs, that is, objects for which the epoch of the TLE would be relatively distant from the analysis interval. Since these objects would have to be propagated forwards in time to a larger offset compared to their TLE epoch (and thus when the measurements of their orbits were made), the resulting ephemerides might show a larger difference from the values in the TLE. In Figure 8.12 it can be seen that although some objects with a distant TLE epoch show to have large errors, an evenly large amount of objects with a large TLE epoch, that is, “fresh” TLEs, can also have large errors. Note: the time interval that was analysed lies at the far right, around day 295.

One might argue that the objects that produce the extreme errors could be excluded from the analysis, but what if one of these objects then were to have a collision with the ISS? Furthermore, these excluded objects might have smaller errors in future catalogs, while other objects will have larger errors.

8.2.4 Modifications for Perturbations

Although the mean perigee and apogee determined directly from the TLE do not generate good results for the perigee-apogee filter, the process described in Hoots et al. (1984) to include perturbations might be modified for use with TLEs.

It is suggested there, that mean effects of the Earth’s oblateness and

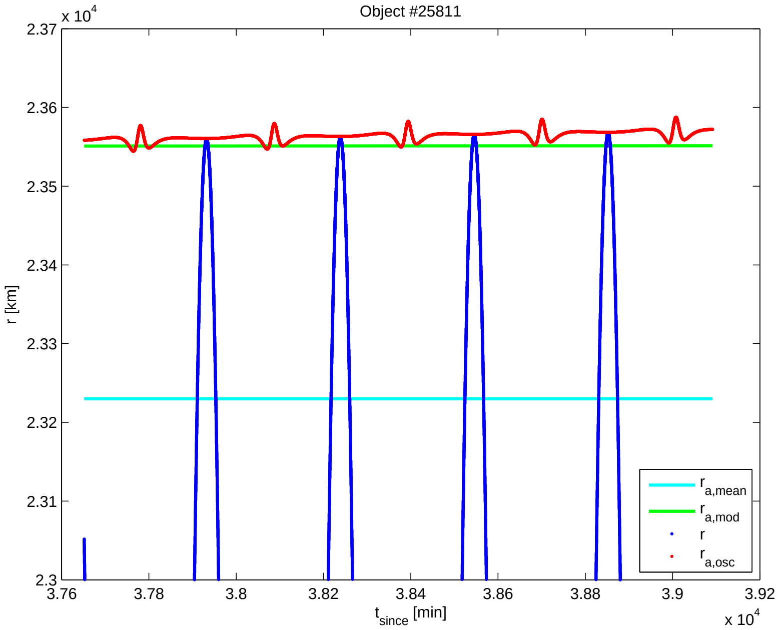


Figure 8.7: Mean (constant) and modified (linear) apogee from TLE, radius and osculating apogee versus time since TLE epoch.

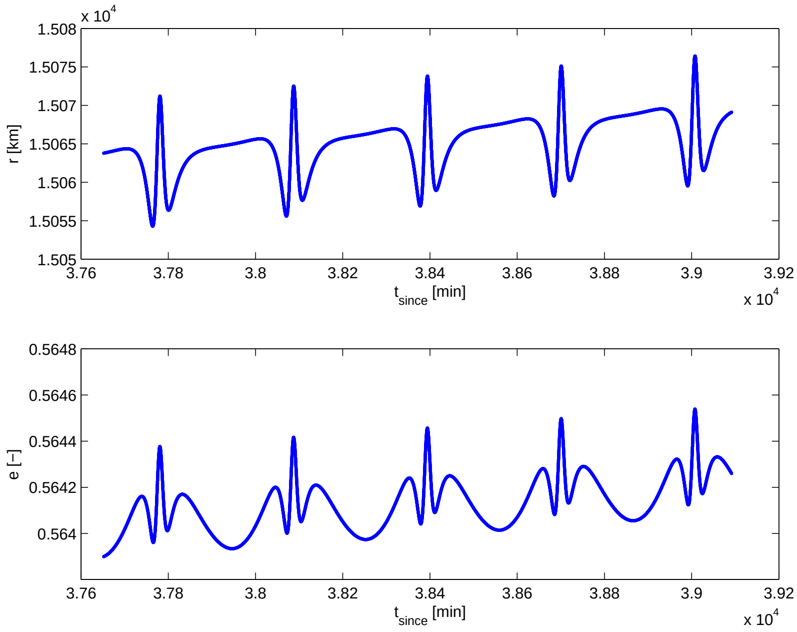


Figure 8.8: Semi-major axis and eccentricity versus time since TLE epoch.

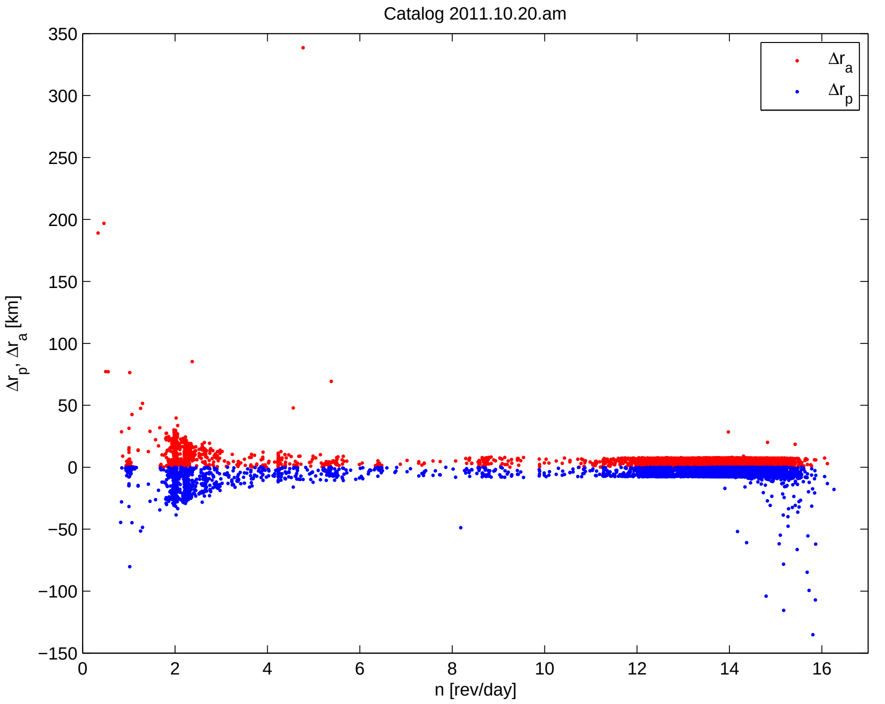


Figure 8.9: Mean perigee and apogee error versus mean motion n.

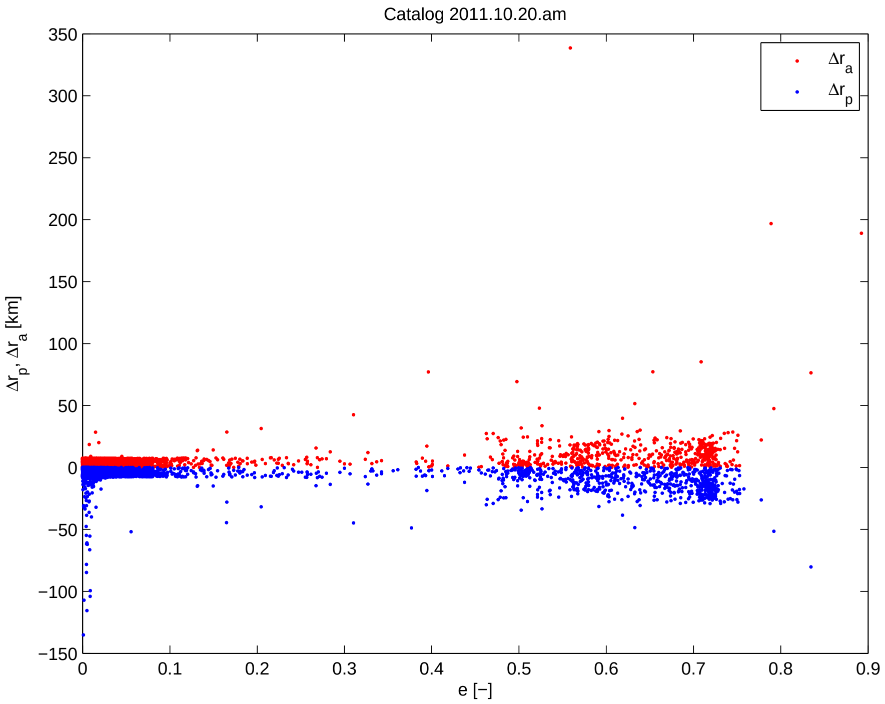


Figure 8.10: Mean perigee and apogee error versus eccentricity e.

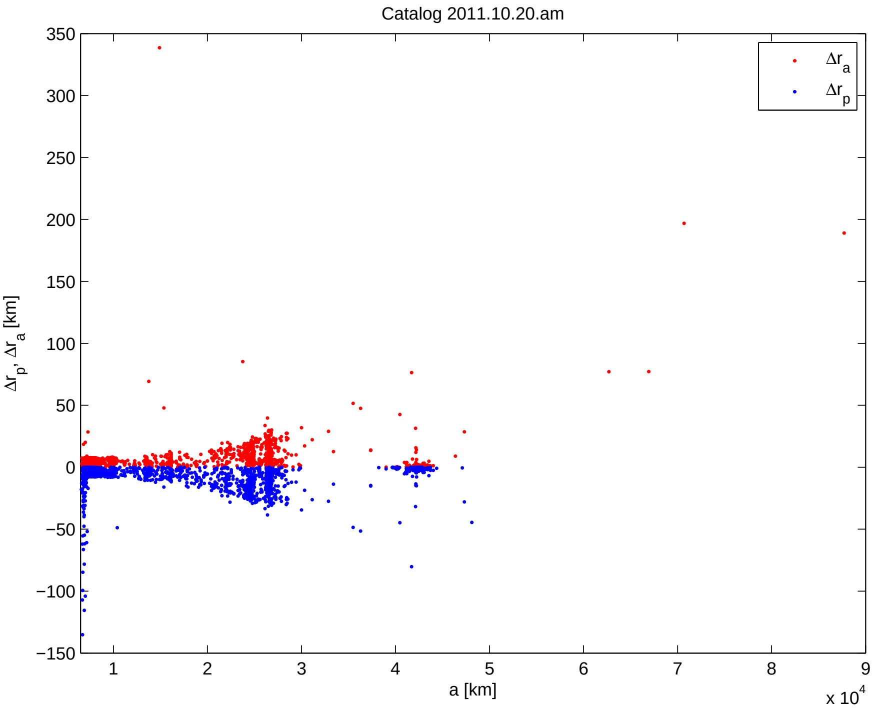


Figure 8.11: Mean perigee and apogee error versus semi-major axis a.

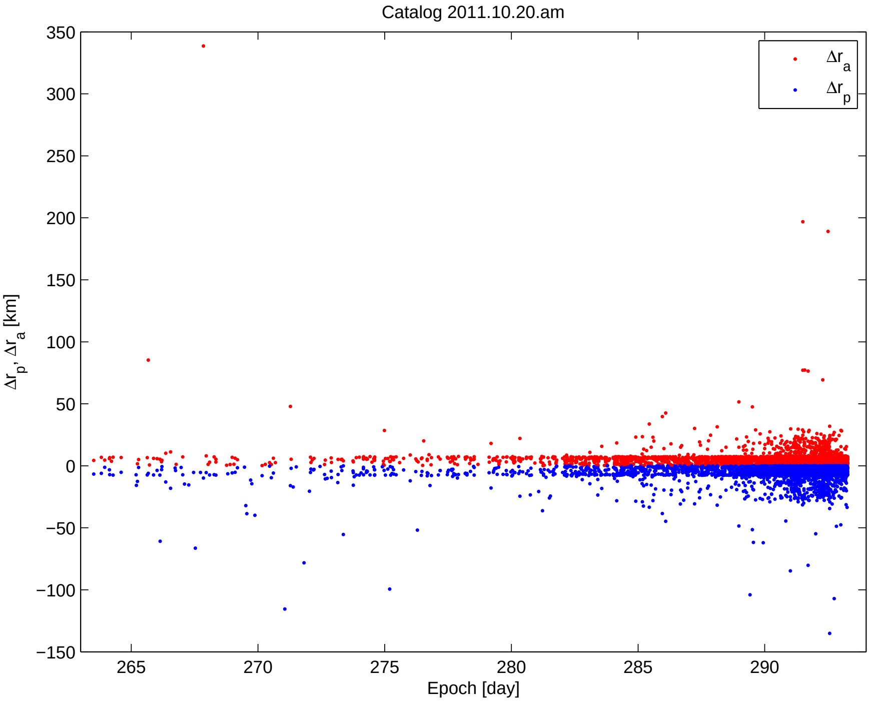


Figure 8.12: Mean perigee and apogee error versus TLE epoch.

atmospheric drag can be approximated by

$$n = n_0 + \dot{n}_0 t \tag{8.16}$$

$$e = e_0 + \dot{e}_0 t \tag{8.17}$$

where the time derivative of the eccentricity is given as

$$\dot{e}_0 = -\frac{2}{3}\frac{\dot{n}_0}{n_0}(1 - e_0) \tag{8.18}$$

Using these equations, the mean motion n and eccentricity e can both be determined at the start and end of the time interval. The time t is then the start and end time minus the TLE epoch, while the mean motion n_0, its time derivative $\dot{n}_0$, and eccentricity e_0 can all be taken straight from the TLE.

Following, the mean perigee and apogee at the start and end time of the interval can be calculated, and the perigee is then set to the minimum of these two values, while the apogee is set to the maximum, as they change linearly with time.

Applying this technique to for instance to object #25811, which generated the largest error above, the modified apogee $r_{a,mod}$ can be seen to follow the maxima of the radius much better, see Figure 8.7, resulting in an error of only 17 km.

However, when applied to the whole catalog for the same 1-day analysis, the largest error made for the apogee is still 189 km while the maximum perigee error is -80 km, resulting in 36.36% of the pairs discarded. The radius history of the object producing the largest error for the apogee now, object #3956, is shown in Figure 8.13.

Furthermore, the errors made for the perigee and apogee for all objects were plotted versus the same variables as in was done in Figures 8.9 - 8.12 (and now also including the time derivative of the mean motion), resulting in the same kind of scatter for which unfortunately no clear relation exists, to make the filter more efficient, and are therefore not included here.

8.2.5 Sampled from Osculating Elements

Another method for an application of the perigee-apogee filter that was investigated is the one presented in Woodburn et al. (2009), as described in Section 5.1. Here an improvement of the filter was discussed, by determining the osculating orbital elements at the start- and endpoint of the time interval, and using these to obtain the minimum perigee and maximum apogee radii of the two time instances.

It was then shown that in doing so, an error still might be made. Thus the difference between the real perigee and apogee and the osculating ones

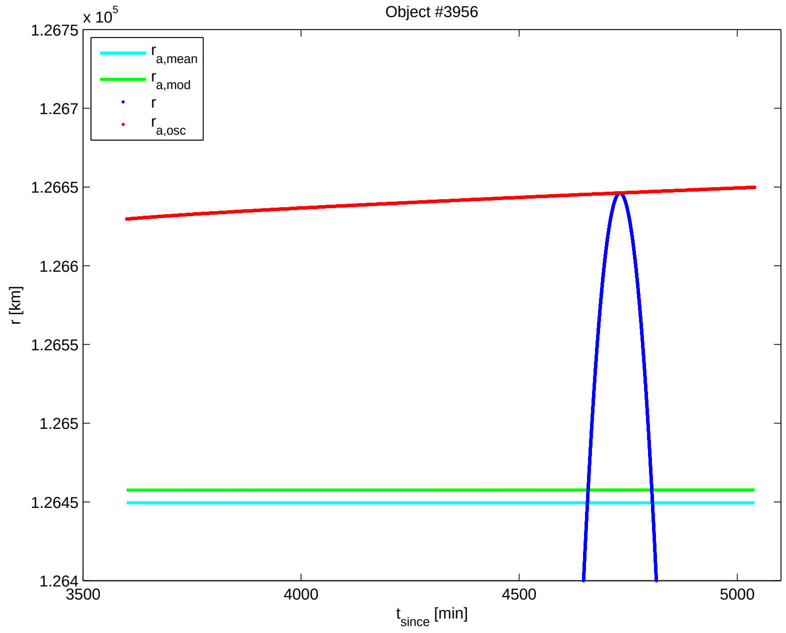


Figure 8.13: Mean (constant) and modified (linear) apogee from TLE, radius and osculating apogee versus time since TLE epoch.

were plotted for all objects, and was shown to have a maximum of about 30 km for both cases. This 30 km was then used as a patch for both perigee and apogee for the application of the filter for *all* future analysis (and thus also different catalogs).

However, while the error that is made by using the osculating elements at the start- and endpoint of the time interval might have a maximum of 30 km in *this* case (catalog, time interval), this certainly will not necessarily hold for *all* future cases.

For instance, when looking at the history of the radius and osculating apogee of GTO object #23687 in Figure 8.14, if the start- and endpoint of the time interval were to both fall precisely into a valley of the osculating apogee graph, the apogee determination would be off by about 46 km in this case. While this is just one of the worst-case examples, it could be that there exist objects for which even larger error are possible.

Applying this method with a patch of 30 km for both the apogee and perigee at the catalog of 2011.10.20.am for a 1-day analysis yields 49.11% of the pairs to be discarded. However, since this patch is shown to be unsafe, running it with a safer patch of 50 km results in 45.57% pairs discarded.

Furthermore, applied to different catalogs and time intervals, this method could result in larger errors as well. And, if one would need to check each

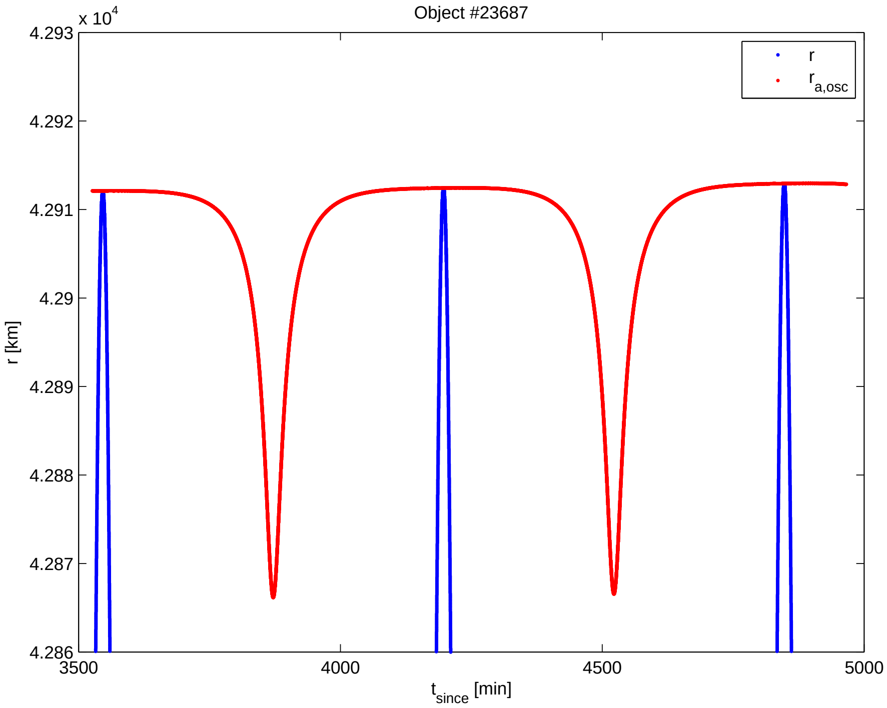


Figure 8.14: Radius and osculating apogee versus time since epoch.

catalog for the maximum error made, which is done by comparing it to the real minimum and maximum radius, why not directly determine the 'perigee' and 'apogee' from the radius history itself?

8.2.6 Determined from Ephemerides

By taking the minimum and maximum attained value for the radius over the time interval under analysis, determined from the ephemerides, and using these for the perigee-apogee filter, one would be sure that no error is made by obtaining these values via other variables. However, an error can still be made due to the fact that these extrema can only be determined with an accuracy dependent on the time step Δt. The smaller this time step, the better a minimum or maximum radius can be approximated.

This can be seen in Figure 8.15, in which the radius history of object #7373 is plotted for time steps of 60, 30 and 1 second. If the minimum radius (perigee) would be determined using a time step of 60 seconds, it would be off by more than 2.5 km, whereas using a time step of 30 seconds would in this case yield a result relatively close to the exact minimum (only 10ths of meters off).

