ENHANCEMENT OF THE PROOF SOFTWARE FOR SPACE DEBRIS OBSERVATION MODELLING AND SIMULATION

Lorenz Böttcher[1], Christopher Kebschull[2]

[1] *IRAS, TU Braunschweig, Hermann-Blenk-Str. 23, 38108 Braunschweig (Germany), l.boettcher@tu-braunschweig.de*
[2] *OKAPI:Orbits, Hermann-Blenk-Str. 23, 38108 Braunschweig (Germany), christopher.kebschull@okapiorbits.space*

ABSTRACT

Within the framework of ESA's Space Situational Awareness program, the software tool Program for Radar and Optical Observation Forecasting will be extended and revised with regard to new functionalities and improvements of the modelling approaches. While the existing software already supports fundamental capabilities in terms of observation planning, the sphere of interest has shifted in recent years towards the assessment of design criteria and evaluation of overall performance of space surveillance, tracking and cataloguing capabilities. The planned and in progress upgrades and new features of the ESA activity are presented. The three major foundations of the updated tool are addressed and a first insight into the ongoing activity is provided. In the context of this proceeding, special attention is allocated to the implementation of the new software framework, the upgrade of observation models and the enhancements in terms of software accessibility.

1 Introduction

The observation of the near-Earth object population takes a prominent role in the context of design and operational phases of space missions, as well as in the validation of debris environment models. The software Program for Radar and Optical Observation Forecasting (PROOF) originally was designed to perform the latter validation tasks for the Meteoroid and Space Debris Terrestrial Environment Reference (MASTER) object population. In this regard, actual observation campaigns are re-modelled, simulated and compared to the real-world data. The comparative validation bases on observation characteristics, as detection rates, object size and orbital element distributions. The core of the currently updated PROOF-2009 software consists of the crossing analysis, which performs the geometrical filtering of objects traversing the Field Of View (FOV) of an Earth- or space-based sensor. In a second step the crossing objects are analysed with respect to the core's radar or optical performance models, which provide detailed information on the detectability and signal characteristics of the crossing object and superimposed background noise [1]. In the context of the mentioned detection performance models, the PROOF-2009 software is able to simulate passive optical and active mono- or bi-static radar campaigns.

Already providing those fundamental observation modelling capabilities, the focus of interest has shifted in recent years towards the assessment of sensor and observation performance characteristics. Especially in the context of planning, designing and conducting costly observation campaigns, the retrieval of such information becomes essential. Furthermore, such an observation modelling tool may support the evaluation and understanding of real-world observation campaigns. This change of targeted use cases and expansion of the application domain has led to a series of proposed enhancements, which currently are implemented by a consortium of European partners (Institute of Space Systems TU Braunschweig, OKAPI Orbits, Deimos Space, Astronomical Institute University of Bern) under ESA contract.

The ongoing activity, leading to a new PROOF-3 software release bases on three major pillars, which are explained in detail within the following sections. The first major contribution to this activity focuses on the introduction of modularity, in order to increase maintainability and traceability from a programming point of view and allow for a high level of customisation of the software by generating new access points. The second major task is committed to the actual update of the underlying observation models and expansion of possible use-case. Finally, the third major contribution consists in the improvement of accessibility, as well as in the verification and validation of the new software release.

2 Modular design

The increased demands in terms of interfaces, modelling capabilities and functionalities led to the decision of re-implementing the entire tool to allow for maximal flexibility, as well as to consider more contemporary standards in software engineering. A high level of flexibility is achieved by two architectural concepts, which are adopted for the implementation of the new software tool.

2.1 Framework

For more flexibility, the architecture is based on a modernised frontend/backend paradigm, where the frontend uses web standards for the user interaction and visualisation. The backend uses a refreshed object-oriented FORTRAN core, which is encapsulated in a shared library ("lib" in Figure 1). The frontend is written in TypeScript (Angular) and relies on Electron to create a desktop application runnable on Windows 10, macOS and Linux. It communicates with the backend using an HTTP interface. The backend application is called PRESTO (PROOF over REST). It is an executable and it uses the PROOF-3 shared library. This way the GUI can directly connect and control PROOF-3. As in previous versions PROOF-3 can also be controlled via Command Line Interface (CLI). A respective executable is provided and called Command Line Tool (CLT). The PROOF shared library can also be called from other third party wrappers, that can bind to shared libraries (*.so, *.dylib or *.dll files respectively). Developers will be able to interact with PROOF-3 using the Getter-Setter-Checker (GSC) Application Programming Interface (API). It allows to access the underlying data structures and manipulate it based on identifiers and the underlying class structure. Functionalities, like the crossing analysis or the detection analysis, are encapsulated in module classes and called plugins. These plugins are referenced in the core and the main loop of PROOF-3. The core supplies a core-model which holds all respective data of the simulation. By this approach a separation of the software workflow and underlying data structures is achieved.

