DOI: 10.13009/EUCASS2017-553

SPOOK - A Comprehensive Space Surveillance and Tracking Analysis Tool

O. Rodriguez[⋆†], *J. Utzmann*[⋆] *and U. Hugentobler*[†]
[⋆]*Airbus DS GmbH*
Claude-Dornier Straße, 88090 Immenstaad
oscar.rodriguezfernandez@airbus.com · jens.utzmann@airbus.com
[†]*Institut für Astronomische und Physikalische Geodäsie, Technische Universität München*
Arcisstr. 21,80333 München
urs.hugentobler@bv.tu-muenchen.de

Abstract

Space Surveillance and Tracking (SST) is the ability to detect and predict the movement of space debris in orbit around the Earth. SPOOK (SPace Object Observations and Kalman filtering) is a versatile and highly configurable software tool developed by Airbus DS GmbH, for the detection, cataloguing and orbit prediction analysis of Earth orbiting objects. It provides an integrated framework for all the activities related with SST sensor architecture simulation. Throughout this paper, the functionalities of the tool are described and examples of different simulation scenarios are provided.

1. Introduction

The Inter-Agency Space Debris Coordination (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, space has become more and more crowded with active satellites and space debris. The US Space Surveillance Network currently catalogues more than 17.000 objects [2], but it is estimated that more than 170 million objects larger than 1 mm are currently orbiting the Earth [3]. The Space Surveillance and Tracking segment is responsible for the detection and prediction of the movement of space debris in orbit around the Earth in order to avoid the degradation of space activities due to collisions. An SST system can be considered as a service oriented system based on the data coming from one or several sensors, typically telescopes or radars. The design of an efficient SST system is critical to maintain and predict the orbits of space debris.
With that goal in mind, the software tool SPOOK (SPace Object Observations and Kalman filtering) has been developed. Throughout this paper, the capabilities of SPOOK are outlined. First, an overview of the tool is given explaining its basic functional architecture. After that, the functionalities of the different modules are described, providing examples of the results that can be achieved by using this tool. Finally, conclusions about the capabilities of the software are drawn and future developments are suggested.

2. Overview of SPOOK

SPOOK is an integrated framework developed to evaluate the performance of SST systems. SST systems can be heterogeneous and consist of one or several sensors distributed across the world or operating in space, tracking different kinds of space debris. The sensor network is linked by a common core in charge of processing the measurements and providing different services. This structure is replicated in the software architecture. It is composed of several interrelated modules that can be divided into three main categories or layers:

- **Simulation Layer**. This includes the Population Generator Module as well as the Sensor Simulator Module. These two modules intend to simulate a population of Earth orbiting objects as well as a network of sensors taking measurements of these objects. These are two highly-configurable modules that bring large flexibility to define different object populations and observation scenarios.

DOI: 10.13009/EUCASS2017-553

- **Analysis Layer**. This category groups the Measurements Analysis module, that computes different metrics to evaluate the performance of the surveillance strategy as well as the Orbit Determination module, responsible of estimating the state of the tracked objects based on the measurements. These two modules can be run independently from the Simulation Layer, using synthetically generated or real world measurement data.
- **Interface Layer**. Interface with the program is established via text files and via our custom designed software visualization tool. Great emphasis has been put into using Consultative Committee for Space Data Systems (CCSDS) standard formats for both measurements and object ephemerides files.

