DEVELOPMENT OF A SOFTWARE SUITE FOR PERFORMANCE ASSESSMENT OF SST SENSOR NETWORKS

G. Purpura[1], A. De Vittori[2], R. Cipollone[3], M. Massari[4], C. Colombo[5], P. Di Lizia[6], S. Cicalò[7], F. Guerra[8], A. Bertolucci[9], A. Di Cecco [10], and L. Salotti [11]

[1]*Politecnico di Milano, Milan, Italy, Email: giovanni.purpura@polimi.it*
[2]*Politecnico di Milano, Milan, Italy, Email: andrea.devittori@polimi.it*
[3]*Politecnico di Milano, Milan, Italy, Email: riccardo.cipollone@polimi.it*
[4]*Politecnico di Milano, Milan, Italy, Email: mauro.massari@polimi.it*
[5]*Politecnico di Milano, Milan, Italy, Email: camilla.colombo@polimi.it*
[6]*Politecnico di Milano, Milan, Italy, Email: pierluigi.dilizia@polimi.it*
[7]*SpaceDyS s.r.l., Navacchio di Cascina (PI), Italy, Email: cicalo@spacedys.com*
[8]*SpaceDyS s.r.l., Navacchio di Cascina (PI), Italy, Email: guerra@spacedys.com*
[9]*SpaceDyS s.r.l., Navacchio di Cascina (PI), Italy, Email: bertolucci@spacedys.com*
[10]*Italian Space Agency (ASI), Rome, Italy, Email: alessandra.dicecco@asi.it*
[11]*Italian Space Agency (ASI), Rome, Italy, Email: luca.salotti@asi.it*

ABSTRACT

The ability to study the performance of different sensor configurations is crucial for the development of any sensor network that provides data for Space Surveillance and Tracking (SST) services. Any software suite devoted to this purpose shall be able to assess the performance of an existing network in terms of effectiveness and robustness, as well as to estimate the advantages and disadvantages of any structural change, such as the addition of new sensors or the upgrade of existing ones.

This paper is devoted to introducing the Space Surveillance Sensor Network SImulation Tool (SΞNSIT) and its contribution to the above task. SΞNSIT is a software suite conceived to perform a statistical analysis of the observational and cataloguing capabilities of a sensor network. The software can model optical, radar and laser ranging sensors and simulate different operational scenarios. The user is provided with the necessary interfaces to define the list of sensors composing the network, along with a reference population of space objects in terms of Two-Line Elements or orbital parameters. The typical sensor properties that can be specified by the user include type, operating mode (survey or tracking), location, measurement accuracy, pointing constraints, sensitivity and time availability. The provided sensor properties and object population are processed to predict the transits that can be successfully detected by each sensor, considering visibility and detectability constraints. The passes are analyzed to assess the network capabilities in terms of number of observations and catalogue coverage. Dedicated plots are provided to compare sensors in terms of coverage, to identify overlapping in the sets of observed objects, and to provide an estimate of the level of complementarity or redundancy.

Afterwards, the software simulates the measurements gathered by the network during the passes and processes them to carry out initial orbit determination and orbit determination refinement, with the aim of assessing the network performance in terms of catalogue build-up and maintenance. All the results are exposed to the user in informative tables and graphs with different levels of detail, starting from a general overview of the network performance up to the complete list of the passes observed by each sensor. The user can also browse the object catalogue of the network, analyzing its evolution in time.

The modularity of the software allows the user to easily modify the properties of the network and to carry out a sensitivity analysis to different parameters, such as the number and location of the sensors or their characteristics. This is expected to ease the setup process of sensor networks for SST, as well as the identification of the most promising upgrades to be recommended. As a side-product, the tool grants the opportunity to show and export all the data associated to the observable passes, including their pointing requirements, allowing for further analyses.

Keywords: Sensor networks; Space object cataloguing; Space Surveillance and Tracking; Space Situational Awareness; Orbit determination .