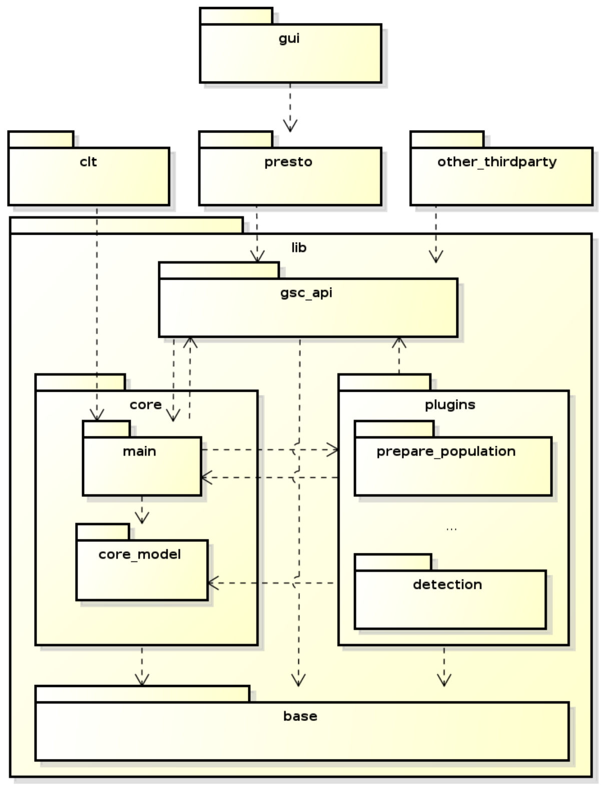


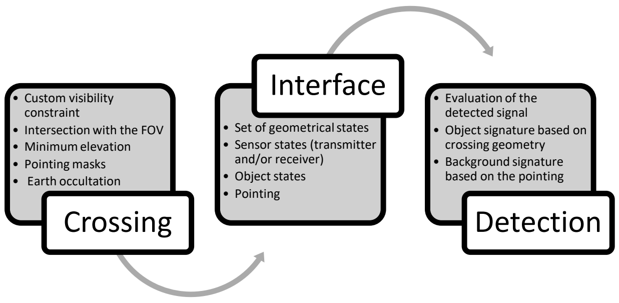
Figure 1: PROOF-3 architecture overview.

2.2 Separation of functionalities

As stated in the introduction, the geometrical crossing analysis and the detection performance models comprise the core of the PROOF software. Accounting for the fact, that most of the crossing and detection related input configurations are not coupled, the decision was made to separate the two plugins, to make them callable independently of each other. In a similar manner the final analysis step, which comprises the evaluation and visualisation of output can be executed independently.

The main goals of such separation are two-folded: On the one hand side it allows for efficient re-evaluation of parts of the software for the case, that only minor configurational changes were applied. On the other side the separation provides additional access points, which may be used to plug-in custom tools, which originally have not been considered within the default use cases of PROOF-3.

The workflow for the separation of the crossing and detection related properties is illustrated by Figure 2.


Custom visibility constraint
Intersection with the FOV
Minimum elevation
Pointing masks
Earth occultation
Crossing
Interface
Set of geometrical states
Sensor states (transmitter and/or receiver)
Object states
Pointing
Evaluation of the detected signal
Object signature based on crossing geometry
Background signature based on the pointing
Detection
Figure 2: Workflow for the separation of the crossing and detection analysis.