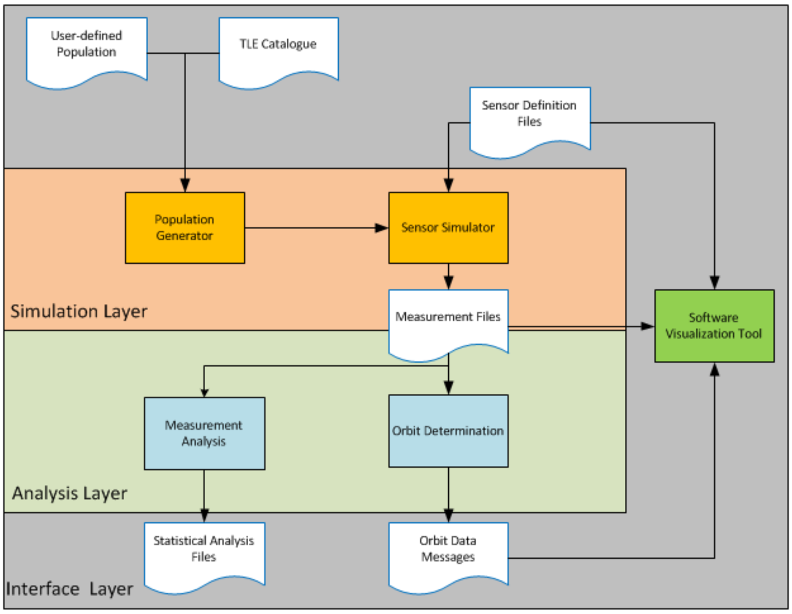


Figure 1: SPOOK functional diagram

Figure 1 shows how all the different modules and layers interact with each other. The tool is meant to be able to simulate the whole SST system, from the population and measurement generation till the analysis of the results both in terms of surveillance performance metrics and/or orbit prediction. A typical simulation would start by generating an initial debris population. This population can be formed by multiple objects in different orbital regions. After that, the measurements are generated taking into account the characteristics of the defined sensors (type of sensor, performance model, localisation,...) and the observation strategy (surveillance, tracking,...). The result of this simulation is a set of measurements emulating the performance of real sensors. These measurements can be output in the form of a text file, following the CCSDS Tracking Data Message format or other proprietary formats, or they can be directly fed into the Analysis Layer. The Analysis Layer will post-process the generated measurements in order to compute different metrics about the performance of the observation strategy such as observation rate, revisit times, etc and/or use the measurements to produce an orbit determination using one of the available algorithms. A representative space debris population can be made of several thousands objects, e.g. public Two Line Element (TLE) catalogues usually account for more than 15000 objects. A simulation of such a large population can become computationally expensive leading to large simulation times. In order to speed up execution, the tool has a parallel mode to take advantage of modern multi-core machines. Each one of the objects can be simulated by independent cores, with substantial reduction in computational time.
It is worth mentioning that the Analysis Layer can be run independently from the simulator layer. Real-world data coming from sensor operators can be used as an input for the Analysis layer to compute performance metrics or perform orbit determination.

DOI: 10.13009/EUCASS2017-553

3. Simulator Layer: Population Generator

The Population Generator module is responsible for defining the characteristics of the objects to be tracked by the sensors. The population can be defined via two different forms:

- TLE files.
- User defined objects. All the features of the objects such as the state vector, size, ballistic coefficient, etc. can be directly defined by the user.

In the case of populations defined via TLE catalogues, extra information about the physical properties such as area or size can be input via a separate file. The use of this file is not necessary for user-defined objects. Besides defining the set of objects to be simulated, this module is also in charge of:

- Classification the objects into the different orbital regions (LEO, MEO, GEO and HEO with distinction between resident and transient objects).
- Application of optional user-defined filters to the objects (processing of specific orbital regions only, size, etc...)
- Generating the reference trajectory for each object, propagating their dynamics during a predefined time-span.

As an example a TLE catalogue with 15404 objects is used to define an initial population. Figure 2 shows the number of objects per orbital region. In figure 3, it is shown how the objects are distributed showing inclination vs semi-major axis on the left and eccentricity vs semi-major axis on the right.