Using a larger time step will decrease the CPU time, while increasing the error made due to approximation. And since the largest of this error

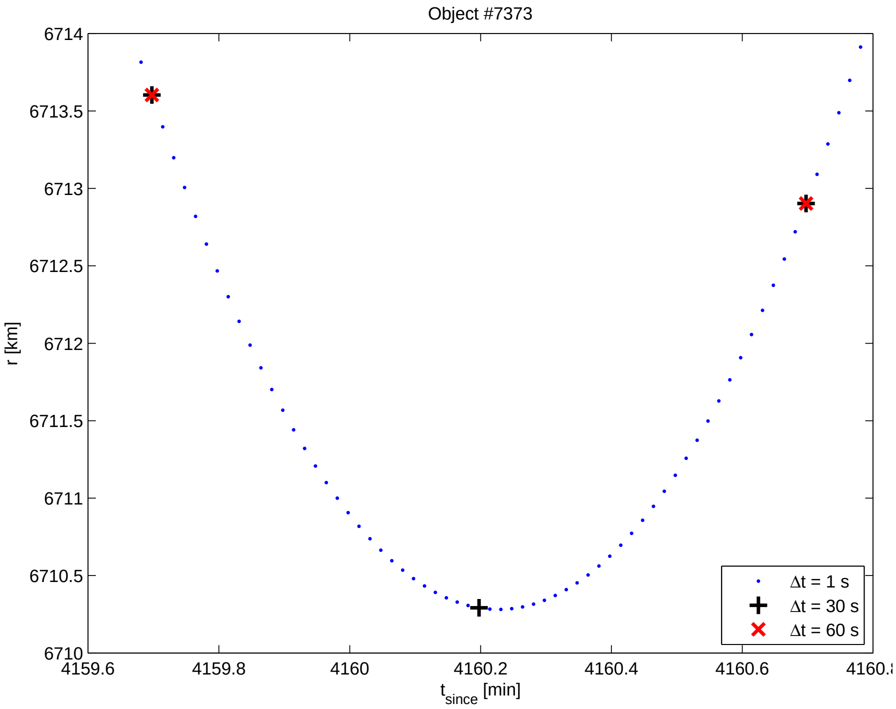


Figure 8.15: Radius versus time from TLE epoch with different time steps.

in the entire catalog has to be used as a patch for all catalog objects when applying the filter, this will also decrease the performance of the filter, in terms of percentage of discarded pairs.

To determine the performance of the perigee-apogee filter based on different time steps Δt, a reference was made based on $\Delta t = 1$ s. For a number of time steps, the ephemerides for the entire 1-day interval were generated for all objects. From these ephemerides the minimum and maximum attained radius per object were taken as r_{min} and r_{max}. These two values were now subtracted from the r_{min} and r_{max} obtained by using $\Delta t = 1$ s, and subsequently, the maximum error made in these two variables of all objects, is indicated in Table 8.11 as ΔR_{min} [m] and ΔR_{max}.

The error can be seen to decrease rapidly, and so using this data, a safe guess on the exact error made by using $\Delta t = 1$, compared to the real r_{min} and r_{max}, would be somewhere below -10 m for r_{min} and below -1 m for r_{max}. A plot of the error made for all objects and $\Delta t = 60, 30, 20, 10$ s can be seen in Figure 8.16.

The performance of the perigee-apogee filter for each of these time steps can now be determined by using the relevant Δr_{min} as a patch for the perigee

Table 8.11: For different time steps Δt, the maximum error made by determination of r_{min} and r_{max} of all objects, with respect to $\Delta t = 1$ s, next to the percentage of rejected over analysed r/a pairs by the resulting perigee-apogee filter, and the CPU time needed for the entire process of determining these patches.

Δt [s]	ΔR_{min} [m]	ΔR_{max} [m]	r/a [%]	t_{CPU} [s]
3600	-6215122	675518	16.01	1
1800	-2062665	212197	19.06	2
600	-249846	21888	36.34	6
360	-98585	7992	45.78	10
180	-21148	2372	54.42	19
60	-2620	306	57.08	56
30	-740	85	57.36	112
20	-334	38	57.42	168
10	-84	8	57.46	334
1	-	-	57.47	3388

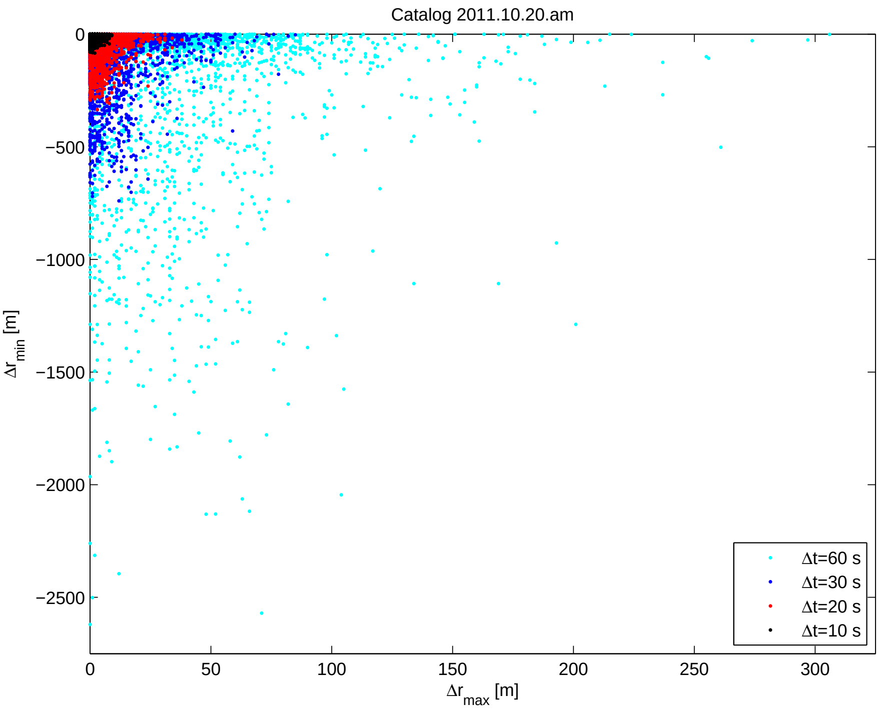


Figure 8.16: Error of r_{min} and r_{max} relative to $\Delta t = 1$ s, per object.

and Δr_{max} as a patch for the apogee, plus adding the assumed maximum error made by $\Delta t = 1$ itself as respectively -10 and 1 m. This results in the percentages of discarded pairs indicated in the next column, while using a conjunction distance D of 25000 m. Furthermore, it is noted that using no patch, and the minima and maxima determined by the ephemerides with Δt=1 s, the result is 57.47% pairs discarded, while using no patch and a conjunction distance of 0 m, 61.04% of the pairs are discarded.

Finally, the CPU time needed for the process of determining the patches for each of the time steps is listed in the table as well. The CPU time needed for merely applying the perigee-apogee filter to the whole catalog with the already determined patches (to filter out the pairs), is in all cases approximately 1.5 seconds, and is not included in the table. A trade-off can now thus be made between the percentage of pairs discarded and CPU time needed. This trade-off can be done when the CPU time needed for the entire conjunction analysis process is known, as it depends on the performance gain it creates for the entire process.

The first three methods analysed in Subsections 8.2.3 - 8.2.5 could be processed within approximately 1, 2 and 4 seconds CPU time respectively. Now comparing the results of Table 8.11 to the 'directly from TLE' method, to gain a performance of about 29% discarded pairs, the CPU time needed to establish the safe filter discussed here, only 4 seconds of CPU time are needed. Furthermore, compared to the method with modifications for perturbations, for a performance of about 36% one would only need 6 seconds here. Finally, comparing it to the filter based on osculating elements, to gain a performance of about 45%, this new method needs on the order of 10 seconds.

8.2.7 Reliability in Permanent Application

The first of the two filters (Subsections 8.2.3 - 8.2.4) yield semi-random errors for the determination of the perigee and apogee, due to the fact that these errors are produced by observation errors of the object when producing the TLE. Although the errors determined for the catalog at hand can be used for a safe patch applied to conjunction analysis for that same catalog, one must be careful in selecting a safe patch for future analyses with different catalogs.

For a determination of these safe patches, one could analyse multiple catalogs in a row, and from the absolute maximum errors of all these catalogs, determine a safe patch for all future conjunction analyses. Furthermore, this could be repeated after some time, say, a number of months, to see if the patches are still valid then; if not, the patches could be updated.

This analysis could be run separately to the real-time ongoing conjunction analysis, therefore not hampering this process in terms of CPU time needed. running parallel to the ongoing conjunction analysis, it would have no practical CPU time limitation.

If, in a worst-case scenario, while analysing a future catalog with patches determined pre-hand using earlier catalogs, an object would unknowingly generate an error which would be larger than the patch size, the probability that this object would have a conjunction which would not be detected is very small. Since this primary object would have to be close to a secondary object at exactly its perigee or apogee, while the secondary object would *also* have to be at its apogee or perigee, respectively, at that same moment, while *also* having an error close to the conservatively determined patch size. And since the occurrence of any of these conditions has a low probability by itself, a combination of these conditions would have a very low probability to occur.

It therefore can be said that while the application of these two methods for the perigee-apogee filter can be executed in a very safe way, it can never reach a perfect application with 100% security to always safely filter only non-conjunctions.

For the third method analysed, depending on osculating elements (Subsection 8.2.5), the maximum size of the local minima of the osculating perigee and apogee could be determined for all objects for multiple catalogs. As the maximum error for the determination of the perigee and apogee for this filter depends on this local minimum, a safe patch can be determined based on the largest local minimum found.

Since these minima are produced by the SGP4 theory, and are large especially for HEO and GTO objects, they are less random than the errors of the former two methods. When multiple catalogs are now analysed, one can be fairly sure that the largest error which can practically be made by the SGP4 theory in this way, is caught.

And since furthermore the same conditions, like described above, must apply for a worst-case scenario conjunction not to be detected, it therefore can be said that a filter using this method can be applied in an even more safe manner than using the previous two methods.

The last method described above (Subsection 8.2.6) yields the errors made in the determination of the perigee and apogee from the inaccuracy of the approximation of the extrema of the radius generated from ephemerides. The maximum error of the catalog for a certain time step like indicated in Table 8.11 will always be a worst-case scenario like shown in Figure 8.15 for $\Delta t = 60$ s, and is most likely to be produced by HEO and GTO objects.

Since the catalog contains a large number of objects, the error produced by the worst-case scenario for a certain time step will always lie very close to the error that could ever possibly be reached by that catalog (i.e. resulting in a worst-case scenario like for $\Delta t = 60$ s in Figure 8.15). Certainly, if multiple catalogs are analysed, this error will lie even closer to that theoretical maximum.

It therefore can be said that the patches for this method can be determined very safely as well, perhaps around the same order of safety of the filter based on osculating elements, due to the fact that these are both produced by theoretical mathematical models, and are not based on measurement errors, like the first two methods.

Since the last method is, of the two most reliable methods, the only one implemented fully here, and can be fine-tuned with respect to the percentage of pairs to be discarded versus CPU time, while operating at comparable performance in terms of computation time, this is the one that is selected in subsequent analysis.

8.3 Sieves

Next to the filters which are applied pre-hand, another way to decrease the conjunction analysis processing time is by the application of (a) sieve(s) to the ephemerides, as discussed in Section 5.1. These sieves can be applied consecutively to the ephemerides generated at each time instance, each time discarding a percentage of the pairs that passed the previous sieve, in order to reduce the search space for conjunctions as quickly as possible.

At first, an implementation of the sieves was made like originally described in (Alarcon-Rodriguez et al., 2002), to test this theory in terms of performance (Subsection 8.3.3). However, the last 'sieve' which accurately determines the conjunctions within the time step, was implemented in a different way here (Subsection 8.3.1). Originally this was done by numerically finding null values for v_{app}^2 to accurately find the point of closest approach. But since the theory to do this was not described in the publication, a different approach was used here, making use of the Taylor series approximation for the minimum conjunction distance and time described in Section 8.1. The implementation of the entire process including the perigee-apogee filter is then discussed with a flow diagram in Subsection 8.3.2.

The sieves can also be applied separately to test their performance, so the optimal combination and order of the sieves can be found (Subsection 8.3.4). Improvements to the existing sieves are described in Subsection 8.3.5. Once the optimum combination of improved sieves is found, the time step can be optimised (Subsection 8.3.6), because the theory for these sieves is variable with Δt.

Following, the chosen perigee-apogee filter can be optimised with its own variable time step, in combination with the now determined optimum conjunction analysis process (Subsection 8.3.7). Then, a comparison is made with the results of previous method described in references (Subsection 8.3.8) Concludingly, some other improvements which could not be implemented during the course of this thesis (due to time constraints), are discussed in Subsection 8.3.9.

8.3.1 New Fine Conjunction Detection Method

When the optimal time step for the conjunction analysis process making use of the sieves is rather large, say, in the order of minutes, the Taylor series approximation will be less accurate, thus making it difficult to compare the results, in terms of conjunctions found, of the sieves method, with the brute force method. Because, when a conjunction is found by the brute force method, which makes uses of the Taylor series to determine a minimum conjunction event within a time step of one second, this conjunction would be more accurately determined, than with a time step of one minute.

So if there is a conjunction found by the brute force method that is just below 25 km, using the same Taylor series method but for a larger time step for the sieves method, the same conjunction could be determined at little over 25 km, and would in the latter case not be count as a conjunction. Both methods must thus have the same order of accuracy in the determination of a conjunction event. This is done as follows.

Since a number of calculations needed for some of the sieves are also used in the Taylor series approximation to determine the minimum conjunction event, they can be easily reused and implemented here. Furthermore, the Taylor series approximation was already applied in the brute force method, so its theory is known.

The method implemented here makes a first estimation of the conjunction event within a larger time step Δt (which could be in the order of minutes, see Section 5.1), making use of the Taylor series approximation method, which was explained before. The accuracy of this method is now relatively poor compared to the method applied within the brute force conjunction analysis, where a time step of 1 second was used.

Thus, the time instant now determined for this conjunction event is used in combination with the TLEs of the two objects, to propagate both objects to this time instant. The TLEs can be obtained per object directly from the TLE reader, where each TLE separately is mapped to its object ID.

This propagation results in new state vectors, which can be used for another Taylor series approximation of the conjunction event, resulting is a more accurate determination.

This method can now be iterated to produce the required accuracy, which is here set to stop when $t_{closest}$ is smaller than 0.5 seconds, as was the case for the brute force method, resulting in a comparable accuracy.

8.3.2 Implementation of Sieves

In Figure 8.17, a flow diagram of the total conjunction analysis process including the application of the perigee-apogee filter and the sieves can be seen.

The process starts off with the ephemerides file generated by the SGP4 propagator which was already treated in Chapter 7, and with the TLE data generated by the TLE reader which was treated in Chapter 6.

Then, the perigee-apogee filter is applied with a patch matching the maximum error in perigee and apogee. The filter needs a file with the minimum and maximum radius of each object. This file is generated beforehand, separately, using the SGP4 propagator with a certain time step on its own. Since for some analyses the perigee-apogee filter was not used, it can also be set off, and will then not be included in the process. The remaining pairs then go to the sieves process, which starts out by setting the critical radius R_{cr}, after which R_{th} and R_{acc} can be determined. For improvements 4-5a, these variables are calculated within the sieves for each pair.

The combination and order of the sieves also needs to be set, after which the ephemerides file can be started to be read in. After the ephemerides of one time instance are read in, the sieves are run on this time instance, and only on the remaining pairs of the perigee-apogee filter.

The pairs that passed all tests, continue to the fine conjunction detection, which loops through the SGP4 propagator to find the conjunction event with minimum distance. Once this has been found, it adds the conjunctions with a minimum distance smaller than 25 km to the list of conjunctions.

This process is then looped until all time instances in the ephemerides file have been analysed, and thus all conjunctions have been found. This data is written to a file. Since multiple defined conjunction events can still exist now, these are filtered out after which the resulting conjunction data is sorted and written to another file.

Furthermore, the performance in terms of average percentage of rejected pairs with respect to the analysed pairs per sieve, and the CPU time of all separate steps, is given as output.

8.3.3 Performance Results of the Original Method

After all of the sieves were implemented successfully like described in Alarcon-Rodriguez et al. (2002) (excluding the time skipping process), with the new fine conjunction method at the end, some performance results could be gen-

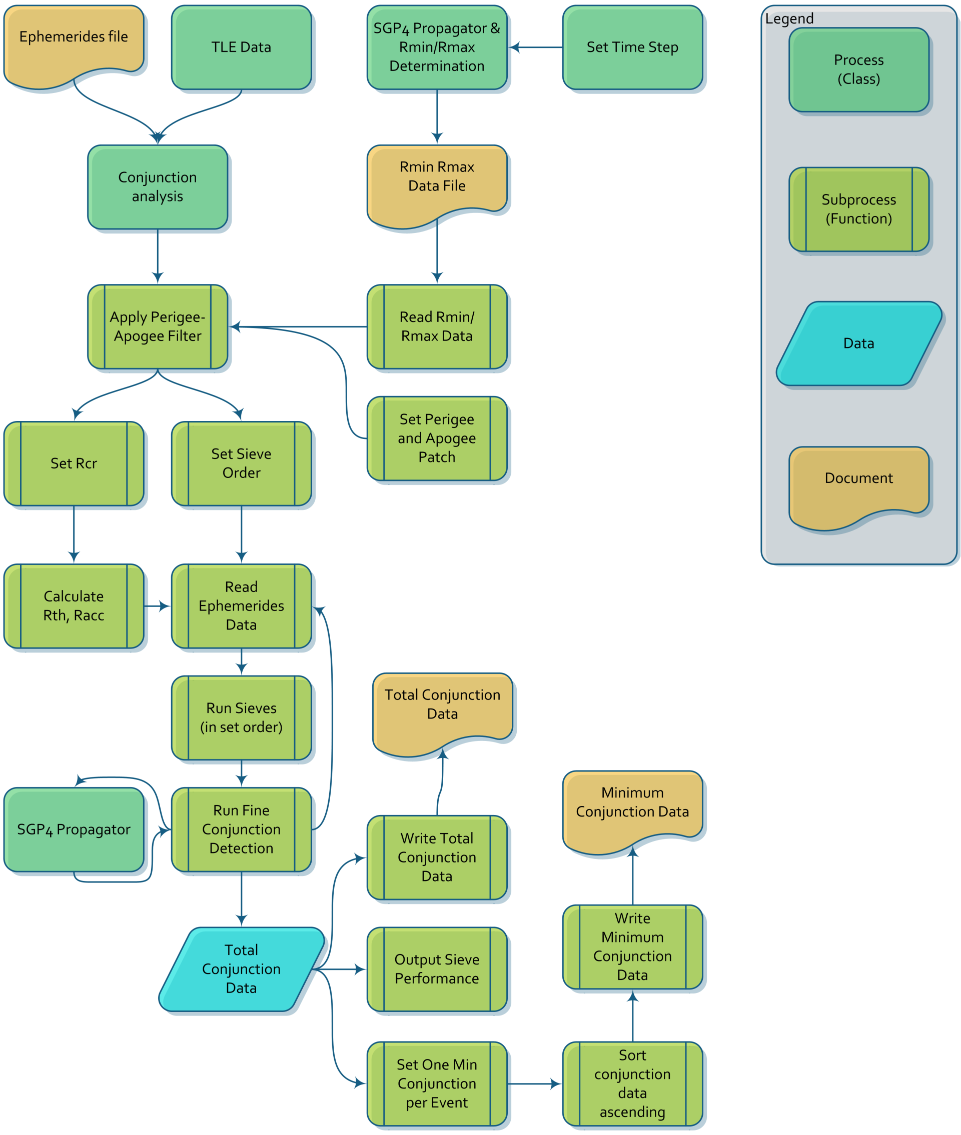


Figure 8.17: Flow diagram of the conjunction analysis process subjected to the application of the apogee-perigee filter and the sieves.

erated, as listed in Table 8.12. The analysis performed here, was with the sieves in the original order and setup, like described in the reference. This is done to get a first reference benchmark, and to see if and how the results compare to the reference document. Later on, in the next subsection, the sieves will be analysed seperately.

The column 'pre' within the tab CPU time t_{CPU} stands for pre-processing, and includes the TLE reader, SGP4 propagation and perigee-apogee filter application processes, of which the first and last for all cases needed approximately 1.6 and 1.5 seconds CPU time, respectively, irrespective of any input parameters.

Furthermore, the conjunction distance was in all cases 25 km, and the best available perigee-apogee filter was applied in each case, resulting in 57.47% of the pairs discarded, before the sieves process started.

The sieves were written as different functions here, where each sieve passes on the pairs that are remaining to the next sieve.

The performance of the sieves is taken as an average of the performance of all time instances, because the sieves are applied to the ephemerides of each time instance, generating different results.

Looking at the entries in the table, the same time intervals T were analysed as for the brute force method, all resulting in the same (number of) minimum conjunction (m.c.) events found, thereby verifying the safe application of the perigee-apogee filter and sieves, in this order at least.

Due to the fact that for the time interval of 1 seconds the time step can also attain a maximum value of 1 seconds, the brute force method is faster than the sieves method, due to the fact that both methods have to analyse the same amount of time instances, while the sieve method has to perform more calculations.

However, for the time interval of 60 seconds, the situation is reversed already, since now the time step of the sieve method can be increased to 60 seconds as well, still yielding the analysis of only two time instances, whereas the brute force method had to analyse 61.

Then, the improvement can be seen to increase until finally resulting in a decrease of CPU time by a factor of more than 50 for a 1-day analysis interval.