Proc. 8th European Conference on Space Debris (virtual), Darmstadt, Germany, 20–23 April 2021, published by the ESA Space Debris Office
Ed. T. Flohrer, S. Lemmens & F. Schmitz, (http://conference.sdo.esoc.esa.int, May 2021)

1. INTRODUCTION

The space environment has become a valuable asset for communication, navigation and observation purposes over the past few years. Due to the increasing number of satellites, potential collisions with other objects and uncontrolled debris reentry, that may endanger populated areas, are of major concern. The population of space objects is mainly located in three orbital regimes [1]:

- Low Earth Orbit (LEO): below 2000 km, where the issue of space debris is presently most urgent,
- Medium Earth Orbit (MEO): between 2000 km and 36 000 km, widely used by navigation constellations such as the Global Positioning System (GPS),
- Geostationary Earth Orbit (GEO): a roughly toroidal region of space close to the equatorial plane at about 36 000 km, the vast majority of telecommunication satellites are in this slot.

Since 1957, more than 4900 space launches have led to an orbital population of more than 23 000 trackable objects with sizes larger than 10 cm [2]. About a thousand of these are operational satellites, while the remaining 94 % are space debris – objects that no longer serve any useful purpose. About 64 % of the routinely tracked objects are fragments from some 250 breakups, mainly explosions and collisions of satellites or rocket bodies. In addition, about 670 000 objects larger than 1 cm and 170 million objects larger than 1 mm are expected to be in orbit. A schematic representation of the entire space debris population is given in Fig. 1.

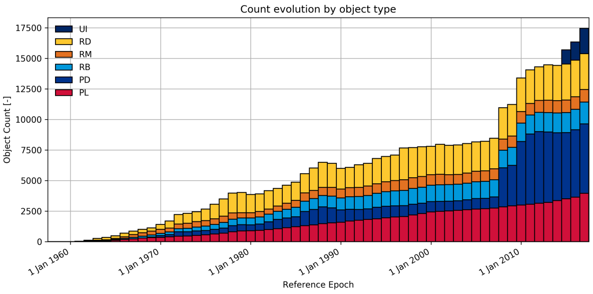


Figure 1. Timeline of the number of space debris in orbit (plot available on the ESA website [3])

To mitigate these risks, surveying and tracking such objects is becoming of primary importance, as well as providing this information to a variety of stakeholders. The amount of catalogued objects in orbit scales with the quality of the available space surveillance systems. Hence, simulating a sensor network can have a significant impact when dealing with catalogue build-up and maintenance.

At European level, two examples of available sensor network simulation tools are the BAS3E (Banc d'Analyse et de Simulation d'un Systeme de Surveillance de l'Espace - Simulation and Analysis Bench for Space Surveillance System) and the S3TOC (Spanish Space Surveillance and Tracking Operations Center) tool. BAS3E is a complete SST simulation framework developed by CNES [4], with the goal to evolve existing Space Surveillance and Tracking (SST) network, both from a software and hardware point of view, and to define major evolutions of existing SST networks. It implements the capability to simulate ground and space based sensors via the integration of the following functions:

- Detection, tracking and generation of observations of space objects
- Object identification and tracking correlation
- Orbit determination
- Maintenance of a space debris catalogue
- Centralized / de-centralized tasking and scheduling

The S3TOC is located in the Torrejón de Ardoz Military Air Base [5], 30 km away from Madrid (Spain). The centre is devoted to the generation of SST end-user products, for which a catalogue of objects is maintained, and orbital information from SST observations obtained by the S3TSN (Spanish Space Surveillance and Tracking Sensor Network) is computed. The S3TOC consists of the following elements:

- Data Processing and Cataloguing
- Service Processing
- Sensor Planning and Tasking
- Fragmentation messages
- Service Provision

The Italian SΞNSIT software provides functionality similar to those of its European counterparts. SΞNSIT is a tool for modeling sensor networks and evaluating their performance in terms of coverage and capability of building and maintaining a catalogue of space objects. Moreover, it allows the user to perform sensitivity analysis of the performance of the sensor network by varying the network configuration.