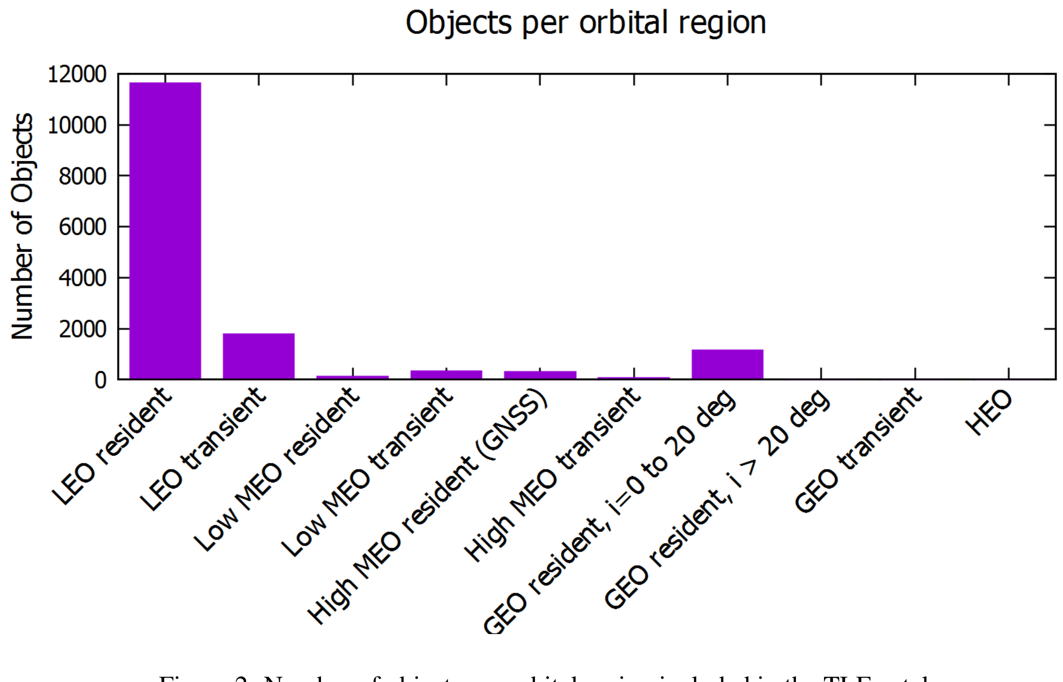


Figure 2: Number of objects per orbital region included in the TLE catalogue

DOI: 10.13009/EUCASS2017-553

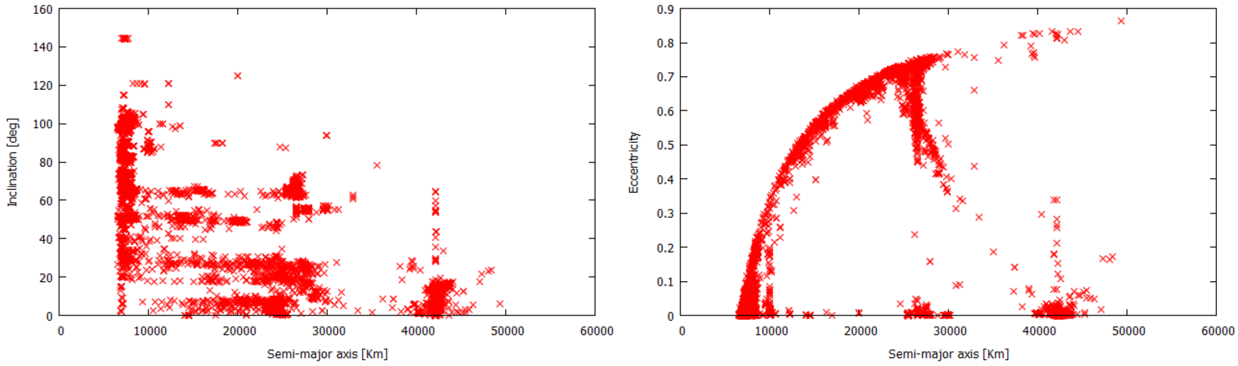


Figure 3: Distribution of the catalogue objects according to their orbital characteristics. Left: inclination vs semi-major axis. Right: eccentricity vs semi-major axis.

4. Simulator Layer: Sensor Simulator

This is one of the most important modules of SPOOK. It generates the observation information (measurements) associated to the simulated objects. It enables great flexibility to simulate different kind of sensors and observation strategies. An observation architecture definition includes the total number of observers, their locations, types, characteristics and the observation strategy to be followed per sensor.
This module can simulate the following measurement types:

- **Ground-based radars**. Radar sensors can generate any combination of the following observables: Azimuth and Elevation angles (deg), Range (km) and Range-rate (km/s) information. The location of the observer is fixed using geodetical coordinates. Both surveillance and tracking type radars can be simulated.
- **Space-based radars**. Same measurement primaries as the ground-based radars, but with a their position bound to a specific user-defined orbit.
- **Ground-based optical measurements**, using telescopes that provide Right Ascension and Declination angles (deg).
- **Space-based optical measurements**, with the orbit defined by an initial state vector or the classical orbital parameters.

The observers can be defined in a highly configurable and flexible way. It allows to define configurations made up of one or several sensors of the different types. The observation strategies can be grouped in three main categories:

- **Surveillance mode**. It uses a given pointing pattern.
- **Tracking mode**. The pointing of the sensor is automatically computed using the reference orbits generated by the Population Generator module.
- **User-defined observation strategies**. It allows to define a pointing of the sensor over time via a pointing definition file or the simulation of more complex observation strategies. The use of custom-design strategies is specially suited for space-based observers.