The crossing analysis can be understood as a geometrical filter considering different visibility conditions. The latter ones mainly comprise the constraint, that the object to observe has to be contained in the FOV of the sensor (modelled as right circular cone or rectangular frustum). Additional constraints at the sensor location are considered: For Earth-based sensors those are natural local horizon/minimum elevation constraints and artificial time-variable azimuth-elevation exclusion masks. For space-based sensors the Earth disk and surrounding thin layer of atmosphere represents a natural limitation of visibility. Based on a numerical step size control the crossing plugin then retrieves periods of the simulation time, for which the respective visibility conditions are fulfilled for an object of the population. The crossing period is sampled by providing a time-tagged set of geometrical information for the retrieved crossing arc. Those geometrical states of object, sensor(s) and pointing of the Line Of Sight (LOS) then represent the actual interface between the crossing and detection plugin.

Based on the geometrical information provided through the interface the detection plugin determines the object signature at the receiver location in accordance with the underlying sensor performance model. Additional signature related constraints are considered (e.g. bright celestial bodies close to the FOV and background signatures for the optical performance model).

Through the implementation of this separation and the associated interface users have the possibility to provide a custom set of geometrical states and perform the detection analysis through PROOF or they can refer to PROOF's crossing analysis and use it within their custom detection models. Referring to the previously explained framework design, the separation of analysis modes is supported by the elaborated plugin structure.

3 Modelling updates

Core of the PROOF-3 software remain the detection performance models and operational modes. In this context, updates to the existing radar and optical performance models, as well as the incorporation of new features and use cases are addressed.

3.1 Optical and radar performance models

The visible and radar wavelength regimes remain the two spectral ranges supported by PROOF, but both detection performance models see a series of upgrades and extensions. Furthermore, the new software considers measurement uncertainties for the evaluation of sensor performance characteristics.

The optical performance model is extended by an additional active Satellite Laser Ranging model (SLR). For this new model a perfect mechanical tracking performance is assumed (modelled by local horizon crossings). The detectable number of photoelectrons n_R then can be determined in analogy to the radar link equation 1 for a single laser pulse [2].

$$n_R = \eta_T n_T \cdot T_\alpha^2 T_C^2 \left(\frac{1}{4\pi R^2}\right)^2 \cdot \sigma \cdot \eta_R \eta_Q A_R \qquad (1)$$

Here, $n_T = G_T E_T \frac{\lambda}{hc}$ is the emitted number of photoelectrons, defined by transmitter gain G_T, transmitter power E_T and wavelength λ. The transmitter's optical transmissivity is represented by η_T. Equation 1 then accounts for the two-way signal path by considering atmospheric and cloud transmissivity T_α and T_C, as well as the signal slant range R. The detectable number of electrons at the receiving telescope furthermore depends on the object's optical cross section σ, the receiver's aperture area A_R and the associated optical transmissivity η_R and quantum efficiency η_Q. The detection probability at the receiver then follows from Poisson statistics (depending on the detector system) and under consideration of the effective signal ratio, relating the required number of signal photoelectrons to the superimposed background photoelectron numbers. In this context, the presence of SLR retro reflectors are considered as additional object attribute for the deterministic population. Apart from the incorporation of the active SLR model, the possibility to generate realistic image frames with passive optical telescopes is reactivated. Here, the image capture rate or specific time-tags can be specified by the user in order to provide full-frame optical outputs.

For the radar performance model a series of different upgrades are foreseen in a similar manner. Those comprise simplified coherent integration of pulses, consideration of side-lobe detections and the possibility to use and/or customise phased array scan patterns as exemplary seen in Figure 3. For this case, the orientation of the LOS is specified as sequence of pointing angles (local azimuth and elevation or inertial right ascension and declination), for which the time dependency is defined by the number of pulses per orientation, pulse duration and the associated repetition frequency. Furthermore, the bi-static restriction is loosened to enable the analysis of multi-static scenarios. In this regard PROOF-3 supports the analysis of entire sensor networks, consisting of a single transmitter and a series of different receivers.

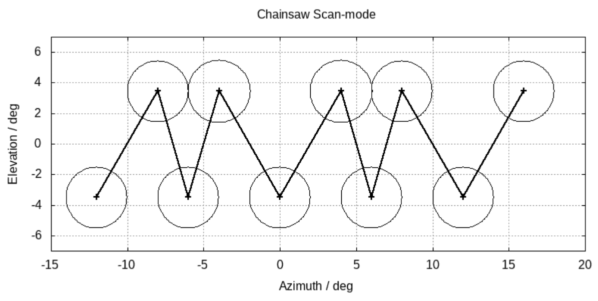


Figure 3: Exemplary phased array scan pattern (chainsaw pattern).