Multiple time steps were tried for the time interval of 1 hour, yielding an approximate optimal result for $\Delta t = 90$ seconds, which was then used for the 1-day interval.

Table 8.12: Performance results of the sieves method, original order and implementation, but with the new fine conjunction detection (f.c.d.) method. The corresponding total CPU time needed for the brute force method ('brute') of Table 8.10 is indicated as well. All cases used a conjunction distance D of 25 km, and the best available perigee-apogee filter was used pre-hand, rejecting 57.47% of the pairs in all cases before the sieves process started.

					t_{CPU} [s]					Pairs rejected/analysed [%]						
T [s]	Δt [s]	# t_i	# t.c.	# m.c.	pre	sieves	post	total	brute	X	Y	Z	r^2	r_{min}	r^2_{fine}	f.c.d.
1	1	2	77	42	4	2	0	6	2	99.58	99.37	88.04	43.01	30.67	25.66	8.33
60	60	2	294	232	4	3	0	6	59	92.02	90.04	64.11	32.73	96.54	56.30	88.84
3600	60	61	13960	10597	11	68	1	80	3442	92.03	90.05	64.59	33.11	96.45	57.29	82.25
3600	90	41	14021	10597	9	62	1	72	3442	88.26	85.92	61.95	32.24	94.26	53.31	93.39
3600	120	31	14326	10597	8	74	1	82	3442	84.55	82.01	60.10	31.79	91.20	49.68	96.96
86400	90	961	338215	255712	125	1476	16	1616	83109	88.26	85.96	61.94	32.24	94.25	53.45	93.18

The total number of conjunctions (t.c.) found is significantly smaller than for the brute force method, as the fine conjunction detection method already filters out all found minimum conjunction with a distance larger than 25 km. The difference with the amount of minimum conjunctions found here is thus purely due to the multiple defined conjunction phenomena, as explained earlier.

The CPU time needed for pre-processing can be seen to increase with time interval T, while decreasing with time step Δt. This is due to the fact that the SGP4 propagation has to propagate to more time instances when *only* the time interval increases, while decreasing when *only* the time step increases.

Furthermore, the CPU time needed for the application of the sieves increases with the time interval as well, while there exists an optimum for a certain time step. Finally, the CPU time needed for post-processing increases with time interval (or rather, total amount of conjunctions) as well.

What then follows are the columns for the performance of the respective sieves in terms of percentage of the amount rejected pairs with respect to the amount of pairs analysed by that sieve. The performance of the first 4 sieves can be seen to decrease with increasing time step, since the threshold distance R_{th} on which these sieves are based increases with Δt.

This also holds for the 5th and 6th sieve, when looking at only the time interval of 1 hour. The reason that this does not completely hold for the time intervals of 1 and 60 seconds, is because the first sieves have now filtered so many extra pairs already, which would also have been filtered by these 5th and 6th sieves, but were now already filtered, thus decreasing their performance.

As the time step increases, less pairs are filtered out by the first 6 consecutive sieves, thus the more pairs end up at the fine conjunction detection, which thus has to filter out more pairs. This last sieve is relatively slow, because it has to perform significantly more calculations, due to the fact that for each pair that is still remaining, SGP4 propagation has to be performed.

This is therefore the reason why there exists an optimum time step for a certain time interval in terms of CPU time needed; increasing the time step leads to less time instances which have to analysed, decreasing the CPU time needed, but at the same time leads to more pairs ending up at the last sieve, increasing the CPU time needed again.

Example Output

An example of the output data of the new fine conjunction detection method is given in Table 8.13, where conjunctions of the ISS and Delfi-C3 are given.

For the full-day analysis, the ISS had 5 conjunctions with a distance smaller than 25 km, while Delfi-C3 had 31. Of the latter only the top-10 of smallest conjunction distances, and the 31^st, is shown.

In the table the looping can be seen until the time offset is smaller than half a second, which happens at the second step (subscript 1) for all cases shown here.

The reason for the difference in the amount of conjunctions is because the average altitude of the ISS (approximately 387 km) is much lower than that of Delfi-C3 (621 km), whereby Delfi-C3 orbits in a much more crowded area in terms of total number of orbital debris objects.

The two top objects that conjunct with the ISS are the docked Soyuz stages again, which were treated earlier. The amount of conjunctions found with the ISS for 1 day and smaller than 25 km was 7 in Alarcon-Rodriguez et al. (2002), and thus quite similar.

Table 8.13: Example output of the new fine conjunction detection system, for conjunctions of the ISS and Delfi-C3 (using their TLE names), for the same full-day analysis as above.

t_i [JD]	ID_p	ID_s	d_{ps} [m]	$t_{closest,0}$ [s]	$x_{closest,0}$ [m]	$t_{closest,1}$ [s]	$x_{closest,1}$ [m]	$t_{con,1}$ [JD]
				ISS (ZARYA)				
2455855.51388888	25544	37396	13475	-22.000	13472	0.000	13466	2455855.51363441
2455855.51388888	25544	37633	13475	-22.000	13472	0.000	13466	2455855.51363441
2455855.51041666	25544	35002	63069	6.208	18873	0.000	18870	2455855.51048851
2455855.95312500	25063	25544	335331	31.909	23750	-0.015	23737	2455855.95349414
2455855.98611111	25063	25544	522524	-49.753	24231	0.052	24272	2455855.98553586
				DELFI-C3 (DO-64)				
2455855.57465277	32150	32789	553189	46.676	1291	-0.042	1328	2455855.57519252
2455856.35069444	29126	32789	515578	50.928	4872	-0.053	4906	2455856.35128328
2455855.71180555	32789	32791	8196	30.600	4971	-0.248	4979	2455855.71215685
2455856.10937500	5133	32789	96956	-6.578	7548	0.000	7544	2455856.10929886
2455856.35416666	27146	32789	253683	23.033	7841	-0.006	7829	2455856.35443319
2455856.17708333	5133	32789	510720	-34.774	8745	0.015	8756	2455856.17668103
2455856.21527777	32376	32789	144324	11.085	8885	-0.001	8884	2455856.21540606
2455856.08333333	27147	32789	16034	-16.323	9072	-0.015	9070	2455856.08314422
2455856.11979166	32789	35687	839735	60.391	10189	-0.085	10184	2455856.12048964
2455855.71006944	23608	32789	494503	46.013	11950	-0.038	11955	2455855.71060155
				...				
2455855.67708333	23608	32789	208960	-19.369	24234	0.003	24239	2455855.67685919

8.3.4 Sieves Applied Separately

When running the sieves consecutively, like above, the performance of each sieve is not immediately clear. Therefore, each sieve was tested on all possible pairs separately, see Table 8.14. However, since the functions of all sieves still store the remaining pairs at the end, sieves that are fast but filter out less pairs, have to store more pairs and thus become slower again.

At the end, when a most efficient order has been established for the final conjunction analysis process, the sieves will be implemented right after each other in one function, using nested if structures. Thus, they then do not need to store the remaining pairs any more, as a pair under analysis will move immediately to the next sieve when it is not filtered out.

Therefore, the sieves are also tested separately *without* storing the remaining pairs at the end. In this way, when combining multiple sieves, in subsequent sieves use can be made of already calculated variables, while these variables had to be calculated multiple times when applying the sieves separately.

Furthermore, the performance of each sieve separately does then not have to be stored any more. The input/output and calculation of this information was not counted as counted as part of the sieves anyway, but rather as part of the post-processing.

Comparing the performance of the different sieves can give an indication for the most efficient combination and order of sieves. However, a sieve that scores well when applied to all possible pairs, could perform poor when applied to the remaining pairs of another sieve. A good example of this are the X, Y and Z sieves, which filter the same objects as the r^2 sieve would filter by itself.

Since the perigee-apogee filter is applied very quickly and only once (not at each time instance), it is always beneficial to run this filter before the sieves process is started. As the sieves might perform differently when applied after the perigee-apogee filter is applied, the performance of the sieves was also tested on the remaining pairs of the perigee-apogee filter.

Results

When looking at the results of the original sieves, one can see that all sieves need less processing time when the perigee-apogee filter is applied first, and also when the pairs do not have to be stored. Applying the perigee-apogee filter results in only a small decrease in percentage of pairs discarded, while the CPU time decreases drastically, and thus proved that it is best applying the perigee-apogee filter pre-hand at all times.

Table 8.14: Performance of different sieves, original implementation, applied to the time interval of 1 hour, both separately (upper 6 entries) and as combinations (bottom 5 entries), and both applied with and without the storing of the remaining pairs at the end of the sieve, and furthermore with and without the application of the perigee-apogee filter up front.

	with pairs storing				without pairs storing			
	without p-a		with p-a		without p-a		with p-a	
Sieve	r/a [%]	t_{CPU} [s]	r/a [%]	t_{CPU} [s]	r/a [%]	t_{CPU} [s]	r/a [%]	t_{CPU} [s]
X	89.28	82	88.26	40	89.28	53	88.26	26
Y	89.58	81	88.61	39	89.58	53	88.61	26
Z	88.19	83	87.57	40	88.19	53	87.57	26
r^2	99.65	73	99.57	34	99.65	61	99.57	29
r_{min}	99.87	146	99.79	66	99.87	135	99.79	61
r^2_{fine}	99.97	235	99.95	103	99.97	210	99.95	92
XYZ	99.46	76	99.37	37	99.46	58	99.37	28
XYZ+r^2	99.65	76	99.57	37	99.65	70	99.57	34
Y+r^2	99.65	78	99.57	38	99.65	68	99.57	33
All-XYZ	99.9945	70	99.9886	33	99.9945	70	99.9886	33
All	99.9945	79	99.9886	38	99.9945	79	99.9886	38
All Sep.	99.9945	92	99.9886	45	-	-	-	-

Furthermore, some sieves are faster, while filtering out less pairs, while others are slower, due to the fact that they are more complex and more calculations have to be performed in their application, but discard significantly more pairs.

Due to the fact that the X, Y and Z sieves have to store significantly more remaining pairs because they filter out less, they perform even worse than the r^2 sieve at the left-hand side of the table, while they are quicker at the right hand-side. When looking at that right-hand side, it can be seen that while the X, Y and Z sieves perform better in terms of computation time than the r^2 sieve, this is only slightly so, and thus a better combination of the sieves than originally described in Alarcon-Rodriguez et al. (2002) might exist.

Therefore, the X, Y and Z sieves were combined in one function (XYZ), so that their combined performance could be compared to the r^2 sieve, and furthermore to the combination of both (XYZ+r^2). Looking at the results (lower part of Table 8.14) it can be concluded that one is better off using *only* the r^2 sieve, while not making use of the X, Y or Z sieves at all.

However, for the last two sieves, r_{min} and r^2_{fine}, the same kind of reasoning does not apply, as their underlying theory is not similar and their CPU times are significantly different as well. Furthermore, when looking at Table 8.12, the r_{min} sieve can be seen to still filter out a relatively large amount of pairs, when applied to the remaining pairs of the r^2 sieve, while it does this twice as fast as the r^2_{fine} sieve would.

Sieves Combined

Therefore, the order of the last 3 of the original sieves is kept as is, and a comparison is made between all sieves implemented in 1 function, both with and without the X, Y and Z sieves. Losing the X, Y and Z sieves shows to be significantly more efficient.

The reason why this is true is because some pairs that are not filtered out by the X, Y and Z sieves, but *are* filtered out by the r^2 sieve, have to go through 4 if statements before they are discarded. Similarly, even more pairs have to go through the X and Y sieves before being discarded by the Z sieve, having crossed 3 if statements. In the same way, a comparable amount of pairs have to go through 2 if statements before being discarded. Thus, even though the r^2 sieve needs a little bit more calculation, it is still faster on its own because only one if statement has to be passed by all pairs.

Another combination which might be more efficient now arises, and that is, just combining one of the X, Y or Z sieve with the r^2 sieve. Since the Y sieve seems to be most efficient here, it was chosen to form the combined

Y+r^2 sieve.

One might expect this sieve to be faster than the XYZ+r^2 sieve as it has less if statements, while the Y sieve on itself is faster than the r^2 sieve.

However, as can be seen in Table 8.14, while the Y+r^2 sieve is a little bit faster than the XYZ+r^2 sieve, it is still not faster than the r^2 sieve by itself.

The reason why the percentage of rejected pairs is listed with more decimals for the last three entries is because when these percentages get closer to 100%, they cannot be compared anymore, as they all would round off to 99.99, and, as we will see later on, even to 100.00%.

To see how much time is lost on storing the remaining pairs, all sieves written as separate functions ('All sep.') were also tested using the original ordering. As can be seen, this resulted in significantly more CPU time, thus proving that it is beneficial not to apply all functions separately. The same test could not be applied when not storing pairs, as then the remaining pairs can then not be sent to the next sieve, hence the '-' signs.

Verification

Next to the above performance testing, all of the original sieves were put first in the entire conjunction analysis process in order to determine whether still the same amount of conjunctions would be found as with the brute-force method. This turned out to be the case, and thus all sieves can be safely applied. That is, their theory holds, and they will not discard pairs which do have a conjunction.

As the author of this report is no computer scientist, the code might not have been written in an optimal way. However, as all sieves were written in the same way, and the sieves are believed to be applied in the most efficient way possible, they can be compared to one another. So, even though one might think of ways to better implement the conjunction analysis process, the same reasoning in terms of performance and resulting combination and order of sieves would still hold.

Although these tests were performed for a time interval of 1 hour, some runs were also made for 1 day, generating similar results: multiplications by approximately 24 for the computing time and approximately the same percentage of discarded pairs. Therefore, the results generated here are deemed good enough for the proper optimisation of the sieve combination and order.

A small test to determine the performance of just the fine conjunction detection method (which was not included in Table 8.14) was also made, resulting

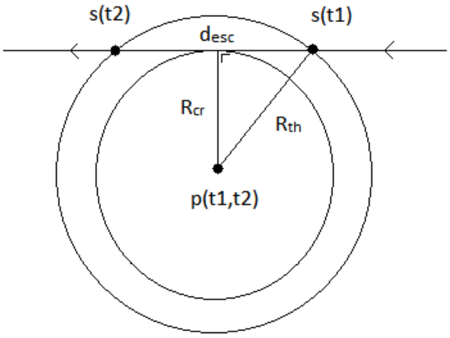


Figure 8.18: Schematic of the new determination of R_{th}.

in 99.9993% of pairs being discarded but needing 47608 seconds of CPU time, for the time interval of 1 hour, using a time step of 90 seconds.

If this sieve would solely be used for the conjunction determination process, it would even need more than 13 times the computing time of the brute force method! It can thus be safely said that it is best that this 'sieve' is placed at the end.

8.3.5 More Improvements

Next to losing the XYZ sieves, four more possible improvements with respect to the original sieves and underlying theory were tested, which will be explained below. Not including the XYZ sieve is called improvement 1 from now on, so the tag 'All+1' will mean all sieves, but with the X, Y and Z sieves removed, in the rest of the document.

Two improvements were made to the underlying theory of the sieves (improvements 2 and 3 respectively), while the other two were made by determining variables per object rather than use a constant value (improvements 4 and 5).

Improvement 2: Better Threshold Radius Definition

The way in which the radius of the threshold volume R_{th} was originally determined was already shown in Section 5.1.

There, see Figure 5.2, the relative path of the secondary satellite is taken perpendicular to the sphere (or circle) spanned by R_{th}, while it could (or should?) be drawn by just touching the sphere spanned by R_{cr}, leading to a smaller value for R_{th}. This is shown in Figure 8.18, and is in fact analogous to the theory set up with help of Figure 8.2.

Originally R_{th} was calculated as

$$d_{esc} = 2v_{esc}\Delta t \tag{8.19}$$

$$R_{th} = R_{cr} + d_{esc}/2 = R_{cr} + v_{esc}\Delta t \tag{8.20}$$

And, since R_{th}^2 is used in the r^2 sieve, results in

$$R_{th}^2 = (R_{cr} + v_{esc}\Delta t)^2 = R_{cr}^2 + 2R_{cr}v_{esc}\Delta t + v_{esc}^2\Delta t^2 \tag{8.21}$$

However, the new proposed way to determine R_{th}^2 follows from Figure 8.18 as

$$R_{th}^2 = R_{cr}^2 + v_{esc}^2\Delta t^2 \tag{8.22}$$

decreasing R_{th}^2 by $2R_{cr}v_{esc}\Delta t$.

Decreasing R_{th} is beneficial for the X, Y, Z and r^2 sieves, as the test for discarding pairs in these sieves depends on this constant. Decreasing R_{th} will increase the amount of discarded pairs, thereby increasing the performance of the sieve. And, since the constant is only calculated once before application of the sieve, it will not increase the computation time of the entire process.

On the downside, due to the fact that Δt is squared in the last term, for a time step of 90 seconds and a critical radius of 25 km, the decrease in R_{th}^2 is only little less than 5% of the original. But this does not take away the fact that extra rejection of pairs can be gained 'for free' here.

Improvement 3: Better Acceleration Radius Definition

The radius of the acceleration safety volume R_{acc} was originally envisioned to be determined as, see also Section 5.1,

$$d_{acc} = \frac{1}{2}(2g_0)\Delta t^2 \tag{8.23}$$

$$R_{acc} = R_{cr} + d_{acc} = R_{cr} + g_0\Delta t^2 \tag{8.24}$$

However, in this way, one actually determines the maximum possible deviation due to acceleration *at the end* of the time step, and thus at the next time instance. But, because the point of closest approach, in a worst-case scenario, occurs at the midpoint of the time step, one could (or should?) determine the maximum curving effect due to acceleration as

$$d_{acc} = \frac{1}{2}(2g_0)(\frac{1}{2}\Delta t)^2 \tag{8.25}$$

$$R_{acc} = R_{cr} + d_{acc} = R_{cr} + \frac{1}{4}g_0\Delta t^2 \tag{8.26}$$

resulting in a significantly smaller radius R_{acc}. For a time step of 90 seconds and a critical radius of 25 km, R_{acc} is decreased by a factor of more than

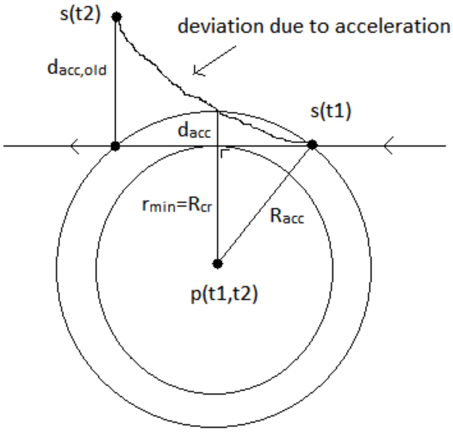


Figure 8.19: Schematic of the improvement of the determination of R_{acc}.

2.3 compared to the original!

Again, a smaller R_{acc} leads to more pairs being discarded, this time by both the r_{min} and the r^2_{fine} sieve.

In the r_{min} sieve, R_{acc} is tested against the value for r_{min}, which is the minimum value that the distance attains at the point of closest approach, and is determined linearly using the relative velocity and position. Since r_{min} is determined at each time instance in the application of the sieve, when a conjunction would occur, there will be one time instance where r_{min} is determined at a time difference smaller than half the time step, thereby proving the safe application of the improvement proposed here.

The improvement for the r_{min} is illustrated in Figure 8.19.

The r^2_{fine} sieve is applied differently however, where the fine threshold distance $R_{th,fine}$ is determined, see Section 5.1, by adding R_{acc} to the distance covered due to the relative approach velocity in half a time step.

This $R_{th,fine}$ is then tested against the distance between the two objects under analysis. This is correct, but due to the decrease in R_{acc} this test will now filter out more pairs as well.

Improvements 4/4a: Better Maximum Relative Velocity Determinations

The application of all sieves is in principle theoretically determined by two constants, v_{esc} and g_0, and one variable, Δt, as all other variables are determined by the ephemerides of the objects. The time step is optimised by performing iterations on the complete conjunction analysis process.

Since the two constants are chosen very conservatively, there might be some room for improvement here.

In determining R_{th}, the escape velocity v_{esc} is used, as this is the absolute maximum velocity an object can attain, while in orbit about the Earth.

However, since for the perigee-apogee filter the minimum radius of each object has already been determined, the maximum velocity *per object* can also be determined, in the time interval under consideration. This has to be done with the vis-viva law, as the minimum radius for an object here is not necessarily the perigee radius of the object, due to the possibility of objects having a larger orbital period than the time interval under consideration.

$$v_{max} = \sqrt{\mu \left(\frac{2}{r_{min}} - \frac{1}{a} \right)} \tag{8.27}$$

If this new value for V_{max} is now to be used in the sieve per object, the threshold distance follows as, using the result of improvement 2,

$$R_{th}^2 = R_{cr}^2 + (\frac{1}{2}(v_{max,p} + v_{max,s}))^2 \Delta t^2 \tag{8.28}$$

$$R_{th} = \sqrt{R_{cr}^2 + (\frac{1}{2}(v_{max,p} + v_{max,s}))^2 \Delta t^2} \tag{8.29}$$

which is calculated per pair.

Since the value of v_{max} determined per object is always smaller than v_{esc}, which was previously applied in all cases, the sieves using R_{th}, namely the X, Y, X and r^2 sieves, will be able to filter out more pairs.