The process of simulating the measurements consists of first determining if the object is detectable by the sensor and then generating the observables. For each object and observer, the detectability is evaluated in several steps at each possible measurement time. Starting from the position of both the object and the observer and using the pointing information of the sensor (externally defined or previously computed based on the simulated observation strategy), a series of filtering steps are sequentially applied. These filters are based on both geometrical and physical constraints. Examples of geometrical constraints are the evaluation of a direct line of sight between object and observer, minimum elevation angle, sensor Field of View (FoV), etc). Examples of physical constraints are illumination conditions of the targets for optical sensors, or a radar performance model based on the detectable radar cross section for radar observers. The use of successive filtering steps allows the early discard of non-detectable measurement times, avoiding expending extra computational effort.

DOI: 10.13009/EUCASS2017-553

After assessing the detectability of a pass, the observables must be evaluated. This is done in three successive steps. The first step generates an initial value of the observables. This initial value is computed taking only into account the relative geometry between the observer and the target. In a second step corrections to the measurements are applied to take into account atmospheric and light speed delay effects. Finally, and to simulate more realistic measurements, noise is applied to the measurements following the characteristics specified for each sensor. Figure 4 shows this measurement generation process.

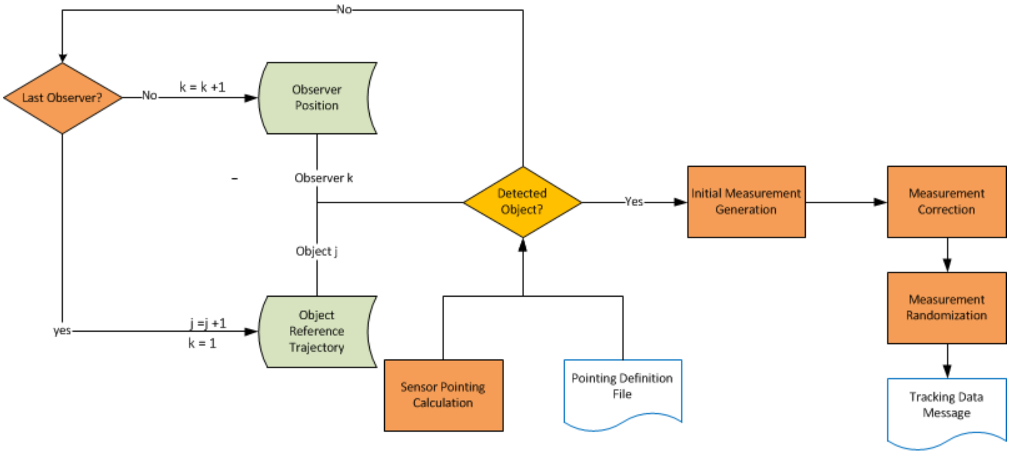


Figure 4: Scheme of the measurement generation process

5. Analysis Layer: Measurement Analysis

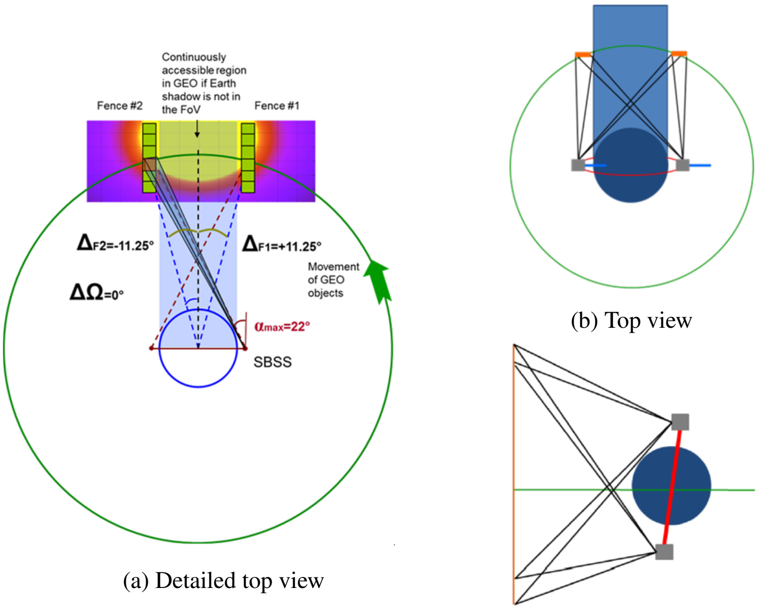
This module computes statistical information about the data generated by the Sensor Simulator or externally provided. It produces information to analyse the performance of the simulated observation strategy without having to run the more computationally expensive Orbit Determination module. It produces fast indicators about the performance based on the availability, duration and frequency of the measurements. All the indicators are produced globally (for all the objects and observers), per orbital region, per object and per observer. The global information allows for a quick assessment of the observation strategy while the segmentation of the information enables for a more in deep analysis and design optimisation. The indicators produced by the code are:

- **Number of detections.** This represents the number of tracks produced by an object. A track is defined as a single crossing of the object per the FoV of the sensor with a successful detection. It allows to discriminate which objects have been detected and how many times.
- **Observability.** Defined here as the duration of the track, the time since the object enters the sensor FoV and it is detected until the last measurement of that track. It plays an important role in getting a first estimation of the orbit. The longer the arc is, the more accurate the Initial Orbit Determination (IOD) will be. The mean, maximum and minimum, as well as the spread of these values (via standard deviation) are computed to capture all the information of the distribution of measurements.
- **Gap time**. Is the duration of the gaps between re-observations of the objects. It has great impact on the capability to maintain an object in a catalogue. Long gaps between set of measurements means a lower rate of orbit update, with the subsequent degradation of accuracy in Orbit Determination. It is specially critical for tracking sensors. The pointing of these observers are based on the propagation of the estimated orbit into the future. Longer propagation periods enhances the risk of loosing the object due to inaccurate initial conditions and/or unmodelled effects.
- **Average Revisit Frequency**. Closely related with the Gap time, it is defined as the average number of times per day that an object is detected by a sensor or by the whole sensor network.

To show the capabilities of this module, an example is presented below using simulated data. This example uses an observation strategy that aims to maximise the coverage of the objects in the GEO-belt. This strategy is called the Geostationary Orbit Fence scenario [4] and it is based in a single space-based optical observer flying in a dawn-dusk LEO object. Figure 5 shows the set up of the observation scenario. The green line represents the orbit of a GEO object

DOI: 10.13009/EUCASS2017-553

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 in Figure 5) to get the best illumination conditions. The FOV of the sensor will move between several fields within the two defined fences as shown in the top view in Figure 5. The effect of the different system parameters (sensor FoV, orbital parameters of the observer, etc...) in the overall performance of the system could be quickly investigated using this module. As an example, the effect of the number of fields per fence on the total achievable coverage will be presented.



(a) Detailed top view

(b) Top view

(c) Side view

Figure 5: GEO Fence Scenario set-up [4]

The same TLE catalogue used in Section 3 will be used. As this strategy is meant to survey the GEO-belt, the option to only process objects belonging to that region is used, speeding up the simulation. From the 15404 simulated objects, 1170 are found to belong to this region. Table 1 links the number of fields or pointing locations within each field with the total coverage achieved of the GEO objects. This analysis shows that to achieve a minimum of 98% of this region at least 5 fields should be selected. Figure 6 shows the temporal evolution of the detection. It can be seen how this observation strategy achieves its maximum coverage after 24h (one orbital revolution of the GEO object).

Table 1: Total coverage of the different GEO Fence strategies

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

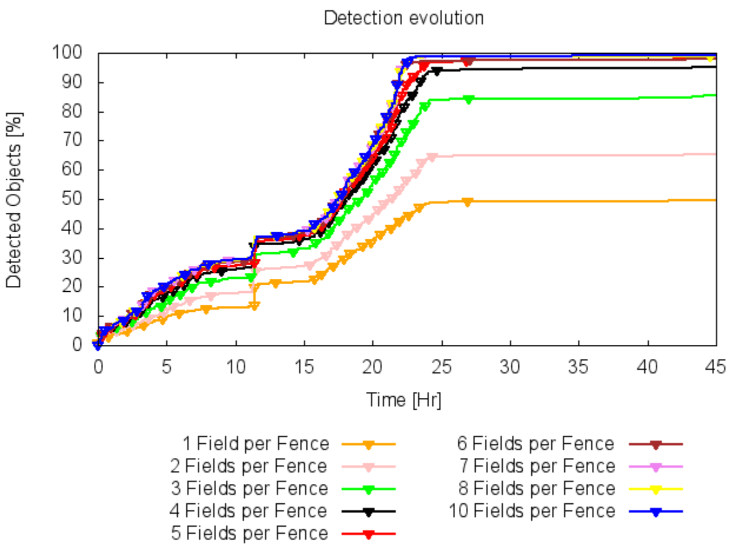


Figure 6: Temporal evolution of the objects detection

DOI: 10.13009/EUCASS2017-553

Figure 7 shows plots of the other metrics computed by the module. The left plot shows the distribution of the detected objects per mean gap time, the central plot shows the distribution per track duration while the last plot represents it per average revisit frequency. In all cases, the plots are computed for the case with 5 fields per fence and a 3° FoV.