2. SΞNSIT

Figure 2. SΞNSIT *logo*

The Space Surveillance Sensor Network Simulation Tool (SΞNSIT) is a software tool conceived by Politecnico di Milano in collaboration with the SpaceDyS company and the Italian Space Agency. SΞNSIT is written in the Python programming language, relies on the NASA SPICE library, and runs on the major operating systems (Windows, MacOS, Linux). The software makes use of YAML files for the configuration and a SQLite database file for internal data storage. It can be used either from the command line or by means of a Graphical User Interface (GUI) based on the Qt library.

Given a list of space objects, a sensor network and a time frame, it performs the following tasks:

- computation of the observable transits of space objects over the selected ground stations
- simulation of the observations and the corresponding measurements
- orbit determination using the simulated measurements and the provided sensor accuracies
- catalogue build-up and maintenance according to the outcomes of the orbit determinations

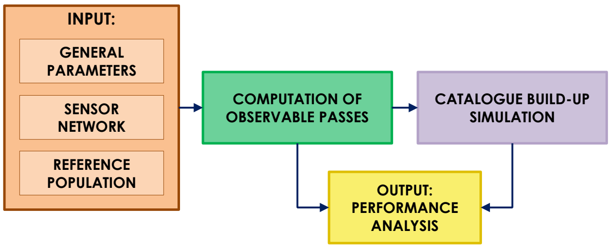

INPUT:
GENERAL PARAMETERS
SENSOR NETWORK
REFERENCE POPULATION
COMPUTATION OF OBSERVABLE PASSES
CATALOGUE BUILD-UP SIMULATION
OUTPUT: PERFORMANCE ANALYSIS

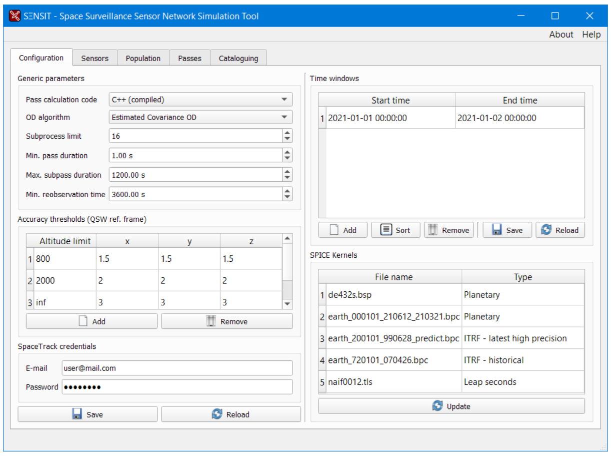
Figure 3. SΞNSIT *scheme*

SΞNSIT is composed of four modules (Fig. 3):

- **Data initialization**: gathers and pre-processes the inputs provided by the user and updates the SPICE kernels (files containing planetary ephemerides and leap second information).
- **Pass computation**: evaluates the observable passes of the objects belonging to the reference population, taking into account various observability constrains.
- **Catalogue build-up**: selects passes to be observed, simulates the measurements and carries out orbit determination, in order to build-up and maintain the network catalogue as the simulation proceeds.
- **Performance analysis**: allows to analyze the data created by the previous modules, by means of tables and charts, providing an overview of the performance of the sensor network.

2.1. Data initialization

2.1.1. Configuration



Generic parameters

Pass calculation code: C++ (compiled)
OD algorithm: Estimated Covariance OD
Subprocess limit: 16
Min. pass duration: 1.00 s
Max. subpass duration: 1200.00 s
Min. reobservation time: 3600.00 s

Accuracy thresholds (QSW ref. frame)

	Altitude limit	x	y	z
1	800	1.5	1.5	1.5
2	2000	2	2	2
3	inf	3	3	3

SpaceTrack credentials

E-mail: user@mail.com
Password: ••••••••

Time windows

	Start time	End time
1	2021-01-01 00:00:00	2021-01-02 00:00:00

SPICE Kernels

	File name	Type
1	de432s.bsp	Planetary
2	earth_000101_210612_210321.bpc	Planetary
3	earth_200101_990628_predict.bpc	ITRF - latest high precision
4	earth_720101_070426.bpc	ITRF - historical
5	naif0012.tls	Leap seconds