There is a catch here however, and that is that in the application of these sieves, R_{th} has to be calculated per pair, while previously this was a constant which could be determined once pre-hand. This will increase the running time of these sieves.

This improvement could be taken one step further (improvement 4a), by taking the actual velocities of each object at each time instance to determine R_{th}^2.

However, as this involves taking the square root to obtain the velocity from its components for both objects, it increases the computational burden significantly, as will be shown later.

Improvements 5/5a: Better Relative Acceleration Determinations

The application of g_0 can be improved in a similar way as in improvement 4, namely, calculating its maximum value pre-hand, per object, for the time interval under consideration, using the minimum radius which was already determined per object for the apogee-perigee filter,

$$g_{max} = \frac{GM}{r_{min}^2} \tag{8.30}$$

resulting in a new R_{acc}, using the result of improvement 3,

$$R_{acc} = R_{cr} + \frac{1}{4}(\frac{1}{2}(g_{max,p} + g_{max,s}))\Delta t^2 \tag{8.31}$$

which has to be calculated per object pair.

The sieves that make use of R_{acc}, namely, the r_{min} and r_{fine}^2 sieves, will be able to filter out more pairs, due to the fact that g_{max} determined per object is always smaller than g_0.

However, also here it is true that these sieves will now need more CPU time in their application, as they have to perform more calculations than before, when R_{acc} was determined as a constant and pre-hand.

This improvement can also be taken one step further (improvement 5a), by taking the actual radius of each object at each time instance to determine g and then R_{acc}. However, this increases the computation time somewhat again.

Results

The performance results of the sieves applied separately without storing the remaining pairs, but including improvements, are shown in Table 8.15, and can now be compared with the results obtained earlier for the original sieves, listed in Table 8.14.

All improvements were tested to find the same conjunctions as the original sieves and thus also as the brute force method, and are thus safe to apply.

For the first 10 sieve entries of Table 8.15, it can be seen that improvements 2 and 3 always increase the amount of pairs that are rejected, while the computation time stays the same.

However, when improvements 4 and 5 are added on top of 2 and 3, the amount of pairs rejected is increased, but the CPU time is also increased. They will therefore have to be tested in a complete conjunction analysis run to determine whether they are worthwhile or not.

Combination of Improvements The results of the complete conjunction analysis process for a number of combinations of sieves and improvements, for a time interval of 1 hour, are shown in Table 8.16. This time, also improvements 4a and 5a were tested.

Table 8.15: Performance of sieves including improvements, for a time interval of 1 hour and a time step of 90 seconds.

	without p-a		with p-a	
Sieve	r/a [%]	t_{CPU} [s]	r/a [%]	t_{CPU} [s]
X+2	89.54	51	88.54	26
Y+2	89.83	52	88.88	26
Z+2	88.44	53	87.83	26
r^2+2	99.67	60	99.59	29
r_{min}+3	99.974	134	99.95	61
r^2_{fine}+3	99.977	210	99.96	92
XYZ+2	99.49	57	99.40	28
XYZ+r^2+2	99.67	70	99.59	34
All+1+2+3	99.9991	71	99.9978	33
All+2+3	99.9991	78	99.9978	38
r^2+2+4	99.87	71	99.8217	34
r_{min}+3+5	99.9785	152	99.9618	68
r^2_{fine}+3+5	99.9775	241	99.9644	106

The number of found conjunctions was 10597 in all cases, being the same conjunctions as were found by the brute-force method. Furthermore, the pre- and post-processing times were approximately 9 seconds in total for all cases, so only the total CPU time is listed here.

Compared to the original, improvements 1, 2 and 3 can be seen to reduce the total computation time significantly. Then, following these results, improvement 4 is seen to add too much extra calculations to a high amount of pairs in the earlier sieves, while improvement 5 decreases the running time slightly again. The same happens for 4a and 5a, where 5a can be seen to perform slightly less again, compared to improvement 5.

Full Conjunction Analysis Process The three most promising combinations, next to the original, are now tested on the full 1-day time interval, generating the results listed in Table 8.17.

It can be seen here that all of these three combinations generate similar results, with the combination of improvements 1+2+3+5 scoring the lowest CPU time.

Furthermore, it can be seen that while the difference in the percentage of pairs filtered does not look like much, with respect to the number of pairs that are left at that point, it is a large number, as can be seen by the percentage which the fine conjunction detection method has to filter out at

Table 8.16: Performance of the original sieves compared to the improvements in a full conjunction analysis process, for a time interval of 1 hour and a time step of 90 seconds.

			r/a [%]	
Sieves	# t.c.	t_{CPU} [s]	sieves	f.c.d.
All orig.	14021	63	99.9886	93.38
All+1	14021	59	99.9886	93.38
All+2+3	12593	50	99.9978	68.87
All+1+2+3	12593	46	99.9978	68.87
All+1+2+3+4	12407	50	99.9979	68.89
All+1+2+3+5	12479	45	99.9982	62.03
All+1+2+3+4+5	12323	50	99.9983	62.02
All+1+2+3+4a	12348	69	99.9979	68.91
All+1+2+3+5a	12473	46	99.9983	61.50
All+1+2+3+4a+5a	12267	69	99.9983	61.52

Table 8.17: Performance of the original sieves compared to the improvements in a full conjunction analysis process, for a time interval of 1 day and a time step of 90 seconds.

			r/a [%]	
Sieves	# t.c.	t_{CPU} [s]	sieves	f.c.d.
All orig.	338189	1413	99.9886	93.19
All+1+2+3	303789	1025	99.9978	67.96
All+1+2+3+5	301201	998	99.9982	60.89
All+1+2+3+5a	301027	1007	99.9983	60.32

the end.

Optimisation of this last sieve can thus also yield performance increases, although the function was implemented here to the best of knowledge as efficient as possible. Implementing a different but faster way to determine the conjunction event within the time step can have beneficial results.

8.3.6 Optimisation of Time Step

Although all of the above analysis was done with a time step Δt of 90 seconds, an optimisation of this time step applied to the entire full-day conjunction analysis process still has to be performed. In Table 8.18 the results of multiple runs with a different time step can be seen for the combination

Table 8.18: Optimisation of time step Δt, for combination 'All+1+2+3+5'.

		t_{CPU} [s]			r/a [%]	
Δt	# t.c.	pre	sieves	total	sieves	f.c.d.
60	314479	187	1202	1405	99.9992	37.60
90	301201	123	859	998	99.9982	60.88
120	296122	95	708	819	99.9962	67.35
130	295391	88	686	791	99.9951	79.97
140	294652	82	670	768	99.9938	82.99
150	294177	77	660	753	99.9923	85.47
160	294121	73	665	754	99.9904	87.57
170	293805	69	670	755	99.9881	89.31
180	293796	65	680	761	99.9854	90.74
210	294546	56	763	836	99.9745	93.83

'All+1+2+3+5', which performed best in Table 8.17.

Post-processing took approximately 16 seconds in all cases, and again the same minimum-conjunction events were found as in the brute-force method.

What can be seen here is that for larger time steps, the pre-processing time decreases, due to the fact that less ephemerides have to be generated, which also decreases the burden on the sieves process, as these have to process the ephemerides at each time instance.

Furthermore, an optimum can be seen at $\Delta t = 150$ s. Due to the fact that more and more possible conjunctions end up at the fine conjunction detection method, the whole progress slows down again for larger time steps.

Something else is noted now as well: the more pairs a combination of sieves discards, the higher the time step where the optimum will be found. And so, the combination 'All+1+2+3+5a', which performed slightly better in terms of pairs discarded, but slightly less in terms of CPU time needed, for a time step of $\Delta t = 90$ s, just might perform better at its time step optimum than the combination analysed here!

Therefore, the optimum time step for the combination 'All+1+2+3+5a' was also determined, see Table 8.19, and for $\Delta t = 150$ s, the CPU time needed is slightly less compared to the optimum value in Table 8.18

To compare the results to the original sieves, the time step had to be optimised there as well, as can be seen in Table 8.20.

The optimum here lies around $\Delta t = 90$ seconds, which results in a CPU time of around 1422 seconds.

Table 8.19: Optimisation of time step Δt, for combination 'All+1+2+3+5a'.

		t_{CPU} [s]			r/a [%]	
Δt	# t.c.	pre	sieves	total	sieves	f.c.d.
120	295861	96	712	833	99.9962	75.89
150	293898	77	657	751	99.9924	85.13
155	293599	75	660	751	99.9916	86.26
160	293806	73	662	752	99.9906	87.27
180	293648	65	677	758	99.9858	90.49

Table 8.20: Optimisation of time step Δt, for the original combination of sieves.

		t_{CPU} [s]			r/a [%]	
Δt	t.c.	pre	sieves	total	sieves	f.c.d.
60	339927	188	1385	1589	99.9972	81.64
80	337720	145	1263	1425	99.9926	90.72
90	338189	126	1280	1422	99.9886	93.17
120	344426	96	1567	1679	99.9666	96.84

The optimum has a lower value here because the original sieves filter less pairs in total, for the same time step. This can be seen when comparing r/a for the time step of 120 seconds in Tables 8.19 and 8.20.

With respect to the optimal time step, compared to the original sieves, the combinations 'All+1+2+3+5/5a' can be seen to be almost twice as fast.

8.3.7 Optimisation of Perigee-Apogee Filter

Looking at the total computation time for the full conjunction analysis process, a proper selection of the time step used for the determination of the minimum and maximum radius per object for the perigee-apogee filter can now be made.

Up until now, the best available perigee-apogee filter (Δt = 1 s) with the smallest patches (10 and 1 m) was used. Choosing a larger time step here decreases the amount of CPU time needed to determine the patches for the perigee-apogee filter, but also decreases the amount of pairs filtered out, which in turn increases the CPU time of the total process again.

An optimum can thus be found here as well, and the results of multiple runs can be seen in Table 8.21.

Table 8.21: Optimisation of time step Δt, for the perigee-apogee filter, using the sieve combination 'All+1+2+3+5a'.

	t_{CPU} [s]			r/a [%]		
Δt	filter	sieves	total	filter	sieves	f.c.d.
1	3388	654	4134	57.4677	99.9924	85.1268
10	334	656	1083	57.4556	99.9924	85.1306
20	168	656	916	57.4189	99.9924	85.1431
30	112	656	860	57.3602	99.9924	85.1615
60	56	658	807	57.0837	99.9925	85.2485
180	19	694	806	54.4187	99.9926	85.7616
360	10	782	886	45.7800	99.9938	85.8586

In between the time step of 60 and 180 seconds for the generation of the data for the perigee-apogee filter, the total conjunction analysis process can be seen to have a minimum total CPU time.

Although less pairs are filtered here by the perigee-apogee filter than for smaller time steps, and the sieves thus have to process more pairs at each time instance, the process to generate the minimum and maximum radii of all objects for the perigee-apogee filter takes considerable less computation time, hence resulting in the optimum.

8.3.8 Comparison with References

Since a relation to compare the results of different methods analysed in different publications could not be found in Chapter 5, the results gained here cannot easily be compared to older methods which had different starting parameters. Most methods used different values for the amount of objects in the catalog (and thus amount of pairs analysed), for the conjunction distance, and for the time interval analysed.

Amount of found Conjunctions

Looking at the results of the found references in Table 5.1, one can see two entries that match the starting parameters used here quite closely, namely, Alarcon-Rodriguez et al. (2002) and Ting & Hai (2008), which used the same time interval and the same conjunction distance. The amount of pairs analysed is significantly less however.

The catalog analysed here contains 14603 objects, resulting in 106616503 pairs analysed, which is approximately 3.3 and 1.5 times the amount of pairs analysed in the two references, respectively. However, the method implemented here found approximately 24 and 2.2 as many conjunction events.

However, no clear relation between these two variables exists either. This can be due to the fact that previous methods might not have found all conjunctions, but this is assumed not to be the case, due to the fact that at least the latter of the two methods also used a reference method for verification. Furthermore, the first of the two methods is the same implemented here, which is verified.

The reason that there exists no natural relation is probably due to the non-randomness of the position and the orbits of the objects in space. In time, most new objects are added to the LEO and GEO regions, generating a disproportionate amount of extra conjunctions.

Finally, comparing Table 8.6 with Table 5.2, the same (approximate) quadratic decrease of number of conjunctions for decreasing conjunction distance can be observed.

Optimal Time Step

When comparing the optimal time step found here with the two references in Table 5.3, they are all different, due to the fact that all conjunction analysis systems are different with different concluding methods, like the fine conjunction detection method here.

Sieve Performance

Furthermore, the percentage of rejected over analysed pairs gained here compared to Table 5.4 is different for all sieves, as different method with different time steps were applied.

However, for the method described in Alarcon-Rodriguez et al. (2002), the same trend for the sieves can be seen for the results here, see Table 8.12, for a time step of 120 seconds, which comes close to the 180 seconds used in the reference. That is, the Z sieve filters out considerably less than X and Y, since the latter sieves already filtered out objects that would have been filtered out by Z as well.

Then, the r^2 sieve performs considerably less again, around 30%, which is about the same that was gained here. Also, the r_{min} and $R_{th,fine}$ sieves show similar numbers.

8.3.9 Possible Other Improvements

There are other improvements that could be made to the sieves process, and two other filters might be applied before application of the sieves, but after the perigee-apogee filter. Unfortunately, there was not enough time in the scope of this thesis to implement and test these additions.

A brief description is given here though, as some research on the possibilities was done.

Improvement 6: Relative Acceleration

Looking back at the equations to determine the maximum relative acceleration between two objects in near conjunction, in subsection 8.3.5, it can be said that this worst case scenario of adding the two gravitational accelerations can actually never be met. Due to the fact that two objects that are in near conjunction have gravitational accelerations that are quite similar to one another, taking the absolute difference of the accelerations determined with improvement 5a might be a better estimate of their maximum relative gravitational acceleration.

However, implementing this resulted in less conjunctions found. For an optimum time step of 200 seconds, the absolute difference needed to be multiplied with a factor 20 in order to find all conjunctions again. However, this did result in a total computation time of only 494 seconds, thus being almost three times faster than the original reference method. In the sieves process, 99.9981% of the pairs were discarded here, compared to the optimum of 99.9924% of Table 8.19.

Seeing as taking the difference needed such a large factor in order to work, the accelerations due to drag might be more important here. As the velocities vectors of both object can have large differences, the accelerations due to drag will not be similar, and generate a large relative acceleration. Furthermore, all other perturbations are quite similar for both objects near the conjunction point. This would require a different approach altogether, resulting in smaller relative accelerations, and consequently a larger amount of discarded pairs by the r_{min} and $R_{th,fine}$ sieves, and might be worth researching more in-depth in the future.

Step Skipping

The process of skipping steps was already described in Section 5.1, and uses the fact that object pairs might be so far apart that they can never reach each other in some n upcoming time steps. At these steps the objects can then skip the sieves process, until the number of steps that could be skipped have passed.

The downfall of implementing this process is that extra information (an integer) has to be stored for each pair, through which the system has to loop at the start of the analysis of each time instance. Each number of steps to be skipped is then decremented by 1, until it reaches 0, and the pair has to be analysed again.

Furthermore, the r^2 sieve would have to calculate the steps to be skipped, and store this extra information, when a pair is discarded.

Although some extra calculations and processes are involved here, the step

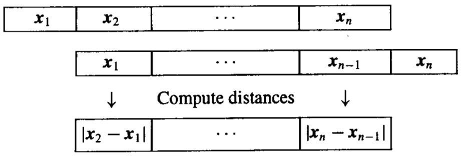


Figure 8.20: Schematic of the forming of pairs with ascendingly sorted arrays with respect to one of the three coordinates. (Healy, 1995)

skipping looks promising. In particular since now, each pair, after the calculation of the steps to be skipped, only has to be checked for an integer, instead of determining its distance each time instance.

Sorting

A sorting theory, set up in Healy (1995), could also improve the computation time of the sieves.

When applied to the X sieve for instance, all objects would have to be sorted ascendingly according to their x coordinate. Then, the now sorted list will move along a copy of this list, like indicated in Figure 8.20, to form the pairs that need to be analysed. In the figure, the bottom array will move one step to the right each time, until the end is reached. In this way, each possible combination of pairs is analysed one time.

There will be a point at which all pairs formed by the the combination of the sorted lists are rejected by the sieve, and thus also all following pairs will be rejected by the sieve, due to the sorting. The sieve can thus stop when this point has been reached.

Although this improvement could be applied to the X, Y and Z sieves, it can unfortunately not be used for any of the other sieves, as these involve testing by use of more than one coordinate, and thus have to be calculated for each pair independently.

Furthermore, it can neither be applied to the perigee-apogee filter, as this depends on both the perigee and apogee radius of both objects in a pair.

Since the X, Y and Z sieves were proven to slow the conjunction process down, this sorting method would have to be so effective in order to make the X, Y and Z sieves efficient enough to take part again.

Two Other Filters

Next to the perigee-apogee filter, there are two other filters initially set up in Hoots et al. (1984), namely, the geometrical filter (also called orbit path filter) and the time filter, which are both much more complex.

It has been shown in Woodburn et al. (2009) and stated in Alarcon-Rodriguez et al. (2002) as well, that these filters are not reliable in their application: they do not guarantee that all conjunctions will be found. In other words, pairs might be discarded by these filters which do have a conjunction in the time interval which is analysed.

In Woodburn et al. (2009) these filters were improved for a reliable application however, but without a detailed description of how this was done. It is therefore not fairly easy to implement these complex filters. Furthermore, it was also stated that at least the time filter would bring a larger computational burden than the amount of pairs it would reject.

Also, in Ting & Hai (2008), a description of a new improvement to the geometrical filter was described, but also not into great detail. This improvement looks promising however, as it is shown to not miss any conjunction while decreasing the CPU time needed substantially.

Chapter 9

Conclusions

A complete and successful orbital debris conjunction analysis system has been set up in C++, which partly was integrated in Tudat as well. This system combines a TLE reader, an SGP4 propagator and a conjunction analysis process which makes use of a perigee-apogee filter and multiple sieves.

The conjunction analysis process was verified by a so-called 'brute force' method, to test if it filters out pairs of objects which do have a conjunction event in the time interval under analysis.

Furthermore, improvements were made to existing filtering methods and sieves, leading to lower CPU times.

9.1 TLE Reader

A TLE reader has been set up and integrated into Tudat, which makes use of publicly available TLE catalog files, which are updated twice a day and can be downloaded with a program called TLERetriever (Kelso, 2010a). This program needs one's account details to gain access to SpaceTrack (USSTRATCOM, 2004), from which the TLE data is retrieved.

The TLE reader can read in a catalog file with 2- or 3-line TLE data, which contains one element set for every object. Variables like the objects' name, ID, launch date and orbital parameters, as well as the epoch of the TLE for which the orbital elements hold, are included. These variables are then stored in a data container, which also stores new variables that have been derived from the TLE variables.

The TLE reader also contains a file integrity check, to test if all TLEs within the catalog file are valid. This is done with a number of rules that have to apply for each TLE. If a TLE is not valid, it will be removed from the database.

Furthermore, a TLE manager was included, which can write selectable TLE data to a file, for analysis. Using the output data of the variables within the TLEs, histograms can be drawn, and multiple variables can be plotted versus each other in one figure.

Besides, it can sort the database with respect to a certain variable, count the number of rocket bodies versus debris or payload objects, and count the amount of objects with the same source or orbital regime.

Furthermore, it can group the TLEs in orbital regimes and make new catalog files from them, as well as take a random percentage or number of objects from the catalog to make a new, smaller database to work with.

9.1.1 Result Examples

A nice example of a histogram can be seen in Figure 6.9, which shows the abundance of different ranges of the mean motion n. The peak at 1 rev/day represents the GEO objects, while the peak around 2 rev/day is due to the satellite constellations and GTO objects. The peaks around 12.5, 13.8 and 14.5 rev/day are LEO objects corresponding to preferred orbital heights for science, weather and remote sensing satellites, respectively.

A good example of a plot of multiple variables is shown in Figure 6.12, where the eccentricity e of each object versus its semi-major axis a is represented with one dot, which is coloured per orbital region. The graph can be seen to have an upper limit, which can be explained to be the limit of eccentricity that an object can have for a certain semi-major axis, for else it would decay towards the Earth and burn up in the atmosphere. The LEO, GTO and GEO objects can be clearly seen in their semi-major-axis bands, with corresponding eccentricities.

Record sources for the amount of debris in the catalog of 2011.01.10 are identified to come from TLE object names ‘FENGYUN 1C DEB’ (2791 objects), ‘COSMOS 2251 DEB’ (1313) and ‘IRIDIUM 33 DEB’ (462), which resulted from the Chinese anti-satellite test (Jan 2007) and the collision of an active and decommissioned payload (Feb 2009).

Categorised into three groups, this catalog contained 9807 (fragmented) debris objects, 1083 rocket bodies, and 3748 other objects, the latter including active and decommissioned payloads. The largest orbital regime is the LEO region, containing 11358 out of 14638 objects in this particular catalog.

The above numbers dropped significantly in a catalog of three months later, due to the decay of objects, as the collision events creating large amounts of debris had occurred only recently.