Independent of the choice of the detection performance model the resolution of the crossing arc of the orbital trajectory is increased. This allows to not only perform the detection analysis for a single point of the crossing period (for PROOF-2009 the point of closest approach), but rather for a sample of the complete crossing arc. This higher resolution is achieved by interpolating in the crossing-detection interface data described in 2.2.

For both detection performance models it is foreseen, to provide and evaluate information on measurement uncertainties. Those uncertainties are modelled as Gaussian noise, where the parameters of the distribution are either directly provided by the user or are determined by the detection performance models in dependency of the actual signal-to-noise ratios (SNR). For the optical performance model the standard deviation of angular measurement errors σ_A can be described in the most general form by equation 2.

$$\sigma_A = \sqrt{\left(\frac{k_1}{SNR}\right)^2 + k_1^2} \qquad (2)$$

Close to the detection limit k_1 becomes the dominant driver of measurement uncertainty, while k_2 is accounting for systematic errors and becomes especially relevant for the high SNR regime.

For the radar performance model angular, as well as range uncertainties are considered [3]. The standard deviation of range uncertainty σ_R is given by equation 3.

$$\sigma_R = \frac{c}{2B\sqrt{2 \cdot SNR}} \qquad (3)$$

It depends on the SNR and the range resolution, expressed by bandwidth B and speed of light c. Angular uncertainties can be approximated by equation 4.

$$\sigma_A = \frac{\theta_A}{k_M\sqrt{2 \cdot SNR}} \qquad (4)$$

Here, θ_A corresponds to the beamwidth in the considered plane in which the angle is measured (typically azimuth and elevation), while k_M represents the difference pattern slope of the sensor.

In order to access sensor performance characteristics, the measurement uncertainties are mapped to orbital state covariances. Those may be evaluated for a single detection, for the entire detected arc of a single crossing or for the complete set of crossings for a single object.

3.2 Use cases

In addition to the upgrades of the performance models the extension and enhancement of use cases mainly address questions in the space surveillance and tracking area. In this regard features from the two ESA tools LIght Space Surveillance radar System Simulation Approach (LIS^{4}A) and Near Earth Object Population Program (NEOPOP) are adopted and migrated to the new PROOF-3 tool. The considered features mainly account for automated parameter studies and the evaluation of observation performance characteristics. Furthermore, new analysis modes are introduced.

The LIS^{4}A tool is used for the quick evaluation of mono-static radar performance characteristics accounting for the detected number of objects of a given population with respect to the overall number of objects [4]. In the context of the here presented activity, the LIS^{4}A feature for performing automated parameter studies is of special interest. This functionality is incorporated into the new PROOF-3 software, allowing to assess the sensor sensitivity by performing systematic parameter variations. In order to evaluate parameter dependencies, the variation of up to two sensor related variables at a time is supported. Performance characteristics then are reflected by the respective crossing and detection frequencies. Furthermore, basic cataloguing capabilities can be evaluated based on the provision of minimum number of detections and a lower bound of detection period.

The considered NEOPOP features comprise the definition of observation strategies and the reactivation of observation performance characteristics. The former one allows to provide a NEOPOP-style survey strategy, which by its part can automatically adapt sensor related parameters for arbitrary instances of the simulation time. This approach prevents the user from having to manually split and configure the simulation whenever adjustments are made to the sensor configuration. The latter observation performance characteristics allow to access statistical information on the coverage of the analysed object population. Those characteristics are either geometrical ones (azimuth-elevation-frequency distribution of objects crossing the local horizon and eventually being detected) or population related ones (crossing and detection coverage of different size regimes of the analysed population) [5].

By the two new processing modes derived from LIS^{4}A and NEOPOP features, the internal workflow and call order of the new PROOF-3 tool can be described as shown in Figure 4.

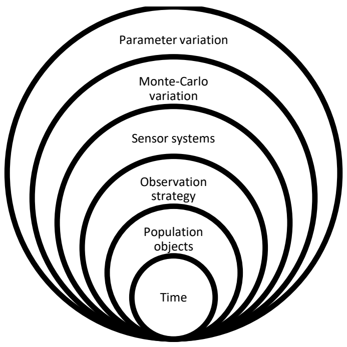


Figure 4: Internal workflow and processing loops starting from the highest hierarchy level in the outermost circle, to the lowest one in the innermost circle.