Figure 4. SΞNSIT *GUI: configuration tab*

The configuration tab of the GUI (Fig. 4) allows to define the following parameters:

- generic parameters to tune the processes,
- accuracy thresholds, that will be used to determine if an object is considered catalogued, according to the covariance of its position,
- time windows for the simulation,
- Space-Track credentials for automatic download of TLEs (optional).

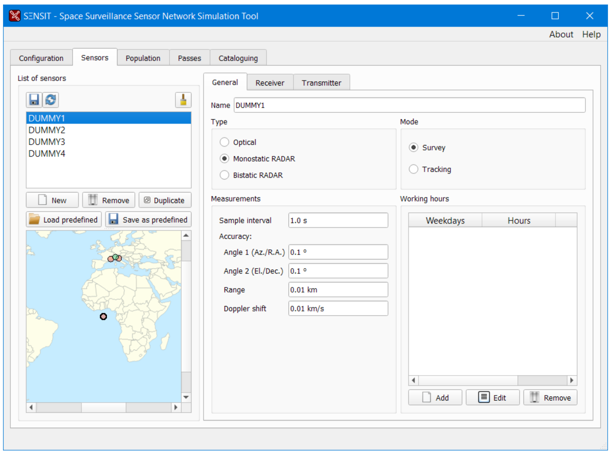
These parameters can alternatively be set in specific files in YAML format.

2.1.2. Sensors

The user shall configure the sensor network either through the GUI (Fig. 5) or by means of a YAML file formatted according to the instructions reported in the software manual.

The parameters to be entered for each sensor are:

- Name
- Type (optical, radar mono/bistatic)
- Mode (tracking, survey)
- Working hours (optional)
- Measurement sample interval
- Measurement accuracies



List of sensors

- DUMMY1
- DUMMY2
- DUMMY3
- DUMMY4

Name DUMMY1

Type
○ Optical
● Monostatic RADAR
○ Bistatic RADAR

Mode
● Survey
○ Tracking

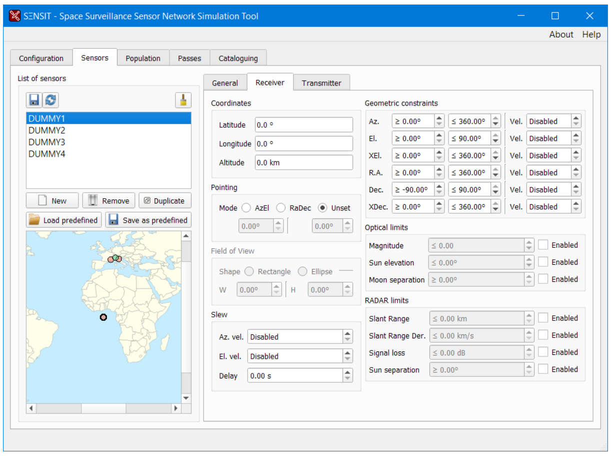
Measurements

Sample interval	1.0 s
Accuracy:	
Angle 1 (Az./R.A.)	0.1 °
Angle 2 (El./Dec.)	0.1 °
Range	0.01 km
Doppler shift	0.01 km/s

Working hours

Weekdays	Hours

Figure 5. SΞNSIT *GUI: sensors tab - general*



List of sensors

- DUMMY1
- DUMMY2
- DUMMY3
- DUMMY4

Coordinates

Latitude	0.0 °
Longitude	0.0 °
Altitude	0.0 km

Pointing

Mode ○ AzEl ○ RaDec ● Unset
0.00° | 0.00°

Field of View

Shape ○ Rectangle ○ Ellipse
W 0.00° | H 0.00°

Slew

Az. vel.	Disabled
El. vel.	Disabled
Delay	0.00 s

Geometric constraints

Az.	≥ 0.00°	≤ 360.00°	Vel.	Disabled
El.	≥ 0.00°	≤ 90.00°	Vel.	Disabled
XEl.	≥ 0.00°	≤ 360.00°	Vel.	Disabled
R.A.	≥ 0.00°	≤ 360.00°	Vel.	Disabled
Dec.	≥ -90.00°	≤ 90.00°	Vel.	Disabled
XDec.	≥ 0.00°	≤ 360.00°	Vel.	Disabled