Sorting the catalog with respect to the apogee radius r_a showed the record holder to be ‘PROGNOZ 6’, which is located more than half-way to the Moon when situated in the apogee of its orbit (201460 km on the 10th of

April 2011).

9.2 SGP4 Propagator

Following the TLE reader, an SGP4 propagation system was implemented, which can be used to determine the ephemerides and orbital elements of an object or entire catalog of objects at any given date and time. SGP4 code already available in C was found online, described in Vallado et al. (2006), which is an improved version of the original SGP4 theory published in Hoots & Roehrich (1980).

This C code was subsequently researched and understood, after which an SGP4 handler was written to provide the proper input/output and functionality for the conjunction analysis.

Different options could now be selected for the version of the SGP4 code, the set of constants used, the time step, and the type of time interval parameters. The latter can be set as minutes from TLE epoch, date and time, or year and day. Also, the combination of output parameters could now be selected, like the epoch type, Cartesian state vector and orbital elements.

Furthermore, different functions to call the SGP4 propagator were implemented.

For instance, one function can process an entire catalog file and determine the ephemerides of all its objects at all time instances within the set time interval separated by the time step. The output could then be listed in the output file in groups per object, or per time instance.

Another functionality needed for conjunction analysis was to determine the ephemerides of one object at one time instance. Also, a function was needed to determine the minimum and maximum radius in a time interval, using a certain time step, for all objects in a catalog, as quickly as possible.

9.2.1 Result Examples

The implementation of the SGP4 propagator has been verified with tables and plots from references, which could be reproduced, if enough data for the creation of these plots was available.

An example of such a verification can be seen in Figures 7.5 and 7.6, where the position components of an object using the improved (corrected) SGP4 theory were recreated.

Next to this verification, orbital parameters and ephemerides of objects from current catalog could now be determined and plotted.

One example of what can be done is shown in Figure 7.12, which shows all objects of a catalog propagated to one time instance. Using a North pole view, the GEO ring and the LEO region are clearly visible to be heavily crowded with objects.

9.3 Conjunction Analysis

After the TLE reader and SGP4 propagator were successfully implemented, a conjunction analysis system was set up. The final goal here was to design a conjunction analysis process which is as efficient as possible.

Therefore, filters and sieves were tested in terms of performance separately first, in order to build up the best combination and order of these filters and sieves. Such that an entire catalog could be tested all-vs-all for a time interval of one full day, as fast as possible. All methods were timed with a CPU timer, in order to test their computational performance.

All analysis was done with the catalog of 2011.10.20.am, analysed from 2011.10.21 00:00:00 until 2011.10.22 at the same time. Of this catalog 14603 objects were used in the analysis.

9.3.1 Brute-Force Method

In order to verify the results of the final process, a baseline conjunction analysis process was setup first, which was called the 'brute-force' method. This method generated the ephemerides of all objects with a time step of 1 second, after which the positions of all possible unique pairings of objects were compared, to determine which conjunction events took place during one day. The method was designed in such way, that all possible conjunctions with a relative minimum distance of less than 25 km were found.

Due to the fact that one conjunction event could be found at multiple time instances (an object pair can have a distance of less than 25 km for the duration of more than one second), these multiple defined conjunction events were filtered out.

Furthermore, for each conjunction event, the minimum distance at the time of closest approach, was determined with a Taylor series approximation, in order to define each conjunction event within the time steps.

The final brute-force analysis resulted in 255712 conjunction events found, while the CPU time needed for this analysis was 83109 seconds in total, which can be subdivided in SGP4 propagation (10793 seconds), conjunction analysis (71742 seconds), and post-processing (575 seconds).

9.3.2 Perigee-Apogee Filter

The first option to reduce the computation time of the conjunction analysis process that was analysed, is the perigee-apogee filter. In essence the filter works by discarding pairs of the database, when the perigee radius of one object is larger than the apogee radius of the other.

The difference between both actually has to be larger than the set conjunction distance. Since the goal of the filter is to eliminate pairs for the entire time interval under analysis, it has to account for perturbations in the orbits of both objects as well.

Four different implementations of this filter were studied in this report, of which two were purposely designed for this thesis (number 1 and 4), one is an improvement of an existing method (3), and one an adjustment of a here-designed method using theory from an earlier method (2).

Directly from TLE

The first method simply determines the apogee and perigee radii of each object directly from the orbital parameters in the TLE. Since the TLE epoch differs from the time interval that is analysed, and the average value determined from the TLE does not always match the 'real' value defined by the ephemerides, the error that is made by this method can be quite significant.

The maximum error that is made in this way was 339 km for the apogee radius and 135 km for the perigee radius. These errors were then used as a patch for all apogee and perigee values of all objects, in order to safely apply the perigee-apogee filter. This resulted in 29.56% of all pairs being filtered out.

Modifications for Perturbations

For the second implementation, an improvement to the first method was made, by approximating the perturbations affecting the apogee and perigee radii, using the theory of Hoots et al. (1984). Equations describing the mean deviation due to atmospheric drag and the oblateness of the Earth were used to determine the perigee and apogee radii at the start and end points of the time interval in a linear fashion.

Now using the minimum and maximum of these values respectively, in the perigee-apogee filter, resulted in 36.36% of all pairs being discarded, as still the maximum error made was 189 km and -80 km for the apogee and perigee radii, respectively.

Sampled from Osculating Elements

For the third option the method described in Woodburn et al. (2009) was analysed, which determined the apogee and perigee radii of each object by determining the minimum and maximum of the osculating perigee and apogee distances based on the ephemerides at the start and end points of the time interval.

It was shown however, that the error that could be made in this way is still at least 46 km for an apogee example, and thus, by using patches of 50 km (the reference suggested patches of 30 km), this method resulted in 45.57% pairs being filtered out.

Determined from Ephemerides

The fourth and last method that was analysed, determined the ephemerides of each object with a certain time step, and from this the minimum and maximum radii of each object in the time interval was determined directly. Then after quantifying the error made in this way for a certain time step, the method can be applied in a safe way.

Furthermore, a trade-off can now be made with respect to the amount of pairs filtered out versus the CPU time needed to apply the entire filter. When generating similar results as the previous three in terms of pairs discarded, this final method only needed slightly more CPU time. However, if needed, it has the ability to go up to 57.47% of all pairs being filtered out.

Results

The latter two methods were found to be the most reliable in long-term application. The fourth method was selected for future analysis, as it is adjustable, and can be tuned to the needs of the entire conjunction analysis process. According to the CPU time needed by the rest of the conjunction analysis process, a trade-off can be made for a suitable time step and resulting percentage of pairs discarded by this method.

9.3.3 Sieves

Then, following the perigee-apogee filter, the sieves described in Alarcon-Rodriguez et al. (2002) were analysed and improved. The strength of the above filter lies in the fact that it can be applied pre-hand and only once, while the now following sieves need to be applied at each time instance.

However, a coarse time step can be used here, after which all pairs that pass all sieves for a certain time instance, go through a fine conjunction detection method, which determined whether or not the pair has a conjunction event within this coarse time step. This method loops the Taylor series approximation with the SGP4 propagator, in order to define a conjunction event with the same accuracy as the brute-force method did.

Sieve Descriptions

Six sieves were tested in terms of performance with respect to the average percentage of filtered pairs, and the CPU time needed for their application.

The first three, the X, Y and Z sieves, compared the difference between these coordinates of both objects, with a so-called threshold radius. This threshold radius is a combination of the critical radius (25 km is used here) and a safety patch, which is set up using the time step and the maximum velocity an object can attain in orbit about the Earth: the escape velocity. Applied in this way, the filtered objects cannot enter the critical sphere, at least, until the next time instance occurs.

The fourth sieve uses the same technique but now the actual distance between the objects is used, instead of just one coordinate.

The fifth sieve compares the minimum distance between two objects determined by the Taylor series approximation, with the acceleration radius, which is a combination of the critical radius and the maximum deflection the objects could make under the influence of Earth's gravity, using the standard gravity at sea-level.

The sixth and final sieve uses the actual approach velocity of the object pair at each time instance, which is used in an addition to the acceleration radius, and is then compared to the distance between the objects.

Improvements

Eight (possible) improvements were made to the theory, application and implementation of the sieves.

The first removed the use of the X, Y and Z sieves, as this proved to be more efficient.

The second and third were improvements to the theory for the set up of the threshold and acceleration safety volumes, leading to more pairs being discarded by all sieves.

The fourth and fifth improvements used the maximum velocity and maximum gravitational acceleration attained by each object at a certain time interval, for the determination of the threshold and acceleration radius. This could be done easily, as the minimum and maximum radii were already determined for each object for the perigee-apogee filter.

The sixth (4a) and seventh (5a) improvements, determined these two values for velocity and gravitational acceleration per object at each time instance using the ephemerides at that point.

These last four improvements (4-5a) resulted in a less conservative approach, compared to the application of the escape velocity and standard gravity. This resulted in more pairs being discarded, albeit at a sometimes high costs in terms of CPU time needed.

After the near completion of this thesis, another improvement (6) was tried, which scaled the difference (instead of addition) of the gravitational accelerations of improvement 5a of an object pair. However, a better implementation of this method still needs to be researched in-depth.

Results

The optimum combination of improvements, after optimising the time step for each method (which resulted in 150 seconds here), turned out to be 'All+1+2+3+5a'. This combination resulted in almost a twice as low value for the computation time needed for the total conjunction analysis process (751 seconds), compared to the original sieves (1422)!

This was due to the fact that, for the same time step, the new combination filtered out a higher percentage of the analysed pairs, compared to the original sieves. And, since for a smaller time step more time instances needed to be analysed, the new optimised method proved to be much more efficient, while still finding the same conjunction events as had been determined by the brute-force method.

The newly designed improved conjunction analysis including the setup of the perigee-apogee filter with an optimised time step of 120 seconds, resulted in a total CPU time of less than 800 seconds. Compared to the brute-force method, this is a decrease by a factor of more than 100 in terms of computation time.

Scaling the difference of the gravitational accelerations of improvement 5a with a factor 20 resulted in a computation time of only 494 seconds, for an optimum time step of 200 seconds, while still being able to find all conjunctions. This is almost three times as fast as the original method.

Leaving the reader with a concluding note: If the original method was called 'Smart-Sieve', what should this improved method be called (pun intended)?

Chapter 10

Recommendations

When designing or applying a new conjunction analysis process, it is recommended to take note of the improvements suggested in this thesis. They could be of great benefit to the efficiency of a new or existing conjunction analysis system.

For the perigee-apogee filter the 4th method described in this document is recommended. Although it takes slightly more CPU time than other methods, is it more reliable and accurate in long-term applications. Furthermore, it can be tuned to the needs of a certain system, in terms of percentage of filtered pairs versus CPU time needed.

When using the sieve system, the combination 'All+1+2+3+5a' performed best in the analysis for this thesis. However, for different implementations of a conjunction detection system, other combinations of sieves and improvements might perform better. The approach followed in thesis can then be a guide in determining the best options.

10.1 Future Research

Next to the research done in this thesis, other research that could benefit the conjunction analysis process could be done, which was considered outside the scope of this assignment. Furthermore, research that can be done in neighbouring areas was found as well, which could benefit conjunction analysis as a whole, or together form a larger analysis spectrum with respect to the understanding of orbital debris conjunctions.

10.1.1 Conjunction Analysis

In Subsection 8.3.9, some suggestions for other improvements not deeply researched in this thesis were given. These include the skipping of steps in the sieve process, or sorting the objects with respect to a coordinate.

Furthermore, two other potential filters could be researched: the geometry and time filter. References to different implementations of these filters were given throughout Section 5.1 and Subsection 8.3.9.

For the perigee-apogee filter, or any other analysis for that part, the TLEs could be categorised into subgroups, like the orbital regions, and according to this be treated differently in the application of the filters and sieves, as the GTO and HEO objects were often seen to create the largest errors.

Furthermore, the fine conjunction detection method used in this thesis was different from the one used in Alarcon-Rodriguez et al. (2002), which were both different from the one used in Ting & Hai (2008). A comparison of the different possibilities here could bring performance increases, as this final method of the conjunction analysis process is the most complex one and thus uses the most CPU time per analysed pair.

Also, the type of implementation here could be seen to greatly influence the optimal time step versus the total amount of pairs rejected.

When all possible filters would be researched, and the best fine conjunction detection method would be found, the ultimate and most efficient conjunction analysis process could be formed in the future.

Just before completion of this thesis, a last improvement was found, which still needs to be analysed more in-depth. When using the difference of the gravitational accelerations of two objects, it was found that other accelerations (most certainly due to drag) are larger in a relative sense. When fine-tuning the sieves that make use of relative acceleration, to gain optimal performance, one has to look at these accelerations as well.

10.1.2 Tudat Additions

Due to time constraints, it was unfortunately not possible to integrate the entire conjunction analysis process, which was designed and implemented in this thesis, in Tudat. If new MSc students pick up this subject however, they for instance might need the SGP4 propagator as well, and could integrate it in Tudat, as a next step.

All code implemented and used in this thesis is available on request, so feel free to contact the author. For contact details see the beginning of this report.

10.1.3 TLE Accuracy

In Section 4.5, the accuracy of TLEs and their propagation was treated briefly. The underlying references found on this subject were not brief however. In fact, this subject could easily span an entire MSc thesis.

The accuracy of the TLEs at epoch and of the ephemerides resulting from propagating with the SGP4 theory, versus time from this epoch, greatly influence the conjunction analysis process as a whole. If one knows the accuracy of the position of an object at certain time, a better judgement on the probability of a collision with respect to a certain conjunction event can be given.

The accuracy could be researched by comparing the propagation data with high-precision ephemerides from tracking data, once they have been converted to a common reference frame. All objects of which this information is freely available (GPS, GLONASS, GRACE, GOCE, etc.), or available to our department, could be used in this research.

Other ways to determine the accuracy, also for non-active objects, are to take as much following TLEs of an object as possible, propagate them to the same time instances, and compare the results.

The last method found to determine the accuracy of TLEs, based on pseudo-observations, might prove a little more challenging however, as this involves the process of simulating inverted tracking station models, which are often not freely available.

10.1.4 Collision Probability

When the accuracy of all TLEs is known, which might be categorised for different object groups, covariance matrices can be set up, which can be used in collision probability analysis. In the literature research Leloux (2010a) on which this thesis was based, this subject was treated as well, and so a large amount of references can be found there. As this research was unpublished, it is available on request.

By combining the accuracy of the propagated TLEs with the conjunction analysis process and the collision probability theory, a nice system could be set up which could result in fast determination of the probability of all conjunctions on a certain day, and could thus single out potential high-risk events.

Furthermore, if the probability of all conjunction events would be added, an estimation of the number of collisions per year could for instance be made. If the evolution of this number throughout history would be determined, an estimation of the start of the Kessler syndrome (e.g. a self-sustaining exponential increase of the amount of space debris) in the future could be predicted.

Bibliography

Alarcon-Rodriguez, J., Martinez-Fadrique, F., & Klinkrad, H. (2002). Collision risk assessment with a 'smart sieve' method. In *Proceedings of the Joint ESA-NASA Space-Flight Safety Conference*, (pp. 159–164). ESA/ESTEC, Noordwijk, The Netherlands. ESA SP-486.

Alarcon-Rodriguez, J., Martinez-Fadrique, F., & Klinkrad, H. (2004). Development of a collision risk assessment tool. *Advances in Space Research*, *34*(5), 1120–1124.

Alfano, S. (1994). Determining Satellite Close Approaches, Part II. *Journal of the Astronautical Sciences*, *42*(2), 143–152.

AS (2011). Tudat project website. `http://tudat.tudelft.nl/projects/tudat/`. Consulted 19.04.2011.

Astronautica (2012). Encyclopedia Astronautica - Prognoz. `http://www.astronautix.com/craft/prognoz.htm`. Consulted 07.01.2012.

Chan, J. C., & Navarro, D. (2001). Comparisons of NORAD two-line elements with INTELSAT orbital elements. In *Proceedings of the Third European Conference on Space Debris*. Darmstadt, Germany.

Christiansen, E. (2006). Hypervelocity Impact Technology Facility. `http://hitf.jsc.nasa.gov/hitfpub/main/index.html`. Consulted 15.05.2010.

Coppola, V. T., Dupont, S., Ring, K., & Stoner, F. (2009). Assessing Satellite Conjunctions for the Entire Space Catalog using COTS Multi-core Processor Hardware. In A. V. Rao, T. A. Lovell, F. K. Chan, & L. A. Cangahuala (Eds.) *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Volume 135, (pp. 1193–1206). Pittsburgh, Pennsylvania: Univelt, Incorporated. AAS 09-374.

Dawes, B., & Stewart, R. (2011). CPU timers - Boost 1.48.0. `http://www.boost.org/doc/libs/1_48_0/libs/timer/doc/cpu_timers.html`. Consulted 27.12.2011.

Donald J. Kessler and Burton G. Cour-Palais (1978). Collision Frequency of Artificial Satellites: The Creation of a Debris Belt. *Journal of Geophysical Research*, *83*(A6), 26372646.

Dong, W., & Chang-yin, Z. (2010). An Accuracy Analysis of the SGP4/SDP4 Model. *Chinese Astronomy and Astrophysics*, *34*(1), 69–76.

Einstein, A. (1915). Die Feldgleichungen der Gravitation (The Field Equations of Gravitation). *Königlich Preussische Akademie der Wissenschaften*, (p. 844847).

Flohrer, T., Krag, H., & Klinkrad, H. (2008). Assessment and Categorization of TLE Orbit Errors for the US SSN Catalogue. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, (pp. 513–524). Wailea, Maui, Hawaii.

Hall, R., Alfano, D. S., & Ocampo, A. (2010). Advances in Satellite Conjunction Analysis. In *Proceedings of the 11th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, Hawaii.

Healy, L. M. (1995). Close Conjunction Detection on Parallel Computer. *Journal of Guidance, Control and Dynamics*, *18*(4), 824–829.

Hoots, F. R., Crawford, L. L., & Roehrich, R. L. (1984). An Analytical Method to Determine Future Close Approaches Between Satellites. *Celestial Mechanics*, *33*, 143–158.

Hoots, F. R., & Roehrich, R. L. (1980). SPACETRACK Report NO. 3: Models for Propagation of NORAD Element Sets. Tech. rep., U.S. Air Force Aerospace Defense Command, Colorado Springs, CO.

Hoots, F. R., Schumacher, P. W., & Glover, R. A. (2004). History of Analytical Orbit Modeling in the U.S. Space Surveillance System. *Journal of Guidance, Control and Dynamics*, *27*(2), 174–185.

IADC (2010). Inter-agency Space Debris Coordination Committee. `http://www.iadc-online.org/index.cgi`. Consulted 17.05.2010.

Kelso, T. (2010a). CelesTrak. `http://celestrak.com/`. Consulted 10.05.2010.

Kelso, T. (2010b). How the Space Data Center is Improving Safety of Space Operations. In *Proceedings of the 11th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, Hawaii.

Kelso, T. (2010c). Iridium 33/Cosmos 2251 Collision. `http://celestrak.com/events/collision/`. Consulted 25.10.2010.

Kelso, T. S. (2007). Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides. In *Proceedings of the 17th AAS/AIAA Space Flight Mechanics Conference*. Sedona, AZ.

Kepler, J. (1609a). *Harmonice Mundi (Harmony of the Worlds)*. First ed.

Kepler, J. (1609b). *Philosophiae Naturalis Principia Mathematica (Mathematical Principles of Natural Philosophy)*.

K.F. Wakker (2007). *Astrodynamics - I (lecture notes AE4-874, Part 1)*. Delft, the Netherlands: Delft University of Technology.

Khutorovsky, Z., Boikov, V., , & Kamensky, S. (1993). Direct Method for the Analysis of Collision Probability of Artificial Space Objects in LEO Techniques, Results, and Applications. In *Proceedings of the First European Conference on Space Debris*, (p. 491508). ESA/ESOC, Darmstadt, Germany. ESA SD-01.

Klinkrad, H. (1993). Analysis of Near-Misses Between Operational Spacecraft and Catalogued Objects. In *Proceedings of the First European Conference on Space Debris*, (pp. 551–557). ESA/ESOC, Darmstadt, Germany. ESA SP-01.

Klinkrad, H. (1997). One year of conjunction events of ERS-1 and ERS-2 with objects of the USSPACECOM Catalog, Second European Conference on Space Debris. In *Proceedings of the Second European Conference on Space Debris*, (pp. 601–611). ESA/ESOC, Darmstadt, Germany. ESA SP-393.

Klinkrad, H. (2006). *Space Debris Models and Risk Analysis*. Springer & Praxis.

Klinkrad, H. (2007). On-orbit risk reduction collision avoidance. *Journal Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *221*(6), 955–962.

Klinkrad, H. (2009). ESA Space Debris Office. `http://www.esa.int/esaMI/Space_Debris/index.html`. Consulted 6.05.2010.

Klinkrad, H., Alarcon-Rodriguez, J., & Sanchez, N. (2005). Collision Avoidance for Operational ESA Satellites. In *Proceedings of the Fourth European Conference on Space Debris*, (pp. 509–515). ESA/ESOC, Darmstadt, Germany. ESA SP-587.