The outermost loop comprises the sensitivity analysis, which is responsible for the systematic variation of sensor characteristics. On the next-lower hierarchy level the Monte-Carlo randomisation of orbital elements of MASTER population objects takes place for the statistical run-mode. For the deterministic processing-mode, which is used to study non-statistical Two-Line Element (TLE) populations, this loop is skipped. The next-inner loop then iterates over all specified sensor systems, whereby multi-static scenarios are resolved by considering one pair of transmitter and receiver at a time (one pair of transmitting and receiving FOV and signal path respectively). The two innermost loops then run the desired analysis modes for all population objects and the respective simulation time.

With regard to new analysis modes, the current set of PROOF-2009 functionalities gets extended by a new tracking feature. The tracking analysis does not aim for the detailed modelling of the sensor tracking motion, but rather provides statistical information on pass frequencies, coverage and detectability of objects crossing the local horizon. As for the SLR a perfect mechanical tracking is assumed. Accordingly, this tracking mode allows for assessing statistical cataloguing performance characteristics based on the chosen sensor network and configuration. Furthermore, the new PROOF-3 software allows for different levels of detail in terms of analysis. The detailed mode of analysis is applicable for deterministic object populations, incorporates a full-force propagator interface and enhanced modelling assumptions (e.g. consideration of atmospheric refraction effects). In order to enable a user to quickly assess supported performance characteristics an additional simplified mode is supported, for which a simplistic, but fast propagator (only accounting for J2-perturbations) is available.

4 Software accessibility

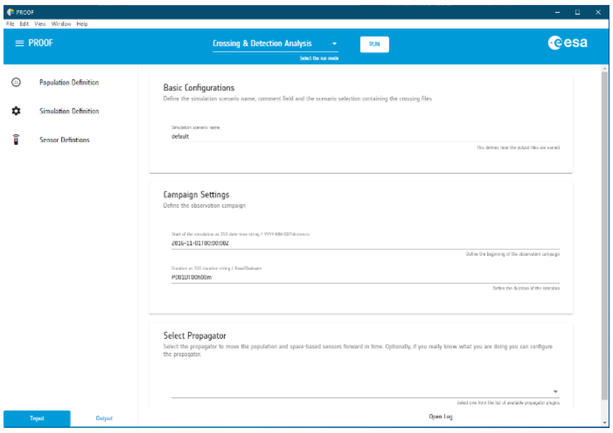
The third major design criterion for the new PROOF-3 tool is to facilitate access to the software in any regard. In this context three aspects are considered, in order to make PROOF-3 available to a larger community. This is mainly achieved by simplifying the use of the software through an intuitive Graphical User Interface (GUI), streamlining used data formats and providing verification and validation test cases.

4.1 GUI

The updated graphical user interface uses web-standards to create a modernised and with ESA's corporate design harmonised look and feel, as shown in Figure 5. The user is greeted with a main window, which shows the top bar, containing the sandwich menu, run-mode selection, run button and the ESA logo. In contrast to previous versions the sidebar on the left serves only a navigational purpose to move between configuration sections. On the bottom of the sidebar the user can switch between the input view and the output view. In the centre of the input view all configurations are performed concerning the simulation, the manipulation of the underlying spacecraft and debris population and detailed sensor configurations.

As part of the usability improvements the radar beam pattern, in its polynomial representation is now visualised (cp. Figure 6.a). The user can configure multiple segments and specify the degree of the polynomial as well as the respective polynomial coefficients.

In a similar fashion the line-of-sight definition is now accompanied by an azimuth-elevation mask. The user can specify in a time-dependent manner when the field of view is blocked. The definition is performed using two azimuth-elevation tuples, specifying the lower and upper bounds of the exclusion area. The spanning area is considered blocked and shown in a polar plot (cp. Figure 6.b).