Optical limits

Magnitude	≤ 0.00	□ Enabled
Sun elevation	≤ 0.00°	□ Enabled
Moon separation	≥ 0.00°	□ Enabled

RADAR limits

Slant Range	≤ 0.00 km	□ Enabled
Slant Range Der.	≤ 0.00 km/s	□ Enabled
Signal loss	≤ 0.00 dB	□ Enabled
Sun separation	≥ 0.00°	□ Enabled

Figure 6. SΞNSIT *GUI: sensors tab - receiver*

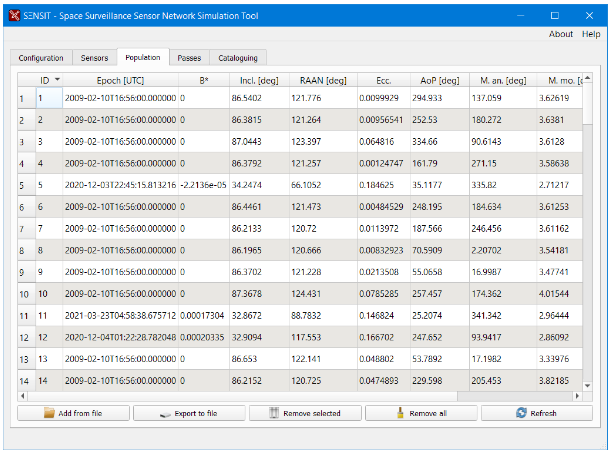
For the receiving station (Fig. 6), it is necessary to enter:

- Geographical coordinates
- Pointing and Field Of View (for survey sensors)
- Slew speed (for tracking sensors)
- Geometrical constraints (e.g., Field of Regard)
- Optical and radar signal limits

If the sensor is a bistatic radar, the user shall also provide the information about the transmitting station. In this case, the fields are the same as those for the receiver, except for the optical and radar signal limits that are not present.

2.1.3. Reference population

The reference population of space objects can be loaded through the GUI (Fig. 7) or using a command line script, using one of the following formats:



	ID	Epoch [UTC]	B*	Incl. [deg]	RAAN [deg]	Ecc.	AoP [deg]	M. an. [deg]	M. mo. [
1	1	2009-02-10T16:56:00.000000	0	86.5402	121.776	0.0099929	294.933	137.059	3.62619
2	2	2009-02-10T16:56:00.000000	0	86.3815	121.264	0.00956541	252.53	180.272	3.6381
3	3	2009-02-10T16:56:00.000000	0	87.0443	123.397	0.064816	334.66	90.6143	3.6128
4	4	2009-02-10T16:56:00.000000	0	86.3792	121.257	0.00124747	161.79	271.15	3.58638
5	5	2020-12-03T22:45:15.813216	-2.2136e-05	34.2474	66.1052	0.184625	35.1177	335.82	2.71217
6	6	2009-02-10T16:56:00.000000	0	86.4461	121.473	0.00484529	248.195	184.634	3.61253
7	7	2009-02-10T16:56:00.000000	0	86.2133	120.72	0.0113972	187.566	246.456	3.61162
8	8	2009-02-10T16:56:00.000000	0	86.1965	120.666	0.00832923	70.5909	2.20702	3.54181
9	9	2009-02-10T16:56:00.000000	0	86.3702	121.228	0.0213508	55.0658	16.9987	3.47741
10	10	2009-02-10T16:56:00.000000	0	87.3678	124.431	0.0785285	257.457	174.362	4.01544
11	11	2021-03-23T04:58:38.675712	0.00017304	32.8672	88.7832	0.146824	25.2074	341.342	2.96444
12	12	2020-12-04T01:22:28.782048	0.00020335	32.9094	117.553	0.166702	247.652	93.9417	2.86092
13	13	2009-02-10T16:56:00.000000	0	86.653	122.141	0.048802	53.7892	17.1982	3.33976
14	14	2009-02-10T16:56:00.000000	0	86.2152	120.725	0.0474893	229.598	205.453	3.82185

