CRANFIELD UNIVERSITY

ÓSCAR RODRÍGUEZ FERNÁNDEZ

Improvement and Validation of a Space Debris Orbit Determination Tool

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
MSc in Astronautics and Space Engineering

Individual Research Project
Academic Year: 2015-2016

Company Supervisor: Dr Jens Utzmann
Academic Supervisor: Dr Stephen Hobbs
September 2016

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
MSc in Astronautics and Space Engineering

INDIVIDUAL RESEARCH PROJECT
Academic Year 2015-16

ÓSCAR RODRÍGUEZ FERNÁNDEZ

Improvement and Validation of a Space Debris Orbit Determination Tool

Company Supervisor: Dr Jens Utzmann
Academic Supervisor: Dr Stephen Hobbs
September 2016

This thesis is submitted in partial (45%) fulfilment of the requirements for the degree of Master of Science in Astronautics and Space Engineering.

Abstract

Since the early days of space exploration, with the launch of Sputnik in 1957, space has become more and more crowded with active satellites and space debris. The European Space Agency estimates that more than 670.000 objects larger than 1 cm orbit the Earth. These objects pose a threat to active satellites, as an impact at orbital velocities can have catastrophic consequences. To avoid collisions, their orbit must be known with sufficient accuracy.

Airbus Defence and Space has been developing SPOOK, an orbit determination tool able to simulate different observation strategies and perform orbit determination of a single space object. The target of this thesis is to enhance this tool. New radar-based sensor types were introduced to produce the measurements of the tracked objects. Also, the measurement generation process as well as the orbit determination algorithms were improved by taking into account the delay introduced by the finite velocity of light. Real world data can now be used to predict the orbit of space objects. Data from a experimental radar was tested with the tool and the state vector of the tracked object could be predicted. Multiple objects can now be tracked in a singular run of the program. Thanks to parallel processing, these objects can be simulated simultaneously, with the subsequent savings in computational times if respective computational resources are available. Lastly, a newly introduced postprocessing mode facilitates the analysis of the data produced by SPOOK, by creating relevant statistics about the simulation in addition of the raw data of the prediction.

All the recently implemented features enhance the capabilities of the tool to analyse different space debris surveillance scenarios and make possible the tracking of space objects with real data supplied by space and ground based observers.

Acknowledgements

I would like to thank all of those who gave me their support and made this thesis possible. Special mention to my supervisor in Airbus Defence and Space, Dr Jens Utzmann, for his guidance and help during the whole development of this project.

Processed data from the Poker Flat Incoherent Scatter Radar (PFISR) was provided by LeoLabs, Inc. PFISR is operated by SRI International on behalf of the US National Science Foundation under NSF Cooperative Agreement AGS-1133009.

Contents

List of Figures

List of Tables

Abbreviations

ECEF	Earth-Centered, Earth-Fixed
ECI	Earth Centered Inertial
EKF	Extended Kalman Filter
EOP	Earth Orientation Parameters
ESA	European Space Agency
GEO	Geostationary Orbit
ICRF	International Celestial Reference System
IOD	Initial Orbit Determination
JD	Julian Date
LEO	Low Earth Orbit
LTD	Light Time Delay
OD	Orbit Determination
OpenMP ARB	OpenMP Architecture Review Board
OTDF	Orbit Tracking Definition Format
PFISR	Poker Flat Incoherent Scatter Radar
RTN	Radial, Tangential and Normal directions
SBLS	Sequential Batched Least Squares
SPOOK	Space Object Observations and Kalman filtering
SPOP	Software Package for Orbit and Performance analysis
STK	Satellite Tool Kit
TLE	Two Line Element
WGS84	World Geodetic System 1984
WLS	Weighted Least Squares

Symbols

Upper Case Symbols

A	Partial Derivative Matrix
A_D	Drag cross-sectional area
A_{SRP}	Solar Radiation Pressure cross-sectional area
C_D	Drag coefficient
C_{SRP}	Reflectivity coefficient
F	Partial Derivative of the state rates matrix
$\vec{F}_g$	Gravity Force
G	Universal gravitational constant
H	Observation Matrix
K	Kalman Gain Matrix
P	Covariance Matrix
R	Measurement Noise Matrix
$R_\oplus$	Radius of the Earth
U	Potential
W	Weighting Matrix
$\vec{X}$	State Vector

Lower Case Symbols

a	Semi-major axis
$\vec{b}$	Residuals vector
e	Eccentricity
el	Elevation
f	Flattening parameter
h	Height
i	Inclination
m	Mass
$\vec{r}$	Position vector
t	Time
$\vec{v}$	Velocity vector
$\vec{y}$	Measurements vector

Greek Letters Symbols

α	Right Ascension
β	Azimuth
δ	Declination
λ	Longitude
$\hat{\lambda}$	Eigenvalue
μ	Gravitational parameter
ν	True Anomaly
ρ	Range Measurement
$\dot{\rho}$	Range-rate Measurement
σ	Standard deviation
Φ	State Error Transition Matrix
ϕ	Latitude
Ψ	Eigenvectors base change matrix
$\vec{\psi}$	Eigenvector
Ω	Right Ascension of the Ascending Node
ω	Argument of Perigee

Chapter 1

Introduction

The Inter-Agency Space Debris Coordination Committee (IADC) defines the term *space debris* as: ”Space debris are all man made objects including fragments and elements thereof, in Earth orbit or re-entering the atmosphere, that are non functional” [1]. Since the early days of space flights with the launch of Sputnik 1 in 1957 space has become more and more crowded with active satellites and space debris. The U.S Space Surveillance Network currently catalogues more than 17.000 objects [2]. However, the real number of objects can only be estimated via scientific models. By 2013, European Space Agency (ESA) estimated that 29.000 objects larger than 10 cm and 670.000 objects larger than 1 cm were orbiting the Earth. If we go down to objects larger than 1 mm, the number of objects is estimated to be greater than 170 million [3].

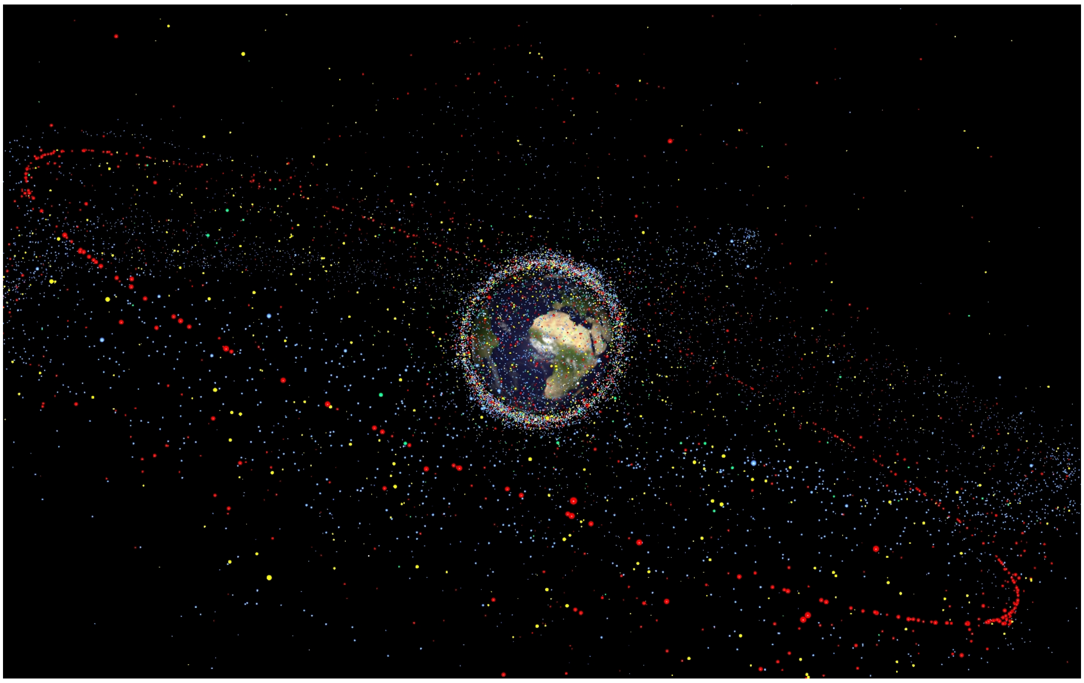


Figure 1.1: Distribution of space debris orbiting the Earth. Image by ESA [4]

Two main events led to a major increase in the number of space debris. In 2007, China destroyed its own satellite Fengyun 1C in an anti-satellite weapon test, leading to an increment of over 2000 objects [5]. The second one, in 2009 was the collision of two satellites: the US Iridium 33 and Cosmos 2251, a non-operational Russian satellite. Over 1500 new fragments were catalogued only from that collision [6].

All this objects pose a threat to the operational satellites that are currently orbiting the Earth. Due to the orbital velocities, even the impact of the smallest object can have catastrophic consequences. In order to be able to predict and avoid collisions, the orbits of objects in space must be known with sufficient accuracy.

Operational satellites can determine their own orbit, e.g using GPS data. However, passive debris such as rocket bodies do not have the means to determine their own orbit and they must be estimated through other means, using external measurements. Different approaches can be used, e.g. radar measurements, telescope observations or laser tracking from ground and from space. By gathering a number of measurements of a target along its orbit one can predict and refine the object's orbital elements over time. This is known as the orbit determination problem and will be the main topic of this thesis.

1.1 Thesis Objectives and Structure

Airbus Defence & Space has been developing Space Object Observations and Kalman filtering (SPOOK), an orbit determination tool (based in Fortran) in order to test different observation strategies for space debris. This tool can simulate measurements of different sensors (optical and radar sensors) and observers (space-based and ground-based observers) and it performs the orbit determination by using techniques such as Kalman filtering or Weighted Least Squares.

The objective of the project is to further develop and improve the tool that simulates observations of space debris orbiting Earth, performs orbit determination and estimates the error of the prediction. The work will be focused in:

- Software Validation.
- Simulation of new radar-based sensors.
- Enabling the tool to use real data coming from tracking stations.
- Implement light time delay compensation for measurement generation and orbit determination.
- Simultaneous tracking of multiple objects.
- Software parallelization.

In chapter 2 the theoretical background of the orbit determination problem is briefly described. Chapter 3 describes SPOOK characteristics and features at the beginning of the project while the implementation of the new observers is described in chapter 4. The Light Time Delay compensation feature is commented in chapter 5 and validation of the synthetic measurements simulated by SPOOK is made in chapter 6. Chapter 7 analyses the data coming from an experimental radar provided by LeoLabs, Inc. and its feasibility to be used for precise tracking of space debris. Finally, the issues regarding the simultaneous tracking of multiple objects can be found in chapter 8, where also a postprocessing feature is introduced. Chapter 9 proposes future enhancements for the code and chapter 10 summarizes the achievements accomplished within this thesis.

Chapter 2

Theoretical Background

This chapter introduces the theoretical background behind the orbit determination problem. It should not be considered as an extensive description of all the available methods currently available but only as a brief introduction to the astrodynamics fundamentals which SPOOK is based on.

2.1 Reference Systems

Within this thesis, two main groups of reference systems are used: Earth Centered Inertial (ECI) and Earth-Centered, Earth-Fixed (ECEF) reference systems serving different purposes. A brief description of these two basic systems is given below.

2.1.1 The Earth Centered Inertial Coordinate System

The ECI system is the most commonly used reference system for Earth-orbiting objects. Within SPOOK this is the basic reference system in which most of the operations will be made. This reference system is defined as an inertial frame with its center in the center of mass of the Earth, the $\hat{I}$ axis pointing towards the Vernal Equinox and the $\hat{K}$ axis points towards the Earth's North Pole. The $\hat{J}$ direction forms a right-handed system as shown in figure 2.1.

The realization of a geocentric inertial reference system is not a trivial problem [8] and different methodologies are used. Due to the Earth's precession and nutation movements, the orientation of the North Pole is not constant. To avoid having different realizations of the ECI systems, the axes are defined in a particular date and time. SPOOK uses the J2000 reference system [9], in which the date chosen is 1st January 2000 at 12:00. Within this document, the use of ECI system is equivalent to the J2000 reference system.

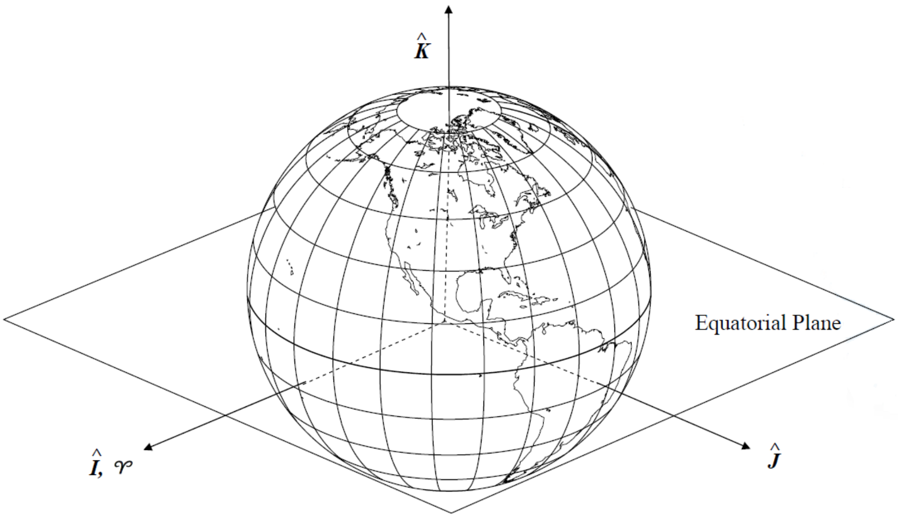


Figure 2.1: The Earth Centred Inertial coordinate system [7]

2.1.2 The Earth Centred Earth Fixed Coordinate System

The use of the ECI system is adequate for Earth-orbiting objects. However, for ground objects such as ground stations the use of a reference system that rotates with the Earth is more suitable. In such system, the coordinates of these objects would be constant over time. This definition corresponds to the ECEF reference system. As the ECI systems, the origin of the ECEF coordinates is the center of mass of the Earth. The $\hat{X}$ axis points towards the Greenwich meridian, the $\hat{Z}$ axis points normal to the Earth's equator in the North direction and $\hat{Y}$ completes the set.

The transformation between the two reference system is described in [7]. Within SPOOK, other intermediate coordinate systems are used. They are always based in a particular geometrical transformation of the ECI or ECEF systems. More information about these intermediate frames can be found in [10].

2.2 Object State

The ultimate objective of the orbit determination is to compute the state of an object at each instant of time. Different ways of representing objects exist. The next section will describe the object state representation methodologies used in SPOOK for both space-based objects (space debris or observers) and ground objects (ground-based observers).

2.2.1 Space-based Objects

SPOOK implements two object state representations for space-based objects:

- Object coordinates.
- Classical orbital elements.

The state of an object in a precise instant of time can be expressed using Cartesian coordinates. For a correct representation a total of three coordinates are needed, three for position $\vec{r}$ and three for velocity $\vec{v}$. The reference frame chosen in SPOOK for the space object representation is the ECI system introduced in section 2.1.1. In 2.1 it can be seen the state vector of the Hubble Space Telescope on 7 August 2016 at midnight [11].

$$\vec{X} = \left\{ \begin{array}{c} \vec{r} \\ \vec{v} \end{array} \right\} = \left\{ \begin{array}{c} r_I \\ r_J \\ r_k \\ v_I \\ v_J \\ v_K \end{array} \right\} = \left\{ \begin{array}{c} -5170.4549591 \text{ Km} \\ 10613.3359010 \text{ Km} \\ -4043.2971919 \text{ Km} \\ -7.0214339 \text{ Km} \cdot \text{s}^{-1} \\ -2.0365443 \text{ Km} \cdot \text{s}^{-1} \\ -3.0060415 \text{ Km} \cdot \text{s}^{-1} \end{array} \right\} \tag{2.1}$$

Space-based objects can also be represented using the 6 classical Keplerian orbital elements [12]. The definition of these elements can be seen in table 2.1 and figure 2.2.

Symbol	Name	Description
a	Semi-major axis	Half the long axis of the orbit ellipse
e	Eccentricity	Describes the shape of the orbit (0: circular,...)
i	Inclination	Angle between the equator and the orbital plane measured at the ascending node
Ω	Right Ascension of the Ascending Node	Angle between the first point of Aries and the ascending node measured in an Easterly direction
ω	Argument of periapsis	Angle between the ascending node and the periapsis position measured in the direction of the object motion
ν	True Anomaly	Angle between the periapsis position and the object's current position

Table 2.1: Classical Keplerian orbital elements

For the same time instance as above, in 2.2 we can see the state representation of the Hubble Telescope using the classical Keplerian orbital elements [11].

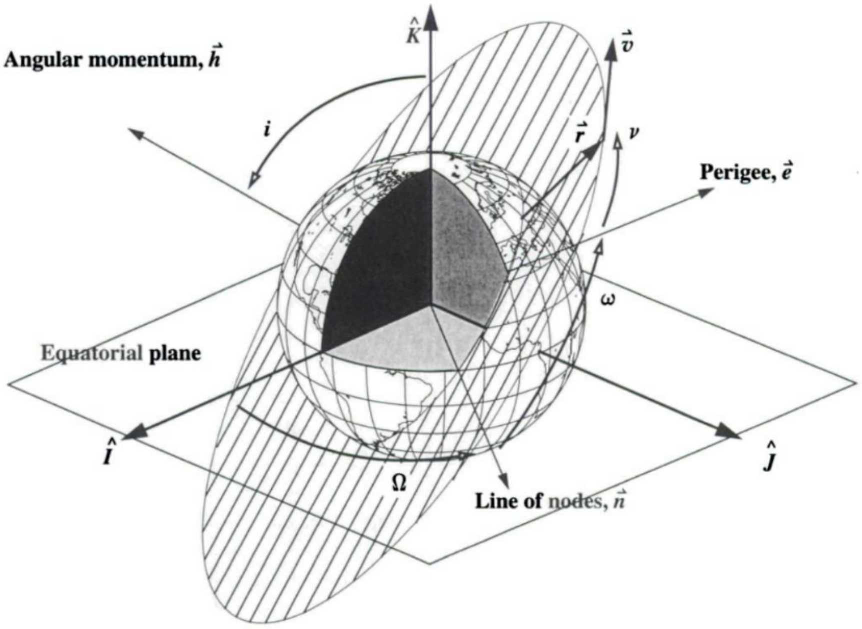


Figure 2.2: Keplerian classical orbital elements [7]

$$\vec{X} = \begin{Bmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \nu \end{Bmatrix} = \begin{Bmatrix} 6925.3597295 \text{ Km} \\ 0.0012866 \\ 28.3967^\circ \\ 282.9733^\circ \\ 134.9545^\circ \\ 36.2234^\circ \end{Bmatrix} \tag{2.2}$$

2.2.2 Ground-based Observers

For ground-based observers the geodetic coordinates longitude λ, latitude ϕ and altitude h above the reference ellipsoid are used. The reference ellipsoid used in SPOOK is the one defined in the WGS84 datum [13].

2.3 Observation Theory

The orbit determination algorithms are based in the use of measurements coming from observers. This observations can be made from Earth (ground-based observers) or directly from other satellites orbiting the Earth (space-based observers). Different observation methods exists for space objects. The two most common ones are radar-based and passive optical observations. The following sections describe the operational principles of these two kind of observers.

2.3.1 Optical Observers

Telescopes have been used for astronomy since the times of Galileo. They are passive observers, basing its functionality in detecting the light reflected by the observed object (e.g. space debris). With the knowledge of the pointing information of the telescope, the direction of the reflecting object can be translated into two angular measurements: topocentric right ascension α and declination δ as defined in figure 2.3

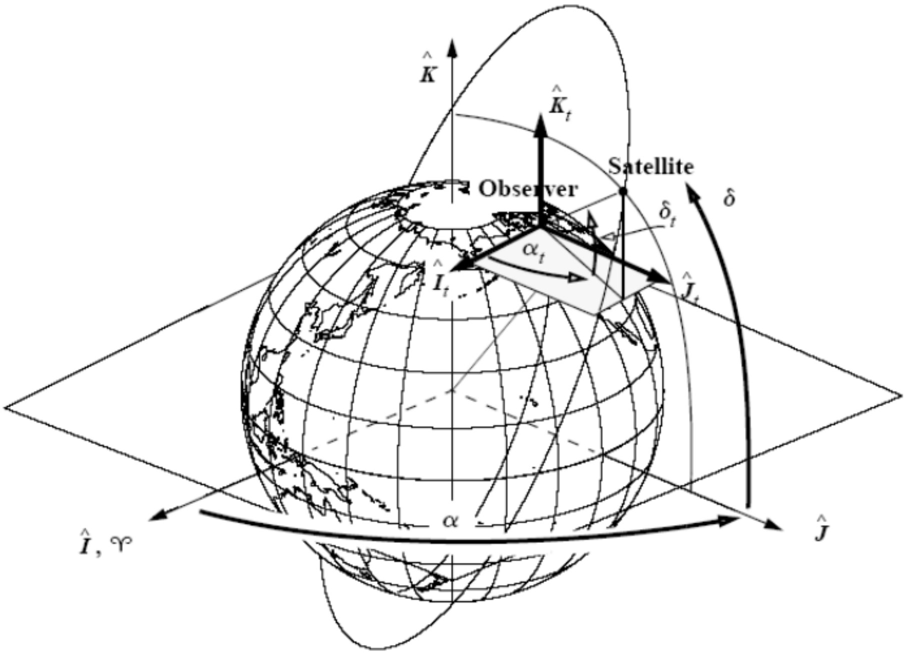


Figure 2.3: Angular measurements of optical observers: topocentric right ascension α and declination δ [7]

2.3.2 Radar Observers

Opposed to passive optical observers, we have radar observers. These are based on actively sent radio waves which are reflected by the object and then detected by the observer sensor. For this reason, these signals are called two-way signals. As the optical observers, information about the object direction is expressed by two angles, in this case the azimuth β and elevation el as defined in figure 2.4. The advantage of the radar observers is that, besides the angular measurements they can provide distance measurements or slant range ρ by measuring the time that the wave needs to travel from the observer and come back once reflected in the object [14]. Also the Doppler effect can be used to measure the change of distance over time or slant range-rate $\dot{\rho}$ [15].

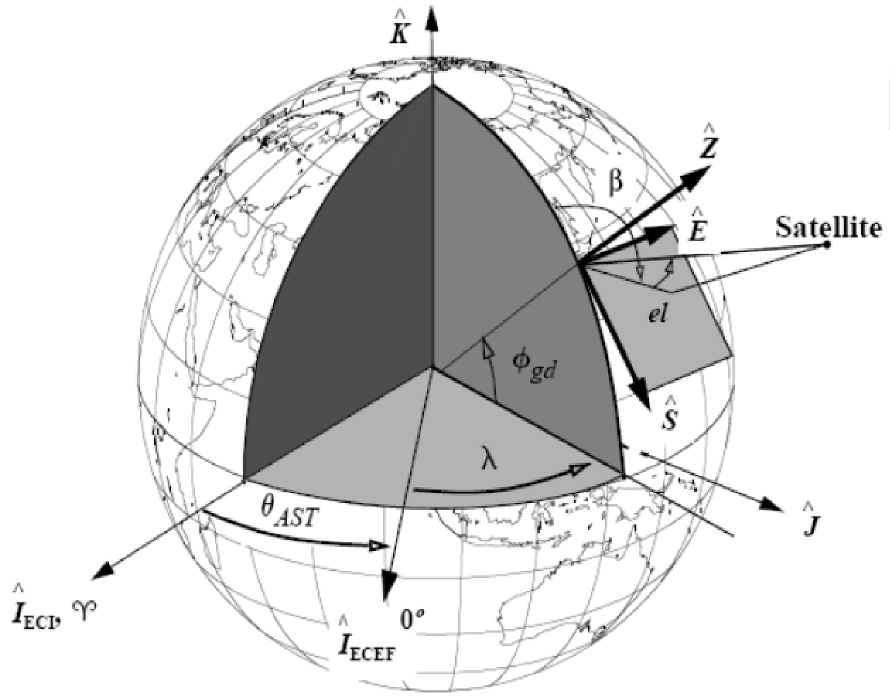


Figure 2.4: Angular measurements of radar observers: azimuth β and elevation *el* [7]

2.4 Dynamic Model

A correct modelling of the dynamics of the tracked object is one of the key points for a successful orbit determination. All relevant forces acting upon the target object must be carefully taking into account. This section describes the forces that are implemented in SPOOK.

2.4.1 The Two Body Equation

According to Newton's law of gravitation, the force acting on an object orbiting the Earth is:

$$\vec{F}_g = -\frac{G m_{\oplus} m_{obj}}{r^2}\left(\frac{\vec{r}}{r}\right) \tag{2.3}$$

This will be the main force acting on the object and defines the Keplerian orbit. The rest of the forces can be considered as orbital perturbations.

2.4.2 Non-spherical Gravitational Field

The two body equation assumes the Earth's mass to be concentrated into a single point, yielding a spherical gravitational field. This assumption is reasonable for a first order approximation. However, for more precise propagations of the object dynamics the real mass distribution of the Earth must be taken into account. In order to consider this force, we introduce a non-spherical Earth's gravity field U. The derivation of this potential can be found in [7].

$$\ddot{\vec{r}} = -\nabla U \tag{2.4}$$

$$U = \frac{\mu}{r}\left[1 + \sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\frac{R_{\oplus}}{r}\right)^{l} P_{l,m}\sin(\phi_{obj})\left[C_{l,m}\cos(m\lambda_{obj}) + S_{l,m}\sin(m\lambda_{obj})\right]\right] \tag{2.5}$$

where $P_{l,m}$ are Legendre functions and $C_{l,m}$ and $S_{l,m}$ are the geopotential model coefficients [16].

2.4.3 Atmospheric Drag with Thermospheric Winds

For objects in Low Earth Orbit (LEO) orbits, atmospheric drag is the strongest perturbation. The computation of this force is one of the major challenges in precise orbit modelling, as it depends of numerous factors as the object orientation or the drag coefficient, which are difficult to determine. Numerous models to accurately represent the atmosphere have been developed. SPOOK uses the MSIS-00 model developed by the US Naval Research Laboratory. More information about this implementation can be found in [17].

According to [7] the acceleration $\vec{a}_{Drag}$ that an object experiments due to the atmospheric drag is:

$$\vec{a}_{Drag} = -\frac{1}{2}C_D\frac{A_D}{m_{obj}}\rho|\vec{v}_{rel}|\vec{v}_{rel} \tag{2.6}$$

where C_D is the drag coefficient, A_D is the object cross-sectional area and m the mass o the object. Here ρ stands for the atmospheric density at the considered altitude. $\vec{v}_{rel}$ is the object velocity relative to the atmosphere. To compute this relative velocity, the rotation of the Earth $\vec{\omega}_{\oplus}$ must be taken into account as well as the so-called thermospheric winds.

$$\vec{v}_{rel} = \vec{v} - \vec{\omega}_{\oplus} \wedge \vec{r} - \vec{v}_{t,winds} \tag{2.7}$$

Thermospheric winds are the winds in the upper part of the atmosphere. For its computation the Horizontal Wind Model developed by NASA is used [18]. Details of its implementation in SPOOK can be found in [10].

2.4.4 Solar Radiation Pressure

When an object is illuminated by the Sun, the impact of the light photons on its surface causes an additional acceleration [19]. This is the so-called Solar Radiation Pressure. It also depends on object specific parameters such as the reflectivity coefficient. Opposite to the atmospheric drag, this perturbation has greater effects in higher orbits. According to [7], this acceleration $\vec{a}_{SRP}$ can be modelled as:

$$\vec{a}_{SRP} = -p_{SRP} C_R \frac{A_R}{m_{obj}} \frac{\vec{r}_\odot}{r_\odot} \tag{2.8}$$

were p_{SRP} is the actual solar radiation pressure, whose value is 4.57·10^{-}6 N·m^{-2}, $r_\odot$ is the relative position of the Sun to the object, C_R the object's reflectivity coefficient and A_r the object's cross sectional area.

2.4.5 Third-body Perturbations

Just as the Earth's gravity field, the gravity acceleration induced by other bodies in the Solar System also affects the dynamics of the orbiting objets. The two main contributions for Earth-orbiting objects are the Sun and the Moon. In SPOOK, besides the perturbations of the Sun and Moon, the rest of the planets of the Solar system are implemented. The value of the induced acceleration from another body is

$$\vec{a}_{body} = -\mu_{body} \left(\frac{\vec{S} - \vec{r}}{|\vec{S} - \vec{r}|^3} - \frac{\vec{S}}{S^3} \right) \tag{2.9}$$

where μ_{body} is the gravitational constant and $\vec{S}$ is the relative position between the object and the body.

2.5 Initial Orbit Determination

The orbit determination methods that will be introduced in section 2.6 require an inital estimation of the object position in order to start. This initial position can be supplied by the user if some knowledge of the object's orbit. In the most general case, no previous information about the orbit is known. For that reason, it is necessary to employ a method to perform a coarse orbit determination using only a few measurements.

In 1801, Carl Friedrich Gauss predicted the orbit of the Dwarf Planet Ceres using only measurements of a fraction of the orbit. The method he used is called the Gauss algorithm. This algorithm has been implemented in SPOOK [17]. The derivation of this method can be found in [7].

2.5.1 Gibbs Algorithm

In 1889 Josiah Gibbs proposed an improvement of the Gauss method. It is a geometrical method that uses three position vectors to derive the velocity in the middle time incident. It's derivation can be found again in [7] and it is implemented in SPOOK [17].

2.5.2 Herrick-Gibbs Algorithm

The Gibbs algorithm fails when the three position vectors are separated by small angles. The Herrick-Gibbs algorithm uses Taylor-expansion series to improve the results for closely-spaced measurements [7]. It has also been implemented in SPOOK [17].

2.6 Orbit Determination Methods

In SPOOK three different orbit determination methods had been implemented [10, 17, 20]. One of the main difference between them is how they use data. The Weighted Least Squares (WLS) uses all the available data at the same time. For this reason, this is the most stable method. The Sequential Batched Least Squares (SBLS) is similar to the WLS but it only uses the data of one tracklet at each time. A tracklet is defined as all the measurements since a pass of the object is first detected by the observer until the last measurement of that pass is done. Finally, the Extended Kalman Filter (EKF) method is the only one that can track the object in real time. However, fine tuning of the filter is needed and an initial covariance matrix must be supplied besides the initial object state needed for the other two methods. These algorithms will be briefly described here.

2.6.1 The Weighted Least Squares Method

As stated before, this algorithm uses all the available measurements to perform the orbit determination. A more in-deep derivation of this method can be found in [7].

The initial state vector of the object $\vec{X}_0$ is propagated to all the time incidents where measurements occur.

$$\vec{X}_j = f(\vec{X}_0, t_j) \tag{2.10}$$

For these predicted state vectors the predicted measurements $\vec{y}_{p,j}$ are evaluated

$$\vec{y}_{p,j} = g(\vec{X}_j) \tag{2.11}$$

This measurements are compared with the real measurements $\vec{y}_{r,j}$ coming from the observers to compute the residuals $\vec{b}_j$.

$$\vec{b}_j = \vec{y}_{r,j} - \vec{y}_{p,j} \tag{2.12}$$

The WLS algorithm computes the correction of the initial supplied state vector $\delta\vec{X}_0$ that minimizes the sum of the square of the residuals.

$$\min\left(\sum \vec{b}^2\right) \tag{2.13}$$

The algorithm also yields a covariance matrix of the predicted state at the initial point. The covariance matrix P_0 at the initial state is computed as:

$$P_0 = \left(\sum^n A_j^T W A_j\right)^{-1} \tag{2.14}$$

where W is a weighting matrix that assures that all the measurements are taking into account with their respective standard deviation and n is the total number of measurements.

$$W = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{\sigma_m} \end{bmatrix} \tag{2.15}$$

where m is the number of measurements produced by the observer in an instant of time (e.g. $m = 2$ for optical observers).

A_j is the partial derivative matrix of the measurements respect to the initial state. It relates how the changes in the initial state affect the observations. It can be split into two matrices: the observation matrix H_j, that relates how the variation of the state affects the observations and the error state transition matrix Φ_j, that translates the initial state to the considered time incident.

$$A_j = \frac{\partial \vec{y}_j}{\partial \vec{X}_0} = \frac{\partial \vec{y}_j}{\partial \vec{X}_j} \frac{\partial \vec{X}_j}{\partial \vec{X}_0} = H_j \Phi_j \tag{2.16}$$

The error state transition matrix is computed solving the system of differential equations defined in

$$\dot{\Phi}_j = F_j \Phi \tag{2.17}$$

where F_j is a matrix formed by the partial derivatives of the state rates with respect to the state.

$$F = \frac{d\dot{\vec{X}}}{d\vec{X}} = \begin{pmatrix} \frac{d\vec{v}}{d\vec{r}} & \frac{d\vec{v}}{d\vec{v}} \\ \\ \frac{d\vec{a}}{d\vec{r}} & \frac{d\vec{a}}{d\vec{v}} \end{pmatrix} \tag{2.18}$$

In SPOOK, equation 2.17 is solved using the Runge-Kutta-Fehlberg [21] or Shampine-Gordon [22] numerical integrators.

Finally, the correction that minimizes the sum of the square root of the residuals is:

$$\delta\vec{X}_0 = \left(\sum^{n} \left(A_j^T W A_j\right)\right)^{-1} \sum^{n} \left(A_j^T W b_j\right) \tag{2.19}$$

Once the algorithm has finished a new improved initial state is obtained.

$$\vec{X}_{0_{new}} = \vec{X}_0 + \delta\vec{X}_0 \tag{2.20}$$

This new improved initial state is supplied again to the algorithm to start the whole process again. This iterative process finishes once the truncation error τ has fallen below a predetermined value.

$$\tau = \sqrt{\frac{\sum^{n} \left(\vec{b}_j^T W \vec{b}_j\right)}{nm}} \tag{2.21}$$

2.6.2 The Sequential Batched Least Squares Algorithm

In [19] a modification of the WLS algorithm is proposed. Instead of using all the information available, it runs each tracklet individually. However, the information of each tracklet relates to the same initial state vector. In order to not lose the information of the measurements of the previously processed tracklets, the SBLS uses two new matrices as inputs $\left(\sum^{n}\left(A_j^T W A_j\right)\right)_{old}$ and $\sum^{n}\left(A_j^T W b_j\right)_{old}$, that come from the previous runs of the SBLS.

For the run of the tracklet k, equations 2.14 and 2.19 are modified as follows:

$$P_0 = \left(\sum^{n_k} A_j^T W A_j + \left(\sum^{n_{k-1}} \left(A_j^T W A_j\right)\right)_{old}\right)^{-1} \tag{2.22}$$

$$\delta \vec{X}_0 = \left(\sum^{n_k} \left(A_j^T W A_j\right) + \left(\sum^{n_{k-1}} \left(A_j^T W A_j\right)\right)_{old}\right)^{-1} \left(\sum^{n_k} \left(A_j^T W b_j\right) + \left(\sum^{n_{k-1}} \left(A_j^T W b_j\right)\right)_{old}\right) \tag{2.23}$$

2.6.3 The Extended Kalman Filter

The EKF algorithm optimizes the state vector and the covariance matrix estimation in real time. The EKF algorithm used in SPOOK is adapted from [7]. The main drawback of this method is that it needs a initial covariance matrix P_0 as an input besides the initial state vector X_0.

The EKF is divided in two steps: The prediction and the update step. In the prediction step, the algorithm propagates both the state vector $\vec{X}_{j-1}$ and the covariance $\vec{P}_{j-1}$ to the next measurement time incident, predicting $\bar{X}_j$ and $\bar{P}_j$. In this new time incident, the Kalman Gain matrix K_j is computed

$$K_j = \bar{P}_j H_j^T \left(H_j \bar{P}_j H_j^T + R\right)^{-1} \tag{2.24}$$

Where H_j is now once again the observation matrix. R is the measurement noise matrix, that accounts for all the uncertainties in the measurements.

$$R = \begin{bmatrix} \sigma_i^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_m^2 \end{bmatrix} \tag{2.25}$$

In the second step, the information of the measurements is used to produce a best estimation of the object's vector state $\vec{X}_j$ and covariance matrix P_j.

$$\vec{X}_j = \bar{X}_j + \delta X_j \tag{2.26}$$

$$\delta X_j = K_j \vec{b}_j \tag{2.27}$$

$$P_j = \bar{P}_j - K_j H_j \bar{P}_j \tag{2.28}$$

Opposite to the WLS and SBLS no iterations are needed and the process continues taking into account the next measurement.

2.7 Orbital Regions

In this section, a classification of different orbit regimens will be defined. Earth-orbiting objects can be classified into different categories regarding their orbital parameters. The definition of these categories is based on the ESA Space Situational Awareness program [23] with some modifications.

Name	Perigee [Km]			Apogee [Km]		
	Min		Max	Min		Max
LEO resident	0	-	2000	0	-	2000
LEO transient	0	-	2000	2000	-	∞
Low MEO resident	2000	-	16000	2000	-	16000
Low MEO transient	2000	-	16000	16000	-	∞
High MEO resident (GNSS)	16000	-	33786	16000	-	33786
High MEO resident	16000	-	33786	33786	-	∞
GEO resident($i \leqslant 20°$)	33786	-	37786	33786	-	37786
GEO resident($i > 20°$)	33786	-	37786	33786	-	37786
GEO transient	33786	-	37786	37786	-	∞
HEO	37786	-	∞	37786	-	∞

Table 2.2: Orbital regions definition

Chapter 3

SPOOK

SPOOK is a tool developed in Fortran able to simulate Earth-orbiting space objects and observers, generate synthetic measurements and perform orbit determination based in the theoretical background explained in chapter 2. In this chapter, the main capabilities implemented in SPOOK before this thesis began will be briefly described to give the reader and idea of how it works. Extra information about the different features available can be found in [10, 17, 20].

3.1 SPOOK structure

At the beginning of this project, SPOOK was already in its 3.0 version. The main structure of the code can be found in the flowchart represented in figure 3.1. A SPOOK run starts by reading the three main configuration files, where all the characteristics of the particular simulation being run will be defined. A general description of this files will be made in the next section. After that, the three main operational modes of SPOOK will be commented. These modes are:

- Covariance Propagation.
- Measurement Generation.
- Orbit Determination.

3.1.1 SPOOK's Configuration Files

SPOOK is based in the use of three main configuration files: a general parameter configuration file, the observers specification file and the target objects specification files.

The main configuration file of SPOOK is named `parameters.ini`. In that file are stated most of the parameters regarding the simulation being run by SPOOK such as the total simulation time, the orbit determination algorithm to use or the names of the files defining the observers and objects characteristics.

The objects to be targeted are defined in the objects configuration file. When objects are simulated, their orbit characteristics can be directly defined in this file or be taken from a Two Line Element (TLE) formatted file, taking advantage of the TLE read-in feature implemented in SPOOK [20]. At the start of the project, only one object could be simulated by SPOOK.