Select the run mode

Basic Configurations

Define the simulation scenario name, comment field and the scenario selection containing the crossing files

Simulation scenario name
default
This defines how the output files are named

Campaign Settings

Define the observation campaign

Start of the simulation as ISO date-time string / YYYY-MM-DDThh:mm:ss
2016-11-01T00:00:00Z
Define the beginning of the observation campaign

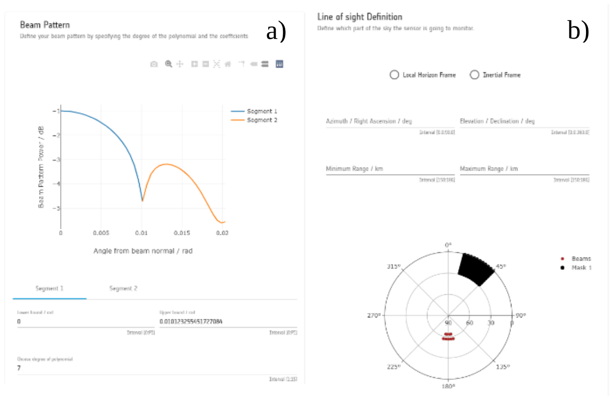
Duration as ISO duration string / PnnnDThhmm
P001DT00h00m
Define the duration of the simulation

Select Propagator

Select the propagator to move the population and space-based sensors forward in time. Optionally, if you really know what you are doing you can configure the propagator.

Select one from the list of available propagator plugins

Figure 5: The main window of the PROOF-3 GUI, with top and side bar and the main input in the centre.



Define your beam pattern by specifying the degree of the polynomial and the coefficients

Lower bound / rad
0
Interval [0:PI]

Upper bound / rad
0.010123255451727084
Interval [0:PI]

Choose degree of polynomial
7
Interval [1:15]

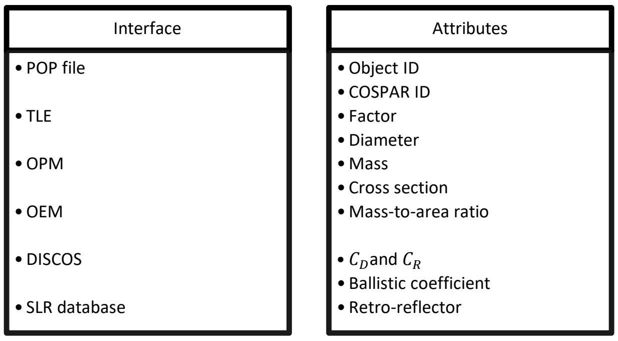
Define which part of the sky the sensor is going to monitor.

Figure 6: Visualisation of a) the radar beam pattern and b) the LOS mask showing the radar beams and the exluded mask area.

4.2 Standardised interfaces

Apart from the backend API and the new GUI described in section 2.1 and 4.1, a series of standardised file formats and interfaces are used by the new PROOF-3 software.

In addition to the already supported TLE format, Orbit Ephemeris Message (OEM) and Orbit Parameter Message (OPM) formats are supported to provide orbital elements and detection related object attributes. When using the GUI, the local object database can be updated via an interface to ESA's Database and Information System Characterising Objects in Space (DISCOS). Each supported input interface has its own initialisation sequence and order by which attributes get associated with the considered object. For information, which are not available in any of the provided interfaces, default values and empirical data are used as fall-back option. The supported interfaces and attributes are summarised in Figure 7.



Interface	Attributes
• POP file	• Object ID • COSPAR ID
• TLE	• Factor • Diameter
• OPM	• Mass • Cross section
• OEM	• Mass-to-area ratio
• DISCOS	• $C_D$and C_R • Ballistic coefficient
• SLR database	• Retro-reflector

Figure 7: Supported input interfaces and object attributes.

To facilitate the exchange of data and simplify post-processing the new PROOF-3 release is able to provide Tracking Data Message (TDM) and Consolidated Laser Ranging Data (CRD) formats. Furthermore, PROOF-3 allows for the integration into ESA's System Performance Simulator.

4.3 Verification and validation

In order to increase acceptance and trust towards the new PROOF-3 release a two-folded verification and validation concept is implemented. It bases on small scale unit tests, comparative tests with PROOF-2009 and external observation simulation tools and the validation by comparing simulated observations of the MASTER population with real-world observation campaigns. The implemented approach allows for a more efficient localisation of inconsistencies and by this increases traceability and maintainability of the tool.