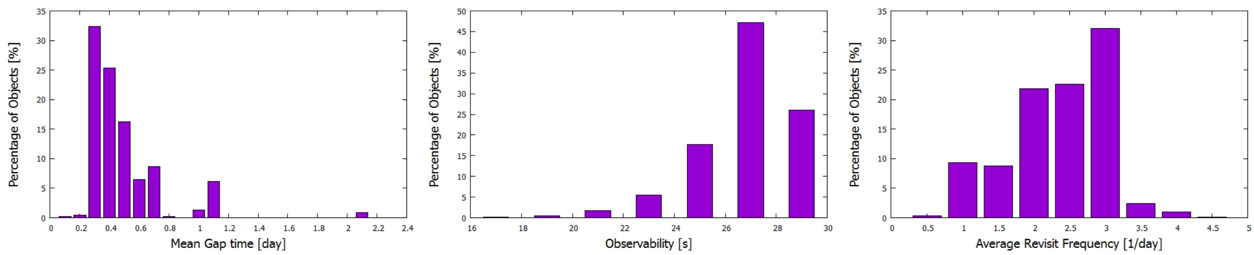


Figure 7: Metrics of the observation strategy using 5 fields per fence

6. Analysis Layer: Orbit Determination

The second module of the Analysis Layer is the Orbit Determination module. This module is in charge of processing the measurements generated by the sensor simulator or externally supplied to produce an estimation of the satellite state vector and its uncertainty. Optionally, object specific parameters such as ballistic or Solar Radiation Pressure coefficients can be estimated as part of the Orbit Determination. This module propagates the object state vector and uncertainty into the future (the future being defined as the time after the last available measurement) that can be used for Collision Avoidance, sensor scheduling, re-entry analysis, etc. This module is made up of 4 basic components.

- **Orbit Propagator**. It uses a numerical integrator to propagate an initial state vector. Besides the spherical gravity field of the Earth, the propagator can take into account other perturbations that affect space debris (non-spherical gravity field, atmospheric drag, Solar Radiation Pressure, Third-bodies perturbations...). This propagator is the same one used by the Population Generator module. However, the force models used by the two modules can be different if selected by the user.
- **Uncertainty Propagation** Knowing the state vector of the object is as important as knowing the uncertainty assigned to that prediction and how this uncertainty evolves. This is done in SPOOK by means of covariance propagation, using the same force models as for orbit propagation. Three main methods of covariance propagation are available:
  - Covariance Propagation based on local linearisation methods.
  - Covariance Propagation based on the Monte Carlo Method.
  - Covariance Propagation based on the Unscented Transformation.
- **Initial Orbit Determination** This component produces an initial estimation of the state vector based on a limited set of measurements with no prior knowledge of position and velocity of the object. The set of measurements used for the IOD method can be manually selected by the user via configuration files. It uses the Gauss algorithm to find three position vectors based on the angular data and range information if available (radar observers) and the Gibbs, Herrick-Gibbs or Lambert-Gauss techniques to find the middle velocity vector.
- **Orbit Determination** This component is in charge of producing a refined estimation of the object state and, optionally, other specific parameters based in all the measurements collected. It uses an initial estimation of the state vector that can come from the Initial Orbit Determination or be defined by the user. Three different methods are available to produce orbit determination:
  - Extended Kalman Filter.
  - Unscented Kalman Filter. A modification of this algorithm to take into account uncertainties in modelling parameters, the Unscented Schmidt-Kalman Filter [5], is also available.
  - Sequential Batched and Non-Sequential Non-linear Weighted Least Squares.

When simulated data is used to perform an orbit determination, the reference trajectory used to simulate the measurements is known. Therefore the code can automatically compute information about the errors of the estimation that

DOI: 10.13009/EUCASS2017-553

allows for a quick assessment of the success of the orbit determination. When only external measurements are supplied, no reference trajectory is available and this information cannot be computed. Among other proprietary formats, the results of the Orbit Determination (state vector, covariance, parameters) can be output using the CCSDS defined Orbit Data Message formats.
As an example of this module a simulation of an Orbit Determination of a Galileo Satellite is performed. The measurements are taken during a period of 15 days by a single ground-based optical telescope in tracking mode. Table 2 shows the location of the ground observer as well as the 1-σ accuracy of the angular measurements taken by the telescope. The value selected for this accuracy is a typical value for a small aperture telescope. Table 3 shows the orbital parameter of a typical Galileo satellite. No prior information about the state of the satellite is known by the orbit determination. An Initial Orbit Determination is performed followed by a Refined Orbit Determination. The algorithm chosen is the Unscented Kalman Filter.