Lastly we have the observers specification file, usually called `observers.dat`, where the characteristics of the sensors tracking the space objects are defined. Parameters such as the kind and number of observers, the location (space-based or ground-based) or the sensors accuracies are defined in this file. This file will define the simulated observation strategy.

3.1.2 Covariance Propagation Mode

In this operational mode, no observation strategy is simulated or real measurements are used. For that reason, no observers need to be defined. This operational mode uses the initial state and the initial covariance defined for the object to propagate them over the defined simulation time. After the propagation is done and the results are output to a file, the program ends.

3.1.3 Measurement Generation Mode

In this mode, the object and the observer (in the case of space-based observers) are propagated over the simulation time to generate synthetic measurements. Details about how the measurement generation process works can be found in [10]. After the measurements are generated and written to an output file, no orbit determination is attempted. This mode is useful to assess the performance of a particular observation strategy in order to track a predefined object. However, version 3.0 of SPOOK only outputs the raw measurement data and some postprocessing would be needed to analyse the results. The measurements are output following the Orbit Tracking Definition Format (OTDF) developed by ESA.

3.1.4 Orbit Determination Mode

This is the main mode of SPOOK, where the different orbit determination algorithms are used. In this case, the measurements can be simulated following the same scheme as in the Measurement Generation Mode or read from external files. At the beginning of this thesis, the feature to read-in measurements in OTDF format had been started and was completed during the development of the project. After the measurements are read, an Initial Orbit Determination can be made with the algorithms described in section 2.5 to evaluate an initial state vector. Otherwise, the initial state supplied in the object configuration file can be used to initialize the orbit determination algorithms. After an initial state is evaluated the orbit determination process begin using one of the three algorithms described in section 2.6.

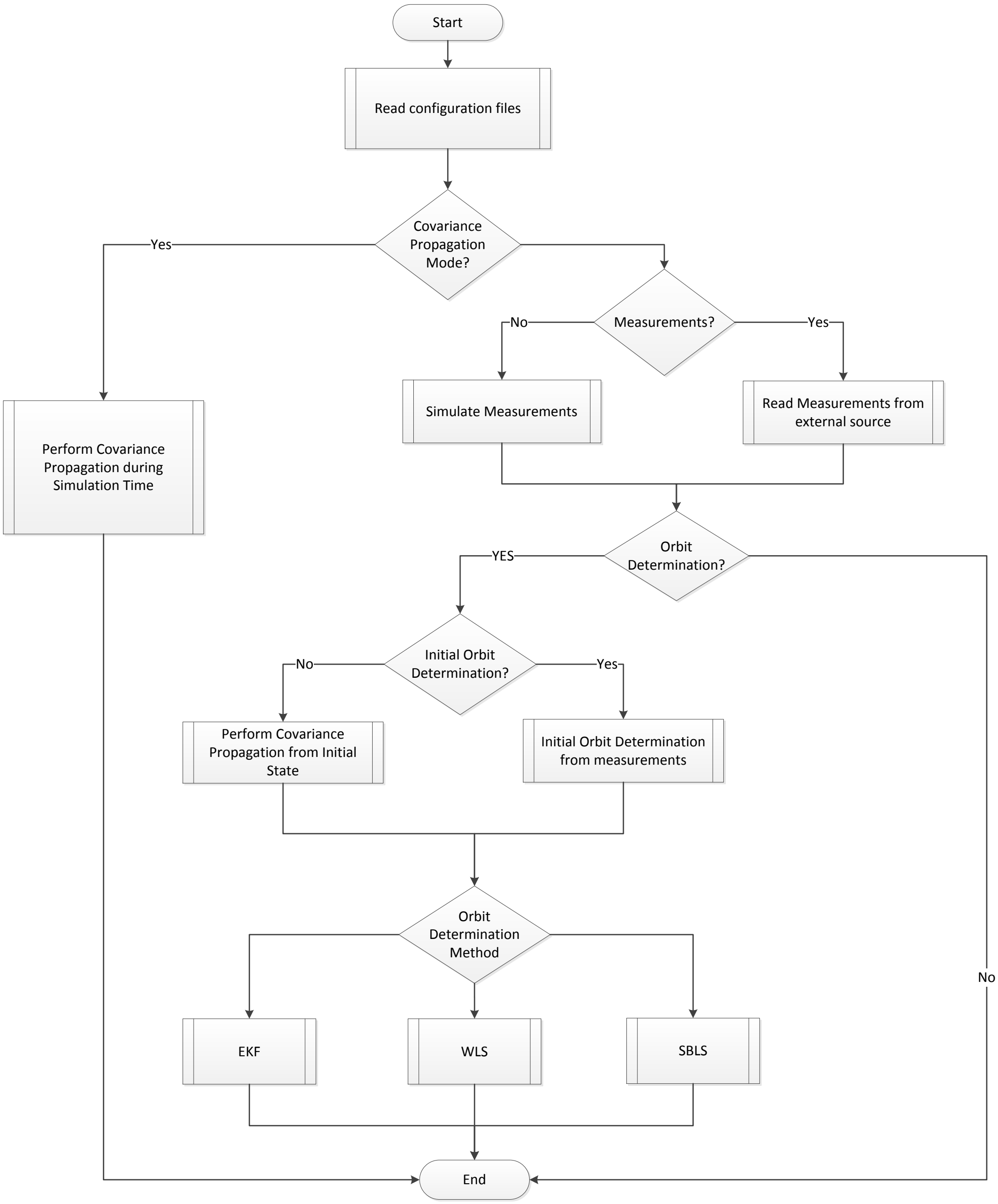

Start
Read configuration files
Covariance Propagation Mode?
Yes
Perform Covariance Propagation during Simulation Time
Measurements?
No
Yes
Simulate Measurements
Read Measurements from external source
Orbit Determination?
YES
No
Initial Orbit Determination?
No
Yes
Perform Covariance Propagation from Initial State
Initial Orbit Determination from measurements
Orbit Determination Method
EKF
WLS
SBLS
End

Figure 3.1: Main structure of SPOOK v3.0

Chapter 4

New Radar Sensor Types

One of the main goals of this project is to enable the use of new kind of observer types in SPOOK. The implementation of this feature enhances the capabilities of the tool to simulate different kind of scenarios and to use observations coming from real data providers.

The new kind of sensors implemented are based in measurements coming from radar-based observers. At the beginning of the thesis, the observers available were only optical observers (providing right ascension and declination measurements) and radar observers (providing azimuth, elevation and slant range measurements). The new sensor types implement the different combination of the radars angular data (azimuth and elevation), slant range and slant range-rate measurements.

Changes in how the measurements are generated were made in SPOOK. Regarding the orbit determination process, the main changes were implemented in how the filters use these new kind of measurements. The aim of this chapter is to derive the relationship between the different kind of available measurements coming form a radar based sensor and the tracked object state vector. By obtaining this relationship, we can evaluate the observation matrix H defined as:

$$H = \frac{\partial Obs}{\partial \vec{X}} \tag{4.1}$$

where X is the state vector of the tracked object expressed in ECI frame coordinates.

$$\vec{X} = \left\{ \begin{array}{c} \vec{r}_{ECI} \\ \vec{v}_{ECI} \end{array} \right\} = \left\{ \begin{array}{c} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{array} \right\}_{ECI} \tag{4.2}$$

The next sections describe how to compute the observation matrix for slant range, slant range-rate and angular measurements. Once these matrices are derived, they can be combined to form the new types of sensors. A validation of these new observer types has also been included via a benchmark example.

4.1 Range Only Measurements

The range is defined as the distance between the observer and the object.

$$\vec{\rho} = \vec{r}_{ECI} - \vec{r}_{ECI_{site}} \tag{4.3}$$

The measurement coming from the sensor is a scalar. Dotting 4.3 by itself we have:

$$\rho^2 = \left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right) \cdot \left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right) \tag{4.4}$$

We can know evaluate the observation matrix for the case where only range data is available

$$H = \frac{\partial Obs}{\partial \vec{X}} = \frac{\partial \rho}{\partial X} = \left[\frac{\partial \rho}{\partial \vec{r}_{ECI}}, \frac{\partial \rho}{\partial \vec{v}_{ECI}}\right] \tag{4.5}$$

It is seen in 4.4 that this measurement is not dependant from the velocity.

$$\frac{\partial \rho}{\partial \vec{v}_{ECI}} = 0 \tag{4.6}$$

To obtain the rest of the matrix, we only have to evaluate the derivatives respect to the position coordinates. Using 4.4 and the chain rule

$$\frac{\partial \rho}{\partial \vec{r}_{ECI}} = \frac{\partial \rho}{\partial \rho^2}\frac{\partial \rho^2}{\partial \vec{r}_{ECI}} \tag{4.7}$$

$$\frac{\partial \rho}{\partial \rho^2} = \frac{1}{\frac{\partial \rho^2}{\partial \rho}} = \frac{1}{2\rho} \tag{4.8}$$

$$\frac{\partial \rho^2}{\partial \vec{r}_{ECI}} = 2\left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right) \tag{4.9}$$

The final observation matrix would be:

$$H = \frac{1}{\rho}\left[\begin{array}{cccccc} x - x_{site} & y - y_{site} & z - z_{site} & 0 & 0 & 0 \end{array}\right] \tag{4.10}$$
$$H = \left[\begin{array}{cccccc} L_x & L_y & L_z & 0 & 0 & 0 \end{array}\right] \tag{4.11}$$
$$\tag{4.12}$$

where $\vec{L}$ is the unit vector in the direction of the range vector.

$$L_x \equiv \frac{x - x_{site}}{\rho} \tag{4.13}$$
$$L_y \equiv \frac{y - y_{site}}{\rho} \tag{4.14}$$
$$L_z \equiv \frac{z - z_{site}}{\rho} \tag{4.15}$$
$$\rho = \sqrt{\left(x - x_{site}\right)^2 + \left(y - y_{site}\right)^2 + \left(z - z_{site}\right)^2} \tag{4.16}$$

4.2 Range-Rate Measurements

The range-rate is the time derivative of the range vector

$$\dot{\vec{\rho}} = \frac{d\rho}{dt} \tag{4.17}$$

Differentiating equation 4.4 with respect to time yields

$$\rho\dot{\rho} = \left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right)\left(\vec{v}_{ECI} - \vec{v}_{ECI_{site}}\right) \tag{4.18}$$
$$\dot{\rho} = \frac{1}{\rho}\left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right)\left(\vec{v}_{ECI} - \vec{v}_{ECI_{site}}\right) \tag{4.19}$$

Once again, we treat separately the derivatives respect with velocity and position. Differentiating 4.19 against velocity

$$\begin{aligned} \frac{\partial\dot{\rho}}{\partial\vec{v}_{ECI}} = & \frac{\partial\frac{1}{\rho}}{\partial\vec{v}_{ECI}}\left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right)\left(\vec{v}_{ECI} - \vec{v}_{ECI_{site}}\right) \\ & + \frac{1}{\rho}\frac{\partial\left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right)}{\partial\vec{v}_{ECI}}\left(\vec{v}_{ECI} - \vec{v}_{ECI_{site}}\right) \\ & + \frac{1}{\rho}\left(\vec{r}_{ECI} - \vec{r}_{ECI_{site}}\right)\frac{\partial\left(\vec{v}_{ECI} - \vec{v}_{ECI_{site}}\right)}{\partial\vec{v}_{ECI}} \end{aligned} \tag{4.20}$$

The first two derivatives on the right size of the equation are 0 while the last derivative is the identity matrix.

$$\frac{\partial\left(\vec{v}_{ECI}-\vec{v}_{ECI_{site}}\right)}{\partial\vec{v}_{ECI}}=[I]_{3x3} \tag{4.21}$$

The derivative respect to the tracked object velocity is

$$\frac{\partial\dot{\rho}}{\partial\vec{v}_{ECI}}=\frac{1}{\rho}\left(\vec{r}_{ECI}-\vec{r}_{ECI_{site}}\right) \tag{4.22}$$

$$\frac{\partial\dot{\rho}}{\partial\vec{v}_{ECI}}=\vec{L}=[L_x,L_y,L_z] \tag{4.23}$$

In the derivative with respect the position, only the derivative in the last term on the right of 4.24 is zero.

$$\begin{aligned}\frac{\partial\dot{\rho}}{\partial\vec{r}_{ECI}}=&\frac{\partial\frac{1}{\rho}}{\partial\vec{r}_{ECI}}\left(\vec{r}_{ECI}-\vec{r}_{ECI_{site}}\right)\left(\vec{v}_{ECI}-\vec{v}_{ECI_{site}}\right)\\&+\frac{1}{\rho}\frac{\partial\left(\vec{r}_{ECI}-\vec{r}_{ECI_{site}}\right)}{\partial\vec{r}_{ECI}}\left(\vec{v}_{ECI}-\vec{v}_{ECI_{site}}\right)\\&+\frac{1}{\rho}\left(\vec{r}_{ECI}-\vec{r}_{ECI_{site}}\right)\frac{\partial\left(\vec{v}_{ECI}-\vec{v}_{ECI_{site}}\right)}{\partial\vec{r}_{ECI}}\end{aligned} \tag{4.24}$$

The derivatives of 4.24 are:

$$\frac{\partial\left(\vec{v}_{ECI}-\vec{v}_{ECI_{site}}\right)}{\partial\vec{r}_{ECI}}=0 \tag{4.25}$$

$$\frac{\partial\left(\vec{r}_{ECI}-\vec{r}_{ECI_{site}}\right)}{\partial\vec{r}_{ECI}}=[I]_{3x3} \tag{4.26}$$

$$\frac{\partial\frac{1}{\rho}}{\partial\vec{r}_{ECI}}=\frac{-1}{\rho^2}\frac{\partial\rho}{\partial\vec{r}_{ECI}}=\frac{-1}{\rho^2}\vec{L} \tag{4.27}$$

If we define the vector $\vec{L}'$ as

$$\vec{L}'\equiv\left[\vec{L}'_x,\vec{L}'_y,\vec{L}'_z\right]=\left[\frac{v_x-v_{x_{site}}}{\rho},\frac{v_y-v_{y_{site}}}{\rho},\frac{v_z-v_{z_{site}}}{\rho}\right] \tag{4.28}$$

we can simplify equation 4.24 using 4.25 to 4.28 as

$$\frac{\partial \dot{\rho}}{\partial \vec{r}_{ECI}} = \vec{L}' - \frac{\dot{\rho}}{\rho}\vec{L} \tag{4.29}$$

The complete observation matrix is for this case

$$H = \left[\left(L'_x - \frac{\dot{\rho}}{\rho} L_x \right), \left(L'_y - \frac{\dot{\rho}}{\rho} L_y \right), \left(L'_z - \frac{\dot{\rho}}{\rho} L_z \right), L_x, L_y, L_z \right] \tag{4.30}$$

The value of $\dot{\rho}$ can be easily computed from the observer's and object's state vectors using 4.19.

4.3 Angular Data Measurements Combined with Range and/or Range-Rate

The derivation of the observation matrix when angular measurements are present can be found at [10], section 3.22. In order to have also the range-rate measurements we only have to add 4.30 to the observation matrix in the relevant position.

4.4 Validation

In this section, the implementation of the new radar based sensors will be tested. Within the Spook tool the sensor type is defined in the `observers.dat` configuration file. Each type of sensor has one unique number assigned. The available obervers types and their correspondent measurments are defined in table 4.1. Type 1 correspond to an optical observer while types 2 to 7 are radar observers.

Observer	**Measurements**					
Type	α	δ	β	el	ρ	$\dot{\rho}$
Type 1	✓	✓				
Type 2			✓	✓	✓	
Type 3			✓	✓	✓	✓
Type 4			✓	✓		✓
Type 5					✓	
Type 6						✓
Type 7					✓	✓

Table 4.1: Observer Types available in SPOOK

For the validation of the new sensor types one of the previous tested cases will be used as a reference. The selected case is the UWE-3 cubesat which describes a Low

Earth Orbit. One ground-based observer placed at the University of Würzburg is tracking the orbit. The newly implemented observers types will be used (types 3 to 7). The target object definition makes use of a state vector for the 1 of August 2015

$$\vec{x}_{\text{initial}} = \left\{ \begin{array}{r} -872.408759 \\ 3487.077087 \\ 6049.576506 \\ 0.376570 \\ 6.548741 \\ -3.659296 \end{array} \right\} \tag{4.31}$$

The coordinates of the ground-based observer are

$$\phi = 49.7813^\circ \tag{4.32}$$
$$\lambda = 9.97394^\circ \tag{4.33}$$
$$h = 0 \text{ m} \tag{4.34}$$

The implemented orbit perturbations are all switched on: Non-spherical gravity field of degree and order 30, atmospheric drag with thermospheric winds, solar radiation pressure and solar and lunar gravity. The 1-σ accuracies of the sensor measurements are (when applicable):

$$\text{Slant range: } \sigma_\rho = 1 \text{ m} \tag{4.35}$$
$$\text{Azimuth: } \sigma_\beta = 0.0001^\circ \tag{4.36}$$
$$\text{Elevation: } \sigma_{el} = 0.0001^\circ \tag{4.37}$$
$$\text{Slant range-rate: } \sigma_{\dot\rho} = 1 \text{ m} \cdot \text{s}^{-1} \tag{4.38}$$

The algorithm used to perform the Orbit Determination (OD) is the Extended Kalman Filter. The accuracies of the defined sensors are higher than for typical space observations, so high-accurate results of the OD are expected.

4.4.1 Sensor Type 3: Azimuth, Elevation, Slant Range and Slant Range-Rate

The total error in position and velocity from the orbit determination with respect to the real object state is shown in figure 4.1.

It is appreciated how the errors in position are in the meter scale while the errors in velocity are below the cm·s^{-1} scale. The confidence level of the prediction, represented by the 3-σ radial, cross track and along track errors is shown in figure 4.2. It can be seen how the error of the prediction is always below the predicted error from the covariance matrix, giving an indication of the success of the orbit determination.

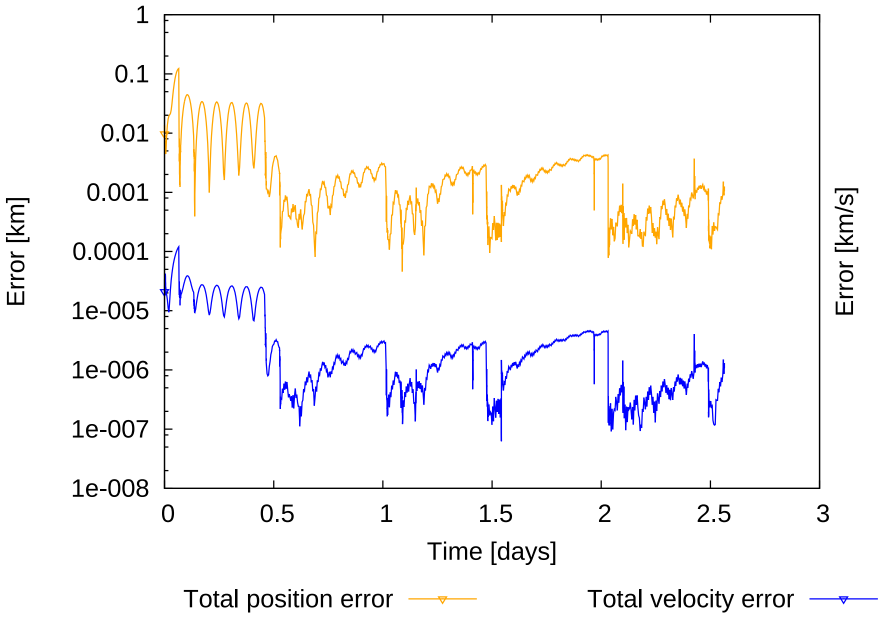


Figure 4.1: Total velocity and position errors for measurements of azimuth, elevation, slant range and slant range-rate

4.4.2 Sensor Type 4: Azimuth, Elevation and Slant Range-Rate

The same plots are produced for a sensor measuring only azimuth, elevation and slant range-rate. In this case, the total position and velocity errors are higher than when all the different kinds of measurements of sensor type 3 are used. This is an expected outcome, as the filter has less information available to make the prediction. However, the errors are still in the meter scale for position and cm·s^{-1} scale for velocity.

It can also be seen how the real error is in line with the predicted 3σ uncertainty from the covariance matrix.

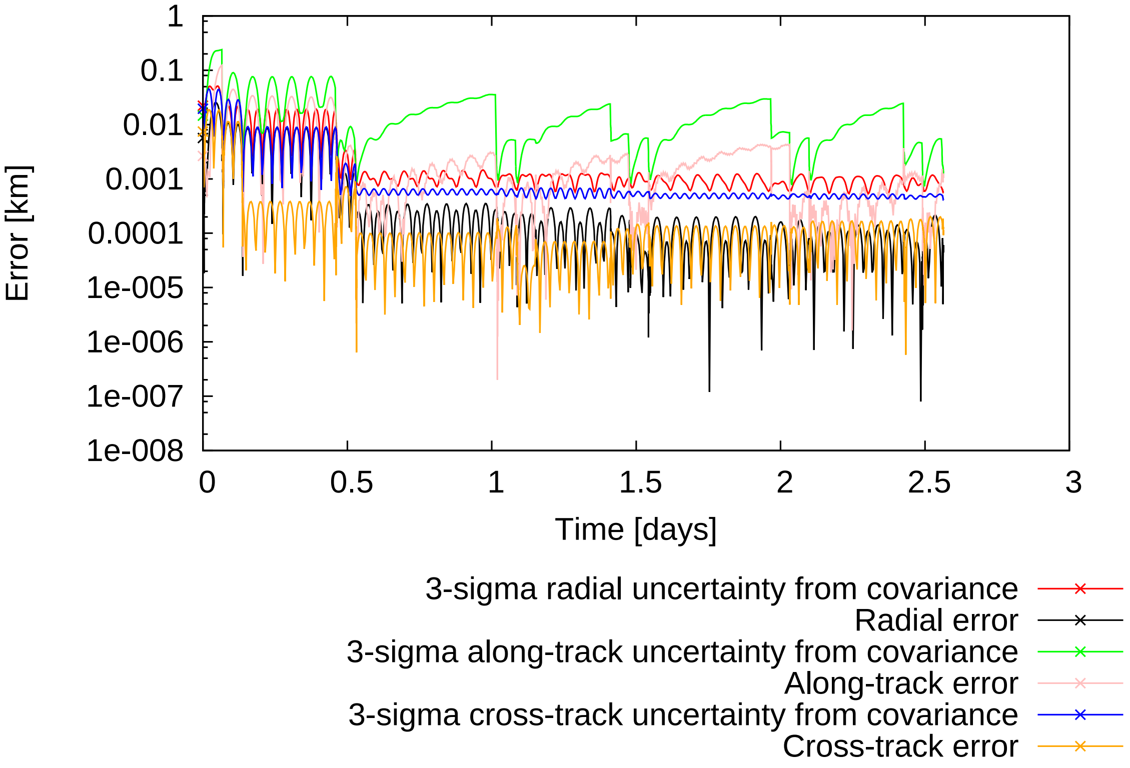


Figure 4.2: 3-σ errors for measurements of azimuth, elevation, slant range and slant range-rate

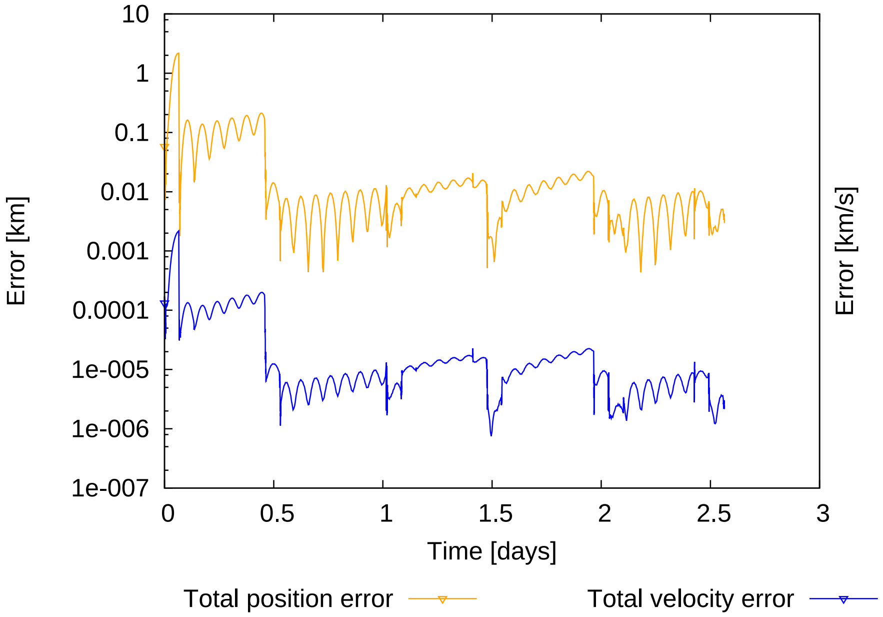


Figure 4.3: Total velocity and position errors for measurements of azimuth, elevation and slant range-rate

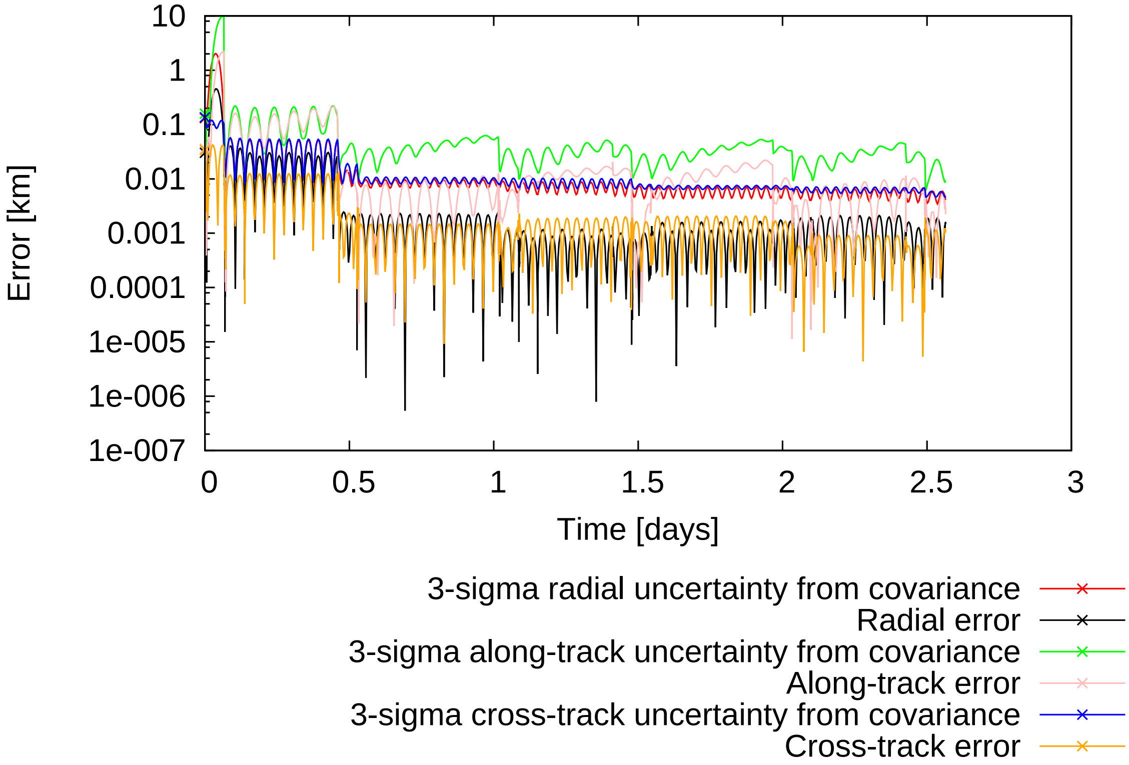


Figure 4.4: 3-σ errors for measurements of azimuth, elevation and slant range-rate

4.4.3 Sensor Type 5: Slant Range

The sensors type 5 to 7 do not have any kind of angle measurements, only different combinations of slant range and slant range-rate. The methods available in SPOOK for performing the Initial Orbit Determination from the measurements are all based in angle measurements. For this reason, an Initial Orbit Determination can not be performed for these sensors. Instead, an initial covariance propagation till the first measurement time has to be performed with the initial state information supplied via the configuration files.

The results of the orbit determination can be seen in figures 4.5 and 4.6. Despite the fact that less information is available, figure 4.5 shows that the errors obtained are lower than in the case where angular data combined with slant range and slant range-rate measurements are used (figure 4.1). This is due to the fact that the initial state provided is more accurate than the one obtained with the Initial Orbit Determination methods. The bigger errors in figure 4.1 are concentrated at the beginning of the simulation, converging towards a more accurate solution as more measurements are available.

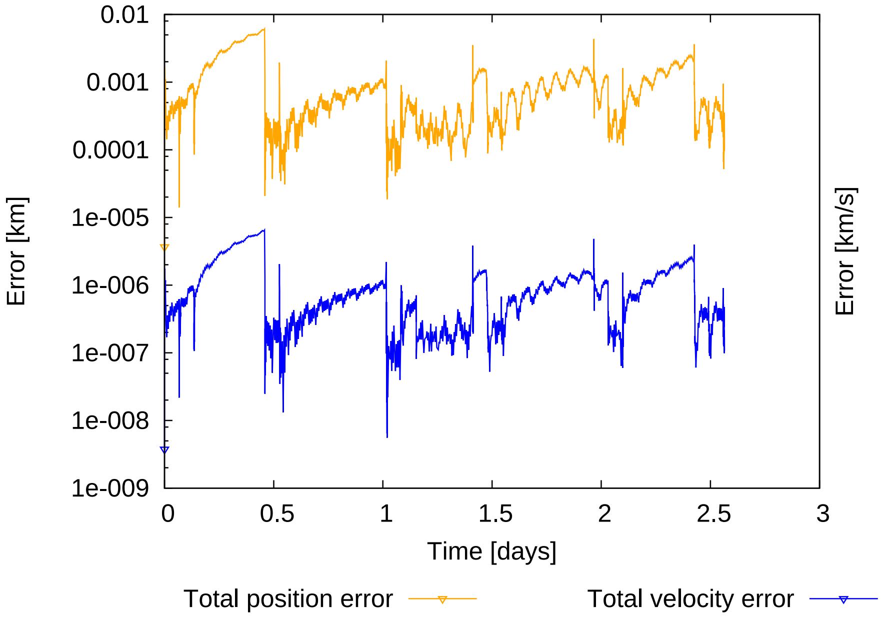


Figure 4.5: Total velocity and position errors for measurements of slant range

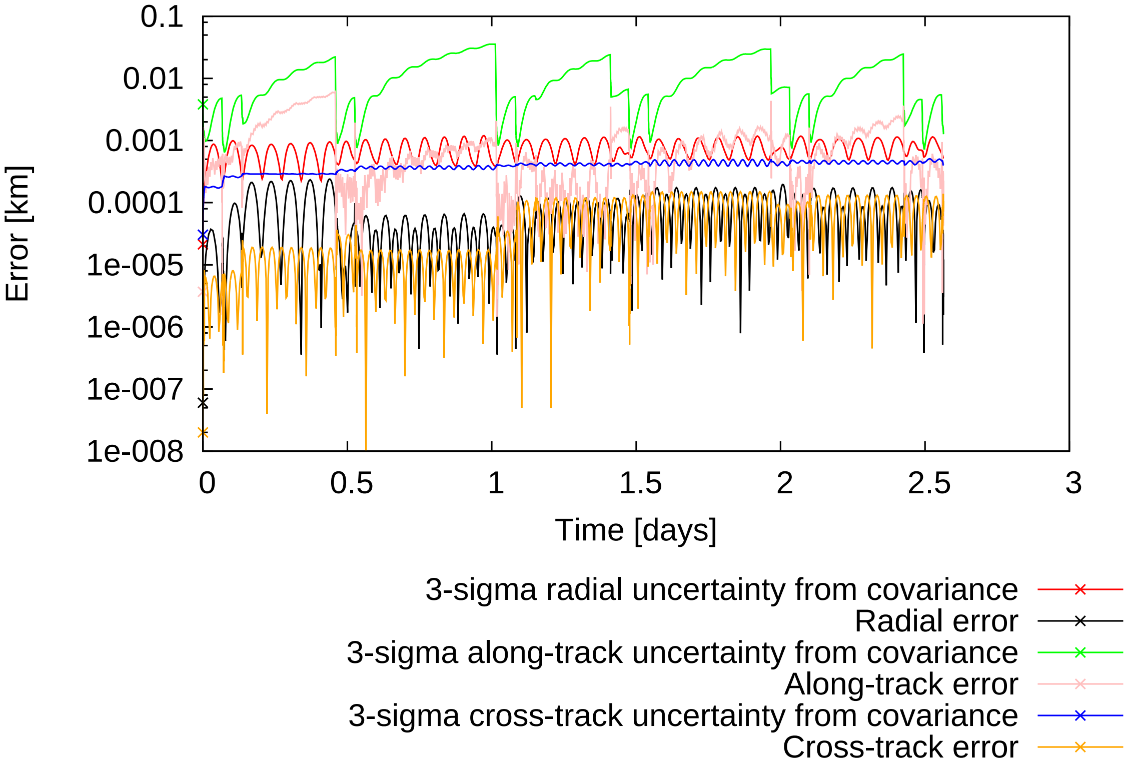


Figure 4.6: 3-σ errors for measurements of slant range

4.4.4 Sensor Type 6: Slant Range-Rate

Figures 4.7 and 4.8 shows the results of the orbit determination using only slant range-rate measurements. In this case, the initial position supplied is way more accurate than the prediction that can be made using only the slant range-rate measurements. This can be seen from the fact that the error at the beginning is very low, as it is relying more on the initial supplied state than in the observations. As the simulation progresses and the memory of the filter fades away, the error increases.

Despite this behaviour, in 4.8 it can be seen how the orbit has been successfully determined, as the real errors in the radial, cross-track and along-track direction are below the 3σ errors predicted from the covariance matrix.

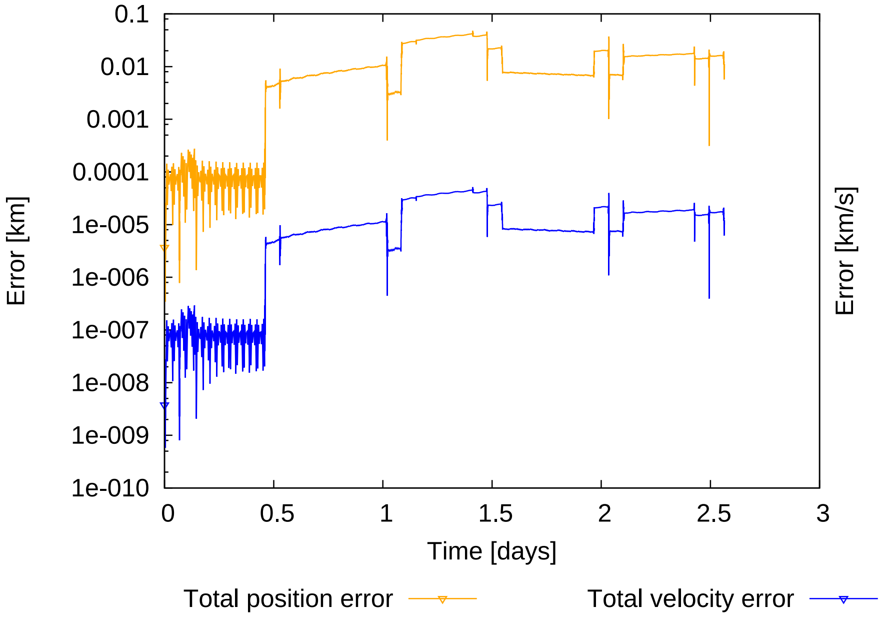


Figure 4.7: Total velocity and position errors for measurements of slant range-rate

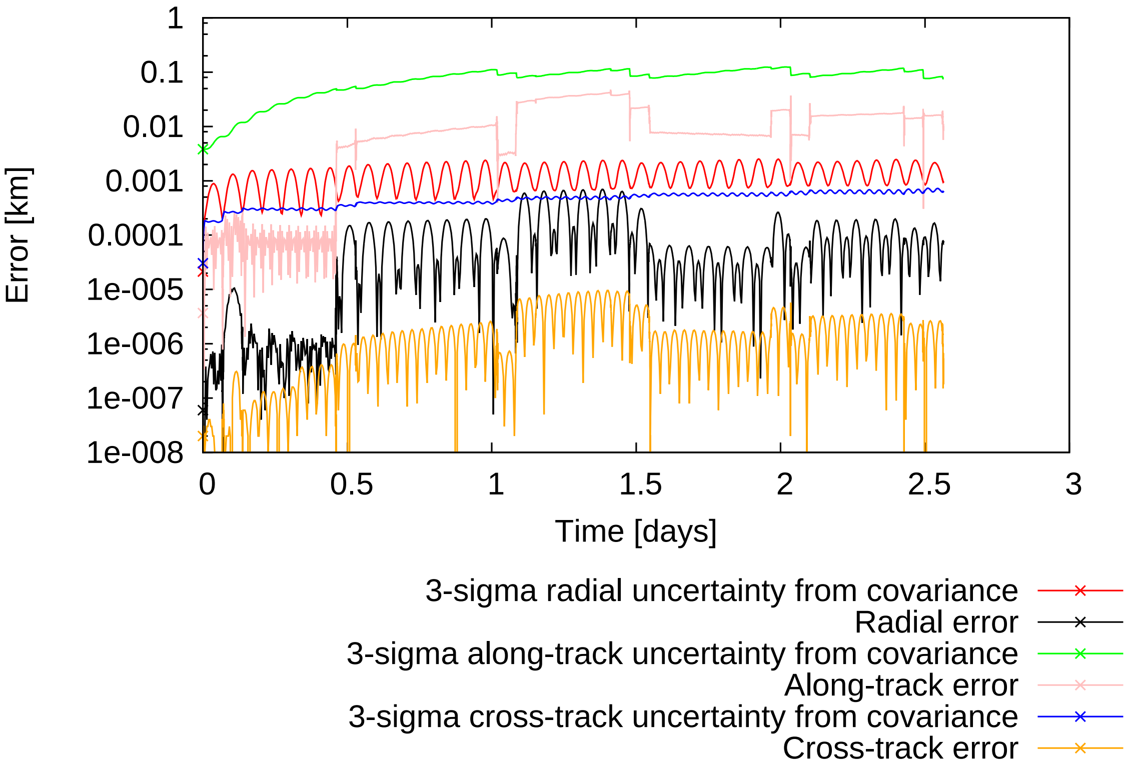


Figure 4.8: 3-σ errors for measurements of slant range-rate

4.4.5 Sensor type 7: Slant Range and Slant Range-Rate

Finally, the sensor using slant range and slant range-rate measurements is evaluated. This kind of sensor is a combination of the sensors type 5 and 6, so an improved performance is expected. The errors plotted in figure 4.9 follows the same pattern as in figure 4.5, indicating that the filter is highly influenced by the slant range measurements. The real errors are once again in line with the 3σ predicted errors from the covariance matrix, indicating the success of the orbit determination.