Considering the low-level tests, a Continuous Integration/Continuous Delivery (CI/CD) pipeline is established, which performs automated building and unit tests of the software. Apart from the automated tests, control output is foreseen, which allows for visual inspection and verification of physical plausibility of the underlying models and implemented algorithms. Figure 8 shows the visualisation of such exemplary control output for the sensor location and pointing of the LOS. Subfigures a) and b) show an Earth-based sensor and a LOS, which is defined with respect to the local horizon coordinate frame. For subfigure b) an additional tracking motion about a fixed tracking pole with constant angular velocity is considered. Such motion can be used to implement and optimise optical observation strategies for specific target orbits (this incorporates an additional modulo operation inducing a repetitive pattern, which is omitted for the shown test case). Subfigures c) and d) show a space-based sensor in low Earth orbit, where for the case c) the LOS is defined in inertial coordinates (inertially stabilised spacecraft), while for case d) the LOS is bound to the local sensor location and orientation (gravity stabilised spacecraft). The visualisation of such features allows for quick and intuitive identification of algorithmic programming or modelling errors. Additionally, it supports the retrieval of minor inconsistencies and bugs in the PROOF-2009 release. Those low-level tests are mainly designed to ensure the correct behaviour of the software from a programming and technical point of view.

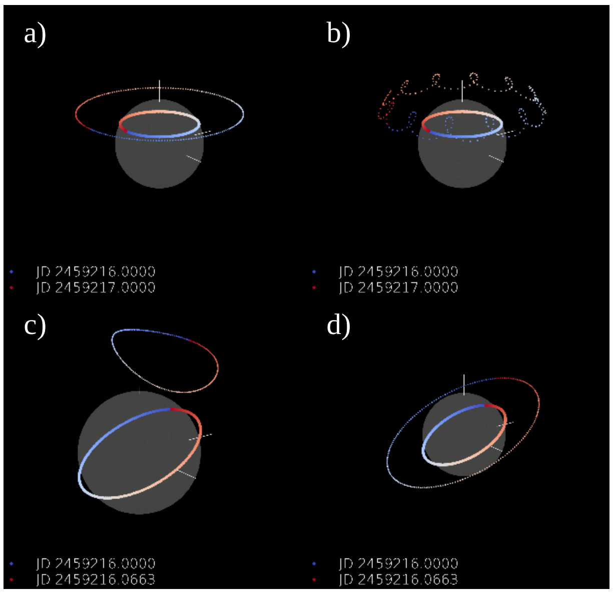


Figure 8: Control output visualisation for the sensor location and pointing of the LOS. The colour gradient represents the temporal evolution indicated by the Julian Dates in the lower left corners. Bold dots represent the sensor location, thin dots the projected LOS.

Apart of those low-level tests intermediate comparative verifications relative to the PROOF-2009 and the external Advance Space Surveillance System Simulator (AS4) software suites are conducted. The high level validation is performed by comparing measurement data with PROOF-3's re-simulation of the real-world observation campaigns. The simulation of observation campaigns refers to the MASTER-8 population. In the context of the radar performance model TIRA (Tracking and Imaging Radar) and EISCAT (European Incoherent Scatter Scientific Association) beam-park experiments are considered. The optical performance model is validated by campaigns of ESA's SDT (Space Debris Telescope). The major observation characteristics as detection rates and size, range and angular distributions of objects are covered by this approach.

5 Summary and outlook

The variety of tasks and the envisaged large domain of application make the ongoing activity a demanding task in terms of software design and implementation. Nevertheless, the expected PROOF-3 release should be a software easy to use for a large community, while providing extensive functionalities and interfaces. In order to accelerate the improvement cycle of the software and to facilitate reuse in other projects, the source code will be made publicly available. The identical considerations apply to the associated Networking/Partnering Initiative Ephemeris Propagation Tool with Uncertainty Extrapolation (NEPTUNE) which is accessed through the Orbital Propagation Interface (OPI) [6]. First scientific results of the activity and more detailed insights into the updated software and underlying models are planned to become available by the end of the year.

6 References

1. Gelhaus, J., Flegel, S., Wiedemann, C. (2011), Program for Radar and Optical Observation Forecasting, Final Report, 21705/08/D/HK
2. Degnan, J. (1993), Millimeter Accuracy Satellite Laser Ranging: A review, *Contributions of Space Geodesy to Geodynamics: Technology*, Vol.25
3. Curry, R. (2012), Radar Essentials, *SciTech Publishing, 10, ISBN: 978-1-61353-007-8*
4. Krag, H. (2008), LIght Surveillance radar System Simulation Approach LISA, User Manual
5. Gelhaus, J., Hahn, G., Müller, S. (2015), Snythetic Generation of a NEO Population, Final Report
6. https://github.com/Space-Systems