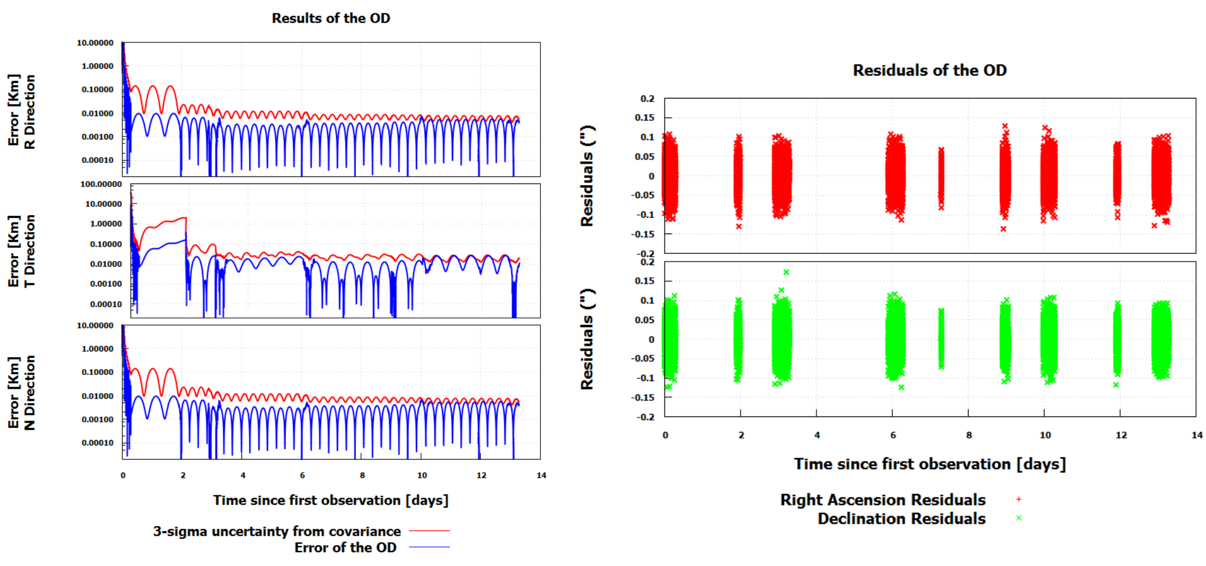
Table 2: Observer Sensor Parameters

Ground-based Optical Telescope	
Latitude	5.17°
Longitude	-52.65°
Altitude	3 m
Right Ascension/ Declination 1-σ	2”

Table 3: Galileo satellite orbital parameters

Galileo Satellite	
Semi-major axis	29599.8 Km
eccentricity	0
Inclination	56°
RAAN	317.632°

The results of the Orbit Determination can be seen in Figure 8. On the left, it shows the errors of in each one of the three directions in the typical Radial-Tangential-Normal frame of reference (in blue). In the same image, in red, the predicted 3-σ uncertainty from the covariance can be seen. The plot shows how the initial error is big, as it is the result of an Initial Orbit Determination, and it decreases with the successive filtering updates. On the right plot the distribution of the residuals is shown. It can be seen how all the residuals are below the 1-σ of the sensor. Very accurate results are obtained as the force model used to simulate the data and to perform the orbit determination is the same.



(a) Errors and predicted uncertainty of the orbit determination estimation

(b) Residuals of the orbit determination

Figure 8: Results of the orbit determination of a Galileo satellite using a single optical observer

DOI: 10.13009/EUCASS2017-553

7. Interface Layer: Software Visualisation Tool

The main mode of interfacing with the program is via the input and output files. However, there is also a custom-built software visualisation tool designed to show the results of the orbit determination. This tool can dynamically show the reference orbit of the object, the sensors location (both for ground and space-based observers), the estimated orbit and the measurement times. Figure 9 shows a static representation of the Galileo orbit estimated in the previous example. In this case, and for a better graphical representation, the orbit has been estimated during 10 days of observations.