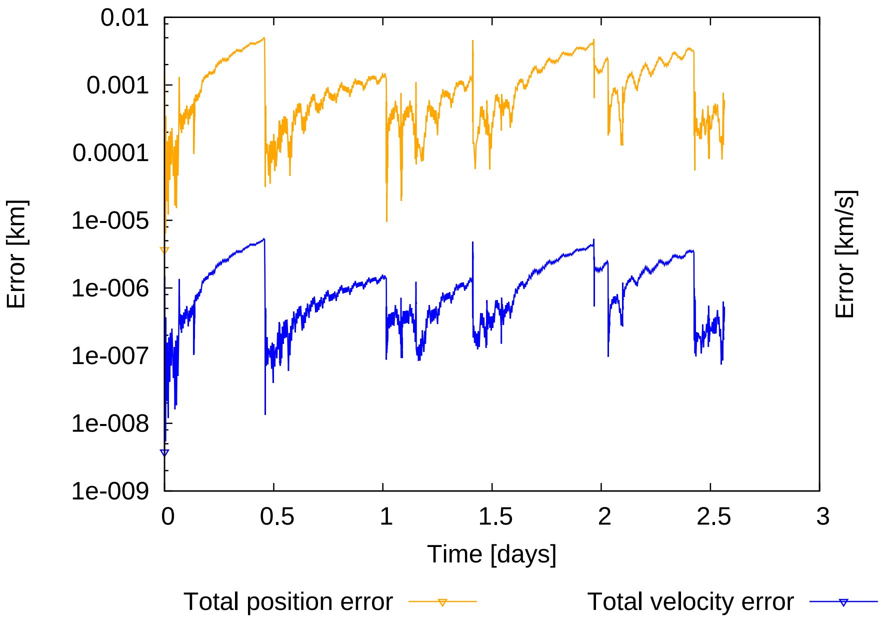


Figure 4.9: Total velocity and position errors for measurements of slant range and slant range-rate

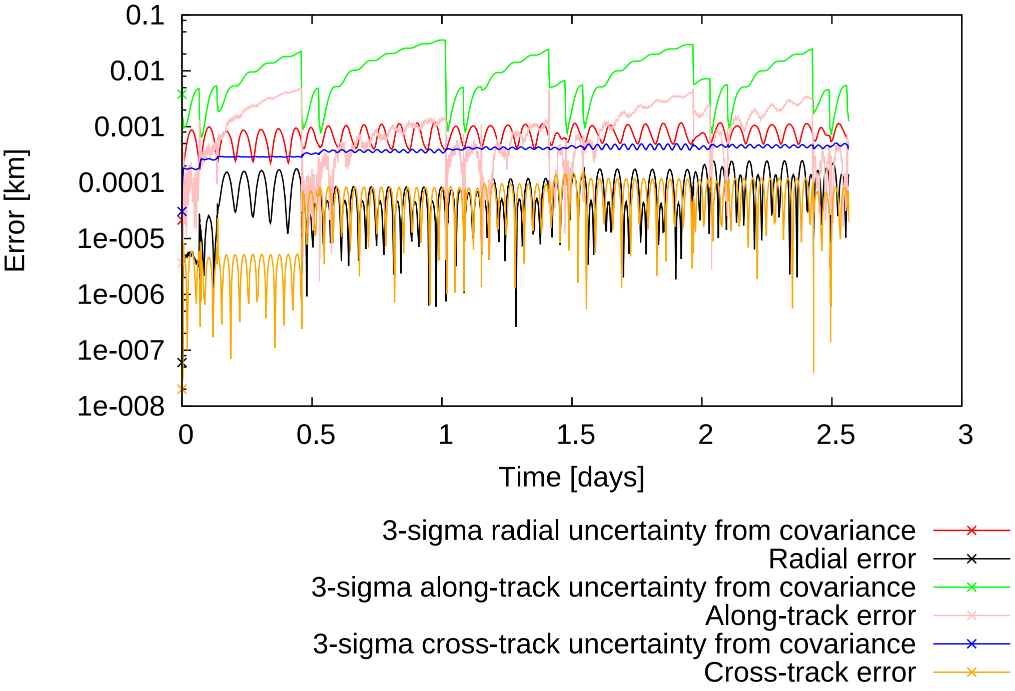


Figure 4.10: 3-σ errors for measurements of slant range and slant range-rate

4.5 Conclusions

The 5 new sensor types based in radar observations have been successfully implemented in SPOOK. This implementation required to derive a new observation matrix to be used by the orbit determination algorithms. This implementation has been validated by analysing the results of the orbit determination of a cubesat. In all the cases, the orbit determination has been successful and the accuracy of the prediction is in line with the expected behaviour. This leads us to validate the implementation of the new observer types in SPOOK.

Chapter 5

Light Time Delay Compensation

5.1 Theoretical Background

The sensors implemented in SPOOK to perform the tracking of the space objects are either optical or radar sensors. These sensors rely on the visible light (optical sensors) or radio (radar-based sensors) waves reflected by the tracked object and detected by the observer. Both visible light and radio waves are assumed to travel at the speed of light in the vacuum ($c = 299792458$ m·s^{-1}).

If we assume that the light speed is infinite, the apparent position of the object from the observer coincides with its true position at the time of measurement. However, in a more realistic implementation, during the time that the light needs to reach the observer, the tracked object will have continued its trajectory making the apparent measured position different from the true position at measurement time. The typical Light Time Delay (LTD) for Earth orbiting objects ranges from 1 μs to 0.2 ms [24].

Taking as reference figure 5.1, we assume that the object reflects the wave at time t_0. Due to the finite velocity of light, the wave does not reach the observer until time t_4. As a consequence, at detection time t_4 the apparent position of the object is not the object's real position (at time t_4) but its position at time t_0. The difference between times t_4 and t_0 is the Light Time Delay Δt_{ltd}.

$$\Delta t_{ltd} = t_4 - t_0 \tag{5.1}$$

In order to compute this time delay, the true light path (ρ_{tlp}) must be evaluated. This light path is defined as:

$$\rho_{tlp} = |\mathbf{r}_{\text{object}}(t_0) - \mathbf{r}_{\text{observer}}(t_4)| \tag{5.2}$$

Taking as a reference the detection time t_4 and combining expressions 5.1 and 5.2 the equation to evaluate the Light Time Delay is:

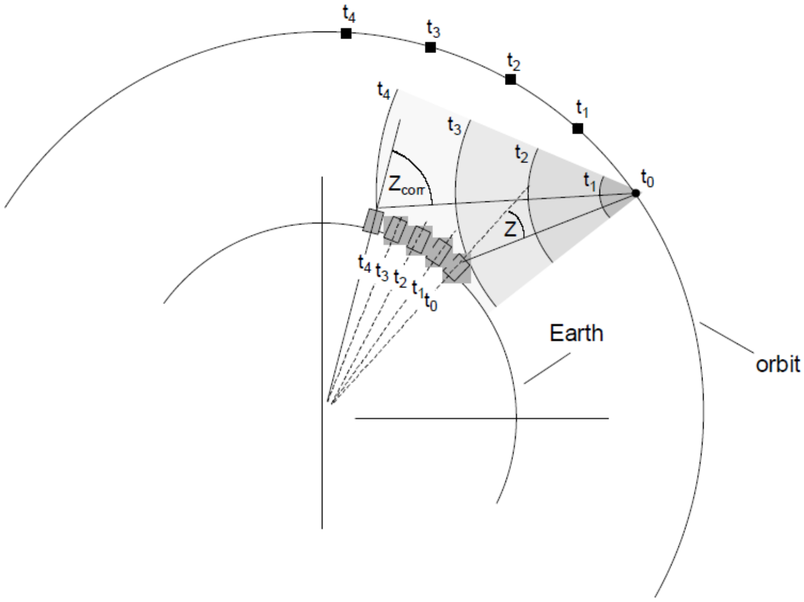


Figure 5.1: Light Time Delay effects [24]

$$\Delta t_{ltd} = \frac{\left|\mathbf{r}_{\text{object}}(t_4 - \Delta t_{ltd}) - \mathbf{r}_{\text{observer}}(t_4)\right|}{c} \tag{5.3}$$

Equation 5.3 is a non-linear implicit equation that must be solved iteratively. Taking as initial point a 0 time delay, at each solving step k the object's position is propagated backwards from t_4 by Δt^k_{ltd} and equation 5.4 is evaluated. The iterative process finishes if the difference between the previous value of the LTD and the next one is below a certain threshold.

$$\Delta t^{k+1}_{ltd} = \frac{\left|\mathbf{r}_{\text{object}}(t_4 - \Delta t^k_{ltd}) - \mathbf{r}_{\text{observer}}(t_4)\right|}{c} \tag{5.4}$$

5.2 Use of Light Time Delay in Measurement Generation

In order to implement the LTD in the measurement generation process equation 5.4 must be solved using the algorithm explained in section 5.1. After having computed the LTD, the object is propagated backwards by Δ_{ltd} and the measurements are evaluated using the object state at reflection time $\mathbf{x}_{\text{object}}(t_0)$ and the observer state at detection time $\mathbf{x}_{\text{observer}}(t_4)$.

The basic flowchart of how the LTD would be included in the measurement generation process is shown in 5.2.

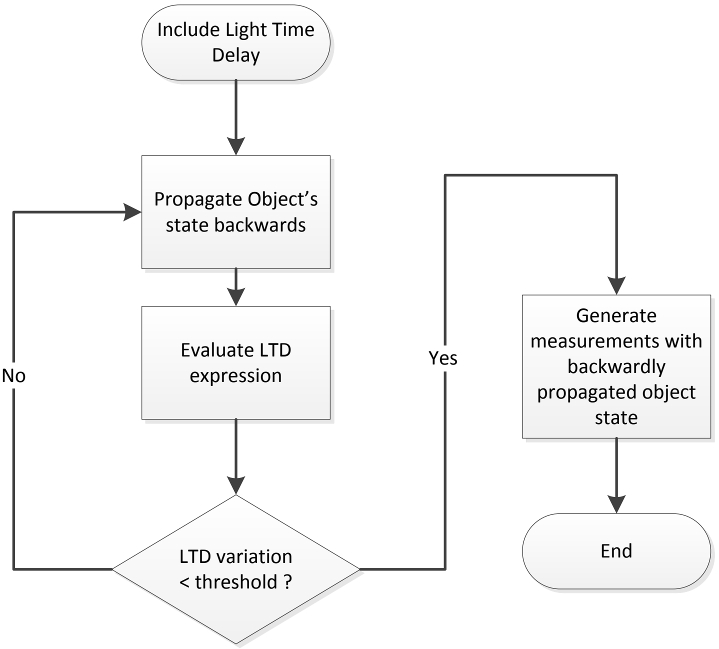

Include Light Time Delay
Propagate Object's state backwards
Evaluate LTD expression
LTD variation < threshold ?
No
Yes
Generate measurements with backwardly propagated object state
End

Figure 5.2: Include Light Time Delay in measurement generation flowchart

5.2.1 Validation using STK

To validate the implementation of this algorithm the generated measurements will be compared against the measurements generated using STK in a similar case. A two-body only propagator will be used. Due to truncation errors, the measurements generated are not exactly identical even when the LTD is not enabled. The comparison between this two tools can be seen in figures 5.3 and 5.4. The differences between the measurements produced with both software are mostly due to truncation errors outputting the results.

To compute the LTD a threshold value of 10^{-05} has been selected. Due to precision errors (SPOOK's propagator uses Double Precision for time) higher values of the threshold value does not improve the solution. The comparison between SPOOK and Satellite Tool Kit (STK) when LTD is enabled is shown in figures 5.5 and 5.6. The results of the comparison when the LTD is enabled are almost identical as in the case where no LTD is taken into account, validating the results of this implementation.

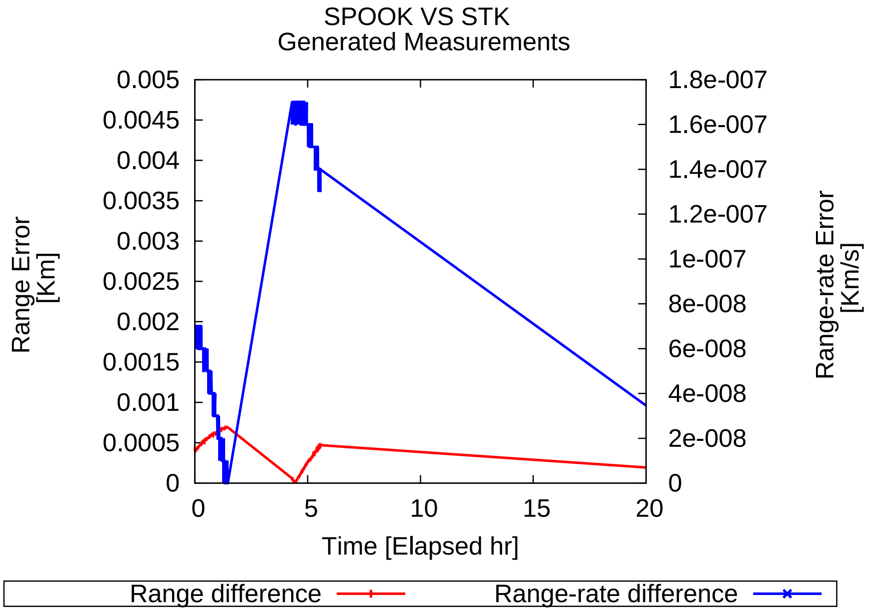


Figure 5.3: Comparison of range measurements between SPOOK and STK without considering LTD

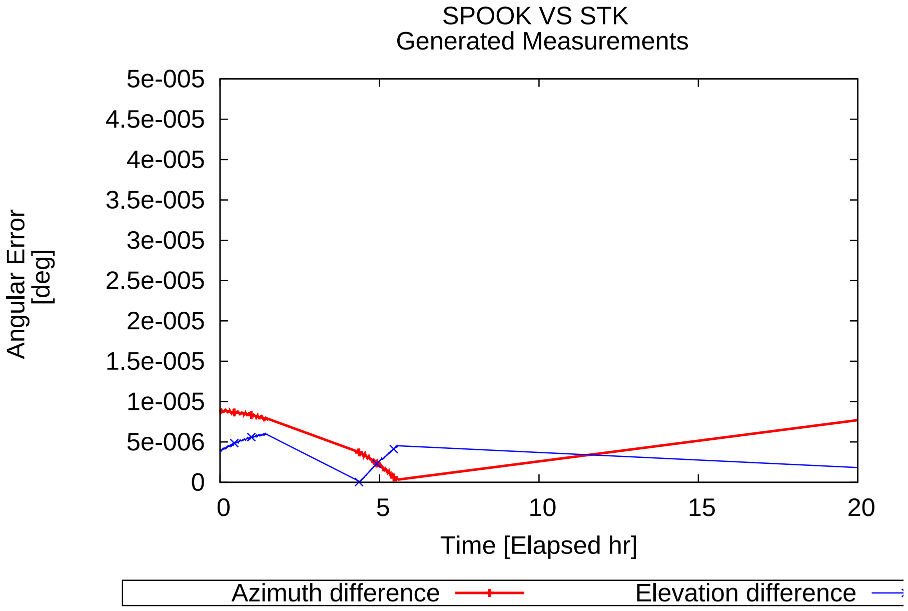


Figure 5.4: Comparison of angular measurements between SPOOK and STK without considering LTD

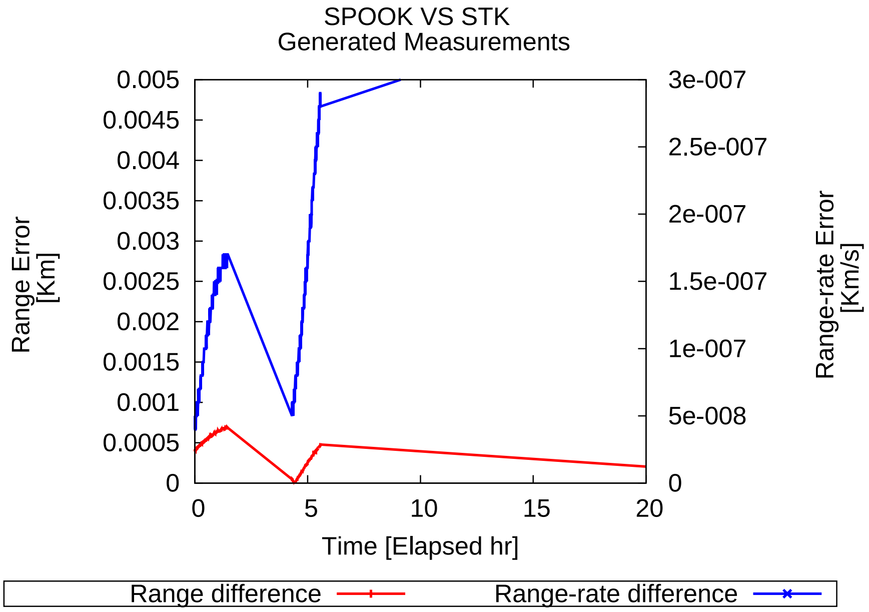


Figure 5.5: Comparison of range measurements between SPOOK and STK considering LTD

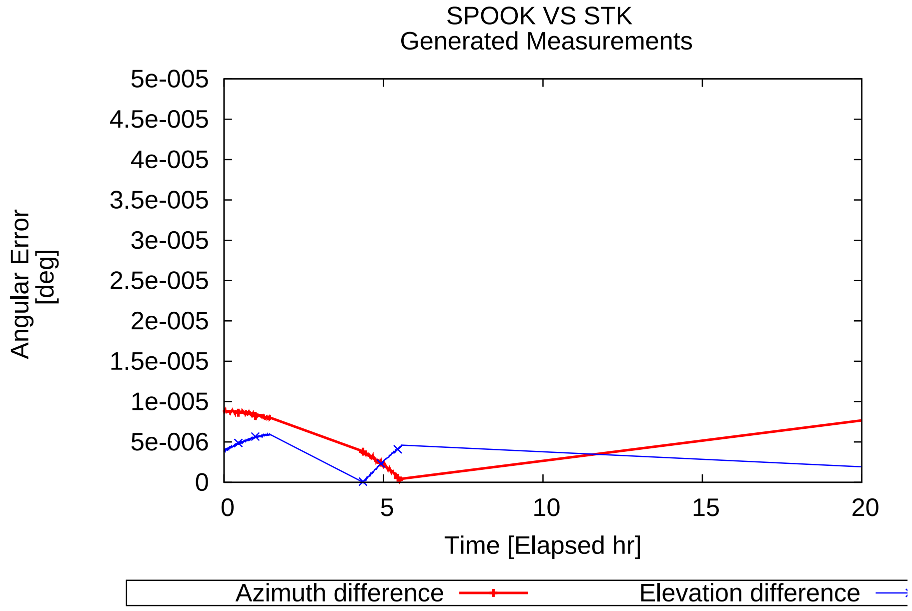


Figure 5.6: Comparison of angular measurements between SPOOK and STK considering LTD

5.3 Use of Light Time Delay in Orbit Determination

Including the LTD in the orbit determination process (EKF, WLS or SBLS) follows a similar algorithm as the measurement generation process. In this case, the object's position is not known and the estimated position coming from the OD method must be used. As the estimated position will change several times before convergence of the OD method is reached, the LTD algorithm will have to be applied several times for each measurement, increasing running time of SPOOK.

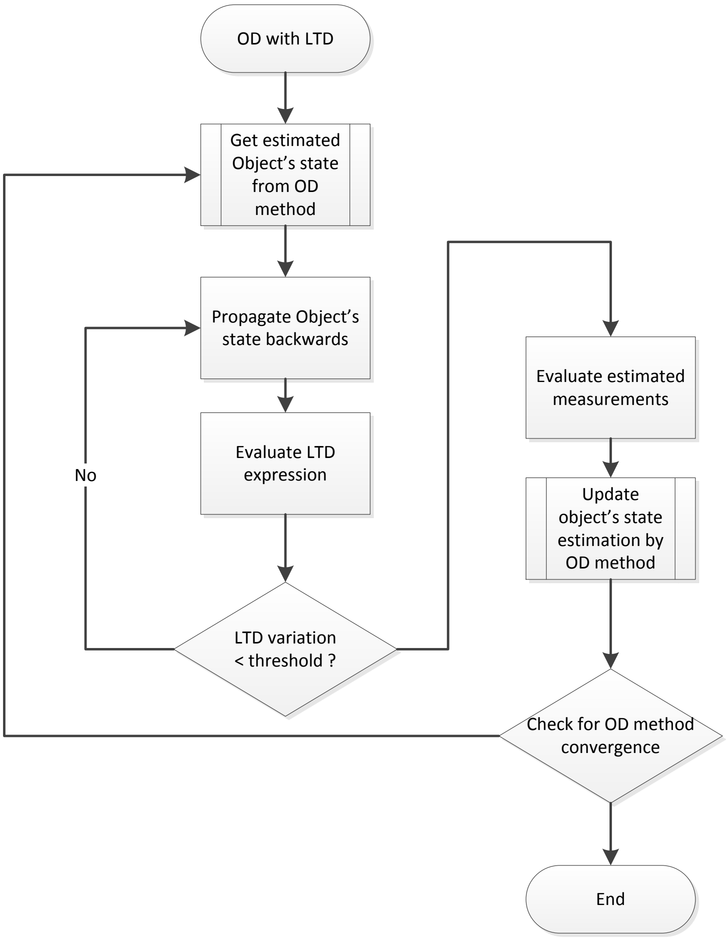

OD with LTD
Get estimated Object's state from OD method
Propagate Object's state backwards
Evaluate LTD expression
No
LTD variation < threshold ?
Evaluate estimated measurements
Update object's state estimation by OD method
Check for OD method convergence
End

Figure 5.7: Include Light Time Delay in the OD method flowchart

5.3.1 Light Time Delay in Extended Kalman Filter

Figure 5.7 shows a general flowchart to include the LTD in any OD method. In figure 5.8 it can be seen how the LTD will be taken into account in the EKF subroutine of SPOOK.

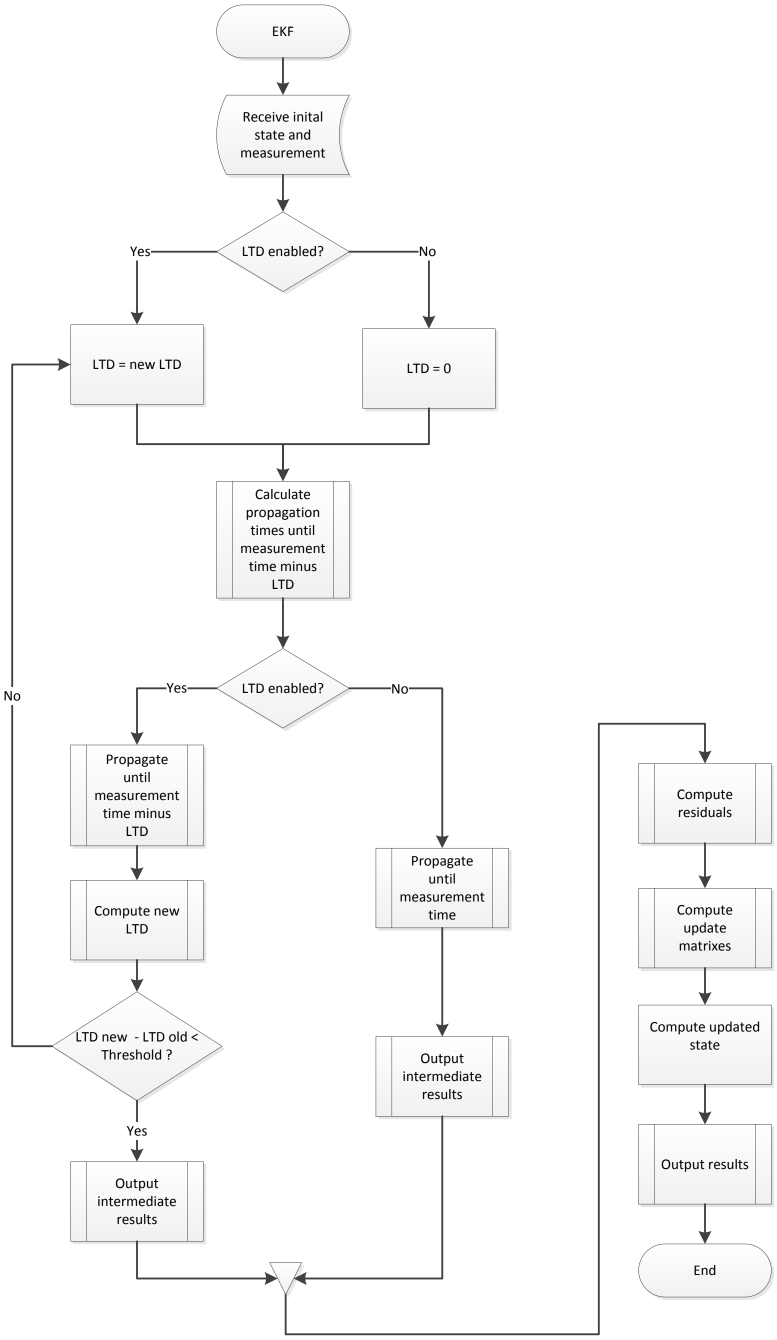

EKF
Receive inital state and measurement
LTD enabled?
Yes
No
LTD = new LTD
LTD = 0
Calculate propagation times until measurement time minus LTD
LTD enabled?
Yes
No
No
Propagate until measurement time minus LTD
Compute new LTD
LTD new - LTD old < Threshold ?
Yes
Output intermediate results
Propagate until measurement time
Output intermediate results
Compute residuals
Compute update matrixes
Compute updated state
Output results
End

Figure 5.8: Flowchart to include the Light Time Delay in the EKF method

When LTD is considered, the estimation of the object's state is no longer made at detection time t_4 but at the estimated reflection time of the object $\tilde{t}_0$ [1]. When the

[1] Due to estimation errors, $\tilde{t}_0$ does not necessarily have to coincide with t_0.

measurements are synthetically generated, the object's true position is known at detection times t_4 only. To calculate the errors between the true object's state and the estimated state an extra backwards propagation of the object true state from t_4 to $\tilde{t}_0$ has to be made. This step is avoided if real measurements are used, as the true position of the object is unknown.

Validation of the LTD implementation in EKF

To validate the implementation of LTD compensation in EKF three variations of the same OD problem have been taking into account.

- LTD is considered both in the measurement generation as in the OD.
- LTD is considered in the measurement generation but not in the OD.
- LTD is not considered neither in the measurement generation or in the OD.

If the implementation of LTD compensation has been successful, the error when the LTD is taken into account only in the measurement generation should be considerable higher than in the other cases. It is also expected that the error when LTD is enabled both in measurement generation and OD is similar as when LTD is not enabled. This can be seen in figure 5.9.

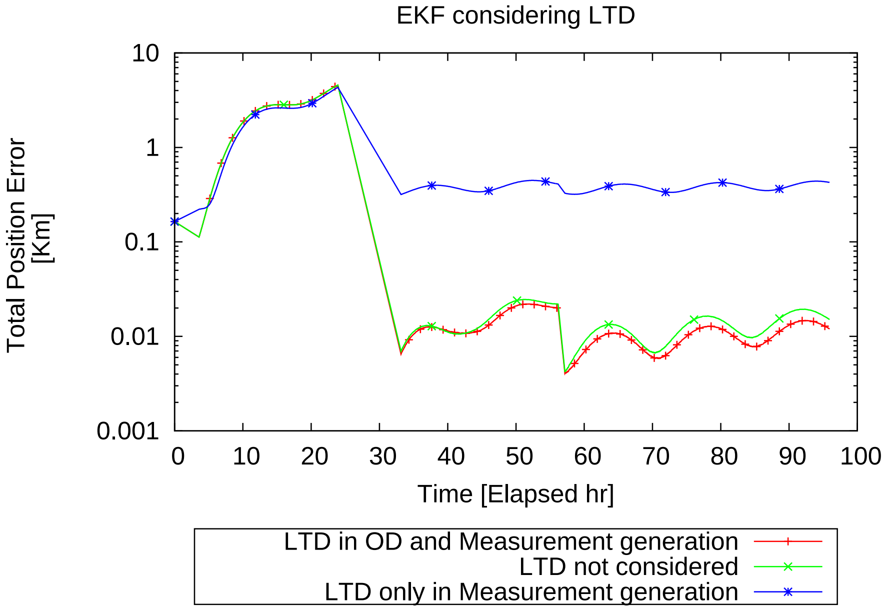


Figure 5.9: Validation of the implementation of LTD

As expected, the error when LTD is only consider in the measurement generation process is bigger than in the other cases. When comparing between enabling LTD and not enabling it at all, the error of the prediction is very similar.

5.3.2 Light Time Delay in WLS and SBLS

The implementation of the LTD compensation for the OD method based in WLS differential correction process (WLS and SBLS) is very similar to the implementation for the EKF method. The difference now is that multiple measurements are processed at the same time so an array of Light Time Delays have to be computed at each iteration of the differential correction process. The LTD compensation stops when the improvement in each and every one of the computed LTD is below the declared threshold. As several iterations are needed for the WLS to converge, the LTD compensation have to be applied several times in each batch of data, increasing computational times. The basic flowchart of the implementation is shown in figure 5.10.

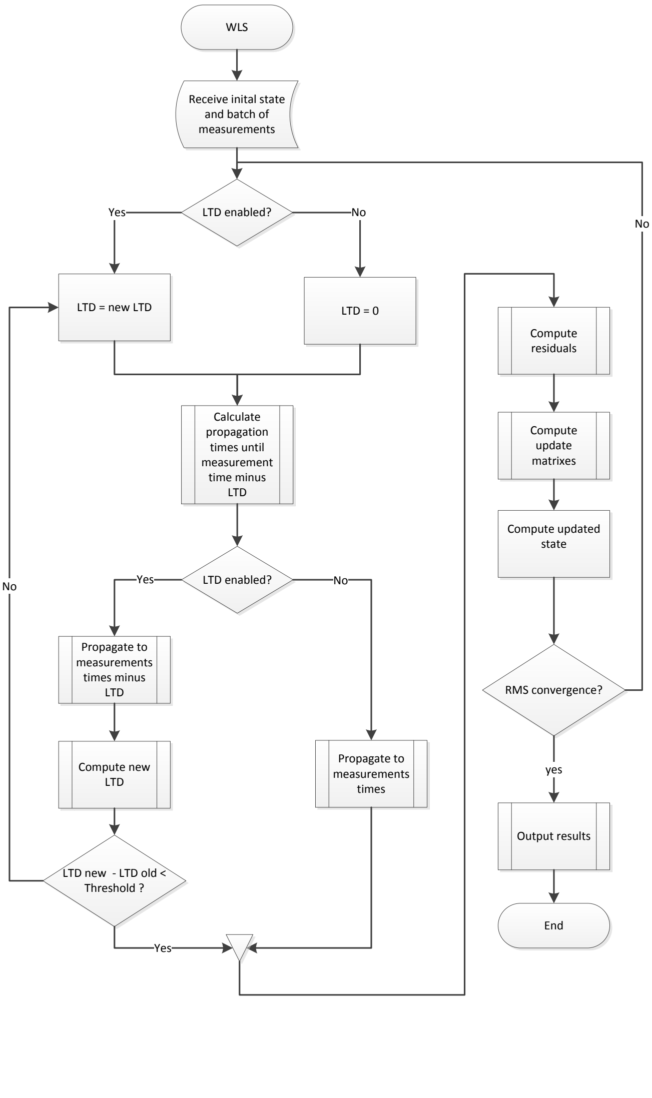

WLS
Receive inital state and batch of measurements
LTD enabled?
Yes
No
No
LTD = new LTD
LTD = 0
Calculate propagation times until measurement time minus LTD
LTD enabled?
Yes
No
No
Propagate to measurements times minus LTD
Compute new LTD
Propagate to measurements times
LTD new - LTD old < Threshold ?
Yes
Compute residuals
Compute update matrixes
Compute updated state
RMS convergence?
yes
Output results
End

Figure 5.10: Flowchart to include the Light Time Delay in the WLS based methods

Validation of the LTD implementation in WLS

For the validation of this implementation, a similar approach as in section 5.3.1 has been followed. Again, three different cases have been taken into account:

- LTD is considered both in the measurement generation as in the OD.
- LTD is considered in the measurement generation but not in the OD.
- LTD is not considered neither in the measurement generation or in the OD.

The results of this three different cases can be seen in figure 5.11. The error when the measurements are only taken into account in the measurement generation process is several orders of magnitude bigger than in the other two cases. When LTD compensation is completely enabled or disabled the error obtained is almost identical.

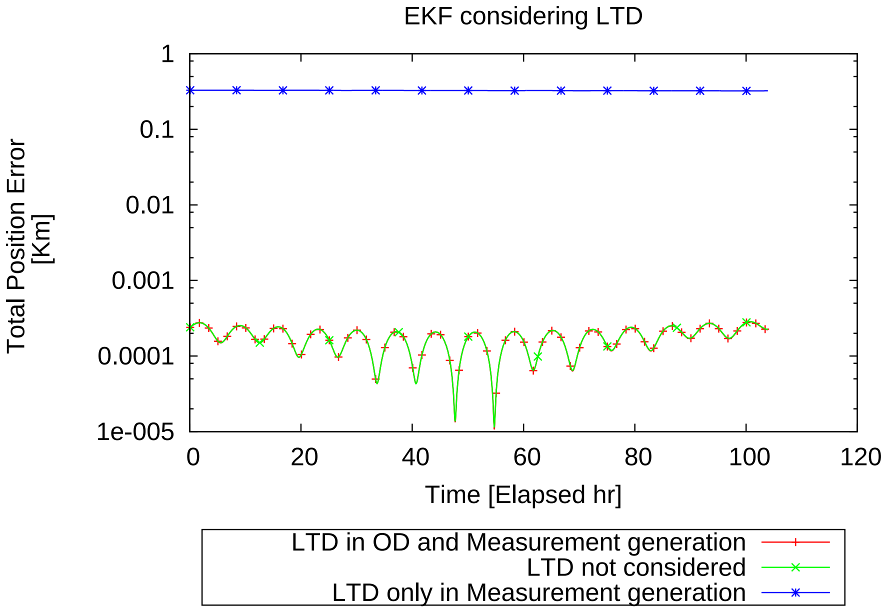


Figure 5.11: Validation of the implementation of LTD

5.4 Conclusions

Light Time Delay compensation has been implemented in SPOOK both in the synthetic measurement generation as in the different OD methods. For the validation of the measurement generation with LTD the results have been compared with the commercial software STK. In order to be able to compare the results, a two body propagation only, with no extra perturbations, has been selected. The differences in the measurements generated by both SPOOK and STK when LTD compensation is enabled are of the same order as when the LTD compensation was not enabled, validating the results. To validate the implementation on the different OD methods, a previously tested and validated case has been used (orbit determination for a Galileo satellite). Different combination of enabling/disabling the LTD compensation in the measurement generation and OD showed that the implementation of this feature has also been correct for the OD methods.

The main source of error of the LTD compensation is due to the propagator used by SPOOK. This propagator uses double precision only floating variables for the time. Besides that, the time format used is Julian Date (JD), limiting the available significant digits. As the typical time delays for LEO objects range from 1 μs to 0.2 ms (from 10^{-11} to 10^{-09} in delays in JD), the limited precision of the propagator will not take into account some of the smallest delays. For this reason, using values below 10^{-05} or 10^{-06} s as the threshold for the LTD compensation convergence will not improve the solution. With a typical orbital velocity of a LEO of 7 Km$\cdot$s^{-1} a difference of 10^{-05} s due to the numerical precision of the LTD would yield a difference in position 7 cm, way below that the accuracy expected to have for collision avoidance applications.

Chapter 6

Validation of Generated Measurements Using STK

6.1 Introduction

During validation of the LTD compensation using STK, some discrepancies between both tools have been found in the measurement generation process. Differences up to 5 m were found for the range measurements. When using a two body propagator only, the difference between the object position between SPOOK and STK were in the sub-meter level so the errors were originated by other sources. The comparison between the measurements in STK and SPOOK can be seen in figures 6.1 and 6.2.

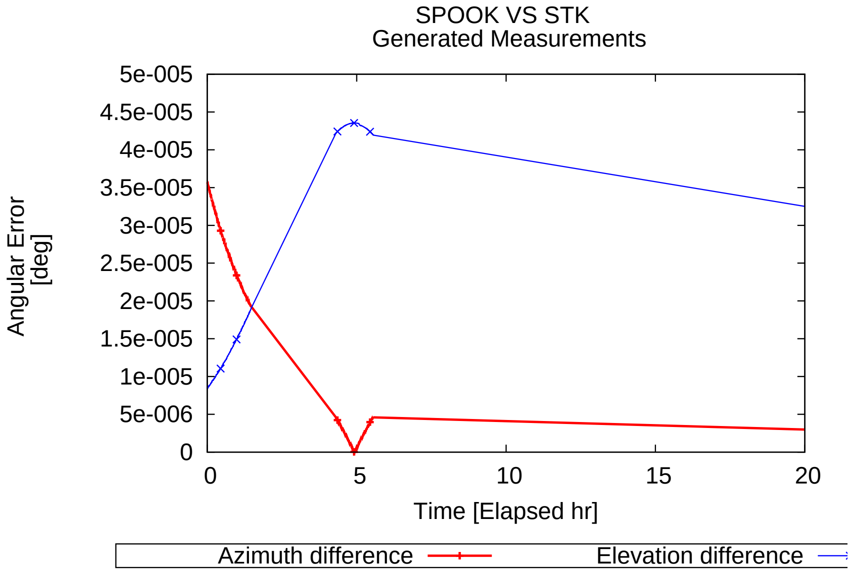


Figure 6.1: Difference between angular measurements generated with SPOOK and STK

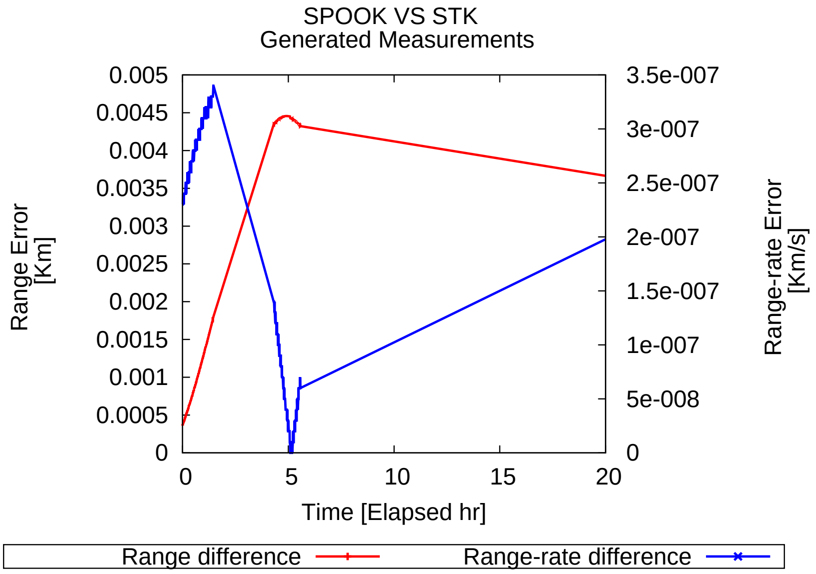


Figure 6.2: Difference between range and range-rate measurements generated with SPOOK and STK

6.2 Observer Position

During validation, it was found out that the main source of error was due to the position of the ground-based observer. In figure 6.3 the error in the observer position in the ECI frame of reference between STK and SPOOK can be seen. This error is in the order of 5 m, the same order of error seen in the range measurements.