Legendre, P., Deguine, B., Garmier, R., & Revelin, B. (2006). Two Line Element Accuracy Assessment Based on a Mixture of Gaussian Laws. In *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Keystone, Colorado.

Legendre, P., Garmier, R., Revelin, B., & Delavault, S. (2008). Improvement of the TLE Accuracy Model Based on a Gaussian Mixture Depending on the Propagation Duration. In *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Honolulu, Hawaii.

Leloux, J. (2010a). Filtering Techniques for Orbital Debris Catalog Conjunction Analysis. MSc Literature Research, Delft University of Technology, Delft, the Netherlands.

Leloux, J. (2010b). Mechatronics in Space-Based Space Surveillance Systems. Essay for the MSc course Spacecraft Mechatronics (AE4S02), Delft University of Technology, Delft, the Netherlands.

Levit, C., & Marshall, W. (2010). Improved orbit predictions using two-line elements. *Advances in Space Research*. Article in Press, Accepted Manuscript.

Mate, A. A., Anton, D. E., & Cambriles, A. P. (2009). Assessing and Mitigating Sstellite Collision Risk With *closeap*. Tech. rep., GMV.

McDowell, J. (2010). Jonathan's Space Report. `http://www.planet4589.org/space/jsr/jsr.html`. Consulted 11.05.2010.

Miura, N. Z. (2009). *Comparison and Design of Simplified General Perturbation Models (SGP4) and Code for Nasa Johnson Space Center, Orbital Debris Program Office*. Master's thesis, Faculty of Aerospace Engineering, California Polytechnic State University, San Luis Obispo, CA, US.

Mohr, P. J., Taylor, B. N., & Newell, D. B. (2008). CODATA Recommended Values of the Fundamental Physical Constants: 2006. *Reviews of Modern Physics*, *80*(2), 633730.

Montenbruck, O., & Gill, E. (Eds.) (2001). *Satellite Orbits - Models, Methods, Applications*. Springer. Corrected 3rd Printing 2005.

Newton, I. (1687). *Philosophiae Naturalis Principia Mathematica (Mathematical Principles of Natural Philosophy)*. Royal Society of London for the Improvement of Natural Knowledge, first ed.

Sanchez Ortiz, N., Bello Mora, M., Graziano, M., Pina Caballero, F., Snchez Prez, J. M., & Klinkrad, H. (2001). Collision risk assessment. In H. Sawaya-Lacoste (Ed.) *Proceedings of the Third European Conference on Space Debris*, Vol. 1, (pp. 447–453). ESA/ESOC, Darmstadt, Germany: Noordwijk, Netherlands: ESA Publications Division. ESA SP-473.

ScienceDaily (2008). Targeting Space Debris Using Networks. `http://www.sciencedaily.com/releases/2008/10/081013112443.htm`. Consulted 21.12.2010.

Stansbery, E. (2009). NASA Orbital Debris Program. http://orbitaldebris.jsc.nasa.gov/. Consulted 6.05.2010.

Ting, W., & Hai, H. (2008). New Method to Determine Close Approaches Between Satellites. In *Proceedings of the Third International Association for the Advancement of Space Safety Conference*. Center for Advanced Defence Studies, Rome, Italy. ESA SP-662.

UCS (2010). UCS Satellite Database. http://www.ucsusa.org/. Consulted 11.05.2010.

UNOOSA (2010). United Nations Office for Outer Space Affairs. http://www.unoosa.org/. Consulted 17.05.2010.

USSTRATCOM (2004). Space Track. http://www.space-track.org/. Consulted 11.05.2010.

USSTRATCOM (2008). USSTRATCOM Space Control and Space Surveillance Fact Sheet. http://www.stratcom.mil/files/STRATCOM_Space_and_Control_Fact_Sheet-25_Feb_08.doc. Consulted 27.05.2010.

Vallado, D. A. (2007). *Fundamentals of Astrodynamics and Applications*. Hawthorne CA & New York NY: Microcosm Press & Springer, third ed.

Vallado, D. A., Crawford, P., Hujsak, R., & Kelso, T. (2006). Revisiting Spacetrack Report #3: Rev 1. In *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*. Keystone, CO.

Vorsmann, P. D. P. (2008). Institute of Aerospace Systems. http://www.ilr.ing.tu-bs.de/. Consulted 17.05.2010.

Wakker, K. (2007). *Astrodynamics - II (lecture notes AE4-874, Part 2)*. Delft, the Netherlands: Delft University of Technology.

Wang, R., Liu, J., & Zhang, Q. (2009). Propagation errors analysis of TLE data. *Advances in Space Research*, *43*(7), 1065–1069.

Wertz, J. R. (2001). *Mission Geometry; Orbit Constellation Design and Management*. Microcosm Press & Kluwer Academic Publishers.

Wikipedia (2010a). Earth Centered Inertial. http://en.wikipedia.org/wiki/Earth-centered_inertial. Consulted 17.06.2010.

Wikipedia (2010b). Nutation. http://en.wikipedia.org/wiki/Nutation. Consulted 17.06.2010.

Wikipedia (2010c). Space Debris. http://en.wikipedia.org/wiki/Space_debris. Consulted 7.05.2010.

Wolfgang O. Schall (1991). Orbital debris removal by laser radiation. *Acta Astronautica*, *24*, 343351.

Woodburn, J., Coppola, V., & Stoner, F. (2009). A Description of Filters for Minimizing the Time Required for Orbital Conjunction Computations. In A. V. Rao, T. A. Lovell, F. K. Chan, & L. A. Cangahuala (Eds.) *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Volume 135, (pp. 1157–1174). Pittsburgh, Pennsylvania: Univelt, Incorporated. AAS 09-372.

Appendix A

TLE Data Format

Table A.1: TLE Format: Field/Column Definition, Description (partly adapted from Kelso (2010a)) and Example

Line 0

Field	Columns	Description	Example
0.1	01-24	SatCat Common object name	CRYOSAT 2

Line 1

Field	Columns	Description	Example
1.1	01	TLE Line Number	1
1.2	03-07	Object Identification Number	36508
1.3	08	TLE Classification	U
1.4	10-11	Int. Designator (Launch year)	10
1.5	12-14	Int. Designator (Launch number of year)	013
1.6	15-17	Int. Designator (Piece of launch)	A
1.7	19-20	TLE Epoch Year	10
1.8	21-32	TLE Epoch Day	130.46204870
1.9	34-43	1^{st} Time Derivative of Mean Motion / 2	.00000003
1.10	45-52	2^{nd} Time Derivative of Mean Motion / 6	00000-0
1.11	54-61	B* drag term	00000+0
1.12	63-63	Ephemeris Type	0
1.13	65-68	TLE number	35
1.14	69	Checksum (Modulo 10)	6

Line 2

Field	Columns	Description	Example
2.1	01	TLE Line Number	2
2.2	03-07	Object Identification Number	36508
2.3	09-16	Orbit Inclination [deg]	92.0231
2.4	18-25	Right Ascension of Ascending Node [deg]	294.5305
2.5	27-33	Eccentricity	0013120
2.6	35-42	Argument of Perigee [deg]	162.0732
2.7	44-51	Mean Anomaly [deg]	198.1059
2.8	53-63	Mean Motion [rev/day]	14.52323328
2.9	64-68	Revolution Number at Epoch [rev]	462
2.10	69	Checksum (Modulo 10)	5

SDC/SCC TWO/THREE CARD ELEMENT SET - INTERNAL FORMAT

CARD NO.	SATNO					CARD NO.	ELEMENT NO.			INTERNATIONAL DESIGNATOR								EL SOURCE	EPHTYPE	ACC IND	EL LIFE		▨	EPOCH REV					▨	EPOCH TIME													▨	M_o MEAN ANOMALY *(Deg)*								▨	Ω_o RIGHT ASCENSION OF NODE *(Deg)*							▨	ω_o ARGUMENT OF PERIGEE *(Deg)*							DREV			SPX ID	DCC	PERIOD *(Min)*						CARD TYPE
										YEAR		LAUNCH NO.			PIECE								▨						▨	Y	D	D	D	•	D	D	D	D	D	D	D	D	▨									▨				•				▨				•													S	E	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1																							▨						▨					•									▨				•					▨								▨																			G
1																							▨						▨					•									▨				•					▨								▨																			G

CARD NO.	SATNO					e_o ECCENTRICITY								i_o INCLINATION *(Deg)*								▨	n_o MEAN NOTION *(Rev/Day)*											$\dot{n}_o/2$ MEAN MOTION DOT/2 *(Rev/Day^2)*											▨	$\ddot{n}_o/6$ MOTION DOT DOT/6 *(Rev/Day^3)*								▨	▨	▨	▨	BSTAR OR AGOM										PERIGEE HEIGHT (KM)						▨	ISTOP	▨	▨	▨	CARD TYPE
																						▨												S	•										▨	S•						S	E	▨	▨	▨	▨	S•								S	E							▨		▨	▨	▨	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
2						•											•					▨			•										•										▨									▨	▨	▨	▨																	▨		▨	▨	▨	G
2						•											•					▨			•										•										▨									▨	▨	▨	▨																	▨		▨	▨	▨	G

CARD NO.	SATNO					COUNTRY	PAYLOAD	MOVER	X SECTION	ORBIT TYPE		SAT CLASS		▨	REMC	FREQ		VMAG LIM			▨	OBSEL	OBSPAN		▨	FTAG	HIPRI	Δ H MULT	BETA MULT	▨	PHANG	SMAG			▨	FPS - 85 TASK				COMMENTS																																							CARD TYPE
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
3														▨							▨				▨					▨					▨																																												G
3														▨							▨				▨					▨					▨																																												G

SDC/SCC TWO CARD ELEMENT SET - TRANSMISSION FORMAT

CARD NO.	▨	SATELLITE NUMBER					CLASS	▨	INTERNATIONAL DESIGNATOR								▨	EPOCH TIME														▨	$\dot{n}_o/2$ (Rev/Day^2) or B (m^2/kg)										▨	$\ddot{n}_o/6$ MOTION DOT DOT/6 *(Rev/Day^3)*								▨	BSTAR (ER^{-1}) or AGOM (m^2/kg)								▨	EPHTYPE	▨	ELEMENT NO.				CK SUM	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
	▨							▨	YEAR		LAUNCH NO.			PIECE			▨	Y	Y	D	D	D	•	D	D	D	D	D	D	D	D	▨	S	•									▨	S•						S	E	▨	S•						S	E	▨		▨						▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	▨							▨									▨						•									▨		•									▨									▨									▨		▨						▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
1	▨							▨									▨						•									▨		•									▨									▨									▨		▨						▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

CARD NO.	▨	SATELLITE NUMBER					▨	i_o INCLINATION *(Deg)*								▨	Ω_o RIGHT ASCENSION OF NODE *(Deg)*								▨	e_o ECCENTRICITY							▨	ω_o ARGUMENT OF PERIGEE *(Deg)*								▨	M_o MEAN ANOMALY *(Deg)*								▨	n_o MEAN NOTION *(Rev/Day)*											EPOCH REV					CK SUM	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
	▨						▨				•					▨				•					▨	•							▨				•					▨				•					▨			•															▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
2	▨						▨				•					▨				•					▨								▨				•					▨				•					▨			•															▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
2	▨						▨				•					▨				•					▨								▨				•					▨				•					▨			•															▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

ELEMENT SET FORMATS

Appendix B

SGP4/SDP4 Model Equations

improvements by Lane and Cranford[12] in 1969. Fitzpatrick gives a brief account of Lane's density modeling technique in Ref. 13.

A very important contribution to analytic satellite theory was made by Lyddane[14] in 1963. Lyddane showed that the Brouwer[5,6] solution based on Delaunay variables could be reformulated in terms of Poincaré variables to avoid the small divisors of eccentricity and the sine of inclination while maintaining the first-order character of the theory.

Operational Implementations: 1964–1979

The transition from journal article to operational implementation took two paths. The tracking operation of NAVSPASUR adopted the entire 1959 solution of Brouwer[5,6] with the modifications developed by Lyddane[14] to avoid small divisors of eccentricity or the sine of inclination. This analytic satellite prediction model is now known as Position and Partials as functions of Time (PPT3). The equations of the original PPT model were implemented on an IBM 7090 computer in 1964 under the guidance of Richard H. Smith, who also provided supplemental equations to account for atmospheric drag. At that time, the results of Brouwer and Hori[8,9] could not be implemented operationally because of computer limitations, and the results of Lane and Cranford[12] were not yet available. Smith adapted ideas from King-Hele[15] in a simple original model that is still in use. His semi-empirical drag model assumes that the effect of atmospheric drag on the mean motion can be represented as a quadratic time function. The linear and quadratic coefficients are treated as solved-for parameters during the orbit determination process. A time rate of change of the eccentricity is represented in terms of the mean motion rate by the following equations:

$$\dot{e}_0 = e_0\left(1 - e_0^2\right)\dot{a}_0/a_0, \qquad \dot{a}_0 = -(4/3)a_0/n_0(\dot{n}_0/2)$$

The integral of the mean motion equation provides the model for along-track drag effect.

PPT retains all long-periodic terms, including the ones with a zero divisor at the critical inclination. However, PPT handles these critical terms in a special way, as described in Appendix A.F. A special feature of PPT is that the "mean" mean motion is defined differently from Brouwer's quantity[5,6] of the same name. Brouwer defined mean motion in terms of mean semimajor axis by essentially the Keplerian formula. However, for PPT, it was decided for computational reasons to define the mean motion as the entire coefficient of time in the linear term of the perturbed mean anomaly. That is, the PPT mean motion includes the zonal secular perturbation rate of mean anomaly that Brouwer derived. As a result, the expression for PPT mean motion explicitly contains perturbation parameters and functions of the other mean elements, similarly to the definition adopted by Kozai.[7] Numerically, the PPT mean motion is closer to Kozai's value than to Brouwer's.

The other path from journal article to operational implementation took place in Colorado Springs. In 1961, the NSSCC was relocated to Colorado Springs, Colorado, and became known as the Space Detection and Tracking System (SPADATS) Center. The NSSCC algorithms were rehosted on a Philco Model 211 computer, and the group at Hanscom began to serve as the backup for the SPADATS Center. Following the rehosting in Colorado Springs, Hilton[16] provided updated documentation of the NSSCC algorithms. In 1960, Aeronutronic had begun developing the astrodynamics basis for a new system. The analytic orbit prediction model was based on the works of Brouwer[5,6] and Kozai.[7] To avoid small divisors of eccentricity or the sine of inclination, Arsenault et al.[17] transformed the solution to a series in non-singular parameters, keeping only the most important terms. They included from Brouwer only those long- and short-period terms in position that do not contain eccentricity as a factor. They also adopted from Kozai the non-Keplerian convention relating mean motion to semimajor axis. The model is known as the simplified general perturbations (SGP) model. A complete documentation of SGP is provided by Hilton and Kuhlman.[18] Atmospheric drag was included in a manner similar to that of Smith except that the time rate of change of eccentricity was derived based on the assumption that perigee height remains constant as semimajor axis shrinks. In addition to becoming the principal analytic prediction model for centralized processing, SGP was also implemented at many of the satellite tracking sites around the world. In 1964, the SGP model became the primary orbital prediction model for the SPADATS system.

The improvement offered by an analytic rather than an empirical density model led to a decision to implement the development of Lane and Cranford.[12] However, by 1969 the number of satellites in the catalog had grown to a point that computers would not be able to manage the extensive terms in the model. Consequently, a simplified version of the Lane and Cranford work, known as SGP4, was developed and implemented operationally in 1970.

The simplifications leading to SGP4 were accomplished by retaining only the main terms that modeled the secular effect of drag. Similarly, the gravitational modeling was shortened by retaining from Brouwer[5,6] only those long- and short-periodic terms in position that do not contain eccentricity as a factor. The details of the derivation of SGP4 from the complete development of Lane and Cranford[12] were documented in 1979 by Lane and Hoots.[19] The SGP4 model was used side by side with the SGP model until 1979 when it became the sole model for satellite catalog maintenance.

Deep-Space Modeling: 1965–1997

In 1965, the first highly-eccentric, 12-h-period satellite was launched. Soon it became apparent that a theory was needed that included terms to account for lunar and solar gravitation, as well as the resonance effects of Earth tesseral harmonics. A semianalytic treatment of this special class of orbits, which included lunar and solar gravity as well as geopotential resonance effects, was developed by Bowman[20] in 1967. By 1977, Hujsak[21] had incorporated portions of Bowman's work in a new first-order model, which included all perturbations treated by Bowman and an extension to geosynchronous satellites. This new model was fully integrated with the SGP4 model for near Earth satellites. This work completed the SGP4 model in use today. A complete listing of the equations was provided by Hoots and Roehrich[22] and is repeated in Appendix B.

In 1997 the lunar, solar, and resonance terms from the SGP4 model were added to the Naval Space Command PPT model to provide improved prediction of higher altitude satellites. This modified model became known as PPT3 and is documented in the work of Schumacher and Glover.[23] A complete listing of the equations is provided in Appendix C.

Conclusions

For nearly a half century the U.S. Space Surveillance system has been tracking and maintaining a catalog of manmade Earth orbiting satellites, now consisting of more than 10,000 objects. The tremendous success of this endeavor has been due in part to independent but complementary efforts by both the U.S. Navy and the U.S. Air Force at their mission centers in Dahlgren, Virginia and Colorado Springs, Colorado, respectively. Today the operational centers still depend largely on the original orbit models and applications of the pioneers of the 1950s and 1960s.

Appendix A: Deep Space Equations

A. Initialization for Secular and Long-Period Coefficients of Lunar and Solar Gravity

The first step in the initialization process is to compute the position of the moon and sun at the epoch time of the satellite element set using the following equations:

$$\Omega_{m_\varepsilon} = \left[\Omega_{m_{\varepsilon 0}} + \dot{\Omega}_{m_\varepsilon}\Delta t + \ddot{\Omega}_{m_\varepsilon}\Delta t^2 + \dddot{\Omega}_{m_\varepsilon}\Delta t^3\right]_{\mathrm{mod}2\pi}$$

$$\cos I_m = \cos\varepsilon\cos I_{m_\varepsilon} - \sin\varepsilon\sin I_{m_\varepsilon}\cos\Omega_{m_\varepsilon}$$

The lunar longitude of perigee referred to the ecliptic is

$$\gamma = u_{0_\varepsilon} + \dot{u}_\varepsilon\Delta t + \ddot{u}_\varepsilon\Delta t^2 + \dddot{u}_\varepsilon\Delta t^3$$

where u_{0_ε} is the epoch longitude of perigee (with respect to the ecliptic).

The lunar right ascension of the ascending node referred to the equator is

$$\sin\Omega_m = \frac{\sin I_{m_\varepsilon}\sin\Omega_{m_\varepsilon}}{\sin I_m}, \qquad \cos\Omega_m = \sqrt{1 - \sin^2\Omega_m}$$

Then

$$\sin\Delta = \frac{\sin\varepsilon \sin\Omega_{m_\varepsilon}}{\sin I_m}$$

$$\cos\Delta = \cos\Omega_m \cos\Omega_{m_\varepsilon} + \sin\Omega_m \sin\Omega_{m_\varepsilon} \cos\varepsilon$$

$$\Delta = \tan^{-1}\left(\frac{\sin\Delta}{\cos\Delta}\right), \qquad \omega_m = \gamma - \Omega_{m_\varepsilon} + \Delta = G_{o_m}$$

$$M_s = M_0 + \dot{M}\Delta t + \ddot{M}\Delta t^2 + \dddot{M}\Delta t^3$$

where Δt is the time since the lunar/solar ephemeris epoch and where the elements of the moon and sun are obtained from equations supplied in Ref. 24 (pages 107 and 98 for the moon and sun, respectively). The constants for calculating lunar and solar positions are defined as follows.

The moon's inclination with respect to the ecliptic, in degrees: $I_{m_\varepsilon} = 5.145396374$.

The obliquity of the ecliptic, in degrees: $\varepsilon = 23.4441$.

Lunar eccentricity: $e_m = 0.05490$.

Solar eccentricity: $e_s = 0.01675$.

Lunar mean motion, in radians per minute: $n_m = 1.583521770 \times 10^{-4}$.

Solar mean motion, in radians per minute: $n_s = 1.19459 \times 10^{-5}$.

Solar inclination, in degree: $I_s = \varepsilon = 23.4441$.

Constants, in degrees:

$$\Omega_s = 0, \qquad \omega_s = 281.2208 = G_{0_s}$$

Lunar perturbation coefficient, in radians per minute: $C_m = 4.796806521 \times 10^{-7}$.

Solar perturbation coefficient, in radians per minute: $C_s = 2.98647972 \times 10^{-6}$.

The lunar and solar elements are epoched at 0.5 January 1900 (Julian date 2415020.0).

For each body X, either the sun or the moon, terms are calculated that depend solely on the epoch satellite orbital elements Ω_0, ω_0, and I_0 and the orbital elements of the moon and sun. In the calculations of these terms, the following conventions apply:

1) Quantities on the right side of the equation with subscript 0 refer to mean elements of the satellite orbit.