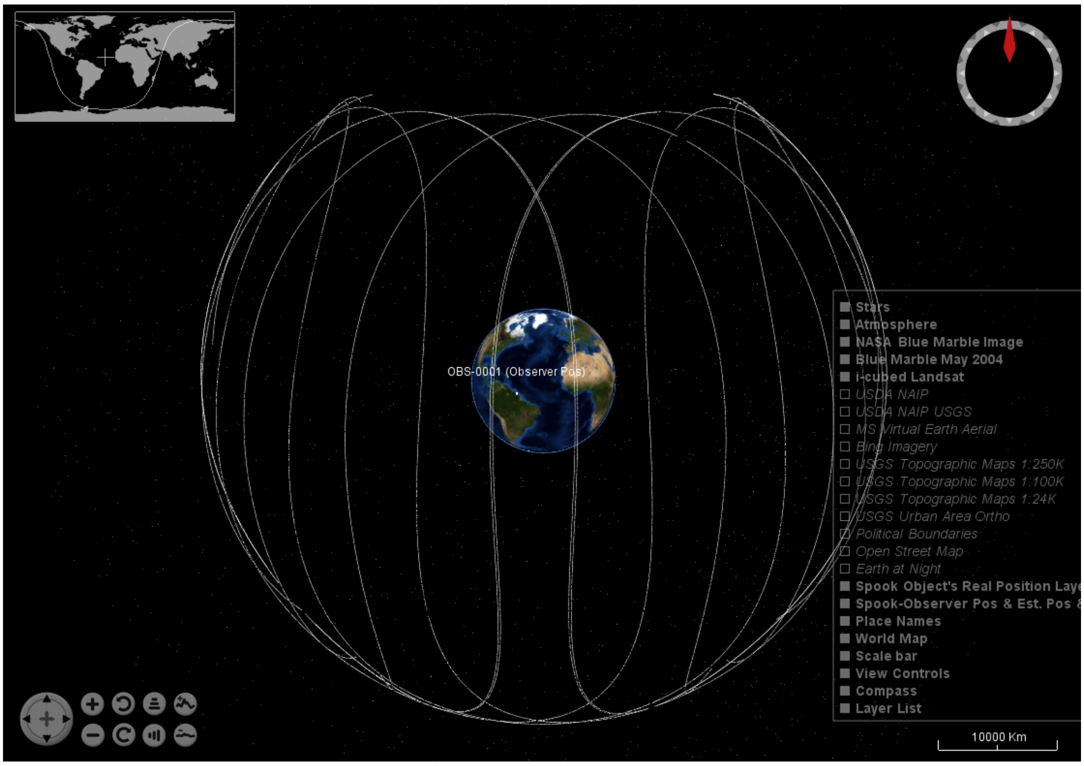


Figure 9: SPOOK visualisation tool

8. Conclusions and Future Developments

Throughout this paper a basic overview of the tool has been given. The tool is able to simulate space debris population defined either via TLE or configuration files. This population can then be used to generate measurements with a simulated sensor network. The tool has great flexibility to define different observer types and surveillance strategies in order to be able to simulate different sensor architectures. The measurements produced by a particular sensor configuration can be analysed via the Measurement Analysis module to produce fast indicators about its performance. Finally, the Orbit Determination module is in charge of predicting both the object state vector and specific parameters as well as the covariances based on simulated or real-world measurements. It is worth mentioning the use of CCSDS standard formats for interfacing with the tool. The software visualisation tool allows to get a graphical representation of the results achieved by the software.

The main area of interest for future developments is to include a correlation module that allows the tool to automatically solve the data association problem between the measurements obtained and the objects to which they belong both for the simulated measurements as well for real-world data.

DOI: 10.13009/EUCASS2017-553

References

[1] Inter-Agency Space Debris Coordination Committee. IADC Space Debris Mitigation Guidelines, 2007.

[2] National Aeronautics and Space Administration. Monthly number of objects in earth orbit by object type. *Orbital Debris Quarterly News*, 2, 2016.

[3] European Space Agency. How many space debris objects are currently in orbit?, 2016.

[4] Jens Utzmann et al. Space-based space surveillance and tracking demonstrator: mission and system design. *65 International Astronautical Congress*, 2014.

[5] Jason Stauch and Moriba Jah. Unscented Schmidt-Kalman Filter Algorithm. *Journal of Guidance, Control, and Dynamics*, 38(1):117–123, 2014.