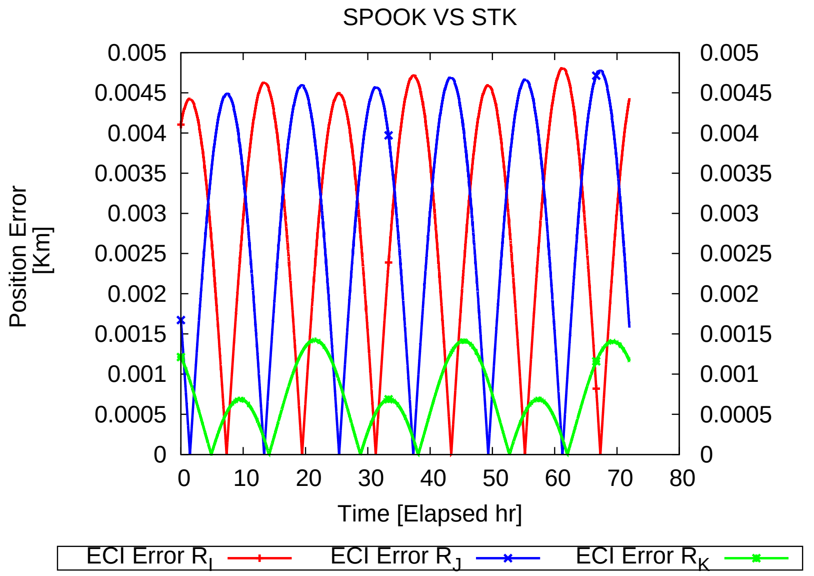


Figure 6.3: Observer position differences in ECI frame between STK and SPOOK

Different sources were contributing to the error. These sources were:

1. Earth Orientation Parameters (EOP) data.
2. Reference datum for the Earth.
3. Inertial Coordinate System Definition.

The contribution of each one of these sources to the global error will be analysed separately.

6.2.1 EOP data

Both in STK as in SPOOK the transformation between the ground-based observer geodetic coordinates (latitude, longitude and altitude) and its position in the ECI frame of reference (used for computing the sensor measurements) is made by using a data file that contains the Earth Orientation Parameters at different instants of time. The EOP data is provided by the International Earth Rotation Service and it is updated daily, with observed parameters since 1962 and predicted parameters for the next 182 days. The files being used by both tools were slightly different, as the updated dates were not the same, being this one of the major sources of error. When both tools used the same files, the errors in the observer position drop to an order of 1 m in all three components (see figure 6.4).

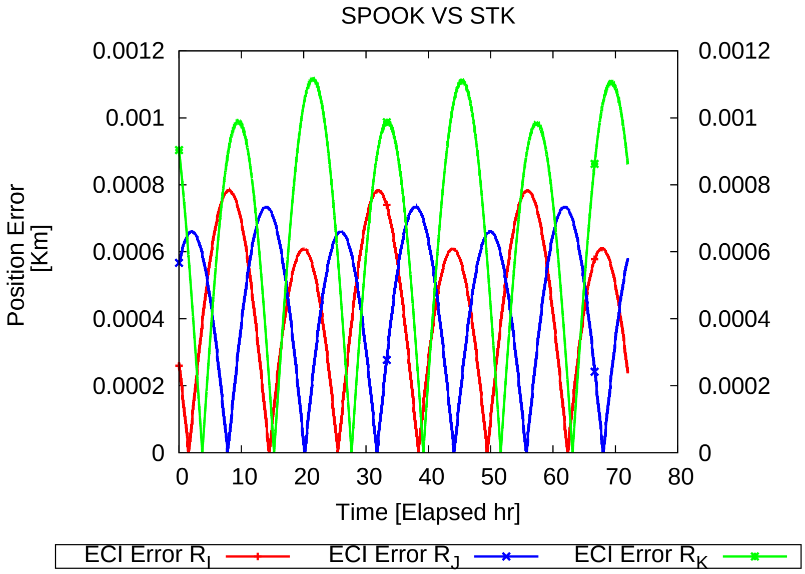


Figure 6.4: Observer position differences in ECI frame between STK and SPOOK with the same EOP data

With this correction, the differences in the generated range measurements have consistently drop to an order of 1 m.

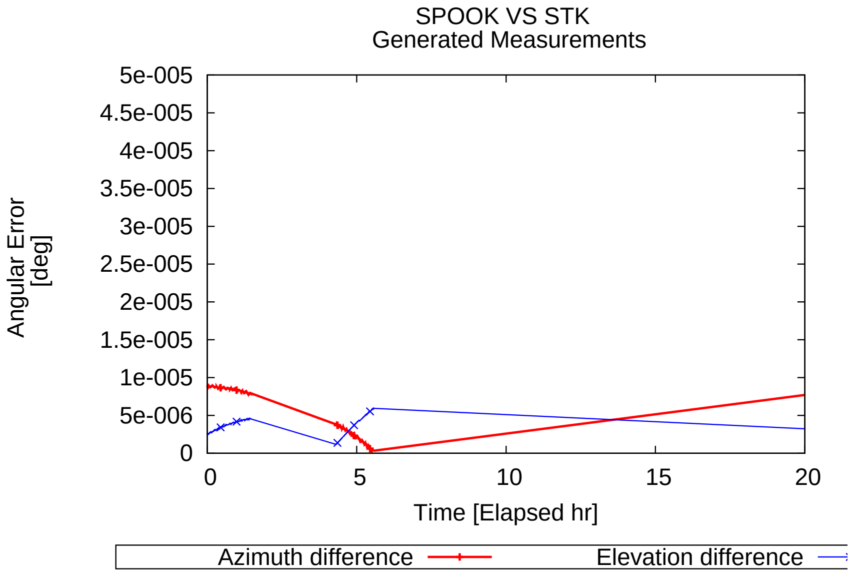


Figure 6.5: Difference between angular measurements generated with SPOOK and STK with the same EOP data

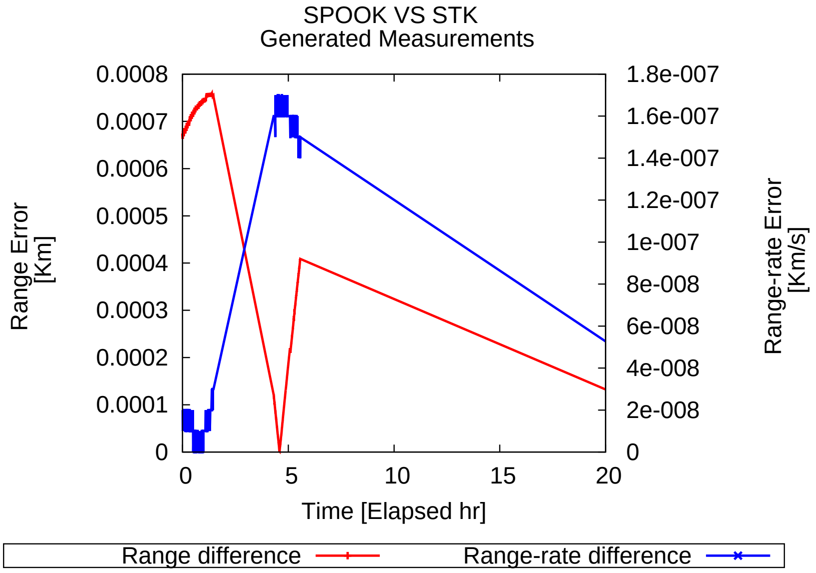


Figure 6.6: Difference between range and range-rate measurements generated with SPOOK and STK with the same EOP data

6.2.2 Reference Datum

With the EOP a transformation between the ECEF coordinate frame and ECI frame can be made. However, the ground-based observer position is specified by its geodetic coordinates. An extra transformation between geodetic coordinates and ECEF coordinates has to be made. This transformation is based on the use of a reference ellipsoid that models the Earth shape and its defined by a pair of parameters (Radius of the Equator $R_\oplus$ and Radius of the Poles or Radius of the Equator and flattening parameter f). The most widely used datum is the World Geodetic System 1984 (WGS84), used in the GPS satellite navigation system. This datum is defined [13] by the following parameters:

$$R_\oplus = 6378137 \text{ m} \tag{6.1}$$

$$f = \frac{1}{298257223563} \tag{6.2}$$

The values of the reference ellipsoid used in SPOOK were not consistent. On one side, it was using the value of the flattening dein the WGS84 reference ellipsoid but it was not using the same value of the radius of the Earth. The value used in SPOOK for the Earth radius was:

$$R_\oplus = 6378136.3 \text{ m} \tag{6.3}$$

$$\tag{6.4}$$

This correction reduces the error in the I and J directions below the meter level. However, a periodic error up to 1 m can be seen for the K component of the position. The period of this error is 12 hours, giving a hint of its relation with the nutation of the Earth.

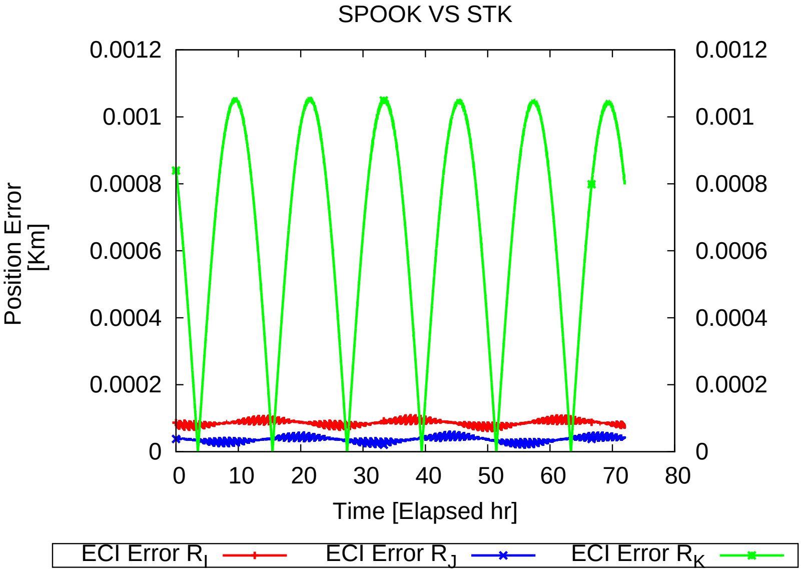


Figure 6.7: Observer position differences in ECI frame between STK and SPOOK using WGS84

As it can be seen in figure 6.9, the error now is below the meter level.

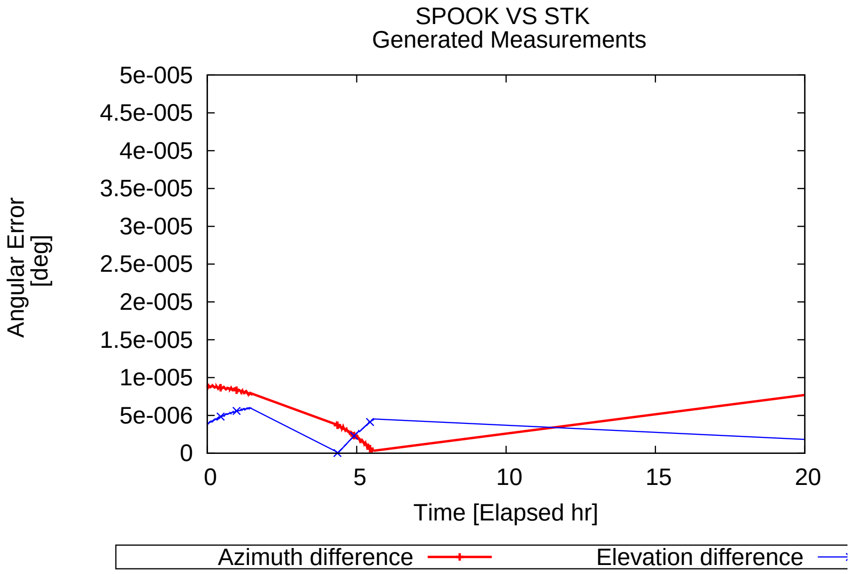


Figure 6.8: Difference between angular measurements generated with SPOOK and STK using WGS84

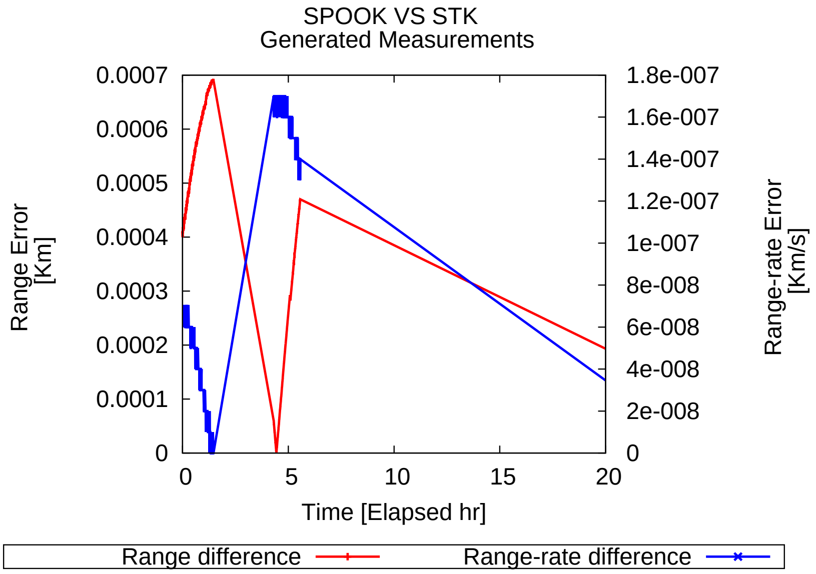


Figure 6.9: Difference between range and range-rate measurements generated with SPOOK and STK using WGS84

6.2.3 ICRF/J2000 Inertial Coordinate Systems

To represent the ground-based observer position in an Inertial coordinate system two options are usually considered: International Celestial Reference System or the J2000 coordinate frame. International Celestial Reference System (ICRF) is the best realization of an inertial coordinate system coordinate constructed so far. J2000 was the previous best realization of an inertial coordinate system and is still widely used. While this two coordinate systems are very close, they are not exactly the same. J2000 is the coordinate system chosen in SPOOK, following the implementation described in[7]. These two coordinate systems are defined using a transformation algorithm from an Earth Fixed System. These definitions are :

- **J2000**[7]. It uses the 1976 IAU Theory of Precession, the 1980 Nutation Model and the Greenwich Mean apparent Sidereal Time.
- **ICRF**[9]. It uses a corrected IAU-76 precession theory, the IAU 2000A [25] nutation model and the Earth rotation angle.

STK defines this two reference systems as [26]:

- ***ICRF***. *International Celestial Reference Frame. The ICRF axes are defined as the inertial (i.e., kinematically non-rotating) axes associated with a general relativity frame centered at the solar system barycenter (often called the BCRF). The IAU (International Astronomical Union) is the authority for the definition of the ICRF. The ICRF is the best realization of an inertial frame constructed to date, and thus represents an improvement upon the theory behind the J2000 frame. While the ICRF and J2000 frames themselves are very close, they are not identical; moreover, the J2000 frame rotates (very slowly) over time with respect to the ICRF frame. Recent star catalogs and celestial body ephemerides are most often expressed natively with respect to the ICRF frame. The ICRF frame is realized by its transformational algorithm between it and the Earth Fixed frame. The current algorithm uses the P03 precession model, the IAU2000A nutation model (as adjusted), and the Earth rotation angle (expressed as a linear function of time in UT1) and became operational on 1 Jan 2009. At present writing (Jan 2009), there is no documentation available from IERS (the International Earth rotation and Reference systems Service) for the current operational model; AGI uses an implementation based upon code available from SOFA (Standards of Fundamental Astronomy), the same code used to produce values in the Astronomical Almanac. The IAU2000A nutation model and the Earth rotation angle are documented by IERS is its Technical Note No. 32, IERS Conventions 2003.*
- ***J2000***. *Mean Equator and Mean Equinox of the J2000 epoch (JD 2451545.0 TDB which is 1 Jan 2000 12:00:00.000 TDB). The J2000 axes were considered the best realized inertial axes until the development of the ICRF. The J2000 frame is realized by the transformational algorithm (also known as the FK5 IAU76 theory) between it and the Earth Fixed frame. The algorithm uses the 1976 IAU Theory of Precession, the 1980 Nutation model, and the Greenwich Mean apparent Sidereal Time (expressed as a function of time in UT1), updated by IERS Technical Note No. 21 to include an adjustment to the equation of the equinoxes.*

The definition of the J2000 coordinate frame is identical for both STK and SPOOK software. However, in figure 6.7 we see a sinusoidal error with a period of 12 hours, identical to the period of the biggest nutation term. Furthermore, if we compare the observer position obtained from SPOOK in J2000 and from STK in ICRF the differences are below the cm level. This can be seen in figure 6.10.

The fact that the errors disappear when two different coordinate systems are used might point to the fact that both tools are making a different use of the same EOP data and how the ECEF system is defined. To fully understand the source of this differences between coordinate frames, further investigations of how STK uses the EOP should be performed before any corrective action is made in SPOOK.

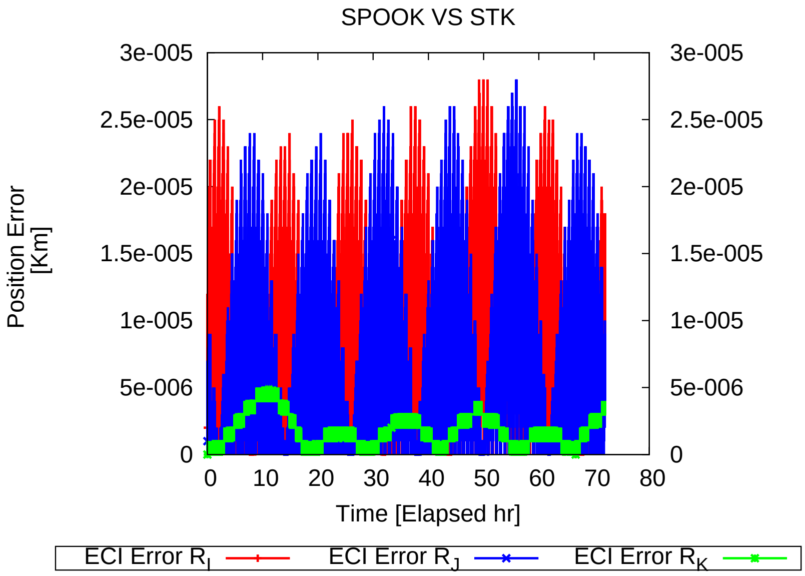


Figure 6.10: Difference between Observer position in STK ICRF and SPOOK J2000

6.3 Conclusions

The measurements generated with STK and SPOOK have been compared for an object following a Galileo orbit and being tracked by a ground-based station using a two-body only propagator. Before the corrective actions taken into account in SPOOK the differences between the measurements generated by both tools had an order of magnitude of 5 m for range measurements. This error have been brought down to below 1 m difference. The two main sources of error were the use of different EOP data files and the difference of the equatorial radius of the Earth. The value of the radius of the Earth has been updated in SPOOK so the reference datum used is WGS84.

A difference between the J2000 realization in SPOOK and STK has also been found. This difference translates into a 1 m periodical error in the K direction of the observer position in this frame reference when it is transformed from geodetic coordinates. This error could be related on how STK uses the EOP data and how the ECEF system is defined in the software. Further investigations should be made on how STK computes transformations to J2000 and ICRF before any actions are taken into account.

Chapter 7

PFISR Data

7.1 Introduction

This chapter documents the use of real world data coming from the Poker Flat Incoherent Scatter Radar (PFISR) to perform orbit determination of space debris using SPOOK. The experimental data has been supplied by LeoLabs Inc. It is emphasized that using PFISR for debris tracking is still in development phase, meaning that calibration of the data is still needed and it is not perfectly characterized. The aim of this chapter is to test SPOOK capabilities to use real data. Through this chapter the characteristics of the PFISR data will be commented as well as the modifications made in the SPOOK tool.

7.2 PFISR Data

The PFISR radar is able to produce angle measurements (azimuth β and elevation el) as well as slant range ρ and slant range-rate $\dot{\rho}$. The observation technique used for the following measurement campaign is of the type step and stare, where the radar does not point directly towards the object but towards a region of the space and waits for the object to cross its beamwidth. For this reason, the resolution of the angle measurements is very low, using the whole beamdwidth as the standard deviation of the angular measurements. From previous evaluations of this data it has been found out that the slant range measurements have a bias of 10 m. This bias is subtracted from the measurement provided by the radar. For these reasons, the first efforts will be focused on using only the slant range information to perform the orbit determination taking into account the new sensor type implemented in chapter 4. The characteristic parameters of the data are shown in table 7.1.

The analysis of the PFISR data will be focused on data acquired from SPOT 6, as both TLE files and ephemerides for this object are available, making it possible to validate the orbit determination process. This data is distributed in several tracklets

	Azimuth	Elevation	Slant range	Slant range-rate
σ	1.1 deg	1.1 deg	20 m	75 cm s^{-1}
Bias	N/A	N/A	10 m	N/A

Table 7.1: Characteristic parameters of the PFISR data

spanning over a time of 9 days. Approximately, two passes of the object (or tracklets) per day are available. In figure 7.1 the available data is shown.

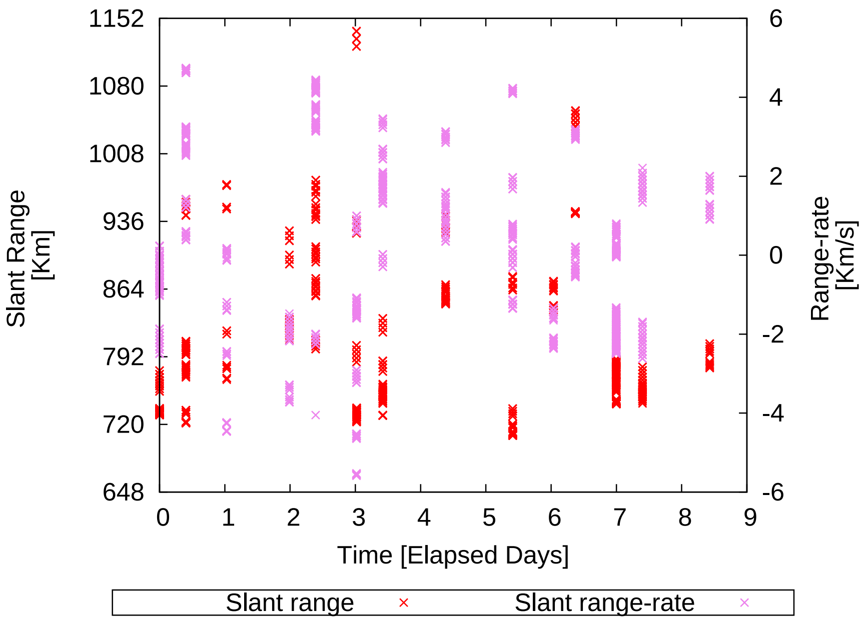


Figure 7.1: Measurements of SPOT 6 by PFISR

In figure 7.2 we can see a detail of two of the data tracklets. The typical duration of a tracklet is around one or two minutes.

As stated before, the first attempts at performing an orbit determination will be made using only range information. Having only range information, an Initial Orbit Determination (IOD) can not be performed with the IOD methods currently availble in SPOOK. To initialize the process, an initial covariance propagation will be made using data from the TLE files and/or from the available ephemerides as an initial state.

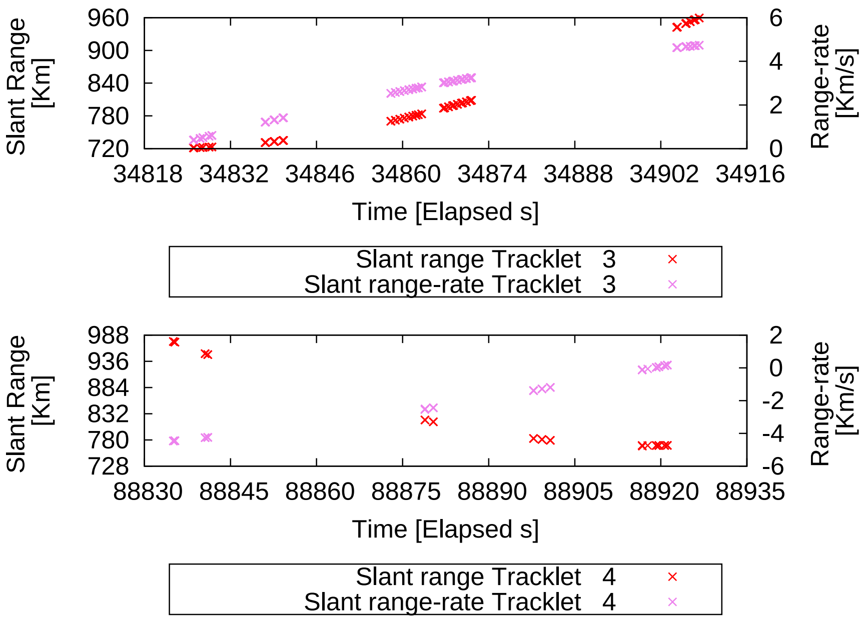


Figure 7.2: Measurements of SPOT 6 by PFISR

7.3 Major Changes in SPOOK

Here, the main changes implemented in SPOOK to be able to use and validate PFISR data are commented.

An import subroutine had to be implemented in SPOOK to enable the use of PFISR data. This subroutine is based on the use of a configuration file (named `PFISR.dat`) that contains the number of folders, the number of files in each folder and the names of the folders and the files. A `.bat` file has also been coded to automatically generate the `PFISR.dat` configuration file. This `.bat` file searches through the folders containing measurements information and writes the name of the files into the `PFISR.dat` configuration file.

For debugging purposes, the capability of writing the synthetic generated measurements into PFISR formatted files has also been added to SPOOK. The same folder structure used by PFISR data is followed here and the `PFISR.dat` configuration file is automatically generated. This capability can be switched on/off in the `parameter.ini` configuration file.

Another added feature to SPOOK is the capability to match certain output times. This helps the validation process, as it allow us to match the same output times as the available ephemerides. This feature is based in the use of two basic parameters: `Para%Target%t0` and `Para%dt_out_covprop`. `Para%Target%t0` serves as the reference for the desired output times while `Para%dt_out_covprop` is the time step (constant) between output times. In the output subroutines, the output times are computed so they match `Para%Target%t0` plus and integer number of `Para%dt_out_covprop`. Adjusting these two parameters the desired output times can be achieved is they have a constant time step between time incidents.

7.4 Evaluation of the Initial State

7.4.1 Comparison between TLE Initialization and Ephemerides

To assess the use of the TLE files as the initial state of the object, a covariance propagation only will be made and the results will be compared against the data from the ephemerides files. The TLE provides the position of SPOT 6 on 19/03/2016. The results from this comparison can be seen in figures 7.3 and 7.4.

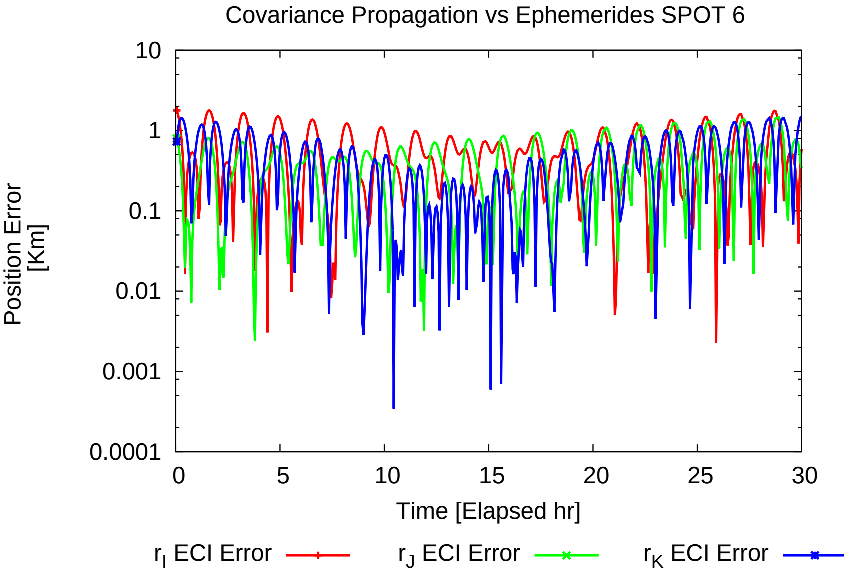


Figure 7.3: Position errors from SPOOK propagation with TLE as initial state compared against SPOT 6 ephemerides

In figure 7.3 can be seen how, after 30 hours of propagation, the errors in position are in the order of 1 Km while for the position are in the order of 1 m/s. The order of magnitude of the error does not change drastically for longer simulations.

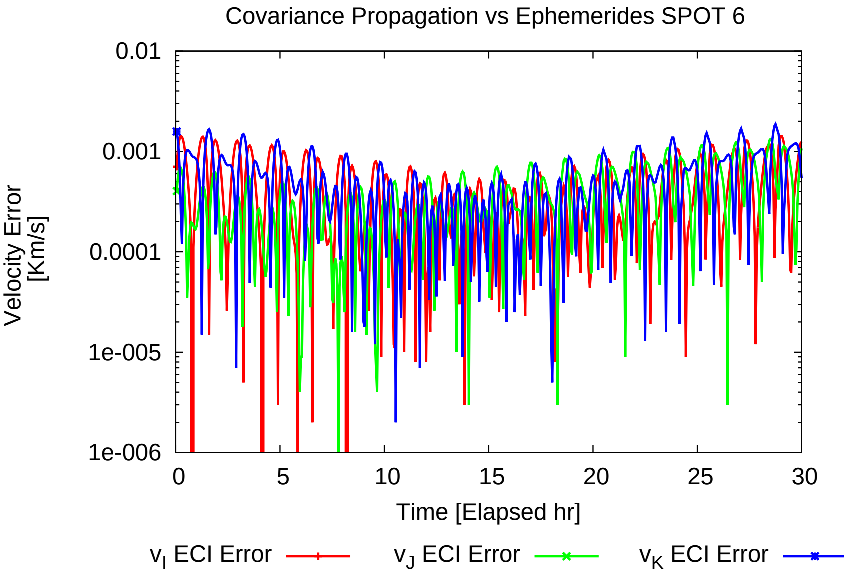


Figure 7.4: Velocity errors from SPOOK propagation with TLE as initial state compared against SPOT 6 ephemerides

7.4.2 Comparison using Ephemerides for Initalization

The same comparison can be performed using as initial point one of the states of the ephemerides files. The results of this propagation can be seen in figures 7.5 and 7.6.

In this case, it can be seen how the errors at the beginning, both for position and velocity, are very low. This is due to the fact that the initial state supplied has no errors at all. After a few hours of propagation, the errors stabilizes around the same order of magnitude as for when TLE set were used. When ephemerides are used for initialization, the errors are still slightly lower than when TLE are used. Both initialization methods are suitable to perform an orbit determination and the results obtained are similar. For this reason, the following analysis will be made using the ephemerides as initialization method. For completeness, the results regarding the initialization via TLE files can be found in appendix B.1.

Figures 7.3 and 7.5 show that there exist differences between the ephemerides files and the model used in SPOOK for orbit propagation in the order of 1 Km for position even when the ephemerides point is used for initialization. For that reason, better accuracies than 1 Km can not be achieved in the orbit determination when comparing against the available ephemerides..

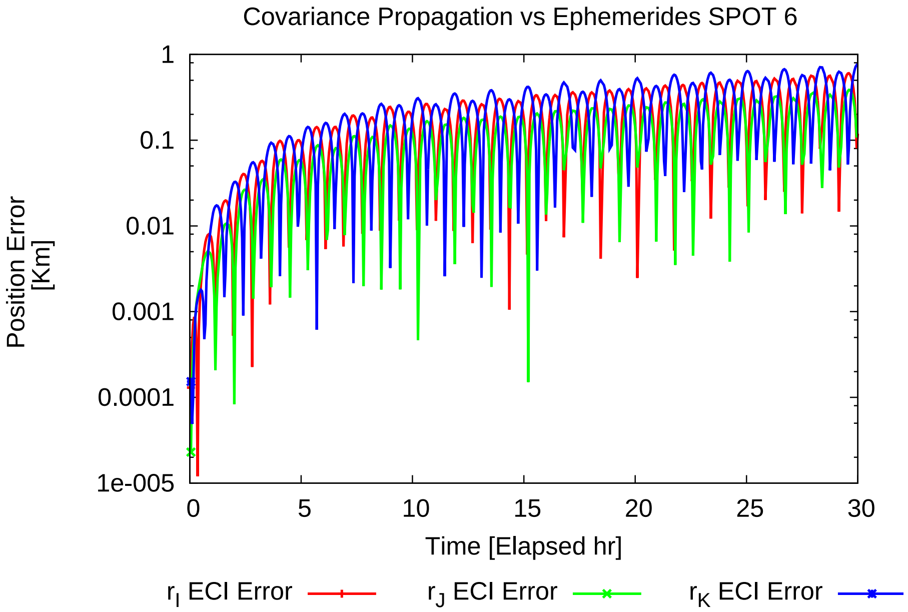


Figure 7.5: Position errors from SPOOK propagation with ephemerides as initial state compared against SPOT 6 ephemerides

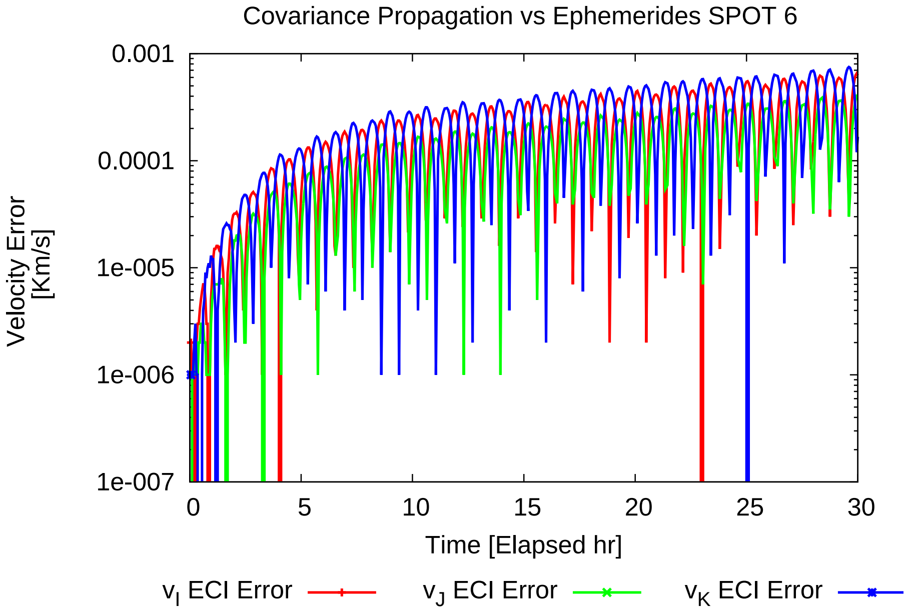


Figure 7.6: Velocity errors from SPOOK propagation with ephemerides as initial state compared against SPOT 6 ephemerides

7.5 Synthetic Measurements

As a first step, synthetic measurements will be generated with the same characteristics as the PFISR data and the same orbit as the SPOT 6. Two Orbit Determination methods will be used: WLS and SBLS. In both cases, we consider the following perturbations: atmospheric drag, Solar Radiation Pressure, Solar Gravity, Lunar Gravity, Thermospheric Winds and a non-spherical gravity field with a degree and order up to 30.

To better analyse the results of the orbit determination, the errors will be plotted using the RTN frame of reference (Radial, Tangential and Normal to the trajectory). The same transformation from ECI to RTN is applied to the data coming from SPOOK and the Ephemerides, using the data from the ephemerides to compute the rotation matrix from ECI to RTN at each time incident.

Using the same parameters as the PFISR data, synthetic measurements have been generated. To make this data as close to the PFISR data as possible, a maximum tracklet duration of 120 s is chosen and a minimum gap of 40000 s (around 11 hours) between tracklets selected. A similar distribution is achieved as the real one using these settings. The data produced is shown in figure 7.5.

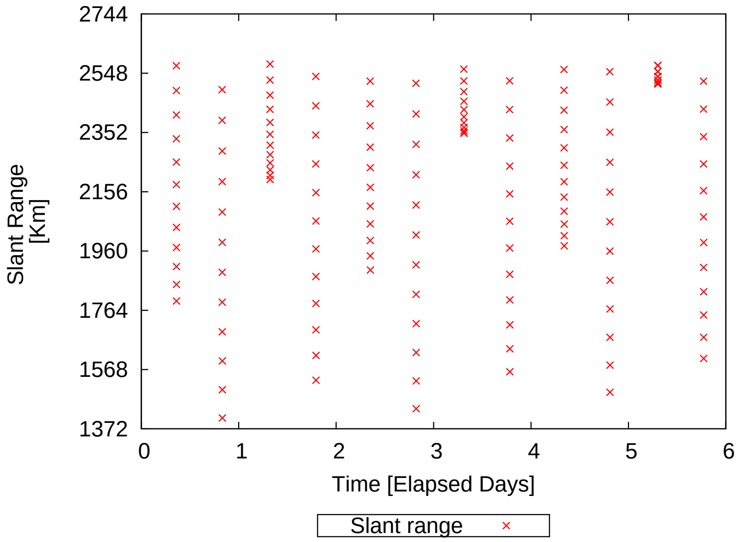


Figure 7.7: Synthetic measurements generated by SPOOK following a similar distribution as PFISR data

7.5.1 Weighted Least Squares

The WLS process converges towards a solution. In figure 7.8 a 3D representation of the orbit estimated by SPOOK with the WLS method is shown. The orbit is plotted in ECEF coordinates.