2) Quantities on the right side of the equation with subscript X refer to the orbit of body X.

3) Quantities on the left side of the equation refer to the satellite's orbit as affected exclusively by body X.

4) Here n_x = mean motion of perturbing body X.

5) All orbital elements of the moon and sun, except mean anomaly, are treated as constant at the epoch of the satellite.

Calculate the constants:

$$a_1 = \cos\omega_x \cos(\Omega_0 - \Omega_x) + \sin\omega_x \cos I_x \sin(\Omega_0 - \Omega_x)$$

$$a_3 = -\sin\omega_x \cos(\Omega_0 - \Omega_x) + \cos\omega_x \cos I_x \sin(\Omega_0 - \Omega_x)$$

$$a_7 = -\cos\omega_x \sin(\Omega_0 - \Omega_x) + \sin\omega_x \cos I_x \cos(\Omega_0 - \Omega_x)$$

$$a_8 = \sin\omega_x \sin I_x$$

$$a_9 = \sin\omega_x \sin(\Omega_0 - \Omega_x) + \cos\omega_x \cos I_x \cos(\Omega_0 - \Omega_x)$$

$$a_{10} = \cos\omega_x \sin I_x, \qquad a_2 = a_7 \cos I_0'' + a_8 \sin I_0''$$

$$a_4 = a_9 \cos I_0'' + a_{10} \sin I_0'', \qquad a_5 = -a_7 \sin I_0'' + a_8 \cos I_0''$$

$$a_6 = -a_9 \sin I_0'' + a_{10} \cos I_0''$$

$$X_1 = a_1 \cos\omega_0 + a_2 \sin\omega_0, \qquad X_2 = a_3 \cos\omega_0 + a_4 \sin\omega_0$$

$$X_3 = -a_1 \sin\omega_0 + a_2 \cos\omega_0, \qquad X_4 = -a_3 \sin\omega_0 + a_4 \cos\omega_0$$

$$X_5 = a_5 \sin\omega_0, \qquad X_6 = a_6 \sin\omega_0$$

$$X_7 = a_5 \cos\omega_0, \qquad X_8 = a_6 \cos\omega_0$$

$$Z_{31} = 12X_1^2 - 3X_3^2, \qquad Z_{32} = 24X_1X_2 - 6X_3X_4$$

$$Z_{33} = 12X_2^2 - 3X_4^2, \qquad Z_1 = 6\left(a_1^2 + a_2^2\right) + \left(1 + e_0^2\right)Z_{31}$$

$$Z_2 = 12(a_1a_3 + a_2a_4) + \left(1 + e_0^2\right)Z_{32}$$

$$Z_3 = 6\left(a_3^2 + a_4^2\right) + \left(1 + e_0^2\right)Z_{33}$$

$$Z_{11} = -6a_1a_5 + e_0^2(-24X_1X_7 - 6X_3X_5)$$

$$Z_{13} = -6a_3a_6 + e_0^2(-24X_2X_8 - 6X_4X_6)$$

$$Z_{21} = 6a_2a_5 + e_0^2(24X_1X_5 - 6X_3X_7)$$

$$Z_{23} = 6a_4a_6 + e_0^2(24X_2X_6 - 6X_4X_8)$$

$$Z_{22} = 6a_4a_5 + 6a_2a_6 + e_0^2(24X_2X_5 + 24X_1X_6 - 6X_4X_7 - 6X_3X_8)$$

$$Z_{12} = -6a_1a_6 - 6a_3a_5 - e_0^2(24X_2X_7 + 24X_1X_8 \\ + 6X_3X_6 + 6X_4X_5)$$

The secular rates are computed separately for both the sun and moon and then are combined into a single third-body secular rate. The secular rates due to the third-body perturbation are

$$\dot{a}_x = 0, \qquad \dot{e}_x = -15C_xn_x\frac{e_0\eta_0}{n_0}(X_1X_3 + X_2X_4)$$

$$\dot{I}_x = \frac{-C_xn_x}{2n_0\eta_0}(Z_{11} + Z_{13})$$

$$\dot{M}_x = \frac{-C_xn_x}{n_0}\left(Z_1 + Z_3 - 14 - 6e_0^2\right)$$

$$\dot{\Omega}_x = \begin{cases} \dfrac{C_xn_x}{2n_0\eta_0 \sin I_0''}(Z_{21} + Z_{23}) & \text{if} \quad I_0'' \geq 3 \text{ deg} \\ 0 & \text{if} \quad I_0'' < 3 \text{ deg} \end{cases}$$

$$\dot{\omega}_x = \begin{cases} \dfrac{C_xn_x\eta_0}{n_0}(Z_{31} + Z_{33} - 6) - \dot{\Omega}_x \cos I_0'' & \text{if} \quad I_0'' \geq 3 \text{ deg} \\ \dfrac{C_xn_x\eta_0}{n_0}(Z_{31} + Z_{33} - 6) & \text{if} \quad I_0'' < 3 \text{ deg} \end{cases}$$

B. Initialization for Resonance Effects of Earth Gravity

If the satellite period in minutes is in the closed interval [1200, 1800], then it is assumed to be in a 1-day resonance condition. The following constants are satellite independent for 1-day period satellites:

$$Q_{22} = \sqrt{C_{22}^2 + S_{22}^2}, \qquad Q_{31} = \sqrt{C_{31}^2 + S_{31}^2}$$

$$Q_{33} = \sqrt{C_{33}^2 + S_{33}^2}$$

where

$$Q_{31} = 2.1460748 \times 10^{-6}, \qquad Q_{22} = 1.7891679 \times 10^{-6}$$

$$Q_{33} = 2.2123015 \times 10^{-7}$$

The three phase angles are

$$\lambda_{22} = \tfrac{1}{2}\tan^{-1}(S_{22}/C_{22}), \qquad \lambda_{31} = \tan^{-1}(S_{31}/C_{31})$$

$$\lambda_{33} = \tfrac{1}{3}\tan^{-1}(S_{33}/C_{33})$$

where

$$\lambda_{31} = 0.13130908, \qquad \lambda_{22} = 2.88431980, \qquad \lambda_{33} = 0.37448087$$

The functions of inclination, F, and eccentricity, G, for 1-day period satellites (which are dependent solely on epoch quantities) are calculated as follows:

$$F_{220} = (3/4)\left(1 + \cos I_0''\right)^2$$

$$F_{311} = (15/16)\sin^2 I_0''(1 + 3\cos I_0'') - (3/4)(1 + \cos I_0'')$$

$$F_{330} = (15/8)(1 + \cos I_0'')^3$$

$$G_{200} = 1 - (5/2)e_0''^2 + (13/16)e_0''^4, \qquad G_{310} = 1 + 2e_0''^2$$

$$G_{300} = 1 - 6e_0''^2 + (423/64)e_0''^4$$

Compute the following coefficients of the resonance terms:

$$\delta_1 = \left(3n_0^2/a_0^3\right)F_{311}G_{310}Q_{31}, \qquad \delta_2 = \left(6n_0^2/a_0^2\right)F_{220}G_{200}Q_{22}$$

$$\delta_3 = \left(9n_0^2/a_0^3\right)F_{330}G_{300}Q_{33}$$

If the satellite period in minutes is in the closed interval [680, 760] and the eccentricity is greater than or equal to 0.5, then it is assumed to be in a 0.5-day resonance condition. The following constants are satellite-independent for 0.5-day period satellites:

$$\sqrt{C_{22}^2 + S_{22}^2} = 1.7891679 \times 10^{-6}$$

$$\sqrt{C_{32}^2 + S_{32}^2} = 3.7393792 \times 10^{-7}$$

$$\sqrt{C_{44}^2 + S_{44}^2} = 7.3636953 \times 10^{-9}$$

$$\sqrt{C_{52}^2 + S_{52}^2} = 1.1428639 \times 10^{-7}$$

$$\sqrt{C_{54}^2 + S_{54}^2} = 2.1765803 \times 10^{-9}$$

$$D_{lmpq} = \frac{3n_0^2}{a_0^l}\sqrt{C_{lm}^2 + S_{lm}^2}\,F_{lmp}G_{lpq}$$

for the following (l, m, p, q) quadruples: (2,2,0,1), (2,2,1,1), (3,2,1,0), (3,2,2,2), (5,2,2,0), (5,2,3,2), (4,4,2,2), (5,4,2,1), (5,4,2,3), and (4,4,1,0). The functions of inclination (dependent on epoch quantities) are as follows:

$$F_{220} = (3/4)(1 + \cos I_0'')^2, \qquad F_{221} = (3/2)(\sin I_0'')^2$$

$$F_{321} = (15/8)\sin I_0''\left(1 - 2\cos I_0'' - 3\cos^2 I_0''\right)$$

$$F_{322} = (-15/8)\sin I_0''\left(1 + 2\cos I_0'' - 3\cos^2 I_0''\right)$$

$$F_{441} = (105/4)\sin^2 I_0''(1 + \cos I_0'')^2, \qquad F_{442} = (315/8)\sin^4 I_0''$$

$$\begin{aligned} F_{522} = {} & (315/32)\big\{\sin^3 I_0'' - 2\sin^3 I_0''\cos I_0'' - 5\sin^3 I_0''\cos^2 I_0'' \\ & + \sin I_0''\left[(-2/3) + (4/3)\cos I_0'' + 2\cos^2 I_0''\right]\big\} \end{aligned}$$

$$\begin{aligned} F_{523} = {} & (105/16)\sin I_0''\big\{1 + 2\cos I_0'' - 3\cos^2 I_0'' \\ & - (3/2)\sin^2 I_0''\left[1 + 2\cos I_0'' - 5\cos^2 I_0''\right]\big\} \end{aligned}$$

$$\begin{aligned} F_{542} = {} & (945/32)\sin I_0''\big\{2 - 8\cos I_0'' \\ & + \cos^2 I_0''\left[-12 + 8\cos I_0'' + 10\cos^2 I_0''\right]\big\} \end{aligned}$$

$$\begin{aligned} F_{543} = {} & (945/32)\sin I_0''\big\{\cos^2 I_0''\left[12 + 8\cos I_0'' - 10\cos^2 I_0''\right] \\ & - 2 - 8\cos I_0''\big\} \end{aligned}$$

and the functions of eccentricity are

$$G_{211} = \begin{cases} 3.616 - 13.247e_0'' + 16.29e_0''^2 & e_0'' \le 0.65 \\ -72.099 + 331.819e_0'' - 508.738e_0''^2 + 266.724e_0''^3 & e_0'' > 0.65 \end{cases}$$

$$G_{201} = -0.306 - 0.44(e_0'' - 0.64)$$

$$G_{310} = \begin{cases} -19.302 + 117.39e_0'' - 228.419e_0''^2 + 156.591e_0''^3 & e_0'' \le 0.65 \\ -346.844 + 1582.851e_0'' - 2415.925e_0''^2 + 1246.113e_0''^3 & e_0'' > 0.65 \end{cases}$$

$$G_{322} = \begin{cases} -18.9068 + 109.7927e_0'' - 214.6334e_0''^2 + 146.5816e_0''^2 & e_0'' \le 0.65 \\ -342.585 + 1554.908e_0'' - 2366.899e_0''^2 + 1215.972e_0''^3 & e_0'' > 0.65 \end{cases}$$

$$G_{410} = \begin{cases} -41.122 + 242.694e_0'' - 471.094e_0''^2 + 313.953e_0''^3 & e_0'' \le 0.65 \\ -1052.797 + 4758.686e_0'' - 7193.992e_0''^2 + 3651.957e_0''^3 & e_0'' > 0.65 \end{cases}$$

$$G_{422} = \begin{cases} -146.407 + 841.88e_0'' - 1629.014e_0''^2 + 1083.435e_0''^3 & e_0'' \le 0.65 \\ -3581.69 + 16178.11e_0'' - 24462.77e_0''^2 + 12422.52e_0''^3 & e_0'' > 0.65 \end{cases}$$

$$G_{520} = \begin{cases} -532.114 + 3017.977e_0'' - 5740.032e_0''^2 + 3708.276e_0''^3 & e_0'' \le 0.65 \\ 1464.74 - 4664.75e_0'' + 3763.64e_0''^2 & 0.65 < e_0'' < 0.715 \\ -5149.66 + 29936.92e_0'' - 54087.36e_0''^2 + 31324.56e_0''^3 & e_0'' \ge 0.715 \end{cases}$$

$$G_{521} = \begin{cases} -822.71072 + 4568.6173e_0'' - 8491.4146e_0''^2 + 5337.524e_0''^3 & e_0'' < 0.70 \\ -51752.104 + 218913.95e_0'' - 309468.16e_0''^2 + 146349.42e_0''^3 & e_0'' \ge 0.70 \end{cases}$$

$$G_{532} = \begin{cases} -853.666 + 4690.25e_0'' - 8624.77e_0''^2 + 5341.4e_0''^3 & e_0'' < 0.70 \\ -40023.88 + 170470.89e_0'' - 242699.48e_0''^2 + 115605.82e_0''^3 & e_0'' \ge 0.70 \end{cases}$$

$$G_{533} = \begin{cases} -919.2277 + 4988.61e_0'' - 9064.77e_0''^2 + 5542.21e_0''^3 & e_0'' < 0.70 \\ -37995.78 + 161616.52e_0'' - 229838.2e_0''^2 + 109377.94e_0''^3 & e_0'' \ge 0.70 \end{cases}$$

C. Secular Updates for Effects of Lunar and Solar Gravity

The secular effects of lunar and solar gravity are included by the following equations:

$$M = M + \dot{M}_{\mathrm{LS}}(t - t_0), \qquad \omega = \omega + \dot{\omega}_{\mathrm{LS}}(t - t_0)$$

$$\Omega = \Omega + \dot{\Omega}_{\mathrm{LS}}(t - t_0), \qquad e = e_0 + \dot{e}_{\mathrm{LS}}(t - t_0)$$

$$I = I_0 + \dot{I}_{\mathrm{LS}}(t - t_0)$$

where the rates with subscript LS are the sum of the effects of lunar and solar perturbations.

D. Secular Update for Resonance Effects of Earth Gravity

Define an auxiliary variable λ for the resonance treatment as

$$\lambda = M + \Omega + \omega - \theta_G$$

for orbits in the 1-day-period band and

$$\lambda = M + 2\Omega - 2\theta_G$$

for orbits in the 0.5-day-period band where θ_G is the longitude of Greenwich. Simultaneously, numerically integrate the resonance equations for mean motion and the resonance variable λ. The numerical integration scheme is the Euler–Maclaurin method with a step size of 12 h (720 min).

At epoch

$$\lambda_i = \lambda_0, \qquad n_i = n_0$$

the Euler–Maclaurin equations are

$$\lambda_i = \lambda_{i-1} + \dot{\lambda}_i(\Delta t) + (\ddot{\lambda}_i/2)(\Delta t)^2$$

$$n_i = n_{i-1} + \dot{n}_i(\Delta t) + (\ddot{n}_i/2)(\Delta t)^2$$

The derivatives are computed as follows.

For 1-day-period orbits:

$$\dot{\lambda}_1 = n_i + \dot{\lambda}_0$$

$$\dot{n}_i = \delta_1 \sin(\lambda_i - \lambda_{31}) + \delta_2 \sin(2\lambda_i - 2\lambda_{22}) + \delta_3 \sin(3\lambda_i - 3\lambda_{33})$$

$$\ddot{\lambda}_i/2 = \dot{n}_i/2$$

$$\ddot{n}_i/2 = (\dot{\lambda}_i/2)[\delta_1 \cos(\lambda_i - \lambda_{31}) + 2\delta_2 \cos(2\lambda_i - 2\lambda_{22}) + 3\delta_3 \cos(3\lambda_i - 3\lambda_{33})]$$

For the 0.5-day-period orbits [using the 0.5-day resonance (l, m, p, q) quadruplets]:

$$\dot{\lambda}_i = n_i + \dot{\lambda}_0$$

$$\dot{n}_i = \sum_{(i,m,p,q)} D_{lmpq} \sin\left[(l - 2p)\omega_i + \frac{m}{2}\lambda_i - G_{lm}\right]$$

$$\frac{\ddot{\lambda}_i}{2} = \frac{\dot{n}_i}{2}$$

$$\frac{\ddot{n}_i}{2} = \frac{\dot{\lambda}_i}{2}\left\{\sum_{(l,m,p,q)} \frac{m}{2} D_{lmpq} \cos\left[(l - 2p)\omega_i + \frac{m}{2}\lambda_i - G_{lm}\right]\right\}$$

where

$$G_{22} = 5.7686396, \qquad G_{32} = 0.95240898, \qquad G_{44} = 1.8014998$$

$$G_{52} = 1.0508330, \qquad G_{54} = 4.4108898$$

and $\omega_i = \omega_0 + \dot{\omega}_0 \Delta t$ is the secularly updated argument of perigee.

The 1-day-period initial conditions are

$$\lambda_0 = M_0 + \omega_0 + \Omega_0 - \theta_0$$

$$\dot{\lambda}_0 = \dot{M}_0 + \dot{M}_{\mathrm{LS}} + \dot{\Omega}_0 + \dot{\Omega}_{\mathrm{LS}} + \dot{\omega}_0 + \dot{\omega}_{\mathrm{LS}} - \dot{\theta}$$

where θ is the Greenwich hour angle.

The 0.5-day initial conditions are

$$\lambda_0 = M_0 + 2\Omega_0 - 2\theta_0$$

$$\dot{\lambda}_0 = \dot{M}_0 + \dot{M}_{\mathrm{LS}} + 2\dot{\Omega}_0 + 2\dot{\Omega}_{\mathrm{LS}} - 2\dot{\theta}$$

When λ_i, n_i are obtained at the time of interest, compute

$$n = n_i$$

$$M = \begin{cases} \lambda_i - \Omega_s - \omega_s + \theta_t & \text{for 1-day period} \\ \lambda_i - 2\Omega_s + 2\theta_t & \text{for 1/2-day period} \end{cases}$$

and Ω_s and ω_s are the mean elements updated with the secular rates of the other perturbations.

E. Update for Long-Period Periodic Effects of Lunar and Solar Gravity

The true anomaly of the perturbing body is approximated by

$$f_X = M_X + 2e_X \sin M_X$$

Define

$$F_2 = \tfrac{1}{2}\sin^2 f_X - \tfrac{1}{4}, \qquad F_3 = -\tfrac{1}{2}\sin f_x \cos f_X$$

We have, for each perturbing body,

$$\delta e_x = -(30\eta_0 C_x e_0/n_0)[F_2(X_2X_3 + X_1X_4) + F_3(X_2X_4 - X_1X_3)]$$

$$\delta I_x = -(C_x/n_0\eta_0)[F_2 Z_{12} + F_3(Z_{13} - Z_{11})]$$

$$\delta M_x = -(2C_x/n_0)\left[F_2 Z_2 + F_3(Z_3 - Z_1) - 3e_x \sin f_x\left(7 + 3e_0^2\right)\right]$$

$$(\delta\omega_x + \cos I_x \delta\Omega_x) = (2\eta_0 C_x/n_0)[F_2 Z_{32} + F_3(Z_{33} - Z_{31}) - 9e_x \sin f_x]$$

$$\sin I_x \delta\Omega_x = (C_x/n_0\eta_0)[F_2 Z_{22} + F_3(Z_{23} - Z_{21})]$$

The long-period periodics are computed separately for both the sun and moon and then combined into a single third-body long-period term.

F. Critical Inclination in PPT3

Brouwer[6] showed that the perturbation theory should remain valid for all inclinations except for an interval of about 1.5 deg on either side of the critical inclination. Within this narrow range, special procedures are required in any implementation of a Brouwer-type theory. In PPT3, the procedure is as follows. First, compute the critical factor

$$x = 1 - 5\cos^2 I''$$

This factor vanishes at about $I'' = 63.43$ deg. Then all occurrences of $1/x$ are replaced by the approximation

$$1/x \approx [1 - \exp(-100x^2)]/x \equiv C(x)$$

Away from the critical inclination, $C(x)$ tends rapidly to $1/x$. However, in the neighborhood of the critical inclination, $C(x)$ is bounded and in fact vanishes at $x = 0$. It can be shown that $C(x)$ has a maximum amplitude of about 6.382 and that there are two extrema having this amplitude, a minimum near $I'' = 61.86$ deg and a maximum near $I'' = 65.08$ deg.