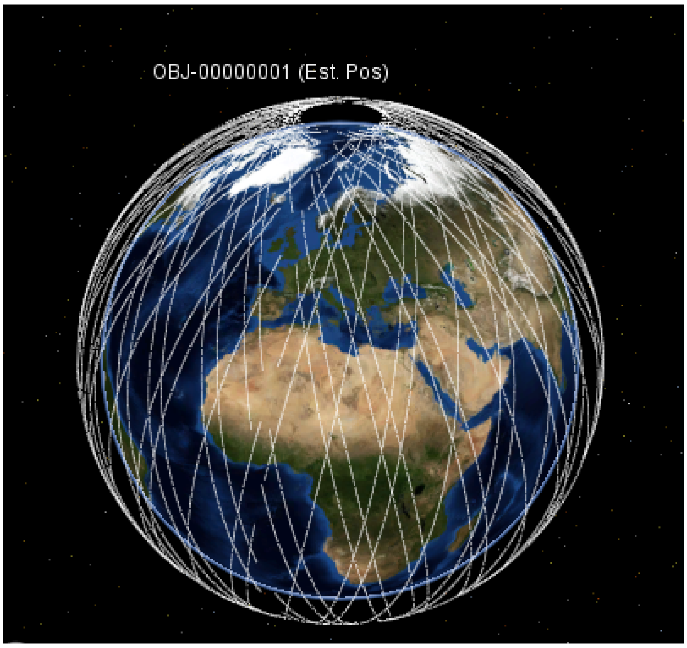


Figure 7.8: 3D representation of the SPOT 6 orbit estimated by SPOOK

When compared the results against the reference solution used to create the synthetic measurements, the errors are below 100 m for position, yielding a very accurate solution as it can be seen in figure 7.9.

The results of this first orbit determination are also compared against the available ephemerides data of SPOT 6 in figures 7.10 to 7.15. The errors of the OD are below the 3σ uncertainties predicted by the covariance matrix for the normal and radial case in the position, and for most of the simulation for the tangential direction. Ideally, for a successful orbit determination, the real errors should be below the predicted uncertainty for the three components. In this case, however, the uncertainty in the normal and cross track direction is one order of magnitude above the error. Regarding the position, the bigger errors are in the tangential direction. This result is expected as the highest velocity is tangential to the trajectory. The order of magnitude of this error is 1 Km, the same error found when the ephemerides were compared against the orbit propagation model of SPOOK.

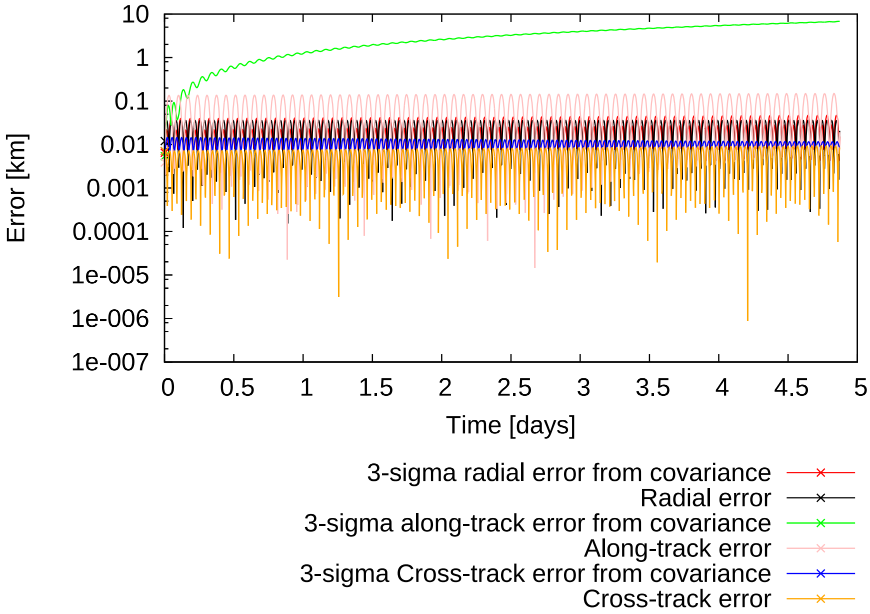


Figure 7.9: Results of the OD with synthetic data compared against the reference solution

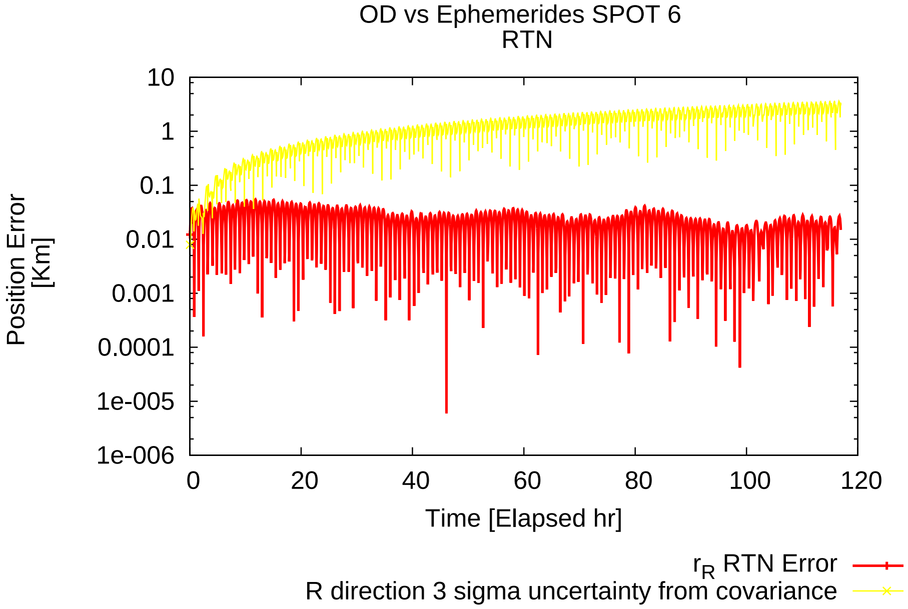


Figure 7.10: Position error in the R direction for the OD using WLS and synthetic measurements

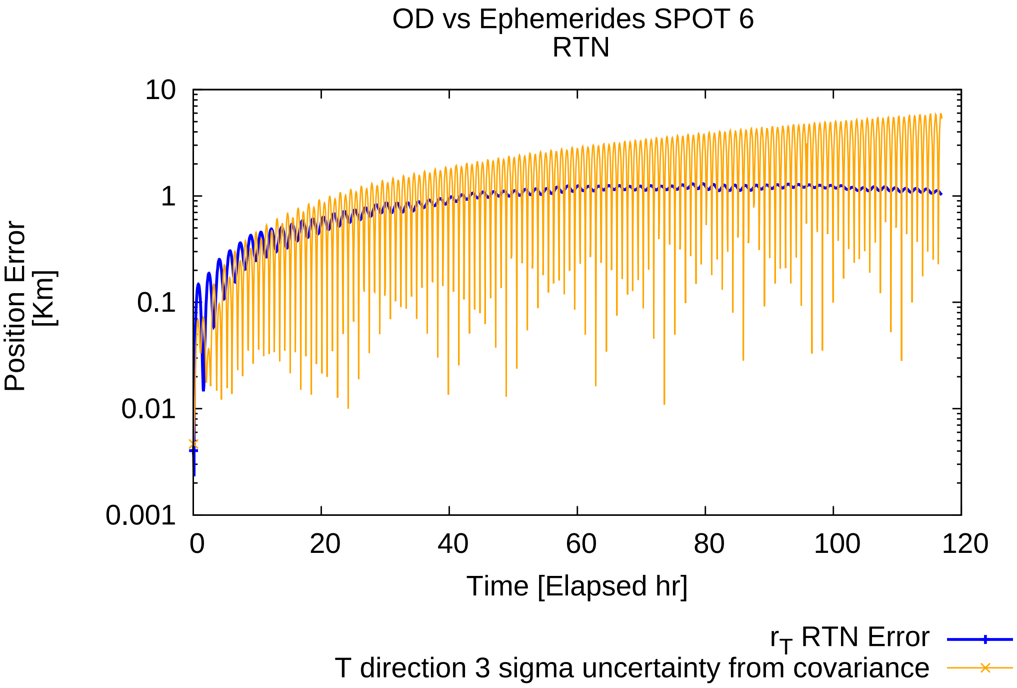


Figure 7.11: Position error in the T direction for the OD using WLS and synthetic measurements

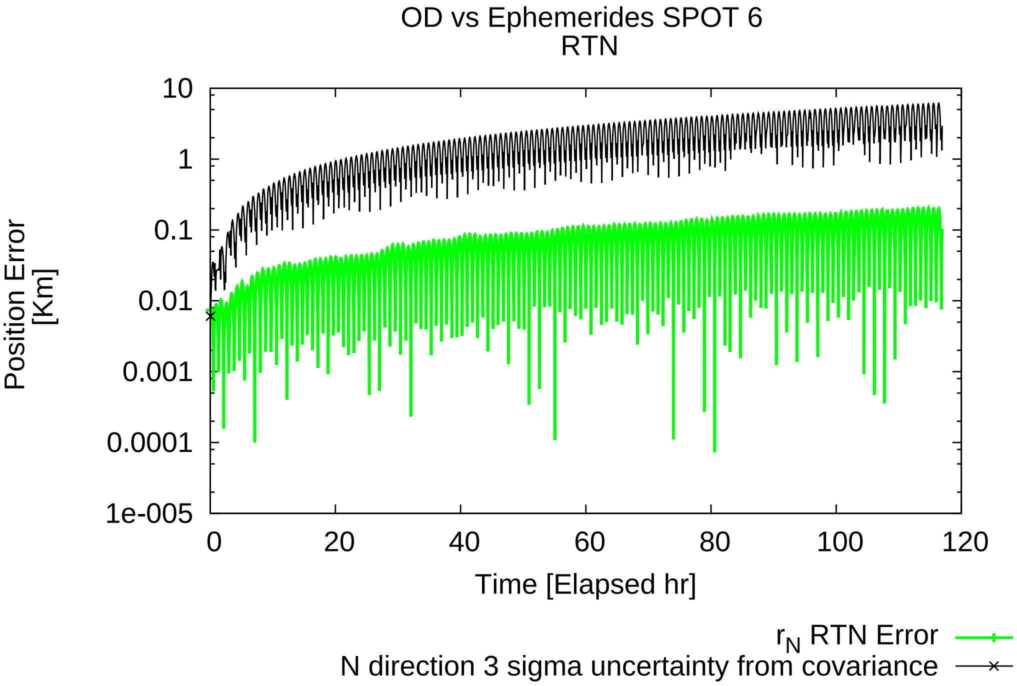


Figure 7.12: Position error in the N direction for the OD using WLS and synthetic measurements

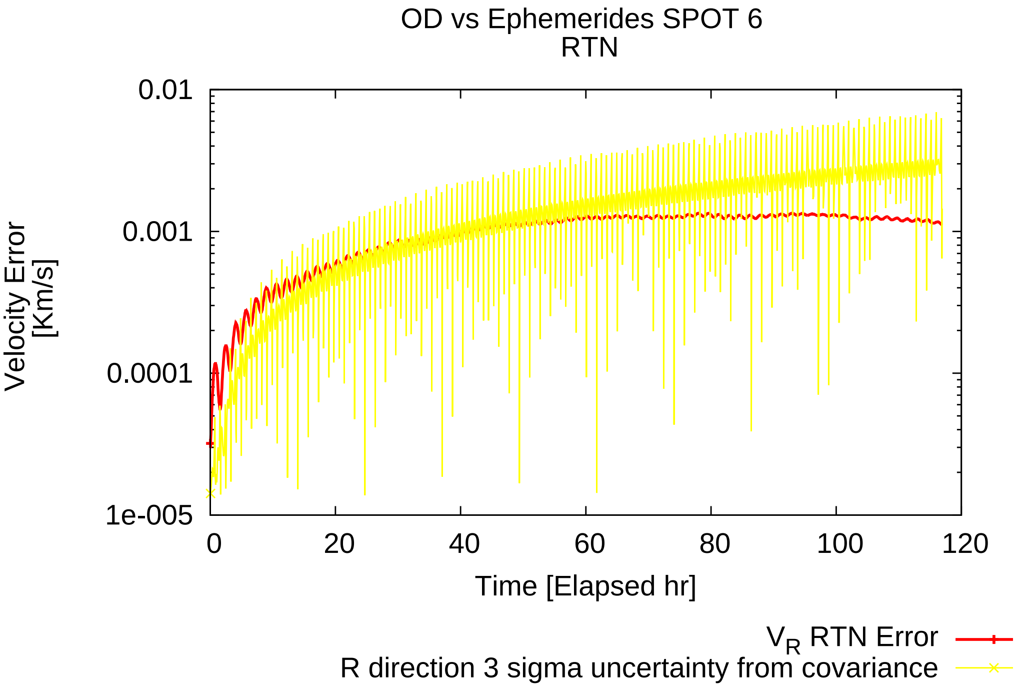


Figure 7.13: Velocity error in the R direction for the OD using WLS and synthetic measurements

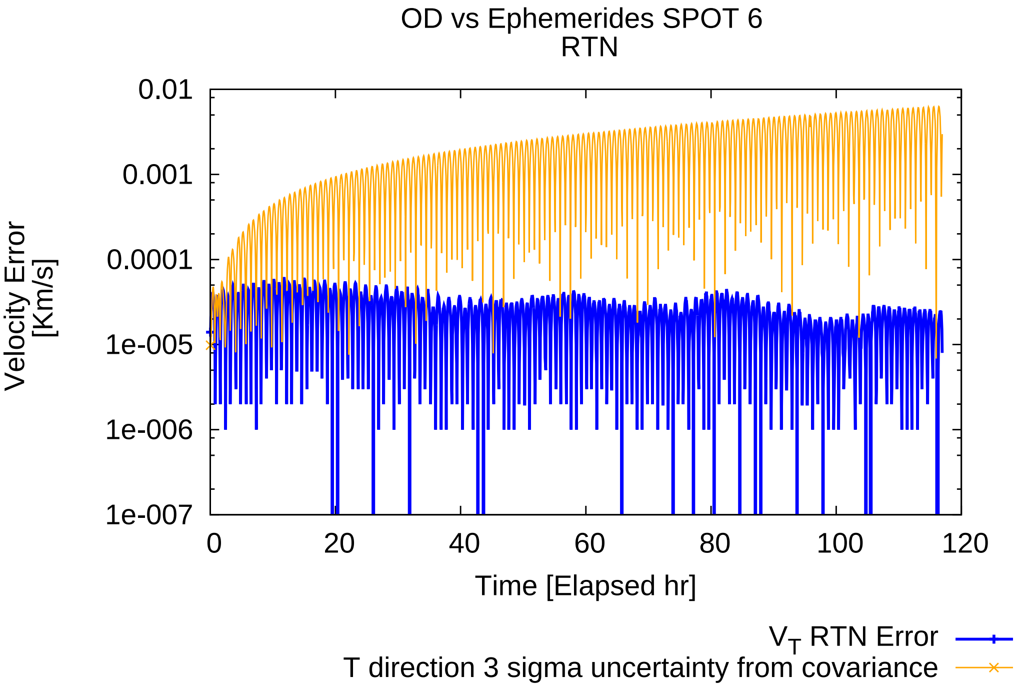


Figure 7.14: Velocity error in the T direction for the OD using WLS and synthetic measurements

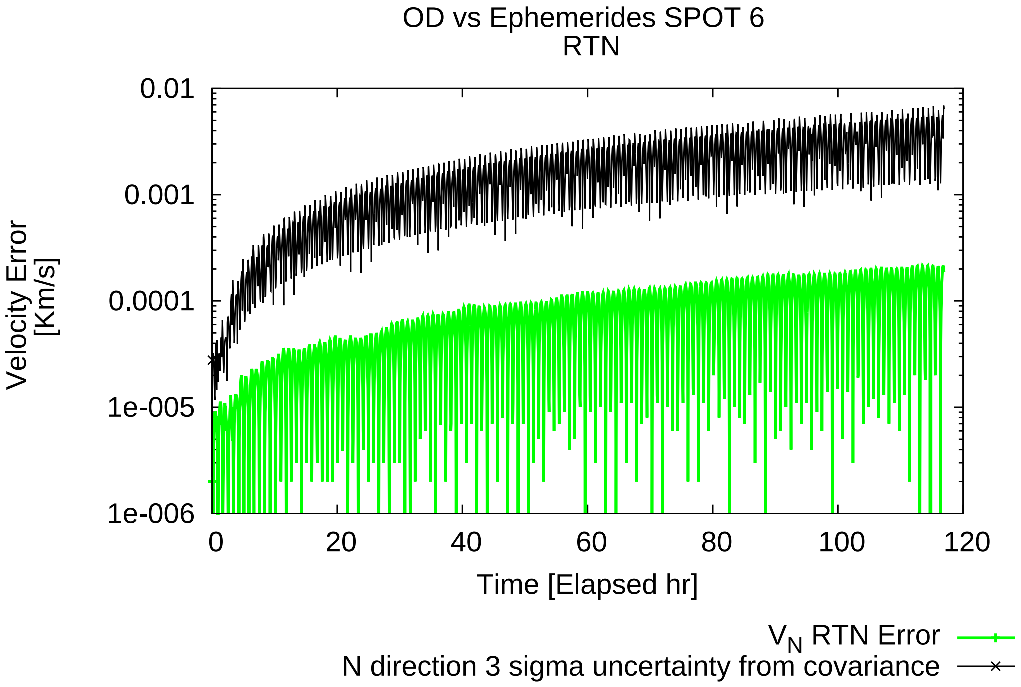


Figure 7.15: Velocity error in the R direction for the OD using WLS and synthetic measurements

7.5.2 Sequential Batched Least Squares

The Sequential Batched Least Squares method uses only portions of the data at each time called global tracklets and applies the same algorithm as the WLS method. Ideally, this global tracklets would correspond to one pass of the object above the detection zone of the station. This can be modified to include more than one tracklet in each subset of data.

If we try to perform an SBLS orbit determination using only one tracklet at each time, the process does not converge with the typical error parameters of the PFISR files. We need to use at least two different tracklets to make the process converge or introduce new measurements as angular measurements.

The results of the SBLS method using two tracklets at each time are shown in figures 7.16 to 7.21. It can be seen how the error decreases as more measurements are taken into account. However, in this case it can not be said that the uncertainty of the prediction is above the obtained error.

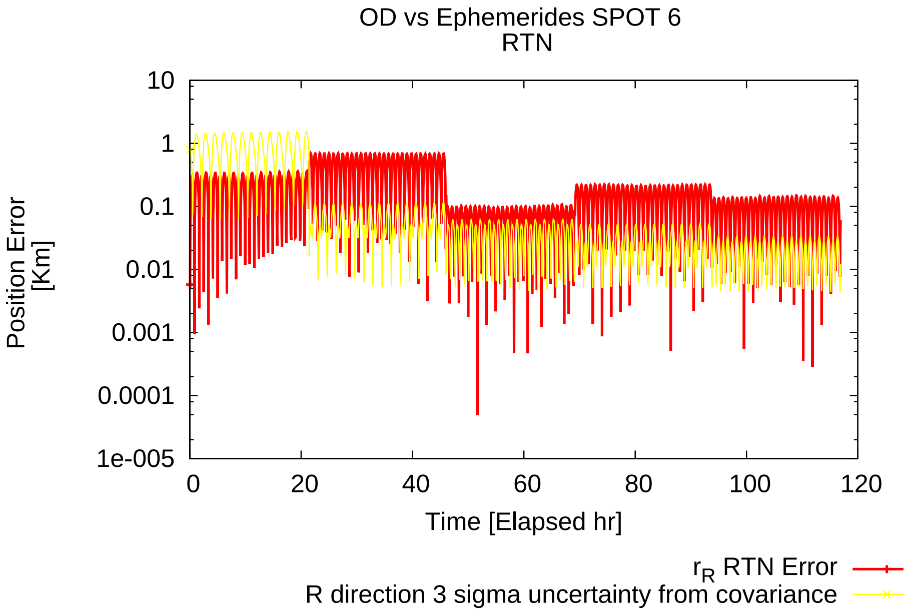


Figure 7.16: Position error in the R direction for the OD using SBLS and synthetic measurements

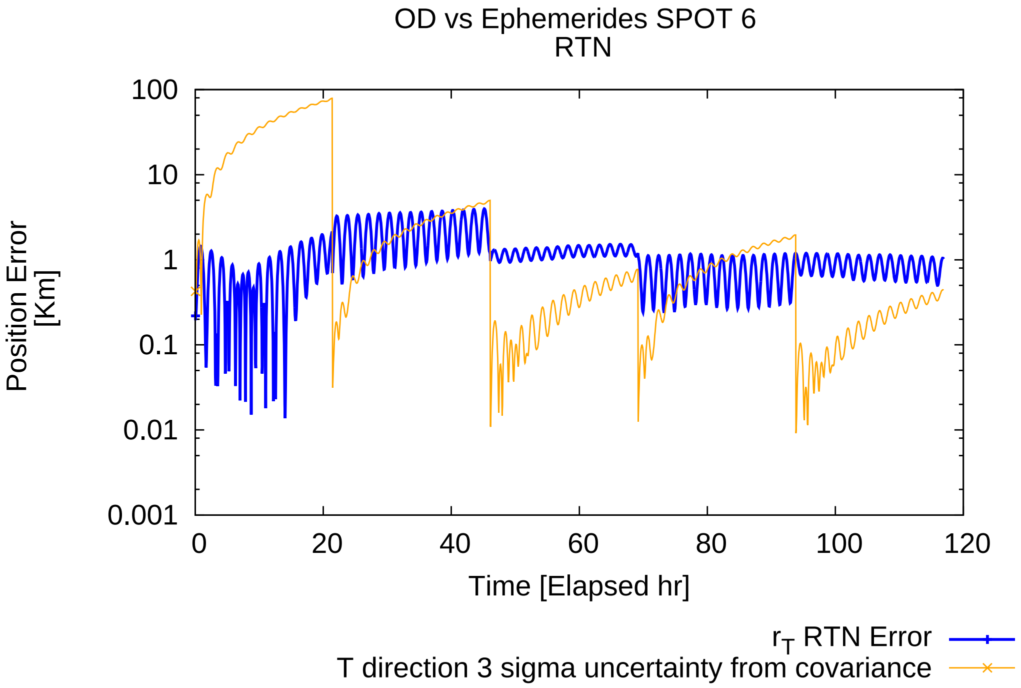


Figure 7.17: Position error in the T direction for the OD using WLS and synthetic measurements

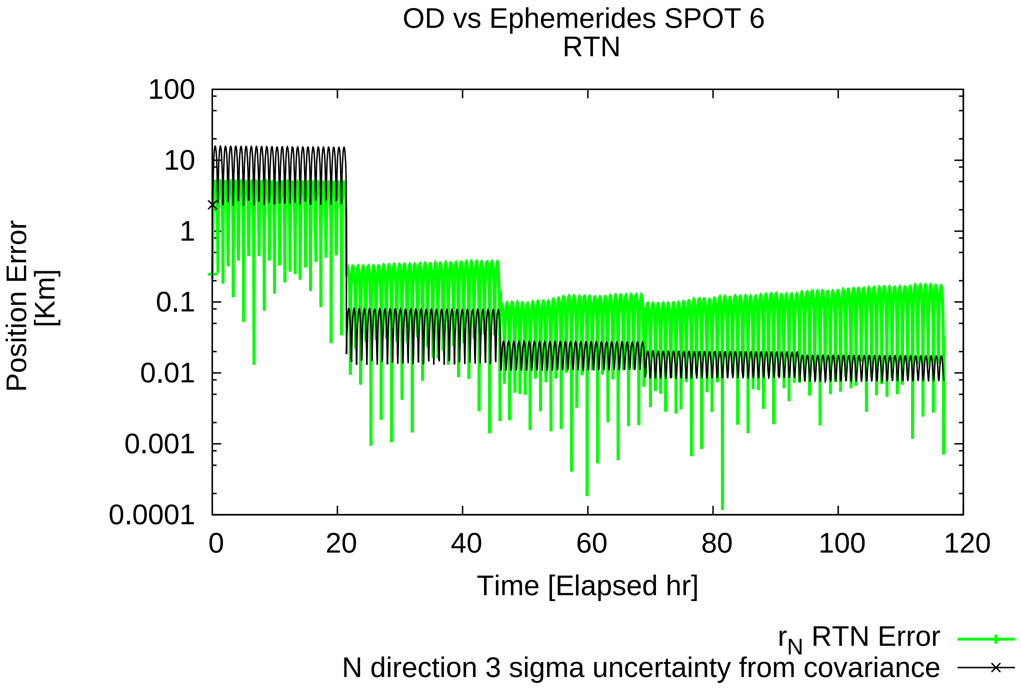


Figure 7.18: Position error in the N direction for the OD using SBLS and synthetic measurements

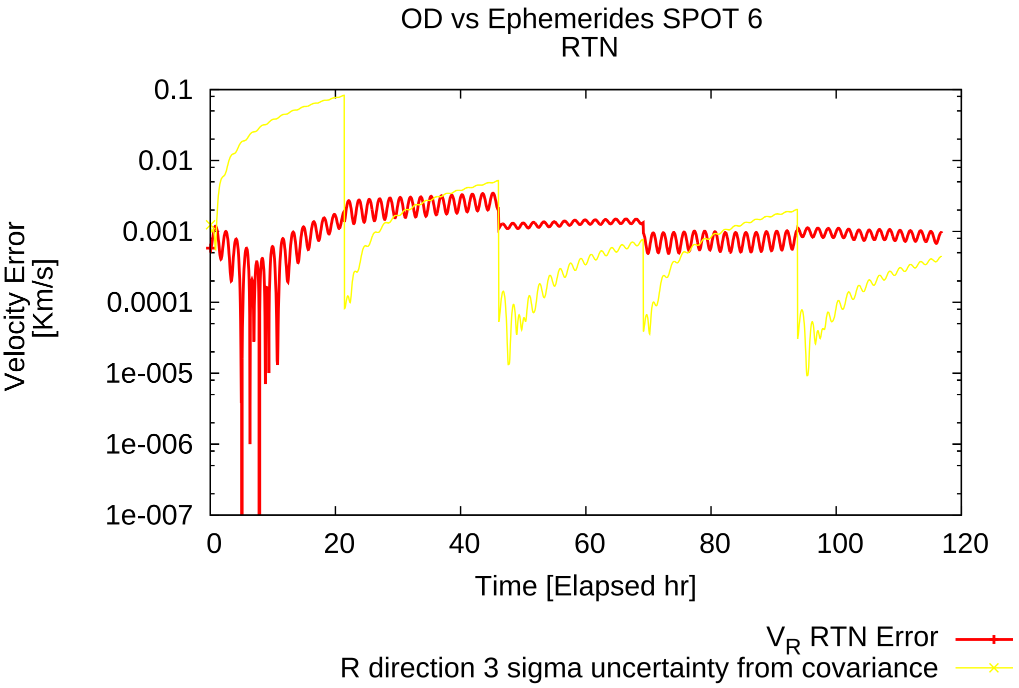


Figure 7.19: Velocity error in the R direction for the OD using SBLS and synthetic measurements

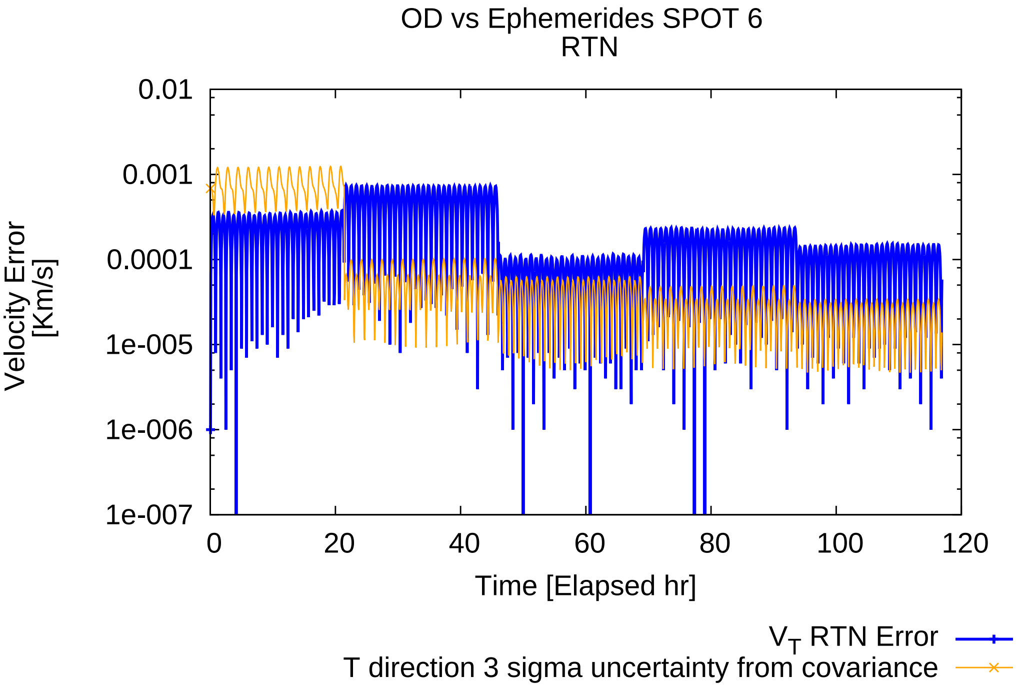


Figure 7.20: Velocity error in the T direction for the OD using WLS and synthetic measurements

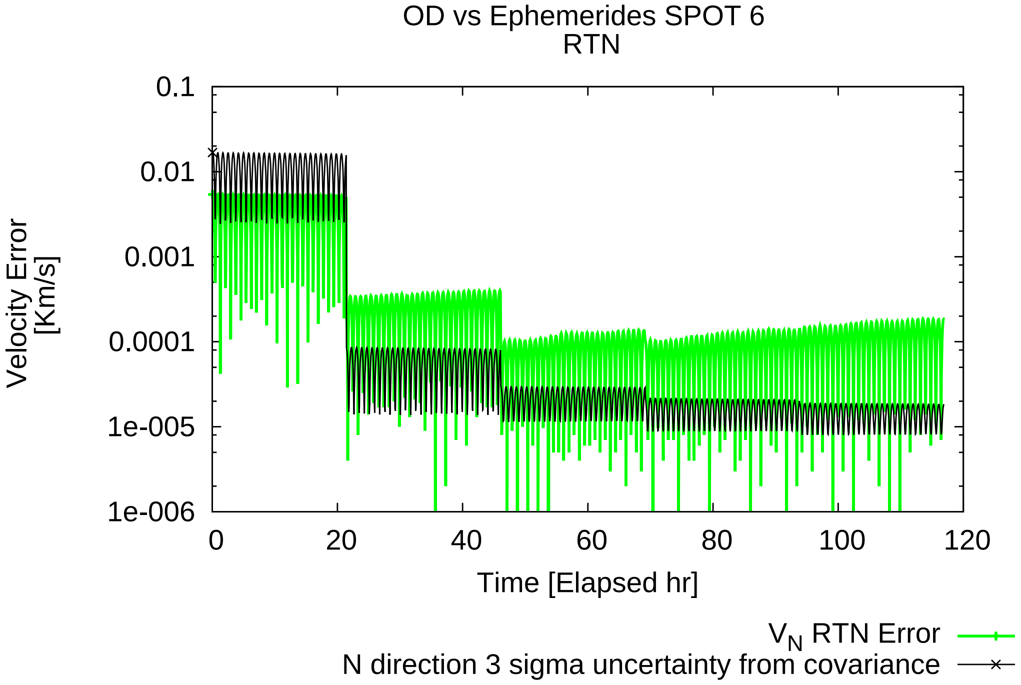


Figure 7.21: Velocity error in the R direction for the OD using SBLS and synthetic measurements

7.5.3 Comments to the Use of Synthetic Data

This section proves that an orbit determination is possible using data with similar characteristics as the PFISR data with the WLS and SBLS algorithms. However, this synthetic data would be the most favourable case, where the measurement errors follow a perfect Gaussian distribution (the same assumed by the OD methods) and their characteristics (bias and standard deviation) are defined without errors. Even in this best case, the orbit determination is not completely successful when compared with the available ephemerides, as the real error is not clearly below the 3σ uncertainty estimated from the covariance matrix during the whole simulation in the case of te SBLS algorithm and it continuously grows for the case. This gives and indication of the difficulty of performing a successful orbit determination based on only ranging data of one ground-based sensor with the real data, as its errors will not be perfectly Gaussian and its characteristics are only experimentally determined.

7.6 Orbit Determination using PFISR Data

The results of the orbit determination using both SBLS and WLS with the real PFISR data are shown here.

7.6.1 Weighted Least Squares

The WLS algorithm is the first method that has been tried with the real PFISR. As it uses all the available data, it is the most stable method and it should yield the best estimations.The results from the orbit determination are shown in figures 7.22 to 7.24 for position and 7.25 to 7.27 for velocity.

The dominating errors in the position are in the along-track direction or "T", and they are in the order of Km. The errors in the other directions are in the order of tens of meters. However, the predicted 3-σ error from the covariance matrix is usually below the error, indicating that, even if the algorithm converges towards a solution, those results could not be trusted for real collision avoidance applications.