The value of $C(x)$ is not computed directly from the preceding expression because of numerical ill conditioning. Even the direct power-series expansion of the exponential function exhibits poor convergence because of the factor of 100. Both problems are avoided by repeatedly factoring the numerator of $C(x)$, expanding one factor in series, and formally canceling x from the denominator. In particular, repeatedly factor the difference of squares to obtain the exact expression

$$C(x) = \frac{1}{x}[1 - \exp(-\beta x^2)] \prod_{m=0}^{10} [1 + \exp(-2^m \beta x^2)]$$

where $\beta = 100/2^{11}$. Then the first factor is computed by a series expansion truncated to a practical number of terms, which is feasible because of the smallness of β. PPT3 currently uses a 12-term expansion:

$$\frac{[1 - \exp(-\beta x^2)]}{x} \cong \beta x \sum_{n=0}^{12} (-1)^n \frac{\beta^n x^{2n}}{(n+1)!}$$

Appendix B: SGP4 Model

The U.S. Space Command two-line element sets can be used for prediction with SGP4. All equations are taken from Ref. 22. The element set consists of the following:

t_0 = epoch time
n_0 = mean motion, revolutions/day
e_0 = eccentricity
i_0 = inclination, deg
ω_0 = argument of perigee, deg
Ω_0 = right ascension of ascending node, deg
M_0 = mean anomaly, deg
B^* = atmospheric drag coefficient, 1/Earth radii

where all orbital elements except mean motion are the mean double-prime quantities defined by Brouwer[6] and where the subscript 0 will indicate the value of a quantity at epoch. The mean motion on the two-line element set follows the convention of Kozai.[7]

A. Initialization

Many terms used in the prediction of SGP4 are independent of time. Thus, the algorithm begins with computation of numerous constant terms. The first step in the initialization is the recovery of the Brouwer mean motion from the Kozai mean motion by the equations

$$a_1 = \left(\frac{k_e}{n_0}\right)^{\frac{2}{3}}, \qquad \delta_1 = \frac{3}{2}\frac{k_2}{a_1^2}\frac{\left(3\cos^2 i_0 - 1\right)}{\left(1 - e_0^2\right)^{\frac{3}{2}}}$$

$$a_2 = a_1\left(1 - \frac{1}{3}\delta_1 - \delta_1^2 - \frac{134}{81}\delta_1^3\right), \qquad \delta_0 = \frac{3}{2}\frac{k_2}{a_2^2}\frac{\left(3\cos^2 i_0 - 1\right)}{\left(1 - e_0^2\right)^{\frac{3}{2}}}$$

$$n_0'' = \frac{n_0}{1 + \delta_0}, \qquad a_0'' = \left(\frac{k_e}{n_0''}\right)^{\frac{2}{3}}$$

where

k_2 = $\frac{1}{2} J_2 a_{\mathrm{E}}^2$, (Earth radii)^2
J_2 = 1.082616×10^{-3}
k_e = $\sqrt{GM} = 0.0743669161$, (Earth radii)$^{1.5}$/min
G = universal gravitational constant
M = mass of the Earth
a_E = equatorial radius of the Earth

The SGP4 model is set in the Fundamental Katalog 4 (FK4) and World Geodetic Survey 72 (WGS72) reference standards, referred to the Julian 2000 (J2000.0) epoch.

From this point on, the mean motion n_0'' and the semimajor axis a_0'' follow the Brouwer convention. Also, all quantities on the right-hand side of equations are understood to be double-prime mean elements.

1. *Initialization for Secular Effects of Atmospheric Drag*

Atmospheric drag modeling is based on a power-law density function[10] given by

$$\rho = \rho_0 (q_0 - s)^4 / (r - s)^4$$

where r is the radial distance of the satellite from the center of the Earth with q_0 and s being altitude parameters of the power-law density function. The parameter q_0 is a constant equal to 120 km plus one Earth radius, whereas s is determined based of epoch perigee height above a spherical Earth. If perigee height is greater than or equal 156 km, the value of s is fixed to be 78 km plus one Earth radius. For altitudes greater than or equal to 98 km but less than 156 km, s is defined to be perigee height minus 78 km plus one Earth radius. For altitudes below 98 km, s is 20 km plus one Earth radius. In the following equations, the parameters q_0 and s should be in units of Earth radii:

$$\theta = \cos i_0, \qquad \xi = \frac{1}{a_0 - s}$$

$$\beta_0 = \left(1 - e_0^2\right)^{\frac{1}{2}}, \qquad \eta = a_0 e_0 \xi$$

$$C_2 = (q_0 - s)^4 \xi^4 n_0 (1 - \eta^2)^{-\frac{7}{2}} \left[a_0\left(1 + \frac{3}{2}\eta^2 + 4e_0\eta + e_0\eta^3\right) + \frac{3}{2}\frac{k_2\xi}{(1 - \eta^2)}\left(-\frac{1}{2} + \frac{3}{2}\theta^2\right)(8 + 24\eta^2 + 3\eta^4)\right]$$

$$C_1 = B^* C_2, \qquad C_3 = \frac{(q_0 - s)^4 \xi^5 A_{3,0} n_0 a_E \sin i_0}{k_2 e_0}$$

$$C_4 = 2n_0(q_0 - s)^4 \xi^4 a_0 \beta_0^2 (1 - \eta^2)^{-\frac{7}{2}} \left\{\left[2\eta(1 + e_0\eta) + \frac{1}{2}e_0 + \frac{1}{2}\eta^3\right] - \frac{2k_2\xi}{a_0(1 - \eta^2)}\left[3(1 - 3\theta^2)\left(1 + \frac{3}{2}\eta^2 - 2e_0\eta - \frac{1}{2}e_0\eta^3\right) + \frac{3}{4}(1 - \theta^2)\left(2\eta^2 - e_0\eta - e_0\eta^3\right)\cos 2\omega_0\right]\right\}$$

$$C_5 = 2(q_0 - s)^4 \xi^4 a_0 \beta_0^2 (1 - \eta^2)^{-\frac{7}{2}} \left[1 + \frac{11}{4}\eta(\eta + e_0) + e_0\eta^3\right]$$

$$D_2 = 4a_0 \xi C_1^2, \qquad D_3 = \frac{4}{3} a_0 \xi^2 (17a_0 + s) C_1^3$$

$$D_4 = \frac{2}{3} a_0^2 \xi^3 (221a_0 + 31s) C_1^4$$

where

$$A_{3,0} = -J_3 a_E^3, \qquad J_3 = -0.253881 \times 10^{-5}$$

2. *Initialization for Secular Effects of Earth Zonal Harmonics*

The secular effects of gravitation are included through the equations

$$\dot{M} = \left[\frac{3k_2(-1 + 3\theta^2)}{2a_0^2\beta_0^3} + \frac{3k_2^2(13 - 78\theta^2 + 137\theta^4)}{16a_0^4\beta_0^7}\right] n_0$$

$$\dot{\omega} = \left[-\frac{3k_2(1 - 5\theta^2)}{2a_0^2\beta_0^4} + \frac{3k_2^2(7 - 114\theta^2 + 395\theta^4)}{16a_0^4\beta_0^8} + \frac{5k_4(3 - 36\theta^2 + 49\theta^4)}{4a_0^4\beta_0^8}\right] n_0$$

$$\dot{\Omega} = \left[-\frac{3k_2\theta}{a_0^2\beta_0^4} + \frac{3k_2^2(4\theta - 19\theta^3)}{2a_0^4\beta_0^8} + \frac{5k_4\theta(3 - 7\theta^2)}{2a_0^4\beta_0^8}\right] n_0$$

where

$$k_4 = -\tfrac{3}{8} J_4 a_E^4, \qquad J_4 = -1.65597 \times 10^{-6}$$

3. Initialization for Secular and Long-Period Coefficients of Lunar and Solar Gravity

For satellites with periods greater than or equal to 225 min, additional terms are included to model the effect of lunar and solar gravitation on the satellite. Such satellites are referred to as deep space satellites. The equations for calculation of the orbital element secular rates and long-period coefficients due to the moon and sun gravitation are provided in Appendix A.A.

4. Initialization for Resonance Effects of Earth Gravity

For orbits with periods that result in repeating satellite position in relation to the Earth's figure, the effects of the nonzonal harmonics can be significant. This resonance condition is treated in the SGP4 model for orbits with 0.5-day (semisynchronous and highly eccentric) and 1-day (geosynchronous) periods. The equations for initialization of the resonance effects of Earth gravity are provided in Appendix A.B.

B. Update

Predictions of satellite motion are performed using the constants computed in the initialization.

1. Secular Update for Earth Zonal Gravity and Partial Atmospheric Drag Effects

The angles M, ω, and Ω are first updated to include the effects of the Earth zonal harmonics and atmospheric drag effects,

$$M_{DF} = M_0 + n_0(t - t_0) + \dot{M}(t - t_0), \qquad \omega_{DF} = \omega_0 + \dot{\omega}(t - t_0)$$

$$\Omega_{DF} = \Omega_0 + \dot{\Omega}(t - t_0), \qquad \delta\omega = B^* C_3(\cos\omega_0)(t - t_0)$$

$$\delta M = -\tfrac{2}{3}(q_0 - s)^4 B^* \xi^4 (a_E/e_0\eta)\big[(1 + \eta\cos M_{DF})^3 - (1 + \eta\cos M_0)^3\big]$$

$$M = M_{DF} + \delta\omega + \delta M, \qquad \omega = \omega_{DF} - \delta\omega - \delta M$$

$$\Omega = \Omega_{DF} - (21/2)\big(n_0 k_2 \theta \big/ a_0^2 \beta_0^2\big) C_1 (t - t_0)^2$$

where $(t - t_0)$ is time since epoch in minutes. Note that when epoch perigee height is less than 220 km or for deep space satellites, the terms $\delta\omega$ and δM are dropped.

2. Secular Updates for Effects of Lunar and Solar Gravity.

For satellites with periods greater than or equal to 225 min, the secular effects of lunar and solar gravity are included as detailed in Appendix A.C.

3. Secular Update for Resonance Effects of Earth Gravity

The resonance effects are applied to mean anomaly, mean motion, and semimajor axis using a numerical integration scheme as detailed in Appendix A.D.

4. Secular Update for Remaining Atmospheric Drag Effects

$$e = e_0 - B^* C_4 (t - t_0) - B^* C_5 (\sin M - \sin M_0)$$

$$a = (k_e/n)^{\frac{2}{3}}\big[1 - C_1(t - t_0) - D_2(t - t_0)^2 - D_3(t - t_0)^3 - D_4(t - t_0)^4\big]^2$$

$$IL = M + \omega + \Omega + n_0\big[\tfrac{3}{2} C_1 (t - t_0)^2 + \big(D_2 + 2C_1^2\big)(t - t_0)^3 + \tfrac{1}{4}\big(3D_3 + 12C_1D_2 + 10C_1^3\big)(t - t_0)^4 + \tfrac{1}{5}\big(3D_4 + 12C_1D_3 + 6D_2^2 + 30C_1^2D_2 + 15C_1^4\big)(t - t_0)^5\big]$$

$$\beta = \sqrt{1 - e^2}, \qquad n = k_e/a^{\frac{3}{2}}$$

where $(t - t_0)$ is time since epoch in minutes. Note that when epoch perigee height is less than 220 km or for deep space satellites, the equations for a and IL are truncated after the linear and quadratic terms, respectively, and the term involving C_5 is dropped.

5. Update for Long-Period Periodic Effects of Lunar and Solar Gravity

The long-period effects due to the third-body perturbation depend on the position of the sun or moon in its orbit. The mean anomaly of the perturbing body at the prediction time is

$$M_X = M_{O_X} + n_X \Delta t$$

where Δt is the time since the lunar/solar ephemeris epoch. The remaining equations for computation of the long-period periodic effects of lunar and solar gravity are provided in Appendix A.E.

The contributions of the sun and moon are combined for each term computed earlier and are applied as follows:

$$e = e + \delta e_{\mathrm{LS}}, \qquad i = i + \delta i_{\mathrm{LS}}$$

For $i > 0.2$ rad,

$$\Omega = \Omega + \delta\Omega_{\mathrm{LS}}/\sin i$$

$$\omega = \omega + (\delta\omega_{\mathrm{LS}} + \cos i\,\delta\Omega_{\mathrm{LS}}) - \delta\Omega_{\mathrm{LS}}\cos i/\sin i$$

$$M = M + \delta M_{\mathrm{LS}}$$

For $i \le 0.2$ rad,

$$\alpha = \sin i \sin\Omega + \sin i \cos\Omega\,\delta\Omega_{\mathrm{LS}} + \cos i \sin\Omega\,\delta i_{\mathrm{LS}}$$

$$\beta = \sin i \cos\Omega - \sin i \sin\Omega\,\delta\Omega_{\mathrm{LS}} + \cos i \cos\Omega\,\delta i_{\mathrm{LS}}$$

$$\Omega = \tan^{-1}(\alpha/\beta), \qquad M = M + \delta M_{\mathrm{LS}}$$

$$\omega = \omega + (\delta\omega_{\mathrm{LS}} + \cos i\,\delta\Omega_{\mathrm{LS}}) - \Omega\sin i\,\delta i_{\mathrm{LS}}$$

6. Update for Long-Period Periodic Effects of Earth Gravity

Add the long-period periodic terms,

$$a_{xN} = e\cos\omega, \qquad IL_L = \frac{A_{3,0}\sin i}{8k_2 a\beta^2}(e\cos\omega)\left(\frac{3 + 5\cos i}{1 + \cos i}\right)$$

$$a_{yNL} = \frac{A_{3,0}\sin i}{4k_2 a\beta^2}, \qquad IL_T = IL + IL_L$$

$$a_{yN} = e\sin\omega + a_{yNL}$$

7. Update for Short-Period Periodic Effects of Earth Gravity

Solve Kepler's equation for $(E + \omega)$ by defining

$$U = IL_T - \Omega$$

and using the iteration equation

$$(E + \omega)_{i+1} = (E + \omega)_i + \Delta(E + \omega)_i$$

with

$$\Delta(E + \omega)_i = \frac{U - a_{yN}\cos(E + \omega)_i + a_{xN}\sin(E + \omega)_i - (E + \omega)_i}{1 - a_{yN}\sin(E + \omega)_i - a_{xN}\cos(E + \omega)_i}$$

$$(E + \omega)_1 = U$$

The following equations are used to calculate preliminary quantities needed for short-period periodics:

$$e\cos E = a_{xN}\cos(E + \omega) + a_{yN}\sin(E + \omega)$$

$$e\sin E = a_{xN}\sin(E + \omega) - a_{yN}\cos(E + \omega)$$

$$e = \big(a_{xN}^2 + a_{yN}^2\big)^{\frac{1}{2}}, \qquad p_L = a(1 - e^2), \qquad r = a(1 - e\cos E)$$

$$\dot{r} = k_e \frac{\sqrt{a}}{r} e \sin E, \qquad r\dot{f} = k_e \frac{\sqrt{p_L}}{r}$$

$$\cos u = \frac{a}{r}\left[\cos(E+\omega) - a_{xN} + \frac{a_{yN}(e\sin E)}{1+\sqrt{1-e^2}}\right]$$

$$\sin u = \frac{a}{r}\left[\sin(E+\omega) - a_{yN} - \frac{a_{xN}(e\sin E)}{1+\sqrt{1-e^2}}\right]$$

$$u = \tan^{-1}\left(\frac{\sin u}{\cos u}\right), \qquad \Delta r = \frac{k_2}{2p_L}(1-\cos^2 i)\cos 2u$$

$$\Delta u = -\frac{k_2}{4p_L^2}(7\cos^2 i - 1)\sin 2u, \qquad \Delta\Omega = \frac{3k_2\cos i}{2p_L^2}\sin 2u$$

$$\Delta i = \frac{3k_2\cos i}{2p_L^2}\sin i\cos 2u, \qquad \Delta\dot{r} = -\frac{k_2 n}{p_L}(1-\cos^2 i)\sin 2u$$

$$\Delta r\dot{f} = \frac{k_2 n}{p_L}\left[(1-\cos^2 i)\cos 2u - \frac{3}{2}(1-3\cos^2 i)\right]$$

The short-period periodics are added to give the osculating quantities,

$$r_k = r\left[1 - \tfrac{3}{2}k_2\left(\sqrt{1-e^2}\Big/p_L^2\right)(3\cos^2 i - 1)\right] + \Delta r$$

$$u_k = u + \Delta u, \qquad \Omega_k = \Omega + \Delta\Omega, \qquad i_k = i + \Delta i$$

$$\dot{r}_k = \dot{r} + \Delta\dot{r}, \qquad r\dot{f}_k = r\dot{f} + \Delta r\dot{f}$$

Then unit orientation vectors are calculated by

$$\boldsymbol{U} = \boldsymbol{M}\sin u_k + \boldsymbol{N}\cos u_k, \qquad \boldsymbol{V} = \boldsymbol{M}\cos u_k - \boldsymbol{N}\sin u_k$$

where

$$\boldsymbol{M} = \left\{\begin{matrix} M_x = -\sin\Omega_k\cos i_k \\ M_y = \cos\Omega_k\cos i_k \\ M_z = \sin i_k \end{matrix}\right\}, \qquad \boldsymbol{N} = \left\{\begin{matrix} N_x = \cos\Omega_k \\ N_y = \sin\Omega_k \\ N_z = 0 \end{matrix}\right\}$$

Then position and velocity are given by

$$\boldsymbol{r} = r_k\boldsymbol{U}, \qquad \dot{\boldsymbol{r}} = \dot{r}_k\boldsymbol{U} + r\dot{f}_k\boldsymbol{V}$$

Appendix C: PPT3 Model

The Naval Space Command two-line element sets can be used for prediction with PPT3. All equations to follow are adapted from Schumacher and Glover.[23] The element set consists of

t_0 = epoch time
n_0 = mean motion, revolutions/day
e_0 = eccentricity
I_0 = inclination, deg
ω_0 = argument of perigee, deg
Ω_0 = right ascension of ascending node, deg
M_0 = mean anomaly, deg
decay1 = $\dot{n}/2$, revolutions/day^2
decay2 = $\ddot{n}/6$, revolutions/day^3

where all orbital elements except mean motion are the mean double-prime quantities defined by Brouwer[6] and where the subscript 0 will indicate the value of a quantity at epoch. The mean motion on the two-line element set follows the convention of Kozai,[7] although PPT3 uses its own convention for mean motion, which is slightly different from Kozai's, as explained in the main text and as will be shown explicitly. (This mathematical incompatibility has long been noted in space surveillance operations and is periodically rediscovered by newcomers. Because the two types of mean motion are numerically close together, the potential incompatibility is easily overcome by special processing at Naval Space Command before two-line elements are transmitted. However, the processing details are beyond the scope of this paper.) The two drag parameters, decay1 and decay2, are empirically determined during the orbit correction process.

The PPT3 orbit theory as implemented at Naval Space Command uses a specific value of the gravitational constant that defines the canonical units of the system:

k_e = $\sqrt{GM} = 0.0743669161$, (Earth radii)$^{1.5}$/min
G = universal gravitational constant
M = mass of the Earth

The PPT3 model is set in the FK4 and WGS72 reference standards, referred to the J2000.0 epoch.

We define the variables that will be used throughout the mathematical development. These definitions also provide the values used in PPT3 for the zonal coefficients. The notation closely follows that used by Brouwer[5,6]:

$$\gamma_2 = k_2/a''^2, \qquad \gamma_3 = A_{3.0}/a''^3, \qquad \gamma_4 = k_4/a''^4, \qquad \gamma_5 = A_{5.0}/a''^5$$

$$\gamma_2' = \gamma_2/\eta^4, \qquad \gamma_3' = \gamma_3/\eta^6, \qquad \gamma_4' = \gamma_4/\eta^8, \qquad \gamma_5' = \gamma_5/\eta^{10}$$

where

$$k_2 = \tfrac{1}{2}J_2R_\oplus^2 = 0.54130789\times10^{-3}$$

$$A_{3.0} = -J_3R_\oplus^3 = 0.25388100\times10^{-5}$$

$$k_4 = -\tfrac{3}{8}J_4R_\oplus^4 = 0.62098875\times10^{-6}$$

$$A_{5.0} = -J_5R_\oplus^5 = 0.21848266\times10^{-6}$$

$$\eta = \sqrt{1-e''^2}, \qquad \theta = \cos I''$$

A. Initialization

Many terms used in the prediction of PPT3 are independent of time. Thus, the algorithm begins with computation of numerous constant terms. The first step in the initialization is the recovery of the Brouwer semimajor axis from the Kozai-type PPT3 mean motion. Form the initial semimajor axis from the Kozai-type PPT3 mean motion

$$a_i = n_0^{\prime\prime-\frac{2}{3}}$$

The semimajor axis is transformed by iterating the following sequence five times. The second and fourth Brouwer gamma prime variables are formed from the current semimajor axis value. Then the semimajor axis is recomputed, using the zonal secular variation of the mean anomaly M defined directly as follows.

For $i = 1, 5$

$$\gamma_2' = k_2\Big/a_{i-1}^2\eta^4, \qquad \gamma_4' = k_4\Big/a_{i-1}^4\eta^8$$

$$a_i = \left[(1+\delta_s M)\Big/n_0''\right]^{\frac{2}{3}}$$

After the fifth iteration, the result is defined as the Brouwer semimajor axis a_0''. From this point on, the semimajor axis a_0'' follows the Brouwer convention. Also, all quantities on the right-hand side of equations are understood to be double-prime mean elements.

1. *Initialization for Secular Effects of Earth Zonal Harmonics*

The secular effects of gravitation are included through the equations

$$\delta_s M = (3/2)\gamma_2'\eta(-1+3\theta^2) + (3/32)\gamma_2'^2\eta[-15+16\eta+25\eta^2$$
$$+ (30-96\eta-90\eta^2)\theta^2 + (105+144\eta+25\eta^2)\theta^4]$$
$$+ (15/16)\gamma_4'\eta e''^2(3-30\theta^2+35\theta^4)$$