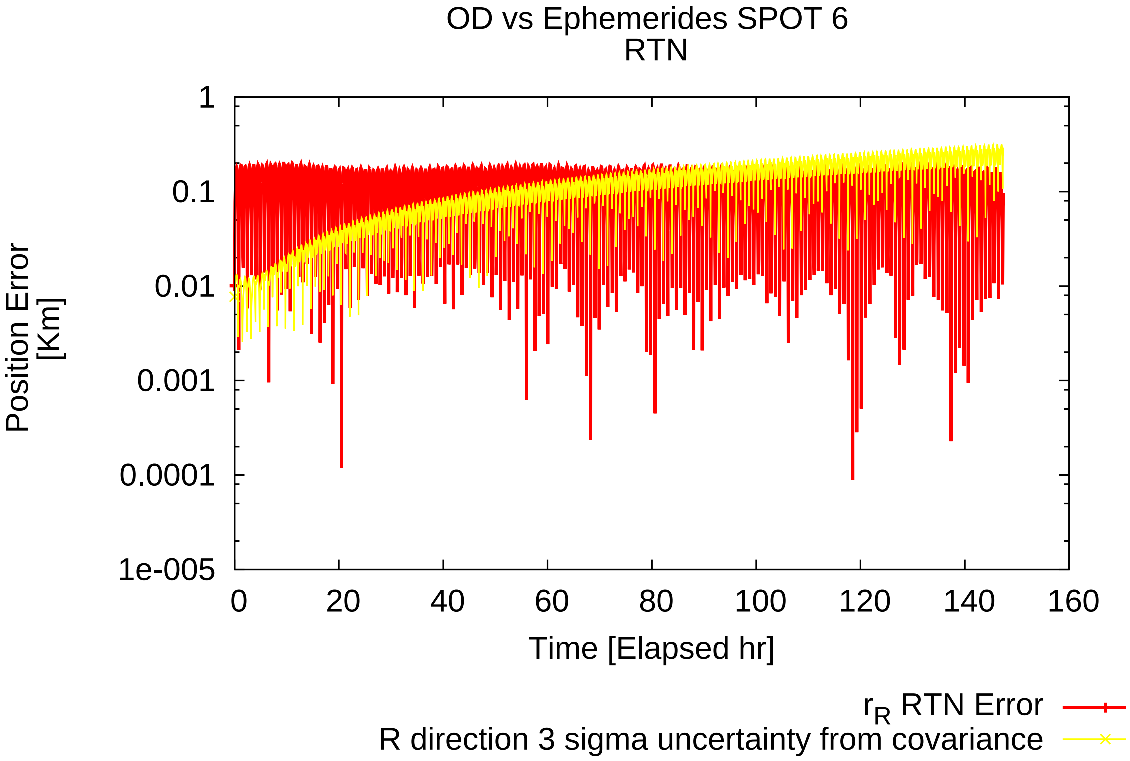


Figure 7.22: Results of the OD using WLS for the position in the R direction

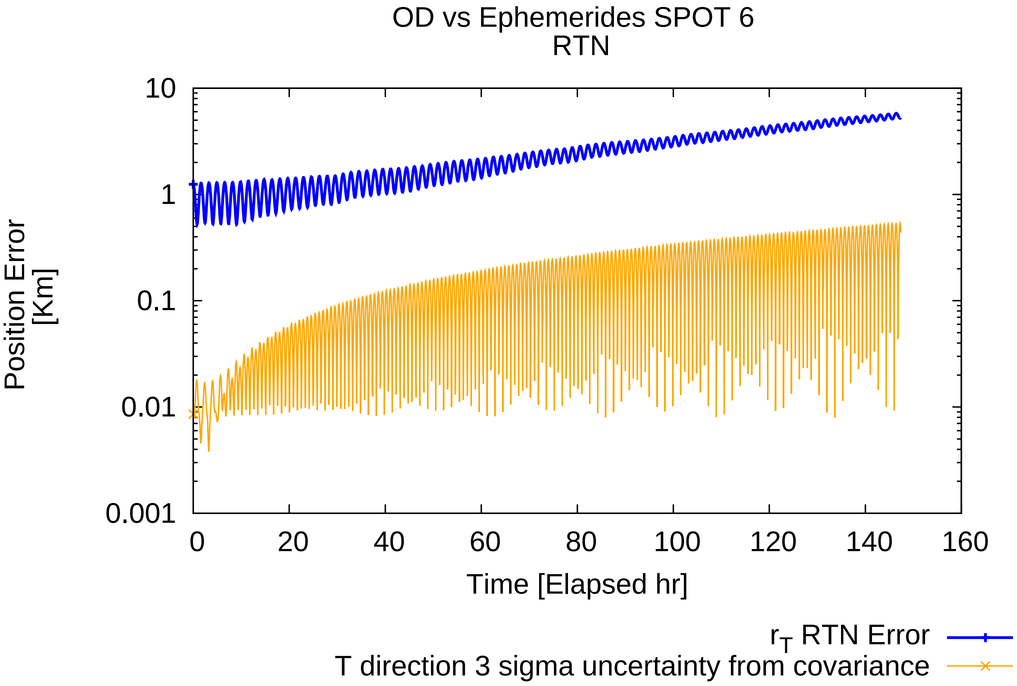


Figure 7.23: Results of the OD using WLS for the position in the T direction

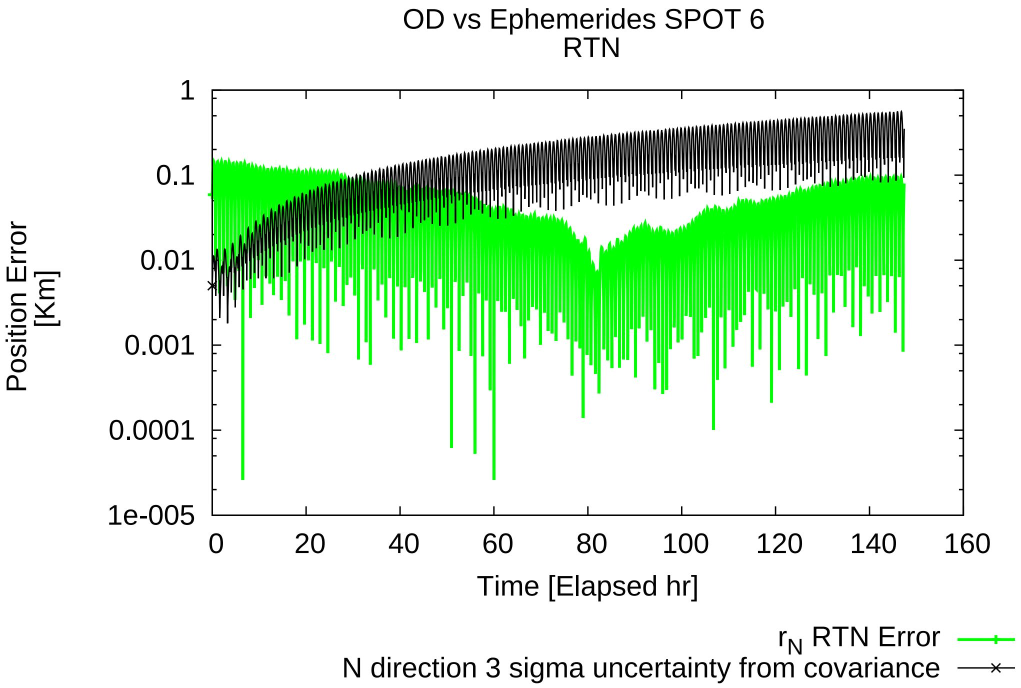


Figure 7.24: Results of the OD using WLS for the position in the N direction

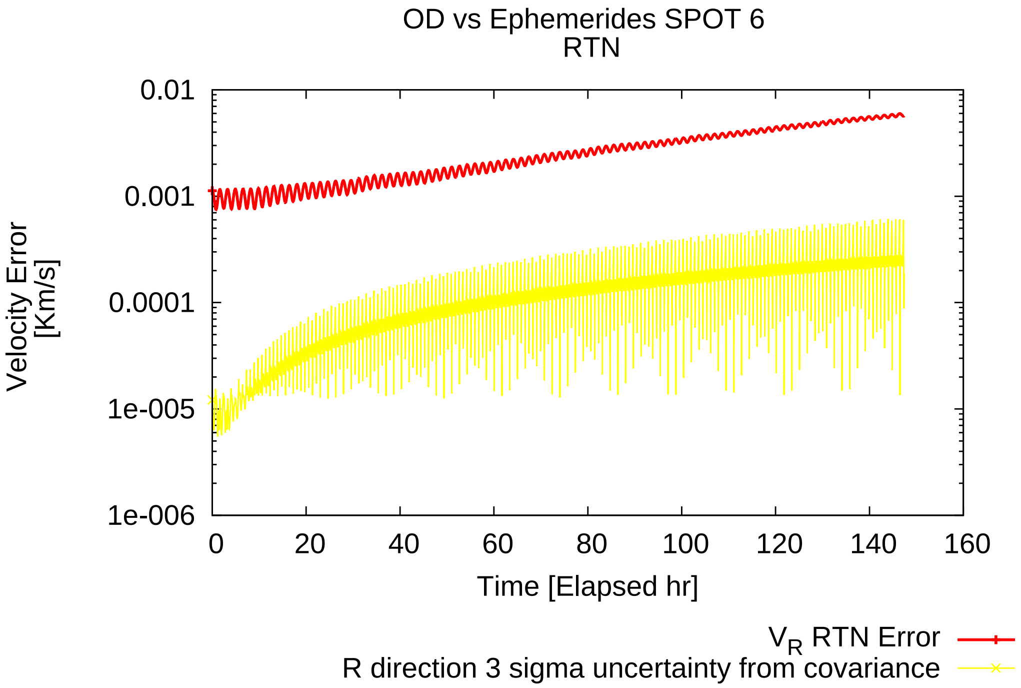


Figure 7.25: Results of the OD using WLS for the velocity in the R direction

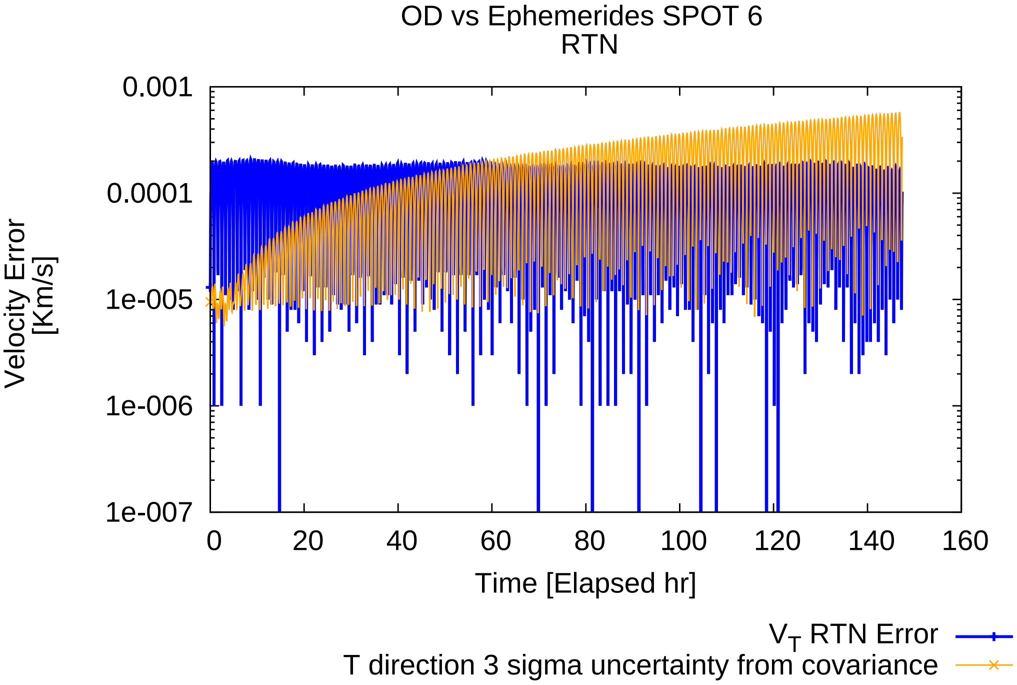


Figure 7.26: Results of the OD using WLS for the velocity in the T direction

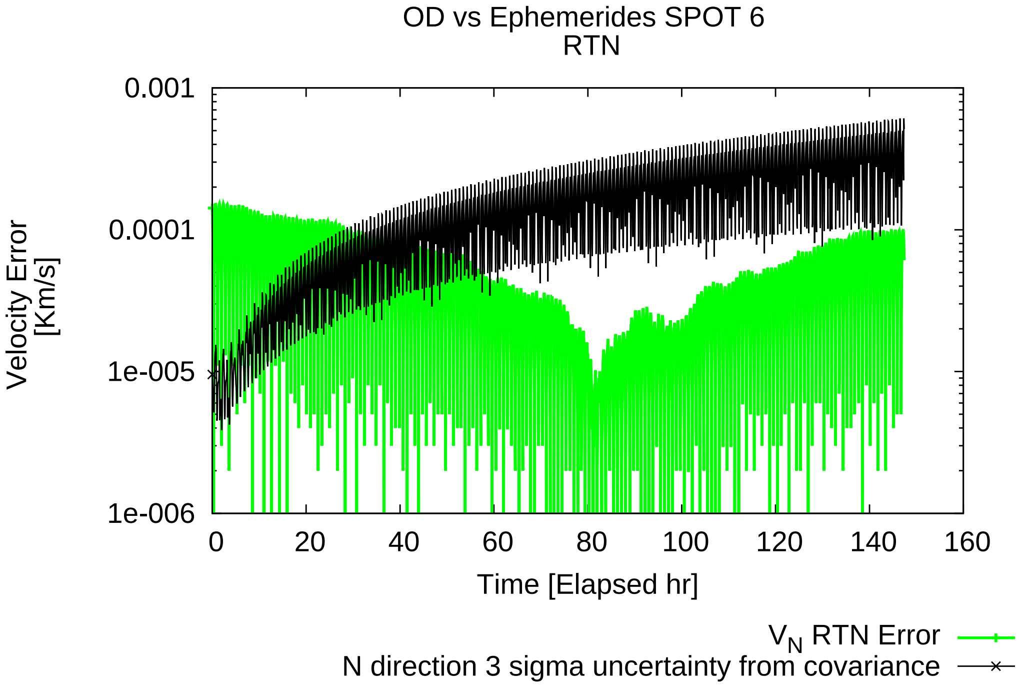


Figure 7.27: Results of the OD using WLS for the velocity in the N direction

7.6.2 Sequential Batched Least Squares

The results shown here have been obtained using the ephemerides files as the initial state and two tracklets into each WLS differential correction process.

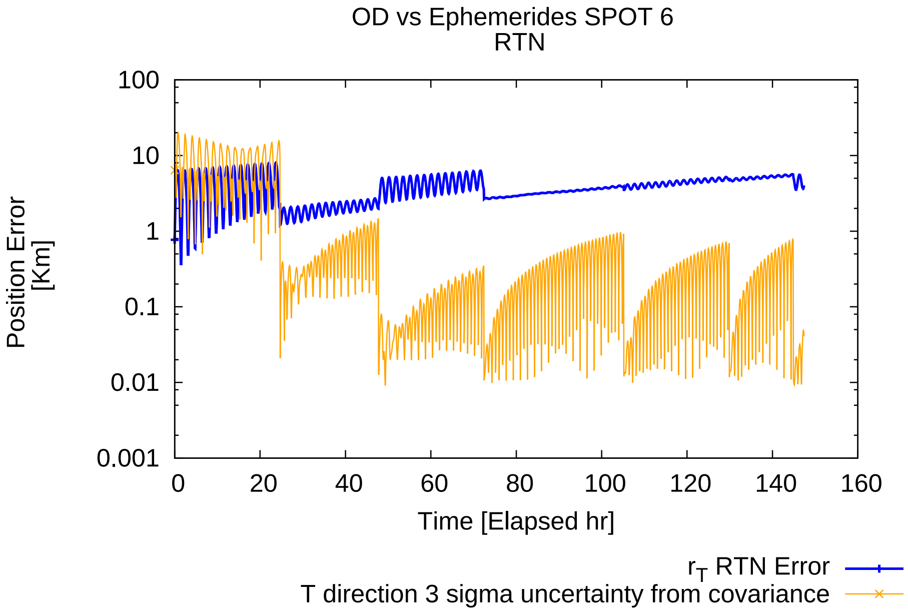


Figure 7.29: Results of the OD using SBLS for the position in the T direction

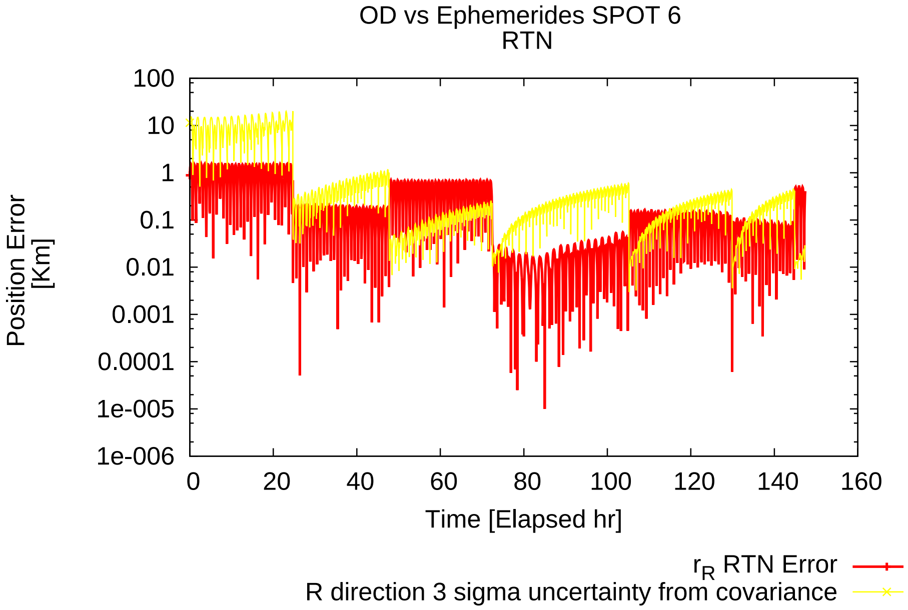


Figure 7.28: Results of the OD using SBLS for the position in the R direction

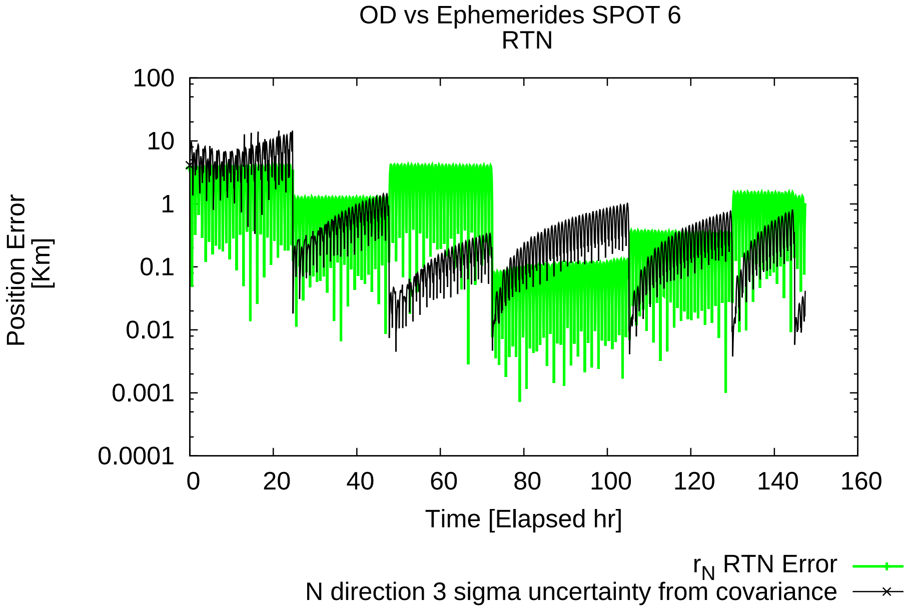


Figure 7.30: Results of the OD using SBLS for the position in the N direction

In this case, the obtained errors are bigger than with the WLS method. All the three components of the position error are in the order of Km while for the velocity are in the order of meters. It can still be seen how the bigger errors are concentrated in the tangential direction T, as in the previously analysed case.

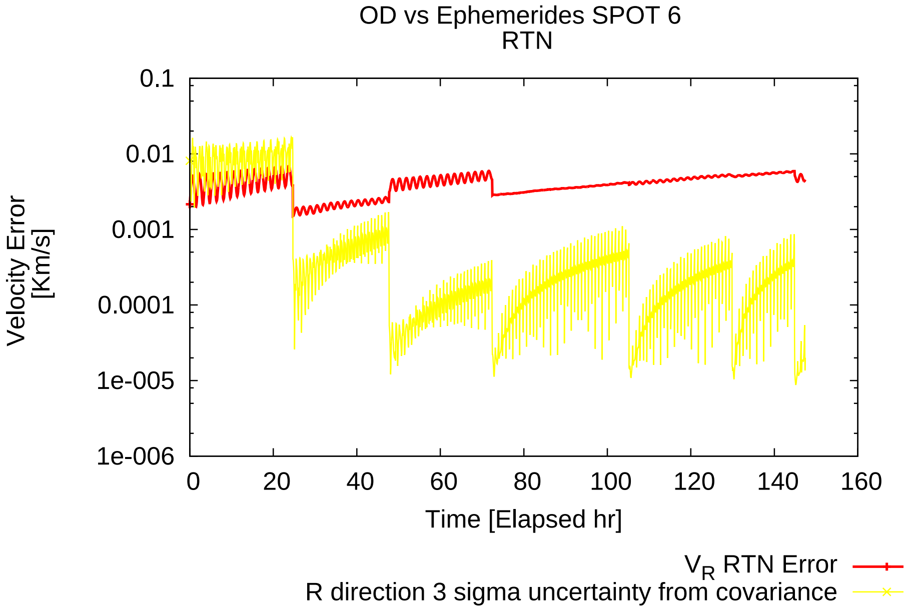


Figure 7.31: Results of the OD using SBLS for the velocity in the R direction

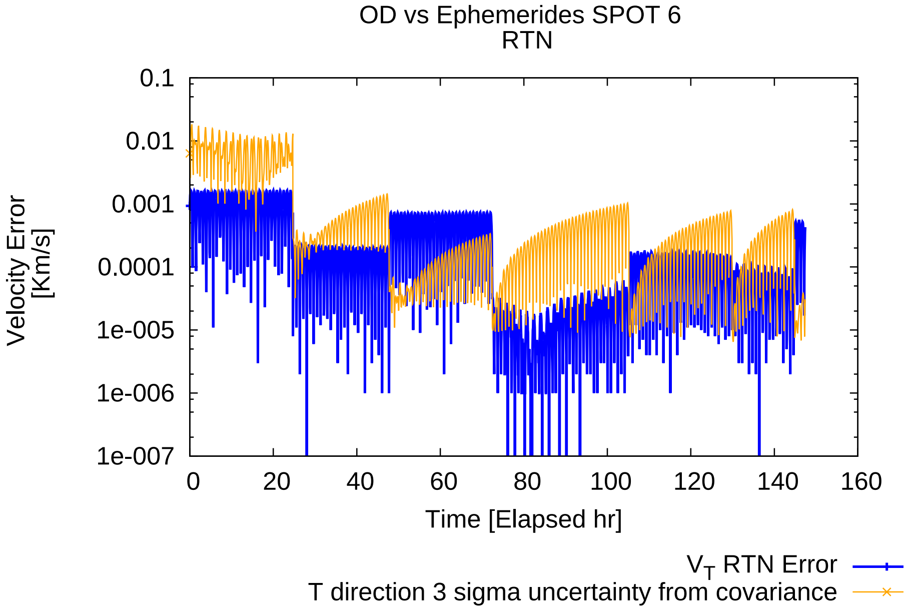


Figure 7.32: Results of the OD using SBLS for the velocity in the T direction

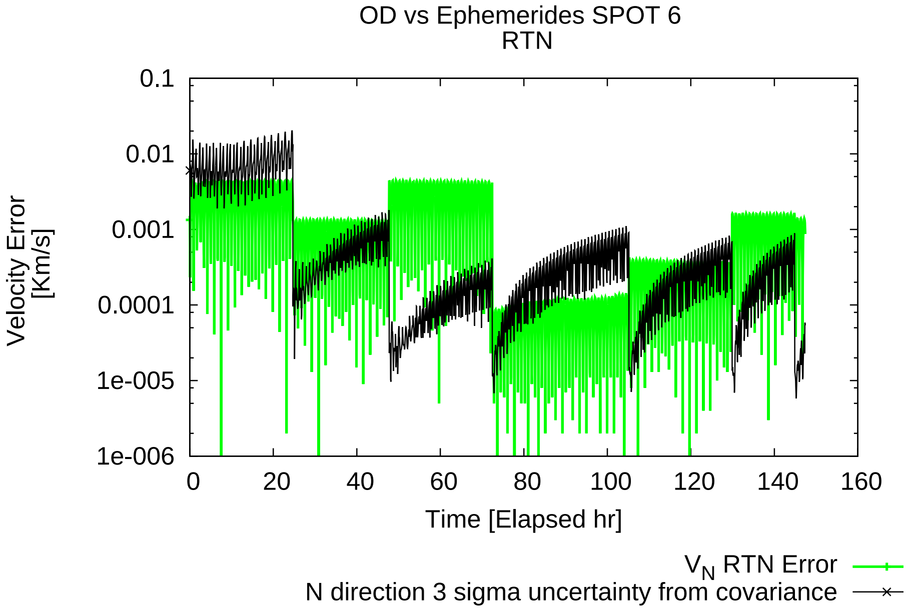


Figure 7.33: Results of the OD using SBLS for the velocity in the N direction

7.7 Conclusions

Orbit determination is possible using only range measurements using WLS or SBLS taking into account various tracklets at the same time Ephemerides files of SPOT 6 are used for validation purposes. Different files are available, and some discrepancies between the state vectors have been found in the order of units of Kms, similar as the errors obtained in the orbit determination. This is due to how the ephemerides were computed. This ephemerides are the result of performing one orbit determination per day to obtain an initial state vector of SPOT 6. After that initial point is obtained, the rest of the state vectors are the result of performing an orbit propagation during the following 8 days. Each ephemerides file uses an initial point computed in a different day. Below are plotted the differences in the SPOT 6 state vector between two of the available ephemerides files.

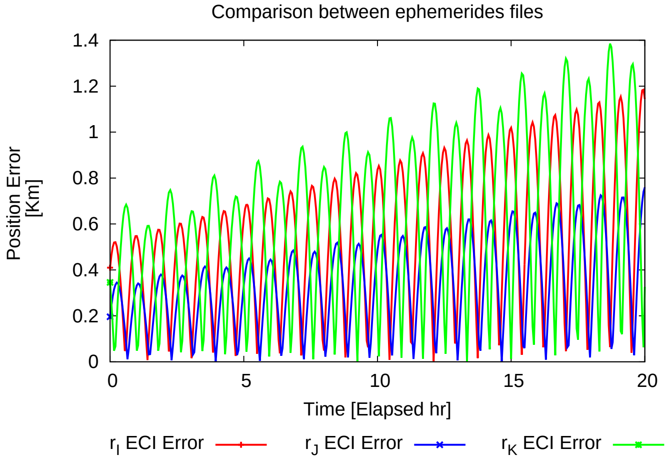


Figure 7.34: Position differences between ephemerides files

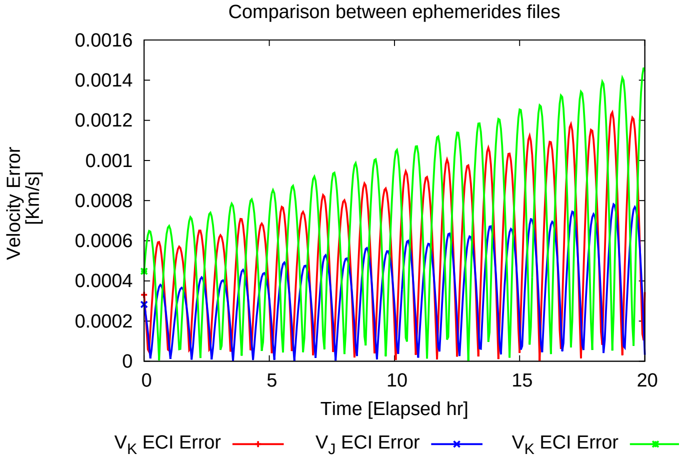


Figure 7.35: Velocity differences between ephemerides files

The results from the synthetic generated measurements and the real data are similar in terms of the prediction error. However, the predicted 3σ uncertainty can not be related to the obtained error. This can be explained as the experimental data is still in the development phase, and has not been used before to produce a successful orbit determination. The characteristics parameters of the data still have to be adjusted. Also, this data has not been yet been filtered and lacks some basic calibration such as atmospheric corrections. The lack of atmospheric corrections introduces a source of non Gaussian error, being one of the explanations behind the uncorrelation between the predicted uncertainty and the real error. Also, the distribution of the data concentrated in small periods of time (tracklets duration of around 2 minutes) and very sparse through time (around two tracklets per day) for SPOT 6 does not help the determination of the object's state. To achieve a better performance more observers could be used that avoid the big gaps between the tracklets. Also other measurements should be taken into account such as angular data. Regarding the filters, new covariance prediction algorithms could be implemented to take into account non-gaussian distributed errors.

Even with this considerations, it has been proven the capabilities of SPOOK to use real world data to perform orbit determination.

Chapter 8

Multiple Object Processing

8.1 Introduction

The previous version of SPOOK only allowed to process one object at each run of the program. One of the main objectives of the project was to enable the analysis of multiple debris objects in the same run of the program. Furthermore, to improve the efficiency of the code in terms of computational time the code has been parallelized using the OpenMP libraries [27] to take advantage of modern computer architectures.

To achieve this, no substantial changes had to be implemented in the way the algorithms that perform the different functionalities of SPOOK (observation simulation, orbit determination, propagation...) works. However, the way of how the different subroutines of the program interact between each other had to be modified. In the following, the changes with respect the code structure to incorporate the multiple objects processing feature are documented.

8.2 Sequential Processing of Multiple Objects

The first step of the development was to enable the software to process more than one object in one run in a sequential way. This was accomplished by reordering the structure of the code. In figure 8.1 it is represented a simplified work flow of how the code works. This structure is based on reading the configuration files at the beginning of the program and then perform the required operations according to the specifications of this files.

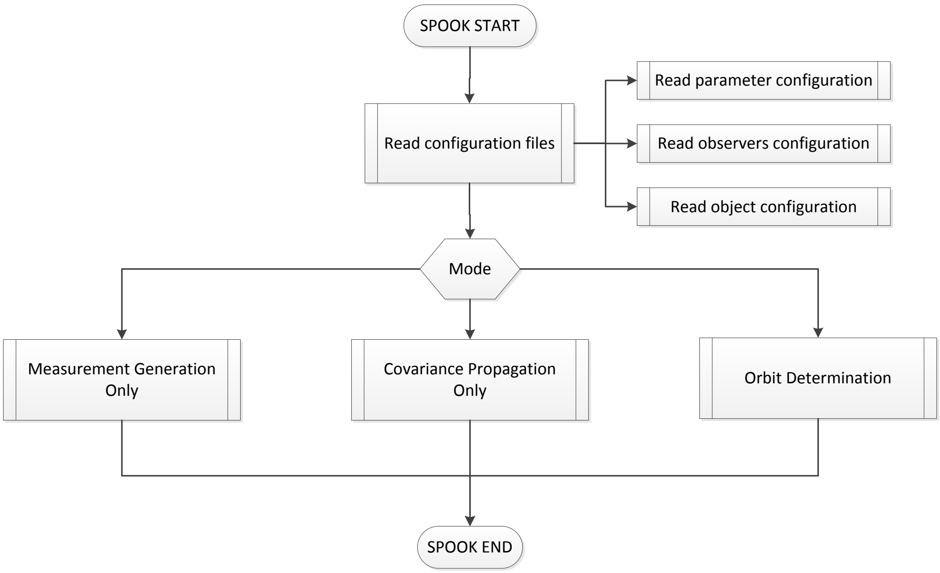

SPOOK START
Read configuration files
Read parameter configuration
Read observers configuration
Read object configuration
Mode
Measurement Generation Only
Covariance Propagation Only
Orbit Determination
SPOOK END

Figure 8.1: Top-level structure of SPOOK v3.0

In the implementation of the multiple object processing feature, it had been decided that the configuration parameters defined in the `parameters.ini` file as well as the observers defined in the observers configuration file will be the same for all the different objects. The different objects are defined in a single object configuration file. At the beginning of this file the total number of objects are specified followed by a block of information for each object containing the definition of all the different object related characteristics.

Taking into account the fact that the configuration parameters and the observers does not change for the different objects, the structure showed in figure 8.2 had been implemented in the newest version of SPOOK. This structure is based in reading first the observers and parameters configuration file and getting the total number of objects in the case. After that, the program sequentially reads the definition of the objects, accessing only the information relevant to the object being considered, and performs the different operations according to the selected SPOOK mode.

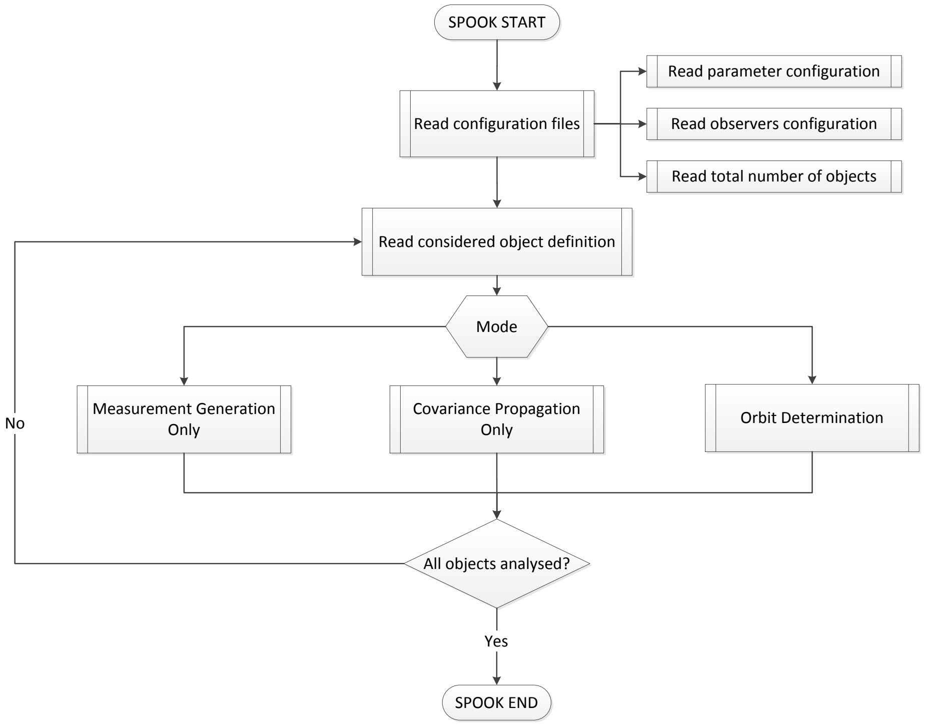

SPOOK START
Read configuration files
Read parameter configuration
Read observers configuration
Read total number of objects
Read considered object definition
Mode
Measurement Generation Only
Covariance Propagation Only
Orbit Determination
No
All objects analysed?
Yes
SPOOK END

Figure 8.2: Top-level structure of SPOOK with sequential processing of multiple objects

8.3 Batched Initialization of Objects via a Single TLE file

The multiple object processing feature can be combined with the object initialization via TLE files [28, Appendix B] capability present in SPOOK [20] to achieve a batched initialization of multiple objects. Space-track.org provides catalogues of space debris using this TLE format. These catalogues can include thousands of objects in a single TLE file, making it inefficient to create a single TLE for each one of the objects in order to use that information, as it was the case in the previous version of SPOOK.

Instead of this approach, multiple objects can be defined and read from a single TLE file and use this file as the single input for SPOOK. Certain information such as an initial covariance matrix or object's specific values such as the Solar Radiation Pressure Coefficient (SRPC) cannot be directly retrieved from the TLE files. These values will be defined in a general way for all the objects on the TLE file using the standard object definition format present in SPOOK. This feature can be enabled in the objects configuration file.

8.4 Software Parallelization using OpenMP

OpenMP is a specification for a set of compiler directives, library routines and environment variables that can be used to specify high-level parallelism in Fortran programs [29]. This specification has been used to parallelize SPOOK, reducing the computational times of large number of objects simulations. The parallelism has been achieved by assigning the computations related to each one of the processed object to a different thread of the machine. Figure 8.3 shows a graphic representation of how the program works in parallel mode. A maximum of N objects can be simultaneously processed, where N is the total number of available threads in the machine. If more than N objects need to be processed, the next object will be dynamically assigned to the first available free thread.

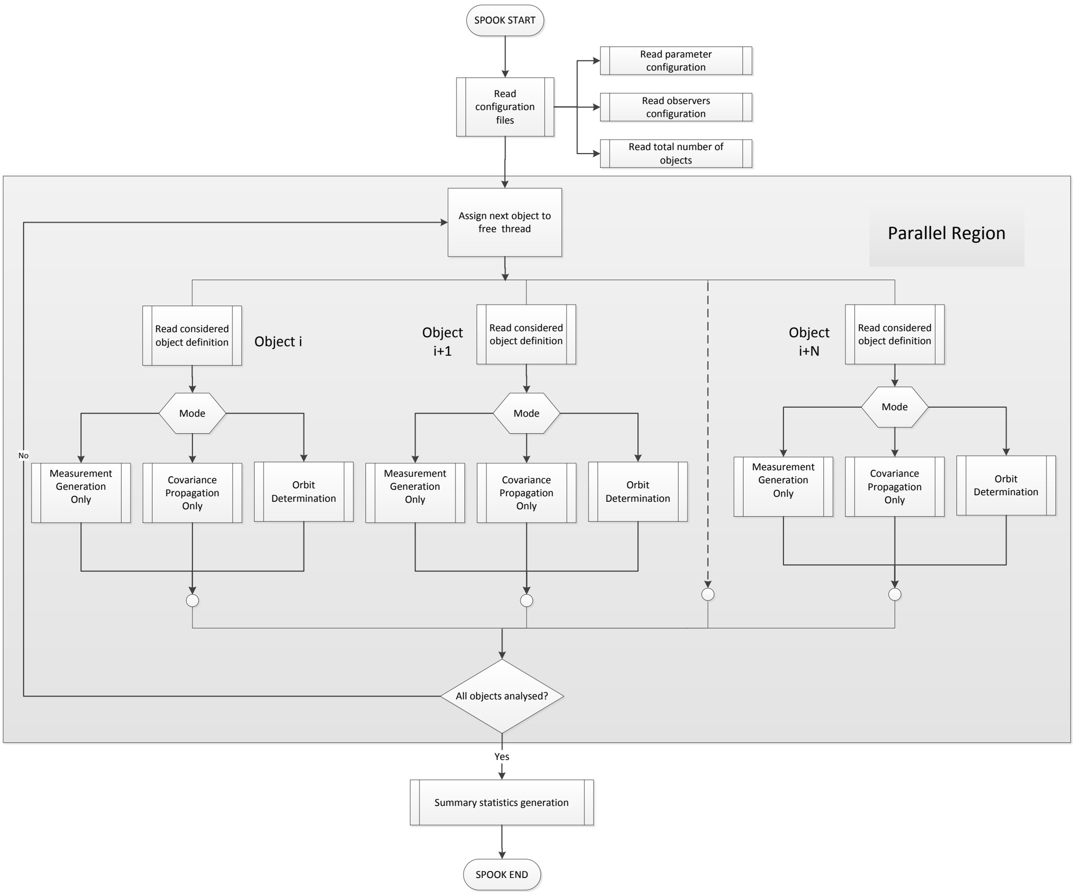

SPOOK START
Read configuration files
Read parameter configuration
Read observers configuration
Read total number of objects
Assign next object to free thread
Parallel Region
Read considered object definition
Object i
Object i+1
Object i+N
Mode
Measurement Generation Only
Covariance Propagation Only
Orbit Determination
No
All objects analysed?
Yes
Summary statistics generation
SPOOK END

Figure 8.3: Top-level structure of parallel SPOOK

The main issues regarding the parallelism implementation will be briefly commented in the following subsections.

8.4.1 Global Variables and the Legacy Code in SPOOK

Parallel programs are based in the use of two basic types of variables present in the main memory heap. Shared variables (that can be accessed from all the different threads) and private variables (each thread will have its own version of this variables) [27]. All the variables defined in the main program must have an assigned type of variable to achieve a correct behaviour of the program. In general, variables directly related to the target object (such as the measurements) will be of type private while more general variables such as the physical constants will be shared between the different threads. The local variables defined in each subroutine are stored in the stack memory and no extra considerations have to be taken into account regarding OpenMP parallelization.

SPOOK uses a series of legacy subroutines written in Fortran 77 which rely in the use of a set of global variables, usually grouped in the form of common blocks, defined inside different subroutines and not in the main program. These variables are stored in the heap memory instead of in the stack. OpenMP does not allow to treat variables not defined in the main program as private. These global variables are treated as shared between the different threads, causing incorrect results of the program.

Specific directives have been developed by the OpenMP Architecture Review Board (OpenMP ARB) to handle these variables. They can be defined as `threadprivate` variables, meaning that they will be shared (global) variables between the different subroutines but private to each active thread (each thread will have its own copy of this global variable). All the variables with the save attribute must be made threadprivate unless they are going to be shared between the different threads. The use of global variables is not advised by the newest Fortran standards and it is known to be a source of conflict in parallel programming, however the risks of changing the tested legacy code of SPOOK were bigger than using the `threadprivate` directive.

8.4.2 Parallel Input/Output

Reading from and writing to external files must be handled with care in parallel programming. Multiple access to a single file is possible in read only mode while not possible for writing. Unique unit identifiers must be defined to assure that different threads are not trying to access to the same file at the same time. This is achieved by defining two global variables, one for input files (`InUnit`) and one for output files (`OutUnit`). These variables contain a series of unique unit identifiers starting from 7 (5 and 6 are the default I/O units [30]). These variables will have as many columns as the maximum number of threads available in the machine executing the code and as many rows as files can be open simultaneously to read/write. Each one of the threads will only be available to access the column corresponding to its thread number. In this way, it is assured that every file opened in parallel for reading/writing will have a unique unit identifier. An example of this algorithm is shown below for a machine with only 4 threads and a maximum of two files opened at the same time for both writing and reading.

$$\texttt{InUnit} = \begin{bmatrix} 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 \end{bmatrix} \tag{8.1}$$

$$\texttt{OutUnit} = \begin{bmatrix} 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 \end{bmatrix} \tag{8.2}$$

In this case, thread number 0 would only be able to use for reading the unit identifiers 7 and 11 and for writing the identifiers 15 and 19.

Messages written in the screen must also be handled with care. The sequential version of SPOOK continuously wrote on screen information about the tasks being performed (name of the task, progress bar, etc). When a large number of objects is being processed, this information ceases to be useful. Besides that, due to the limitations of the output terminal, the messages relative to the different objects would be mixed together. For that reason, when parallel processing is enabled the output messages on the screen are suppressed. Instead of that, only one message per object will be printed on the screen. This message will state the object being processed and the thread processing it. This will serve as an indication of the progress of the simulation. And example of the output message of parallel SPOOK can be seen in figure 8.4.

```
#----------------------------------------------------------------------#
# Processing object number        4 of      250                        #
# Running on thread number       12                                    #
#----------------------------------------------------------------------#
```

Figure 8.4: Example of the output message written in the screen of SPOOK in parallel mode

8.4.3 Error Handling

Error handling must be also modified for parallel processing. In the single object version of SPOOK, any kind of error that would prevent the program to reach the desired results (such and integration error, ill-defined parameters of the object, etc) was handled following the same structure:

1. Error identification.
2. Write error message on the screen.
3. Pause the program to read the screen (optional, only if *debug mode* is selected).

4. Program ends.

This structure meant that the program would always abort in case a fatal error was encountered as no other operations remained to be done. This structure can not be used anymore for parallel processing. In an example simulation of 10000 objects, using this way of handling errors would mean that an error in the first object would prevent the other 9999 objects from being processed.

Errors encountered processing one object should not affect the rest of the objects of the simulation. Instead, if an error was encountered in one of the objects, it should be recorded for future analysis and the next object in line should be processed. To implement this approach an `Error_log.dat` output file is created. This file will record the information relative to the error, the object and the subroutine that caused it. An example of this `Error_log.dat` file can be seen in appendix A.1.

After the error has been recorded in the `Error_log.dat`, an error flag, private to the thread processing the object, will be raised. This flag will indicate to the active nested subroutines to return back to the main program without performing any extra operations in order to process the next object in line. This idea is graphically expressed in figure 8.5, where the green line represents the normal return path and the red line represents the return path when an error is encountered. The dashed line indicates that other subroutines can be present in the middle of the path.

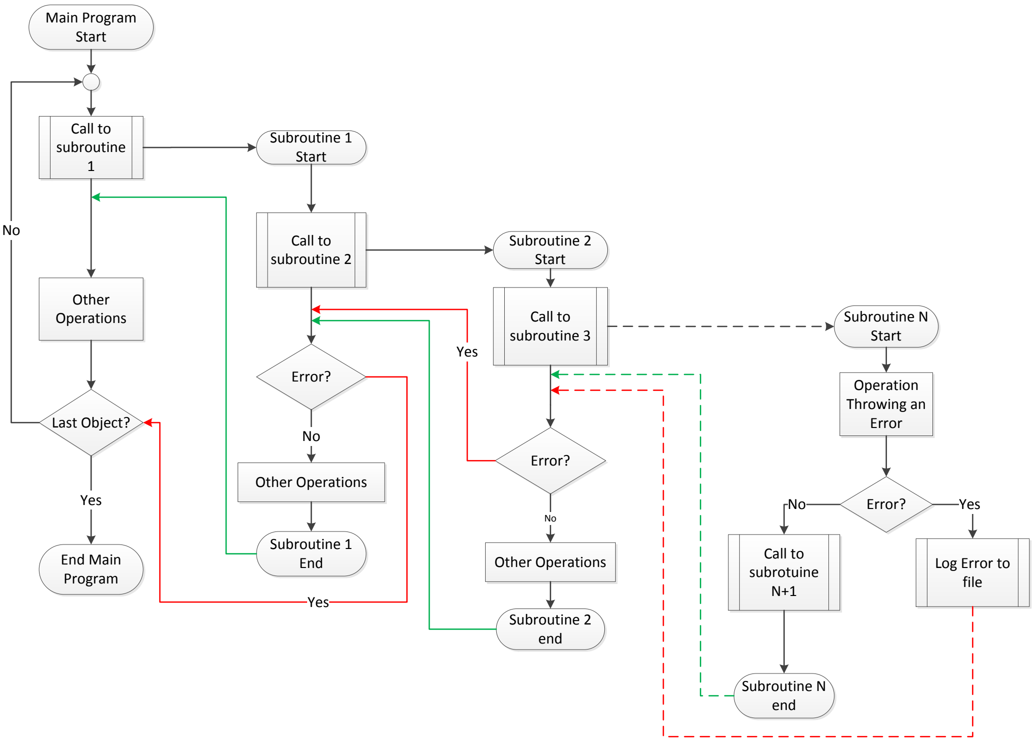


Figure 8.5: Error handling structure of parallel SPOOK

8.4.4 Scaling Test

To asses the performance of the code working in parallel, a scaling test will be carried out. A simulation scenario will be defined (an observation strategy with multiple objects). This scenario will be run multiple times, varying the number of threads used for each run. To asses the performance, the total simulation time of each run will be compared.

The observation strategy chosen is the Geostationary Orbit Fence scenario, based on a study carried out by Airbus Defence & Space [31] to achieve maximum observation coverage for Geostationary Orbit (GEO) objects with only one optical observer. This scenario is based on the use of a space based observer flying in a dawn-dusk LEO. Extra information about this scenario can be found in [10, 20]. To assure the maximum repeatability between the different runs all the objects will have the same characteristics. Also, the random number generator (used to randomize the simulated measurements following a Gaussian distribution) will be controlled to yield the same measurements for all the different objects. The object being selected follows a Geosynchronous orbit with the orbital parameters shown in table 8.1. The simulation start epoch is: 06/04/2014 at 06h 34' 35".

a	e	i	Ω	ω	ν
[Km]	[-]	[deg]	[deg]	[deg]	[deg]
42165	$2.5 \cdot 10^{-4}$	1	60	0	180

Table 8.1: Orbital parameters of the object used for the scaling test

The machine selected to perform the scaling test has a maximum of 36 different cores. Each core can run a single thread. For this reason, 36 objects will be simulated in each run. The number of threads used in each run will vary from 1 (sequential run) to 36 threads (each object will run in an independent thread). The results of this scaling test can be seen in table 8.2 and figure 8.6

Threads	Execution Time
[-]	[s]
1	754.09
4	247.40
8	120.98
16	81.90
32	70.50
36	54.33

Table 8.2: Results of the scaling test

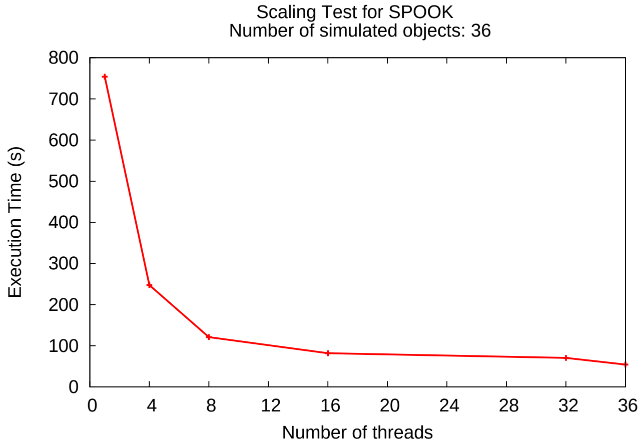


Figure 8.6: Scaling test

It can be seen how the total execution time drastically decreases from almost 755 s in the sequential mode to less than 55 seconds (a total reduction of 92.8% os simulation time). This is achieved for a modest case where only 36 objects where simulated. In a more realistic case, where a whole space debris catalogue could be used (around 17000 objects), time reduction is key factor for quick analysis of surveillance strategies.

8.5 Coverage Analysis Mode

The multiple object processing feature enhances the capabilities of SPOOK to perform the analysis of different observation strategies. Instead of testing the observation strategy against a unique object, which would have to be representative of the kind of objects to be tracked, the entire known debris population can be simulated. However, with no further changes to the code, the amount of data produced by such a large simulation (the U.S space Surveillance Network currently catalogues more than 17000) would make its analysis infeasible. For that reason, a postprocessing feature has also been added in the scope of this thesis. This postprocessing will define SPOOK's Coverage Analysis Mode, where simplified outputs will be produced with relevant statistics about the performance of the observation strategy. The aim of this mode is, instead of obtaining a data file with the predicted state vector and covariance matrix at each instant of time per object, produce only one line per object with relevant statistics. If wished, the detailed output can still be produced. The next sections will describe the newly added postprocessing mode and the generated outputs.

8.5.1 Orbit Determination Convergence Criterion

One of the main uses of SPOOK is to test different observation strategies to catalogue space debris. One object can be catalogued with a particular observation strategy if the orbit can be successfully determined with the generated observations.

So far, in the single object version of SPOOK the success of the orbit determination process was individually analysed using the information of the output files. The 3σ errors from covariance and the real errors (when synthetic measurements were used) in the radial, cross-track and along-track were plotted during the whole simulation time. By visual inspection of this graph, it could be assessed if the Orbit Determination had been successful or if the prediction of the position was not realistic enough.

This evaluation method is not efficient when large numbers of space objects are simulated (the space debris population is in the order of hundreds of thousands of objects). For that reason, a convergence criterion for the orbit determination must be developed.

Focusing on the information related only to the satellite position produced by SPOOK, we have a three component vector of the estimated position $\vec{r}_{estimated}$, a three component vector of the real position $\vec{r}_{real}$ and a three by three symmetric covariance matrix of the position P_r. These sum in total 12 parameters that have to be analysed for each simulation time instant(3 from each position vector and 6 from the covariance matrix, as it is symmetric). A metric must be developed that converts these large number of parameters into a binary check (success or not success).

From the estimation theory, it is known that the predicted position does not locate the object, but the most probable position of the object [32]. The covariance matrix P_r defines the 1-σ error ellipsoid centred in the predicted position, giving an indication of how precise the knowledge of the position is. The dimensions of the axes of this ellipsoid are the square root of the eigenvalues of P_r ($\hat{\lambda}_i$), while its orientation is defined by the eigenvectors [32]. In the 3D case, the 1-σ ellipsoid defines the region where the estimated variable has a 19.872% probability of being. If instead the 1-σ we use the 3-σ ellipsoid (by appropriate scaling of the axis) the probability increases up to 97.071%.

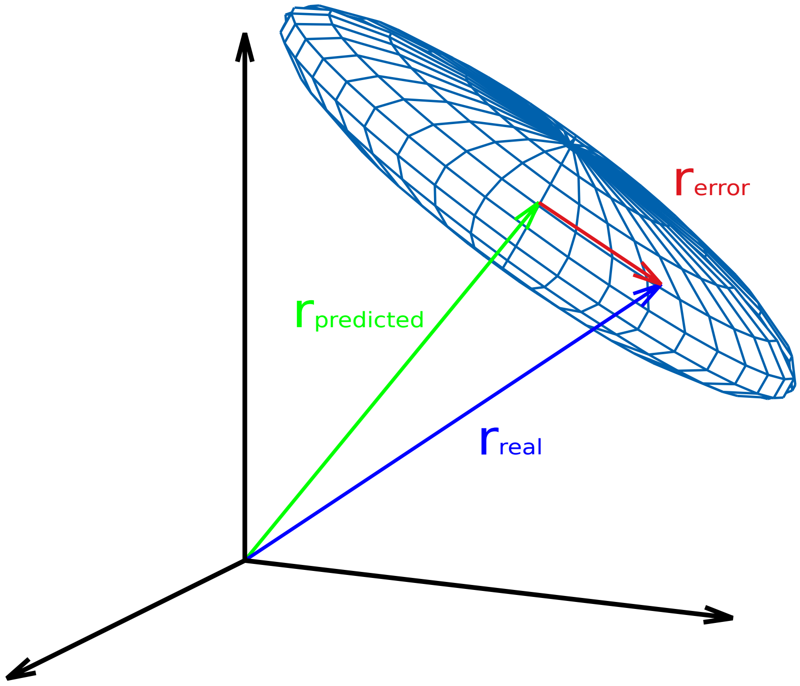


Figure 8.7: The error ellipsoid

By using the predicted position (in green in figure 8.7) and the real position (in blue), we can compute the error of the prediction (red). If this error lies inside the 3σ error ellipsoid, we know that the error predicted by the orbit determination is in line with the real error. If the error is outside the ellipsoid, it means that the estimated error is smaller than the real one and the position is not correctly estimated. This idea is graphically represented in figure 8.7.

To implement this criterion, we must first compute the eigenvalues $\hat{\lambda}_i$ and the eigenvectors $\vec{\psi}_i$ of the covariance matrix P_r.

$$P_r \longrightarrow \hat{\lambda}_i \quad , \quad \vec{\psi}_i \tag{8.3}$$

The eigenvectors $\vec{\psi}_i$ will define the rotation matrix Ψ between the ECI frame and the ellipsoid principal axis (EPA). Using this matrix the error $\vec{e}$ in the ellipsoid principal axis can be computed

$$\{\vec{e}\}_{EPA} = \Psi^T \left(\vec{r}_{real_{ECI}} - \vec{r}_{pred_{ECI}}\right) \tag{8.4}$$

With the ellipsoid equation expressed in its principal axis 8.5, it can be checked if the error lies outside(>) or inside (⩽) the predicted error.

$$\left(\frac{\vec{e}_1}{3\sqrt{\lambda_1}}\right)^2 + \left(\frac{\vec{e}_2}{3\sqrt{\lambda_2}}\right)^2 + \left(\frac{\vec{e}_3}{3\sqrt{\lambda_3}}\right)^2 \lesseqgtr 1 \tag{8.5}$$

This check has to be done for each instant of time where the position is evaluated. To reduce that to a simply binary check, we compute the fraction of the simulation where the error lies inside the 3σ ellipsoid. Ideally, we would like that the error lied inside the ellipsoid during the whole simulation (100%). However, it is possible that the first predictions of the orbit are not precise enough due to the lack of enough data, converging towards a good solution when more observations are taken into account. To avoid categorizing that as unsuccessful orbit determination, a fraction level is established via the configuration files (e.g. 90%). If the fraction of the simulation that the error lies inside the 3σ ellipsoid is higher than the established level, the orbit determination is assumed to be successful. This criterion can be checked only for the position or also for the velocity. The derivation of the criterion is completely analogue using the velocities and the covariance submatrix relative to the velocity.

The proposed metric reduces the analysis of 12 parameters per time incident to an eigenvalue and a geometrical problem. To solve the eigenvalue problem, we take advantage of the fact that the covariance matrix is symmetric for computational efficiency. We use the Householder method to do a first tridiagonalization of the matrix and then we solve for the eigenvalues and eigenvectors with the QL algorithm [33, 34].

If a more detailed analysis of the OD success is desired, the uncertainty in the direction of the error (for position and velocity) can be plotted against the total error. Figure 8.8 shows the performance of the orbit determination for the case defined in section 4.4. Six diagrams had to be analysed with the previous version of SPOOK. This feature reduces these six diagrams to the two graphs shown in figure 8.9.

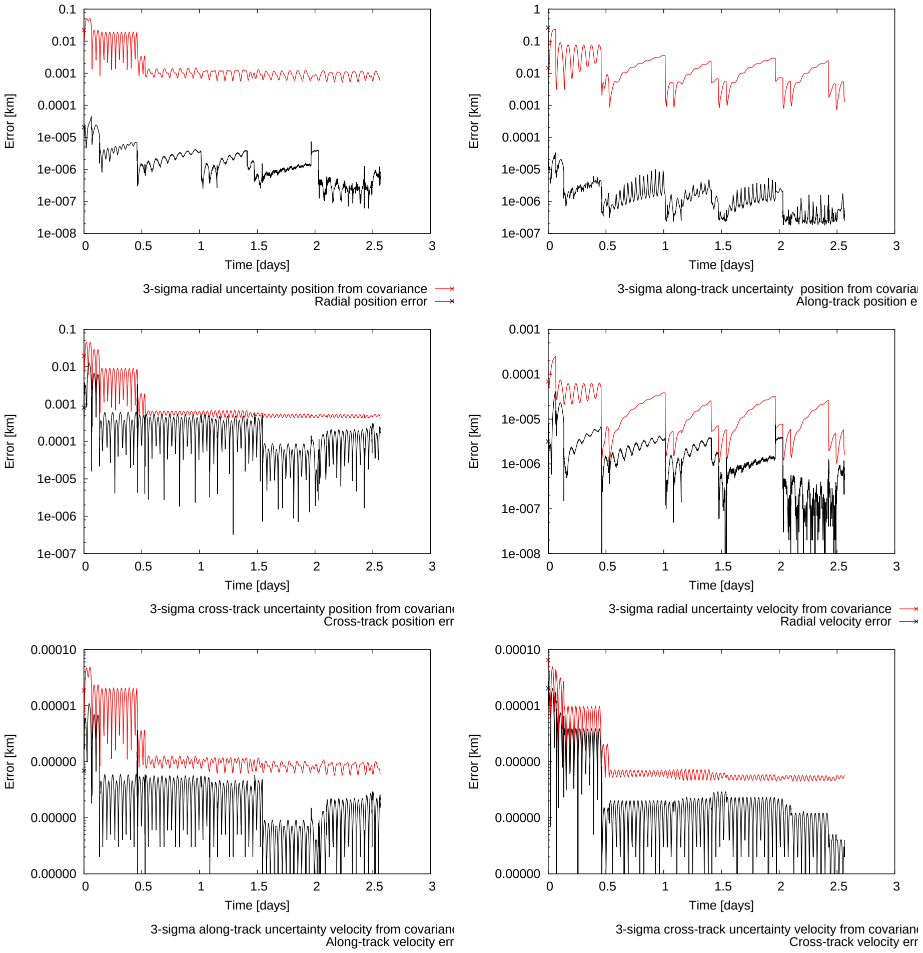


Figure 8.8: Assessment of the OD success using 6 diagrams

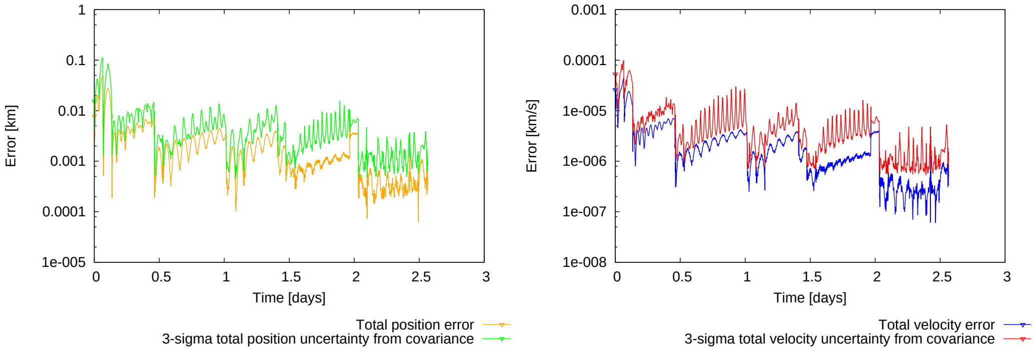


Figure 8.9: New assessment of the OD success using 2 diagrams

8.5.2 Accuracy of the prediction

A relevant parameter of the performance of an orbit determination method is the accuracy achieved. In general terms, accuracy accounts for the total error of the prediction. However, this error is variable and changes at each instant of simulation time. For a quick assesment of the method's performance, a characteristic accuracy must be defined. For Space Surveillance and Tracking Systems aiming at collision avoidance activities the state vector of the tracked object should remain within the defined accuracy region in order to make reliable closest approach forecasts. The closest approach forecast consist in a propagation of the object's state vector and covariance during a predefined timespan after the last measurement of the observation campaign. A characteristic accuracy of the orbit determination method could be defined as the maximum uncertainty obtained during this propagation after the last measurement is done. The length of this propagation can be defined in SPOOK through the configuration files (e.g. 48 h).

8.5.3 Orbital region selection

It is possible that a surveillance strategy has been designed to track objects of a certain orbital region (e.g. GEO objects, LEO objects, etc.). However, the TLE catalogue of space debris might contain objects distributed across all orbital regions. Simulating the objects for which the observation strategy has not been designed would greatly increase the execution times. For that reason the capability to select which objects will be processed has been added into SPOOK . Via the objects configuration file, the desired orbital regions to be analysed are selected. After an object has been read, a first classification into one of the orbital regions defined in section 2.7 will be made using the initial state. If the object belongs to one of the desired orbital regions, it will be processed. If not, it will be skipped and the next object in line will be processed.

8.5.4 Statistics generation

After the data generated by SPOOK has been automatically processed the following statistics will be produced per each object.

- **Mean orbital elements of the predicted object**. The mean semi-major axis a, the eccentricity e, inclination i, Right Ascension of Ascending Node Ω and argument of Perigee ω will be calculated averaging the instantaneous orbital elements computed for each simulated instant of time. With this mean elements, the apogee and perigee altitudes will be computed.
- **Orbital Region**. Using the mean elements computed before, the object will be classified in one of the categories defined in section 2.7.

- **Success of the orbit determination**. If an orbit determination has been attempted, indication if the criterion established in section 8.5.1 has been met is given.
- **Accuracy of the prediction**. The Root Mean Square of the characteristic accuracy defined in section 8.5.2 of all the different MonteCarlo runs performed. If only one MonteCarlo run is performed, the Root Mean Square is equivalent to the absolute value of the characteristic accuracy for that run.
- **Number of detections**. Total number of times that the object has been detected by an observers.
- **Time of first detection**. Time when the object was detected for the first time by an observer.
- **Revisit time frequency**. The inverse of the time between object detections. Maximum, minimum, mean and standard deviation of the revisit time frequencies are obtained.
- **Observability**. The observability is defined as the total time an object is in view of an observer. As with the revisit time, maximum, minimum, mean and standard deviation values are produced.

The number of detections, time of first detection, revisit frequencies and observability will be produced globally for all observers and per observer for each object.

Finally, a separate summary file will be produced. This summary file contains statistics about the total number of detected objects, number of successful orbit determinations, the accuracy of the prediction for the catalogued objects and the revisit frequencies and observabilities globally and per orbital region. This summary statistics file allows to quickly assess the success of the tested observation strategy to catalogue space debris objects.

8.6 Coverage Analysis: A Test Case

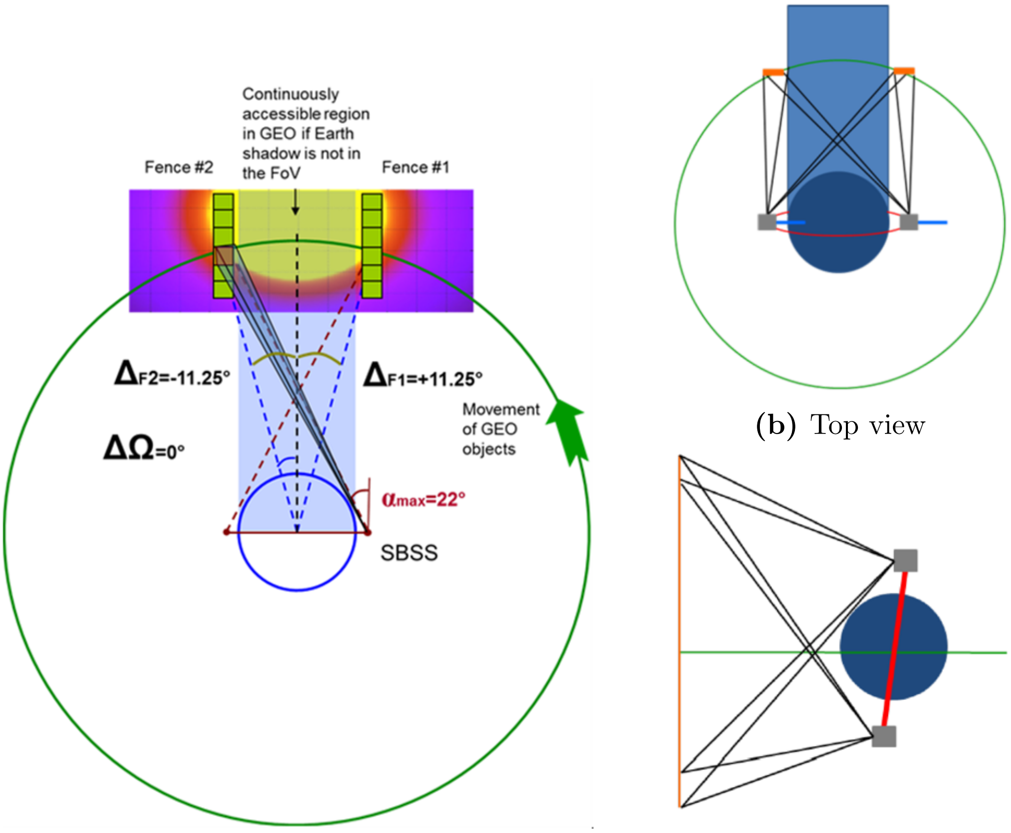
This section serves as an example for the multiple object processing and the coverage analysis mode introduced in SPOOK. A catalogue with more than 15.000 objects will be simulated in parallel to test the observation strategy. These objects are distributed across the 10 defined orbital regions.

8.6.1 Observation Strategy

The same surveillance strategy as in section 8.4.4 will be used here, the Geostationary Orbit Fence scenario [31]. As explained before, this strategy maximizes the coverage for GEO objects using a space-based optical observer flying in a dawn-dusk

LEO orbit. As it is has been designed to track objects in GEO orbit, only the region of GEO resident objects will be analysed.

Figure 8.10 shows the set up of the observation scenario. The green line represents the orbit of a space debris while the red one is the orbit of the space-based optical observer. Two fences are defined close to the edges of the Earth shadow (blue cylinder in figure 8.10). The FOV of the sensor will move between several fields within the two defined fences as can be seen in figure 8.11.



(b) Top view

(c) Side view

(a) Detailed top view

Figure 8.10: GEO Fence Scenario set-up

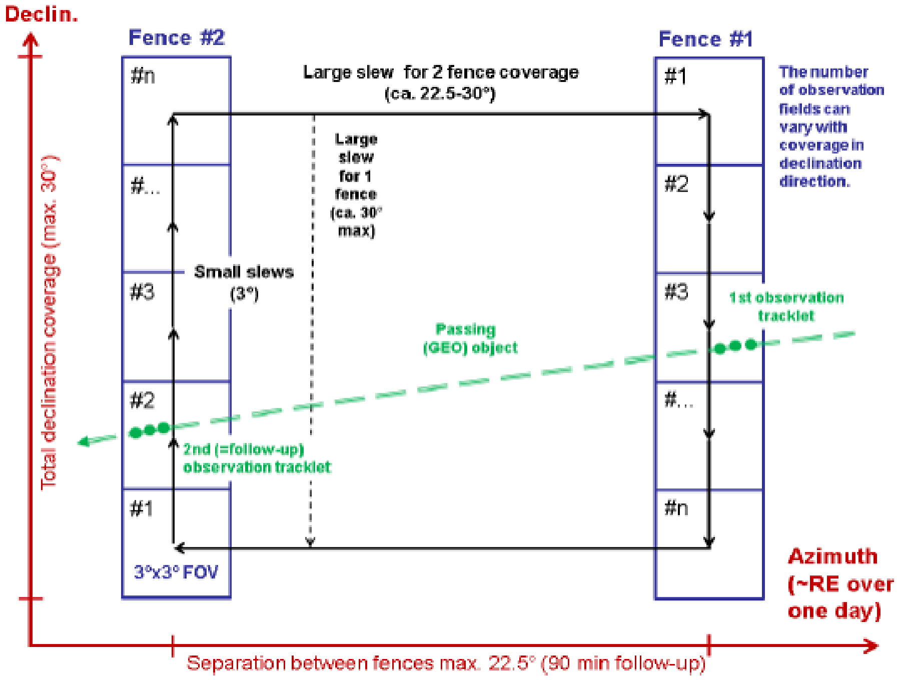

Declin.
Fence #2
Fence #1
#n
#...
#3
#2
#1
3°x3° FOV
Large slew for 2 fence coverage (ca. 22.5-30°)
Large slew for 1 fence (ca. 30° max)
Small slews (3°)
The number of observation fields can vary with coverage in declination direction.
1st observation tracklet
Passing (GEO) object
2nd (=follow-up) observation tracklet
Total declination coverage (max. 30°)
Azimuth (~RE over one day)
Separation between fences max. 22.5° (90 min follow-up)

Figure 8.11: Fields in the GEO fence

8.6.2 Results of the Geostationary Orbit Fence Scenario: Coverage Analysis

The observation strategy describe in the previous section was evaluated against a catalogue of 15404 objects. Of those objects, only 1170 were found to belong to the GEO resident region. The coverage analysis mode allowed to evaluate the performance of this strategy to detect this objects. A variable number of fields per fence was simulated varying from 1 to 10. Table 8.3 presents the results of the different variants of the GEO Fence surveillance strategy.

Number of fields	Declination coverage [°]	Detected objects [%]
1	3.0	50.77
2	6.0	65.58
3	8.9	89.32
4	11.9	95.56
5	14.9	98.55
6	17.9	98.55
7	20.8	98.72
8	23.8	98.72
10	29.7	> 99

Table 8.3: Total coverage of the different GEO Fence strategies

In figure 8.12 can be seen the temporal evolution of the objects detection. The maximum coverage of the particular observation strategy is achieved only after 24 h of observations.

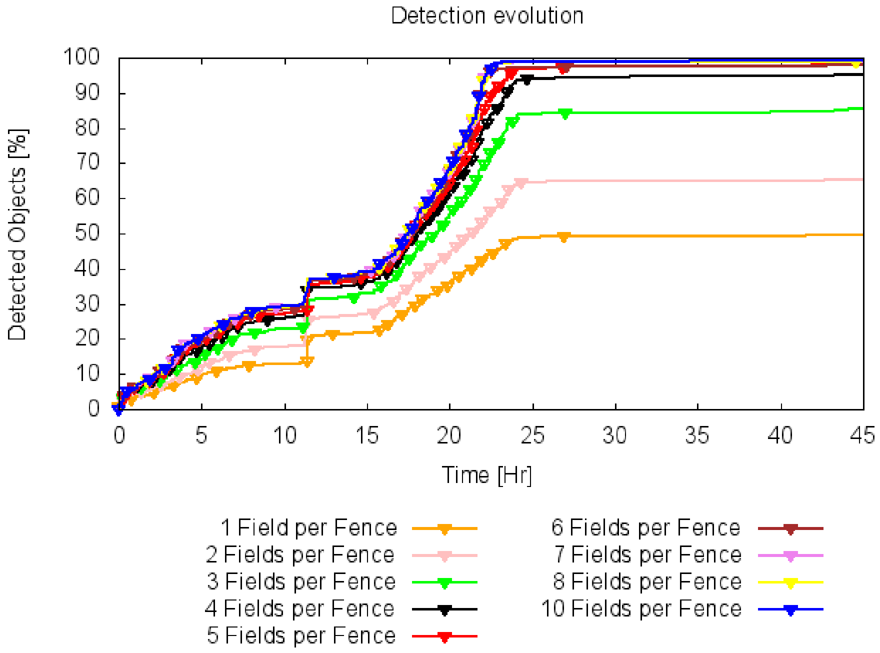


Figure 8.12: Temporal evolution of the objects detection

8.6.3 Results of the Geostationary Orbit Fence Scenario: Orbit Determination

For the GEO Fence scenario with 3 fields per fence an orbit determination was tried using the WLS to see how many objects could be catalogued. SPOOK was able to successfully predict the orbit for 94 objects using the previously derived OD criterion with an average accuracy of 230 m. Samples of the output files produced by SPOOK for this case can be found in appendix C.

Detected	Converged	Undetected	Total
1045 (89.32%)	94 (8.03%)	125 (10.68%)	1170 (100%)

Average Revisit Frequency [1/day]	Average Observability [s]	Accuracy of OD RMS [Km]
29.0751	71.34	0.231

Table 8.4: Results of the Covariance Analysis case

8.7 Conclusions

Multiple objects can now be processed by SPOOK in a single run. Furthermore, these objects can be analysed sequentially or in parallel with the subsequent saving in computational time (92.8% in the example scaling test). The feature greatly enhances the capabilities of SPOOK to analyse different surveillance strategies, being able to test them against a big sample of the space debris population.

The coverage analysis postprocessing feature facilitates the analysis of the data produced by SPOOK. To test this feature, different variations of the GEO Fence scenario were carried out. A 15404 objects catalogue was used as input, but only 1170 were processed (the ones in the GEO region). The total coverage achieved varied from 65% till more than 99%. For one of the test cases, an orbit determination was attempted. As a result, the orbit of 94 objects could be precisely determined with a mean accuracy of 230 m.

Chapter 9

Future Work

The new features added to SPOOK greatly improve its capabilities to predict orbits of space debris and to test different observation strategies. However, the development of the tool is not yet concluded.

The orbit determination feature using real measurements should be validated using tested and calibrated data. Active satellites usually have other means to determine their own orbits (such as GPS data) besides external measurements, being able to generate high precision ephemerides. Using measurements from external tracking stations and comparing the results with the ephemerides, a correct evaluation of the algorithm performance can be made.

An orbit determination convergence criterion was developed in the scope of this thesis. However, this criterion can only be applied for simulated scenarios, as it relies in knowing the true position of the targeted object. When real measurements are used, the convergence of the orbit determination must be evaluated by the user. Another convergence criterion must be developed for this case.

In the implementation of the multiple objects processing, it is assumed that the observers can track all the objects at the same time (e.g. every observer would have as many sensors as objects in the simulation scenario, which could be pointed independently). In reality, each observer can only track a limited number of objects at the same time. A work-load feature should be implemented in SPOOK to account for this limitation.

The currently implemented IOD methods need to use angular data to produce an initial state of the tracked object. However, with the newly implemented sensors, measurements consisting only in slant range and /or slant range-rate can be used in the orbit determination algorithms. Introducing new IOD algorithms would avoid the need of manually supplying an initial state vector when this kind of observers are used.

Real world data can now be used to perform an orbit determination. However, it was shown that the obtained covariance was not correctly linked with the obtained errors when compared against the ephemerides. A more realistic covariance representation algorithm should be implemented to take into account non-Gaussian distributed sources of noise.

The parallelization of the software was achieved by assigning the operations related to each object to a different core. 8.4.4 showed an improvement of the efficiency of 92.8%. However and contrary to what one could think, the efficiency improvement does not scale linearly with the number of cores used. A more detailed analysis of the performance of the parallel code should be carried out.

Chapter 10

Conclusions

Within the scope of this thesis, new types of radar-based sensors were introduced, enriching SPOOK's capabilities to use different types of data. Also the Light Time Delay is now taken into account both in the synthetic measurement generation as well as in the orbit determination methods, producing more accurate simulations of real tracking data.

It has been shown that real world data can now be used to predict the orbit of real space objects, using the data coming from the Poker Flat Incoherent Scatter Radar. Being able to perform orbit determination of real space objects was one of the main goals when the development of SPOOK began.

At the beginning of the activity only one object could be taken into account by SPOOK. Now, multiple objects can be processed in a single run of the program. This greatly enhances the capabilities of the tool to test different observation strategies, as different representative objects can be simulated, even the whole known space debris population, in the same run. Furthermore, all the objects are treated in parallel taking advantage of modern computer architectures, saving computational times as a result. To help the analysis of the data generated by such large simulations, a postprocessing mode was added to SPOOK, where relevant statistics are produced to allow a quick assessment of the results of the simulation.

References

[1] Inter-Agency Space Debris Coordination Committee. *IADC Space Debris Mitigation Guidelines*. 2007. URL: http://www.iadc-online.org/Documents/Docu/IADC_Mitigation_Guidelines_Rev1_Sep07.pdf (visited on 04/08/2016).

[2] National Aeronautics and Space Administration. Monthly Number of Objects in Earth Orbit by Object Type. In: *Orbital Debris Quarterly News* 2 (2016).

[3] European Space Agency. *How Many Space Debris Objects are currently in Orbit?* 2016. URL: http://www.esa.int/Our_Activities/Space_Engineering_Technology/Clean_Space/How_many_space_debris_objects_are_currently_in_orbit (visited on 04/08/2016).

[4] European Space Agency. *Distribution of Debris*. 2016. URL: http://www.esa.int/spaceinimages/Images/2013/04/Distribution_of_debris (visited on 04/08/2016).

[5] Johnson NL, Stansbery E, Whitlock DO, Abercromby KJ, Shoots D. *History of On-Orbit Satellite Fragmentations*. Technical Report. National Aeronautics and Space Administration, 2008.

[6] Wang T. Analysis of Debris from the Collision of the Cosmos 2251 and the Iridium 33 Satellites. In: *Science & Global Security* 18.2 (2010), pp. 87–118.

[7] Vallado DA. *Fundamentals of Astrodynamics and Applications*. Ed. by Wertz JR. 4th. Hawthorne, CA: Microcosm Press, 2013.

[8] Arias E, Charlot P, Feissel M, Lestrade JF. The Extragalactic Reference System of the International Earth Rotation Service, ICRS. In: *Astronomy and Astrophysics* 303 (1995), pp. 604–608.

[9] *Implementation Issues Surrounding the New IAU Reference Systems for Astrodynamics*. 16 AAS/AIAA Space Flight Mechanics Conference. AAS. Tampa, Florida.

[10] Valentin M. Enhacement of an Orbit Determination Simulation Tool for Space Debris. MSc thesis. Universität Würzburg, 2015.

[11] National Aeronautics and Space Administration. *JPL Horizons ephemerides provider*. URL: http://ssd.jpl.nasa.gov/horizons.cgi (visited on 07/08/2016).

[12] Battin RH. *An Introduction to the Mathematics and Methods of astrodynamics.* AIAA, 1999.

[13] *Department of Defense World Geodetic System 1984.* Technical report NIMA TR8350.2, 3rd ed, Amendment 1. National Imagery and Mapping Agency.

[14] Melvin WL, Scheer JA. *Principles of Modern Radar, Volume 3 - Radar Applications.* Institution of Engineering and Technology, 2014. ISBN: 978-1-89112-154-8.

[15] Schetzen M. *Airborne Doppler Radar - Applications, Theory, and Philosophy - Progress in Astronautics and Aeronautics, Volume 215.* American Institute of Aeronautics and Astronautics, 2006. ISBN: 978-1-56347-828-4.

[16] Klinkrad H. *Space Debris.* Wiley Online Library, 2010.

[17] Dreyer H. Development of an Orbit Determination Simulation Tool based on an Extended Kalman Filter. MSc thesis. Universität Stuttgart, 2015.

[18] Hedin AE. *Horizontal Wind Model.* URL: `http://ccmc.gsfc.nasa.gov/modelweb/atmos/hwm.html` (visited on 07/08/2016).

[19] Montenbruck O, Gill E. *Satellite Orbits. Models, Methods and Applications.* Heidelberg: Springer, 2000.

[20] Winzen M. Enhacement of an Orbit Determination Simulation Tool for Space Debris. MSc thesis. Aachen University, 2015.

[21] Fehlberg E. *Low-order Classical Runge-Kutta Formulas with Stepsize Control and their Application to some Heat Transfer Problems.* Technical Report. National Aeronautics and Space Administration, 1969.

[22] Shampine LF, Gordon MK. *Computer Solution of Ordinary Differential Equations.* Freeman, 1975.

[23] ESA SSA Team. *Space Situational Awareness - Space Surveillance and Tracking System Requirements Document.* Technical Report. European Space Agency, 2013.

[24] Gelhaus J, Flegel S, Wiedemann C. *Program for Radar and Optical Observation Forecasting.* Technical report. Technische Universität Braunschweig.

[25] McCarthy D, Petit G. *IERS Conventions 2003.* IERS Technical Note 32. International Earth Rotation and Reference Systems Service.

[26] *Technical Notes on Central Bodies. STK Help.* Analytical Graphics Inc.

[27] Hermanns M. *Parallel Programming in Fortran 95 using OpenMP.* School of Aeronautical Engineering, Universidad Politécnica de Madrid. Madrid, Spain.

[28] *Revisiting Spacetrack Report #3.* 6753. American Institute of Aeronautics and Astronautics. 2006.

[29] OpenMP Architecture Review Board. *OpenMP.* 2016. URL: `http://openmp.org` (visited on 22/07/2016).

[30] Resch MM. *Fortran for Scientific Computing.* Ed. by Küster U. 7th. 2014.

[31] Utzmann J. *Design Definition File for SBSS Demonstrator System.* Technical Report. Astrium GmbH, 2013.

[32] Wiesel WE. *Modern Orbit Determination.* Aphelion Press, 2003.

[33] Press W, Teukolsky S, Vetterling W, Flannery B. *Numerical Recipes in Fortran 77: The Art of Scientific Computing.* 1992.

[34] Press W, Teukolsky S, Vetterling W, Flannery B. *Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing.* 2002.

Appendix A

Example files

A.1 Error log file

Below can be seen an example of an `Error_log.dat` configuration file.

```
========================================================
ERROR INFO:

Object ID:  14985
Object Nr: 00003548
Processed by thread Nr: 00000002
In subroutine/function: WeightedLeastSquares

ERROR MESSAGE:
Error: Differential Correction didn't converge within the maximum Number of Iterations chosen
========================================================

========================================================
ERROR INFO:

Object ID:  22253
Object Nr: 00005526
Processed by thread Nr: 00000029
In subroutine/function: WeightedLeastSquares

ERROR MESSAGE:
Error: Differential Correction didn't converge within the maximum Number of Iterations chosen
========================================================

========================================================
ERROR INFO:

Object ID:  22911
Object Nr: 00005814
Processed by thread Nr: 00000035
In subroutine/function: SelectMeas
```

```
ERROR MESSAGE:
 Error: IOD Measurements Nr 1 and 2 are identical. Program will abort.
========================================================

========================================================
ERROR INFO:

Object ID:  22506
Object Nr: 00005675
Processed by thread Nr: 00000014
In subroutine/function: SPOP

ERROR MESSAGE:
 Error in SPOP integration routine
========================================================
```

Appendix B

PFISR data

B.1 OD results using TLE for intialization

In this appendix it can be found the results of the Orbit Determination of SPOT 6 using TLE as the initialization method. As commented in chapter 7 the analyis is equivalent as when ephemerides are used.

B.1.1 Weighted Least Squares

Results of the Orbit Determination using real PFISR data and the WLS algorithm.

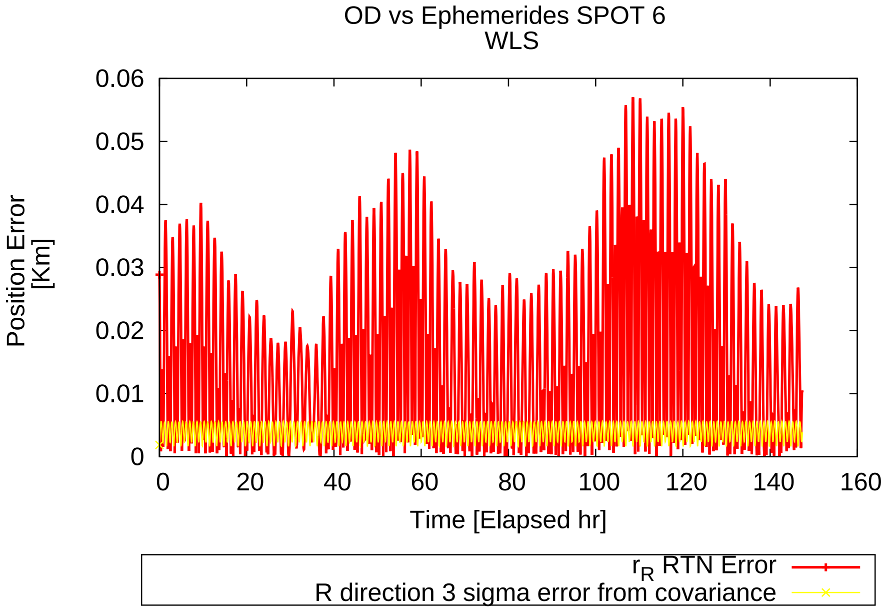


Figure B.1: Results of the OD using WLS for the position in the R direction using TLE for initialization

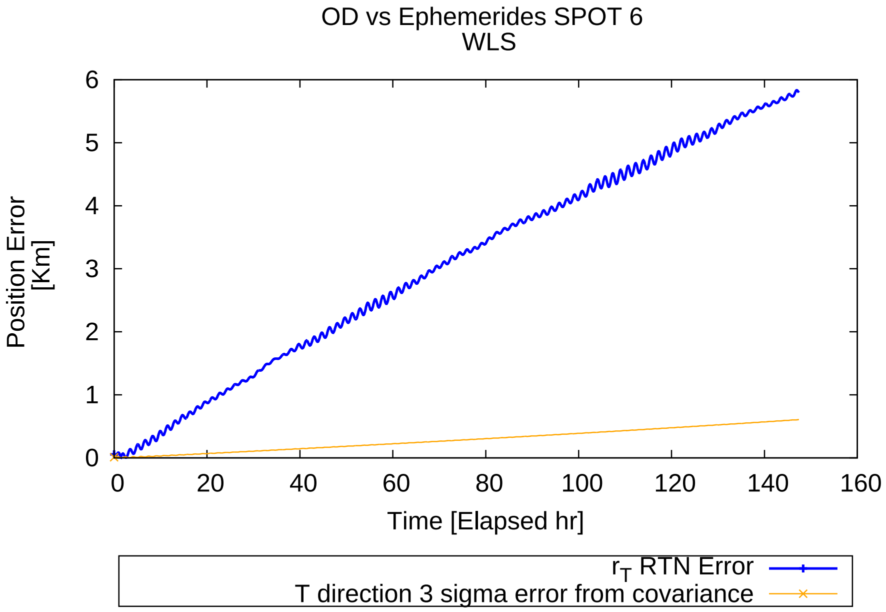


Figure B.2: Results of the OD using WLS for the position in the T direction using TLE for initialization

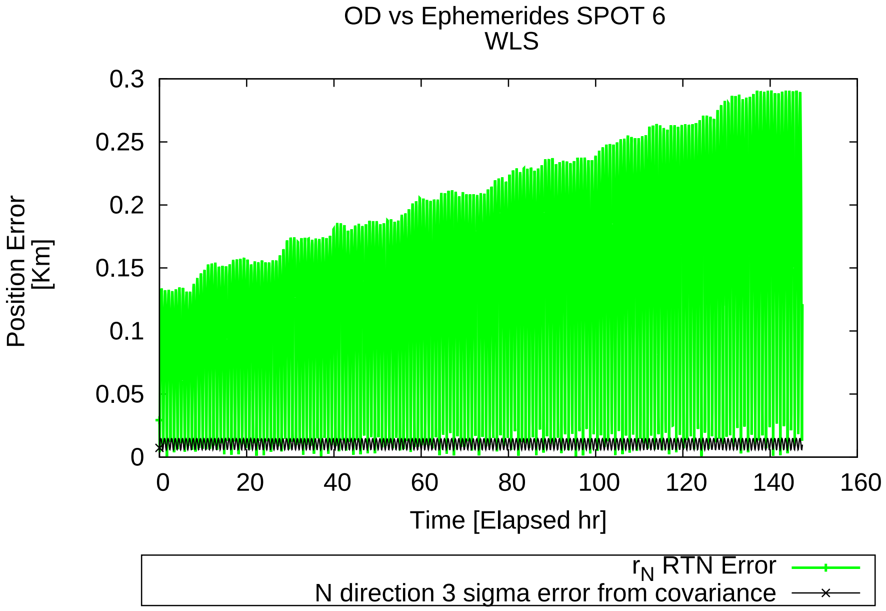


Figure B.3: Results of the OD using WLS for the position in the N direction using TLE for initialization

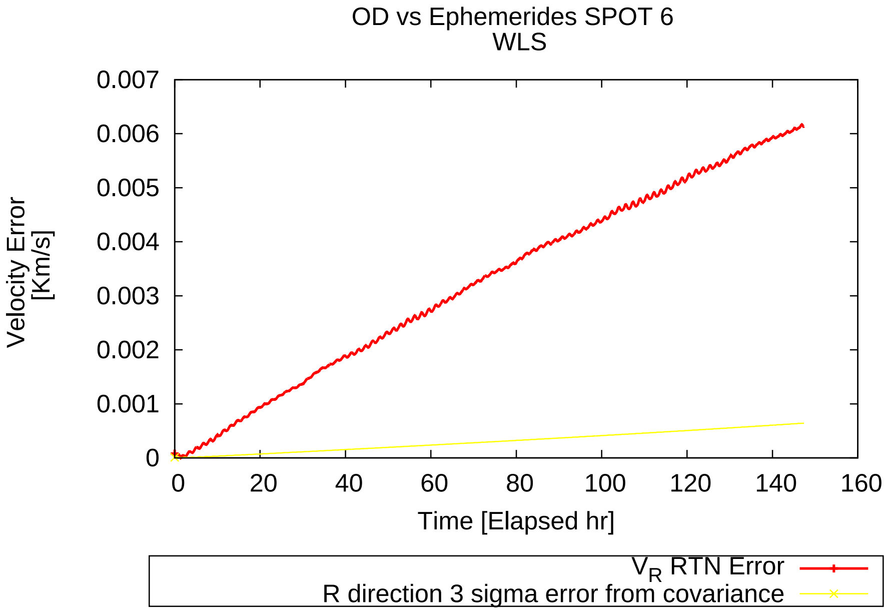


Figure B.4: Results of the OD using WLS for the velocity in the R direction using TLE for initialization

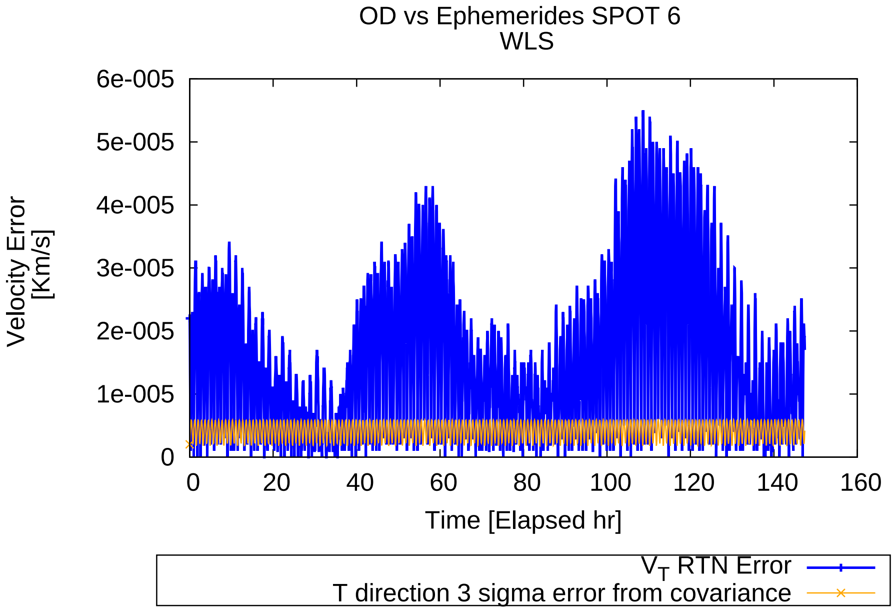


Figure B.5: Results of the OD using WLS for the velocity in the T direction using TLE for initialization

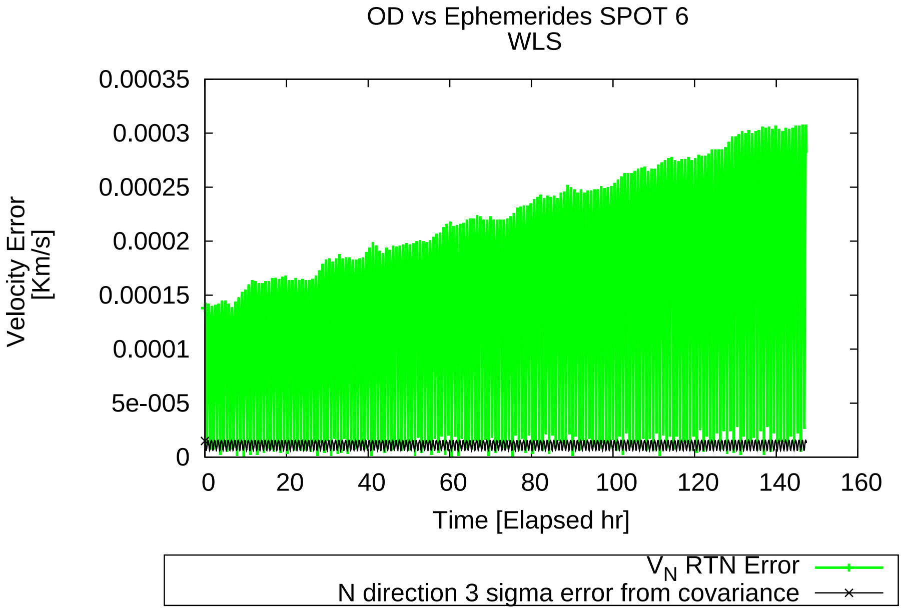


Figure B.6: Results of the OD using WLS for the velocity in the N direction using TLE for initialization

B.1.2 Sequential Batched Least Squares

Results of the Orbit Determination using real PFISR data and the SBLS algorithm using two tracklets for each differential correction process.

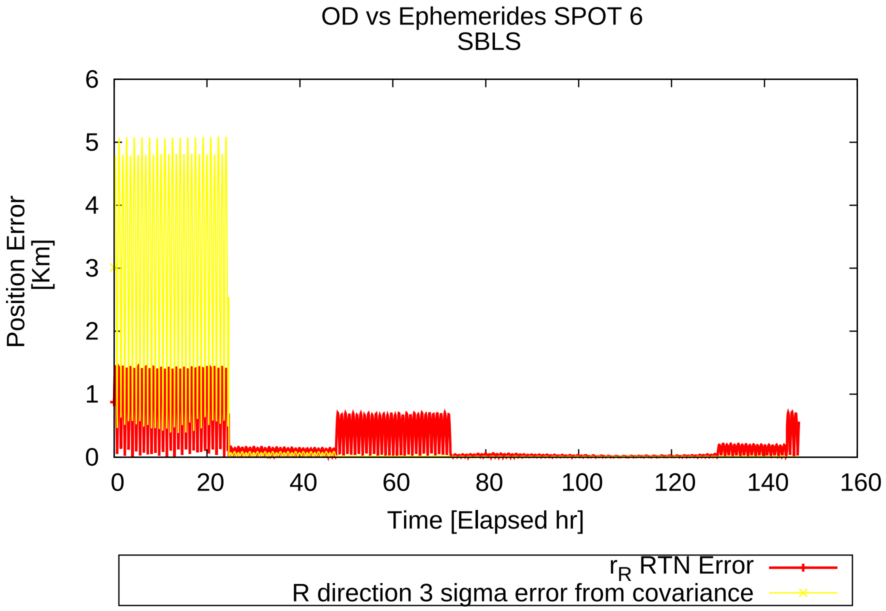


Figure B.7: Results of the OD using SBLS for the position in the R direction using TLE for initialization

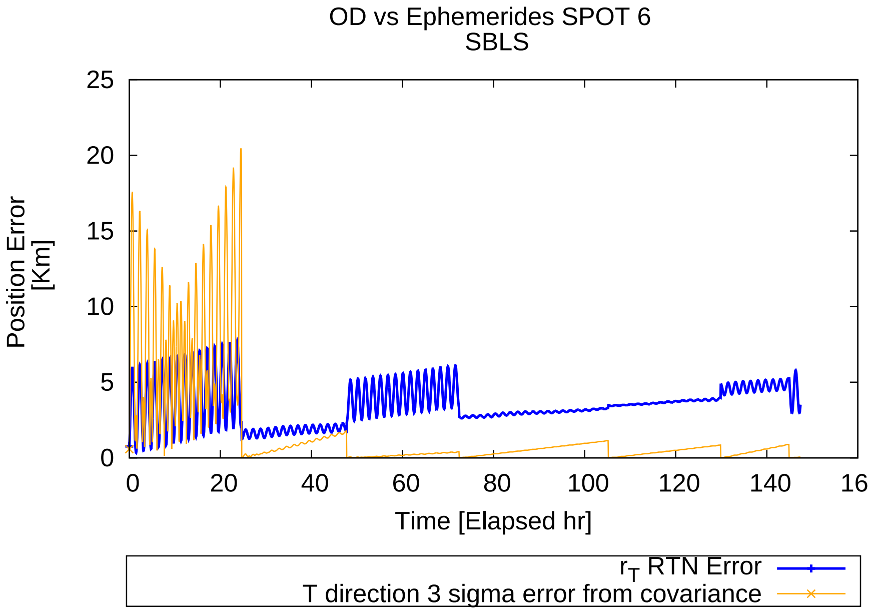


Figure B.8: Results of the OD using SBLS for the position in the T direction using TLE for initialization

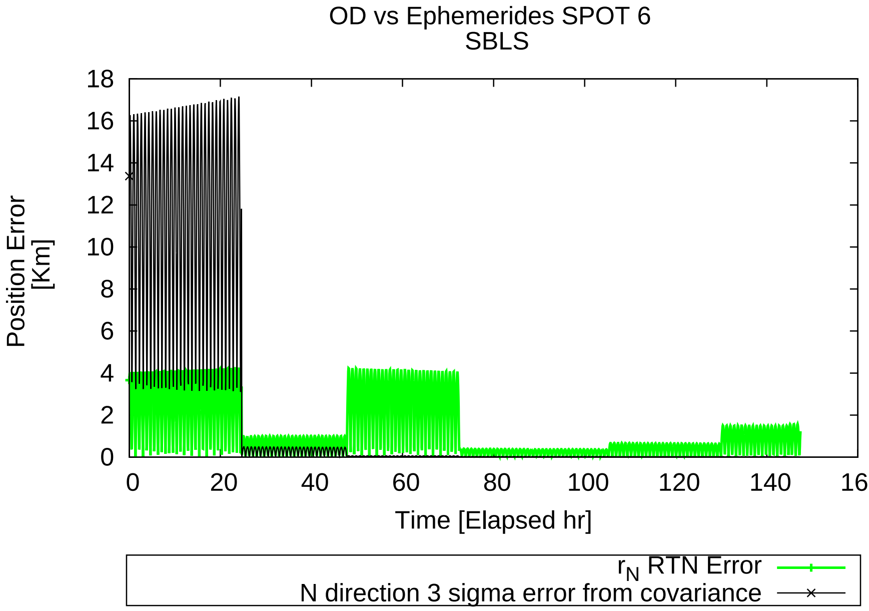


Figure B.9: Results of the OD using SBLS for the position in the N direction using TLE for initialization

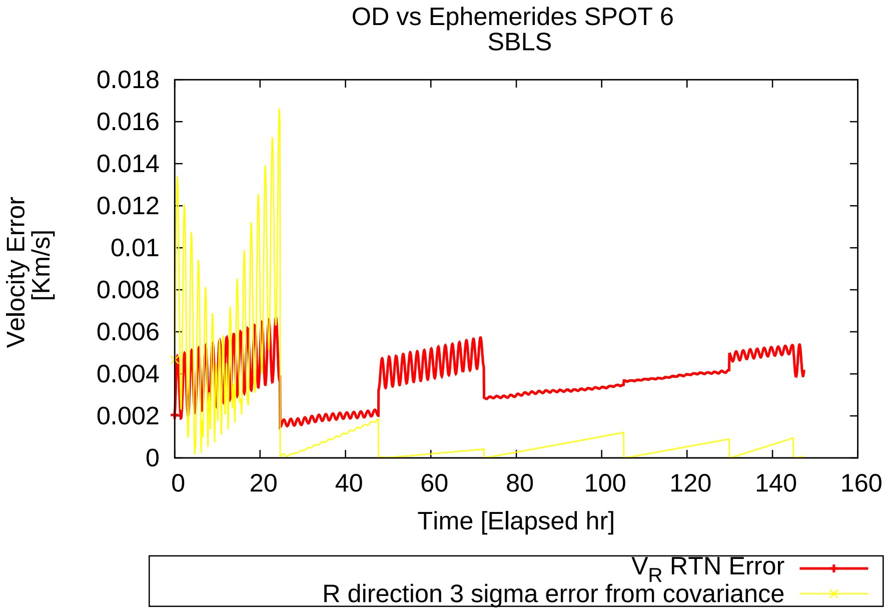


Figure B.10: Results of the OD using SBLS for the velocity in the R direction using TLE for initialization

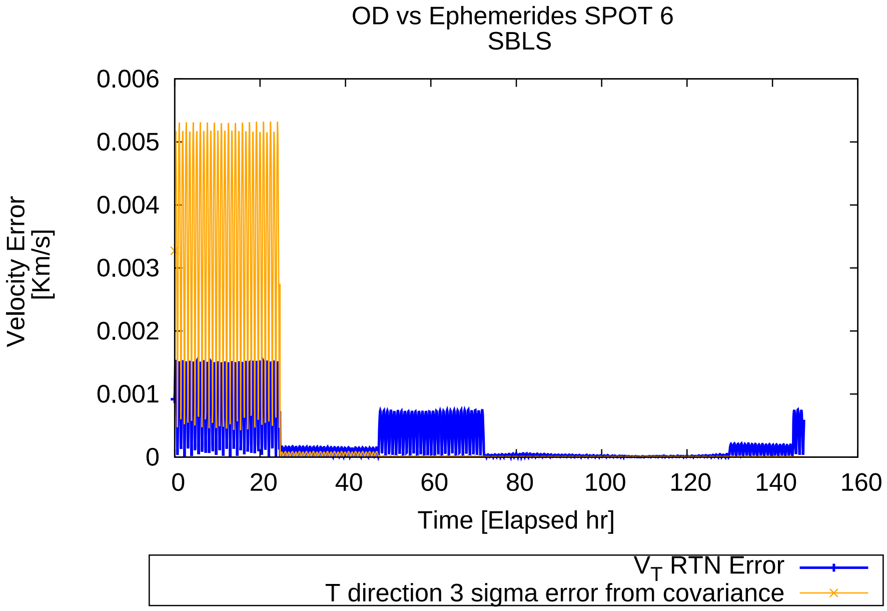


Figure B.11: Results of the OD using SBLS for the velocity in the T direction using TLE for initialization

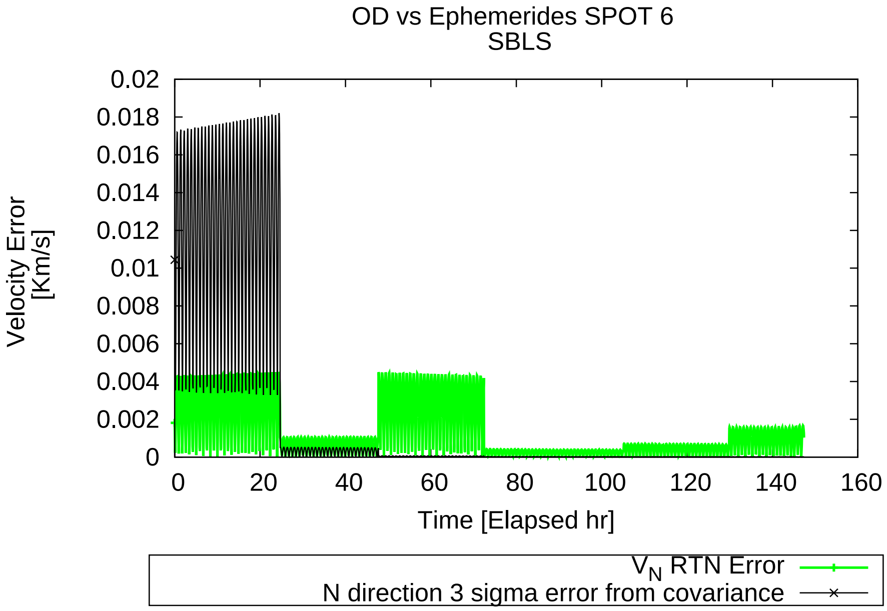


Figure B.12: Results of the OD using SBLS for the velocity in the N direction using TLE for initialization

Appendix C

Coverage Analysis Results

In this chapter can be found some of the output files produced by SPOOK in the coverage analysis mode. Three sample output files can be found.

1. Summary statistics of the coverage analysis. These are the global statistics that asses the performance of the observation strategy.
2. Statistics of the detected objects. One line per object statistics for the detected objects.
3. Statistics of the undetected objects. One line per object statistics for the undetected objects.

Due to the output format with long lines, each file will be split into several parts.

C.1 Summary statistics of the Coverage analysis results

```
#-----------------------------------------------------------------------------------------------------------------------
# Statistics Summary
#-----------------------------------------------------------------------------------------------------------------------
#  #   Orbital Region               |    #Detected Obj.  |    #Converged Obj. |    #Undetected Obj.|     #Total Obj.
#                                   |     [abs. & %]     |     [abs. & %]     |     [abs. & %]     |     [abs. & %]
#-----------------------------------------------------------------------------------------------------------------------
   0   TOTAL (LEO to HEO)                1045 ( 89.32%)          94 (  8.03%)         125 ( 10.68%)        1170 (100.00%)
   1   LEO resident                         0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
   2   LEO transient                        0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
   3   Low MEO resident                     0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
   4   Low MEO transient                    0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
   5   High MEO resident (GNSS)             0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
   6   High MEO transient                   0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
   7   GEO resident, i=0 to 20 deg       1042 ( 90.06%)          94 (  8.12%)         115 (  9.94%)        1157 (100.00%)
   8   GEO resident, i > 20 deg             3 ( 23.08%)           0 (  0.00%)          10 ( 76.92%)          13 (100.00%)
   9   GEO transient                        0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
  10   HEO                                  0 (  0.00%)           0 (  0.00%)           0 (  0.00%)           0 (  0.00%)
```

```
-----------------------------------------------------------------------------------------------------------


-----------------------------------------------------------------------------------------------------------
|  *************************  Revisit frequency for detected objects [1./day]  ***********************
|        min_ave   ,   min_sigma   |       max_ave    ,   max_sigma    |       mean_ave   ,  mean_sigma
-----------------------------------------------------------------------------------------------------------
        1.024349          0.063924       89.124865       162.112533       29.075100          50.124039
        0.000000          0.000000        0.000000         0.000000        0.000000           0.000000
        0.000000          0.000000        0.000000         0.000000        0.000000           0.000000
        0.000000          0.000000        0.000000         0.000000        0.000000           0.000000
        0.000000          0.000000        0.000000         0.000000        0.000000           0.000000
        0.000000          0.000000        0.000000         0.000000        0.000000           0.000000
        0.000000          0.000000        0.000000         0.000000        0.000000           0.000000
        1.024429          0.064004       88.457818       160.592533       28.842933          49.528696
        0.997812          0.000070      312.141136       458.627830      106.696243         156.291741
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
```

```
----------------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------------
|  ****************************  Observability for detected objects [s]  ****************************
|        min_ave    ,   min_sigma    |       max_ave    ,   max_sigma     |       mean_ave    ,  mean_sigma
----------------------------------------------------------------------------------------------------------
       53.536354         28.882528        77.678482          3.897387        71.343331          8.838103
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
       53.549415         28.865914        77.677556          3.902961        71.341440          8.846301
       48.999976         41.469872        78.000006          0.000023        71.999995          6.408007
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
        0.000000          0.000000         0.000000          0.000000         0.000000          0.000000
```

```
--------------------------------------------------------------------

--------------------------------------------------------------------
|  *******  RMS of OD accuracy for converged objects [Km]  *******  |
|    Mean_RMS    |     min_RMS    |     max_RMS    |   Sigma_RMS    |
--------------------------------------------------------------------
     0.23186E+00      0.27947E-01      0.87020E+00      0.16706E+00
     0.00000E+00      0.00000E+00      0.00000E+00      0.00000E+00
     0.00000E+00      0.00000E+00      0.00000E+00      0.00000E+00
     0.00000E+00      0.00000E+00      0.00000E+00      0.00000E+00
     0.00000E+00      0.00000E+00      0.00000E+00      0.00000E+00
     0.00000E+00      0.00000E+00      0.00000E+00      0.00000E+00
     0.00000E+00      0.00000E+00      0.00000E+00      0.00000E+00
     0.23186E+00      0.27947E-01      0.87020E+00      0.16706E+00
     0.00000E+00      0.00000E+00      0.00000E+00      0.00000E+00
```

16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

C.2 Statistics of the detected objects

```
#-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
# Statistics for detected objects  TOTAL (LEO to HEO)
#--------------------------------------------------------------------------------------------------------------------------------------
#- Cat. No           :   NORAD Catalogue Number
#- Catalogued        :   Indication if the Orbit Determination has been successful
                         * Yes: The error is inside the 3-sigma ellipsoid from covariance during at least 90.00% of the simulation.
                         * No:  The error is not inside the 3-sigma ellipsoid from covariance during at least 90.00% of the simulation
                         * N/A: Converge criterion not computed (real measurements used, measurement generation mode only)
                         * Err: Error in the simulation of the object.
#- Orbit category    : Orbital regimen of the object
                     1: LEO resident             2: LEO transient            3: Low MEO resident
                     4: Low MEO transient        5: Hig MEO resident (GNSS)  6: High MEO transient
                     7: GEO resident (i<20 deg)  8: GEO resident (i>20 deg)  9: GEO transient
                    10: HEO                     11: OTH (all other)
#- Revisit Frequency : Inverse of the time between tracklets
#- Observability     : Tracklet duration
#--------------------------------------------------------------------------------------------------------------------------------------
#  1       2           3            4         5           6          7        8        9           10              11          12
#                                                                                                                                    |
#     |  SPOOK   |            | Accuracy | Semimajor |           |        |        |        |               |               | Orbit  |
# Cat.|  Object  | Catalogued |   RMS    |   Axis    |  Eccentr. |  Incl. |   RAAN |   ArgP |     h_peri    |     h_apo     | Categ. |
# No. |  Number  |    [-]     |   [Km]   |   [km]    |    [-]    |  [deg] |  [deg] |  [deg] |      [km]     |     [km]      |   [-]  |
#--------------------------------------------------------------------------------------------------------------------------------------
 28912       834      YES       0.379E-01   42168.28   0.0002427    1.41    68.79   254.83       35779.912       35800.382      7
 28935       836      YES       0.279E-01   42164.36   0.0002059    0.09   110.50   260.13       35777.540       35794.903      7
 28937       837      NO        0.126E+01   42165.92   0.0004190    0.08   106.77   262.64       35770.116       35805.448      7
 28945       838      YES       0.506E+00   42164.04   0.0001882    0.07   123.66   204.21       35777.965       35793.832      7
 28902       831      Err          -        42165.37   0.0001342    0.09   110.90   260.12       35781.573       35792.888      7
 28946       839      NO        0.127E+01   42165.86   0.0001715    0.07   117.08   265.48       35780.492       35794.957      7
 29045       841      NO        0.182E+01   42165.69   0.0001550    0.06   104.99   259.60       35781.015       35794.082      7
 29055       842      YES       0.205E+00   42166.13   0.0004889    0.02    24.05   319.30       35767.375       35808.609      7
 29155       844      NO        0.664E+00   42164.69   0.0006344    0.28   264.02   140.62       35759.803       35813.305      7
 29162       845      YES       0.340E+00   42164.14   0.0002428    0.09    89.66   283.33       35775.764       35796.238      7
 29230       847      YES       0.870E+00   42458.76   0.0002976    5.53    59.81   123.23       36067.987       36093.259      7
 29163       846      YES       0.421E+00   42167.02   0.0006141    0.07   103.11   222.41       35762.994       35814.781      7
 29236       849      Err          -        42165.74   0.0002762    0.08   118.32   238.18       35775.957       35799.252      7
 29270       850      NO        0.179E+01   42165.81   0.0005556    0.07    56.59   296.93       35764.247       35811.102      7
 29272       851      NO        0.984E+00   42165.94   0.0002000    0.08   100.97   254.87       35779.367       35796.230      7
```

```
----------------------------------------------------------------------------------------------------------------------------------------------















----------------------------------------------------------------------------------------------------------------------------------------------
      12              13                   14                   15                   16                   17                   18
                                                                                                    TOTAL                                |
  Number of  |   Time of first    |     Revisit          |     Revisit         |  Revisit            |  Revisit            |  Observability    |
  detections |   detection        |     frequency MIN    |     frequency MAX   |  frequency MEAN     |  frequency SIGMA    |      MIN          |
     [-]     |   [yyddd.dddddd]   |     [1./day]         |     [1./day]        |  [1./day]           |  [1./day]           |      [s]          |
--------------------------------------------------------------------------------------------------------------------------------------------
          6        16084.241829             1.063640            15.880893             9.953992             8.115744            77.999979
          7        16084.423889             1.073085           122.640167            28.738170            46.570865            22.499979
          6        16083.852882             1.006078           118.356167            30.439423            49.704022            17.999983
          6        16083.977014             1.063640            15.880893             9.953992             8.115744            77.999979
          6        16084.390787             1.063640            15.880893             9.953992             8.115744            77.999979
          6        16083.753576             1.063640            15.880893             9.953992             8.115744            77.999979
          6        16084.407338             1.063640            15.880893             9.953992             8.115744            77.999979
          6        16083.588067             1.063640            15.880893             9.953992             8.115744            77.999979
          6        16083.977014             1.063640            15.880893             9.953992             8.115744            77.999979
          6        16083.977014             1.063640            15.880893             9.953992             8.115744            77.999979
          5        16083.510625             0.989600           118.844567            59.736349            67.830842            12.000015
          6        16084.200451             1.063640            15.880893             9.953992             8.115744            77.999979
          7        16084.233553             1.006078           122.379605            46.042415            57.763433            19.499995
          6        16083.662546             1.063640            15.880893             9.953992             8.115744            77.999979
          6        16084.258380             1.063640            16.017798             9.981373             8.140931            31.499970
```

```
--------------------------------------------------------------














--------------------------------------------------------------
           19                  20                  21
                                                             |
|  Observability    |    Observability   |   Observability   |
|      MAX          |        MEAN        |      SIGMA        |
|      [s]          |        [s]         |      [s]          |
--------------------------------------------------------------
         78.000019           78.000013            0.000016
         77.999979           65.357125           22.547566
         78.000019           67.499989           24.279624
         78.000019           78.000013            0.000016
         78.000019           77.999986            0.000016
         78.000019           78.000006            0.000021
         78.000019           78.000013            0.000016
         78.000019           77.999993            0.000021
         78.000019           78.000013            0.000016
         78.000019           78.000013            0.000016
         78.000019           52.500001           34.953529
         78.000019           77.999986            0.000016
         77.999979           57.214271           27.528547
         78.000019           77.999986            0.000016
         78.000019           70.249984           18.983553
```

C.3 Statistics of the undetected objects

```
#--------------------------------------------------------------------------------------------------------------
# Statistics for undetected objects  TOTAL (LEO to HEO)
#--------------------------------------------------------------------------------------------------------------
#  - Cat. No          :   NORAD Catalogue Number
#  - Orbit category   :   Orbital regimen of the object
                          1: LEO resident            2: LEO transient            3: Low MEO resident
                          4: Low MEO transient       5: Hig MEO resident (GNSS)  6: High MEO transient
                          7: GEO resident (i<20 deg) 8: GEO resident (i>20 deg)  9: GEO transient
                         10: HEO                    11: OTH (all other)
#--------------------------------------------------------------------------------------------------------------
#  1       2          3            4          5       6        7            8                9           10
#                                                                                                            |
#     |  SPOOK   | Semimajor |           |        |        |        |                |                | Orbit  |
# Cat.|  Object  |   Axis    |  Eccentr. |  Incl. |  RAAN  |  ArgP  |     h_peri     |     h_apo      | Categ. |
# No. |  Number  |   [km]    |    [-]    |  [deg] |  [deg] |  [deg] |      [km]      |      [km]      |   [-]  |
#--------------------------------------------------------------------------------------------------------------
  4068        12   42385.39   0.0006019    8.57   309.72   255.53       35981.746        36032.767       7
  5589        22   43053.13   0.0150989   11.83   318.28    79.89       36024.939        37325.049       7
  4376        16   42178.54   0.0002981    6.37   302.31   304.05       35787.832        35812.977       7
  3692        11   42929.65   0.0154616    7.54   307.82   109.58       35887.748        37215.270       7
  4881        18   42572.73   0.0012633   12.58   322.15   341.83       36140.813        36248.375       7
  4250        13   42166.31   0.0024481    8.09   310.83   218.45       35684.945        35891.395       7
  7324        35   42055.06   0.0025871   11.13   316.26   243.22       35568.127        35785.728       7
  4297        14   42482.16   0.0011012    7.65   307.55   342.59       36057.242        36150.801       7
  5775        24   42287.78   0.0001114   13.96   334.18   350.91       35904.934        35914.354       7
  5587        20   42165.10   0.0004163   10.07   313.56   233.74       35769.410        35804.520       7
  7392        36   42172.31   0.0010205    9.87   312.86   159.95       35751.142        35837.213       7
  6052        25   42204.35   0.0003649   13.38   329.39   193.45       35810.810        35841.609       7
  4353        15   42163.35   0.0002792    8.70   314.55   214.94       35773.436        35796.982       7
  6974        30   42955.59   0.0071290   12.89   323.55   337.97       36271.220        36883.679       7
  7229        31   42024.97   0.0040649    9.28   312.92   312.47       35476.005        35817.656       7
  5588        21   42153.32   0.0003332   10.03   313.88   326.45       35761.135        35789.223       7
  7298        33   42637.15   0.0017023   12.75   312.15   319.88       36186.427        36331.590       7
  6973        29   42915.84   0.0030164   13.43   322.26   133.16       36408.256        36667.157       7
  5709        23   42350.17   0.0009726   13.85   331.82     7.04       35930.846        36013.222       7
  4902        19   42165.43   0.0002566    9.65   312.17   137.65       35776.471        35798.109       7
  6278        26   42575.32   0.0012565   13.68   331.67   102.80       36143.689        36250.680       7
  7318        34   41693.03   0.0029971   10.27   313.85   206.12       35189.934        35439.847       7
  7547        39   42182.40   0.0000863   11.60   322.97   111.98       35800.620        35807.897       7
  8513        49   42171.43   0.0007932   11.15   317.59   173.30       35759.842        35826.744       7
  7578        40   42258.21   0.0001732   12.66   323.61   251.04       35872.751        35887.389       7
```

42	7648	41	42381.08	0.0012656	13.34	327.07	169.01	35949.310	36056.582	7
43	8132	44	42247.58	0.0004747	12.21	321.84	289.49	35849.385	35889.495	7
44	8357	46	42179.08	0.0009100	10.78	315.70	64.14	35762.562	35839.331	7
45	8747	54	42165.59	0.0022383	14.96	90.53	70.49	35693.077	35881.833	7
46	8751	55	42738.18	0.0146485	13.92	96.04	345.60	35733.994	36986.094	7
47	8366	47	42163.84	0.0004129	12.64	323.93	247.58	35768.296	35803.114	7
48	8585	50	42152.37	0.0016063	12.14	321.05	162.43	35706.528	35841.945	7
49	8832	58	42739.54	0.0150474	13.92	96.05	344.22	35718.284	37004.521	7
50	8746	53	42167.02	0.0016091	15.00	90.50	27.61	35721.028	35856.730	7
51	9416	63	42160.31	0.0027905	11.77	319.58	282.23	35664.522	35899.817	7
52	9503	65	42169.75	0.0061391	11.83	319.79	71.87	35532.732	36050.500	7
53	9852	67	42239.11	0.0002894	13.10	327.05	259.05	35848.754	35873.200	7