Figure 7. SΞNSIT *GUI: population tab*

- Two-Line Elements in a text file,
- Cartesian states in CSV format,
- List of Satellite Catalog Numbers (NORAD IDs), for which TLEs will be automatically downloaded from Space-Track.org,
- ESA MASTER population file (*.pop),
- SGP4 elements (in CSV or JSON format).

The inputs are automatically converted to SGP4 elements (if necessary) and are saved in the internal SQLite database of the application.

Furthermore, it is possible to load the Radar Cross Section and the intrinsic brightness of the objects (in CSV or JSON format): these are used to compute the radar signal loss and the optical magnitude, respectively. The signal loss is computed according to Eq. 1.

$$\begin{aligned} loss_{dB} = &+ 20\log_{10}(\rho_{RX}\cdot\rho_{TX}) - 10\log_{10}(\sigma) \\ &+ 30\log_{10}(4\pi) - 20\log_{10}(c) \end{aligned} \tag{1}$$

with ρ_{RX} and ρ_{TX} the distance of the target from the receiver and the transmitter in m, σ the RCS in m^2 and c the speed of light in $\mathrm{m/s}$.

Considering the link budget equation, the value of signal loss limit to use can be determined from the characteristics of the sensor, as in Eq. 2.

$$\begin{aligned} losslim_{dB} = &+ 10\log_{10}(P_{RX}) - 10\log_{10}(P_{TX}) \\ &+ 10\log_{10}(G_{RX}) + 10\log_{10}(G_{TX}) \\ &- 20\log_{10}(f) \end{aligned} \tag{2}$$

with P_{RX} minimum detectable value of received power in W, P_{TX} transmitted power in W, G_{RX} and G_{TX} receiver and transmitter gains, f carrier frequency in Hz.

Magnitude is computed according to Eq. 3.

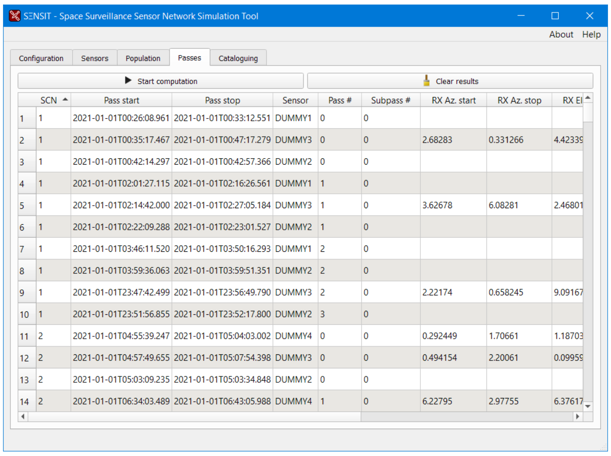
$$
\begin{aligned}
m = &+ ibr \\
&- 2.5 \log_{10}\left((\pi - \phi)\cos(\phi) + \sin(\phi)\right) \\
&+ 5 \log_{10}(\rho) - 15 - ext
\end{aligned}
\tag{3}
$$

with ibr intrinsic brightness, ϕ phase angle in radians, ρ distance in km and ext atmospheric extinction, computed with Eq. 4 [6].

$$
ext = \frac{0.1451e^{-h/7.996} + 0.120e^{-h/1.5} + 0.016}{\sin(el) + 0.025e^{-11\sin(el)}} \tag{4}
$$

with h altitude of the ground station in km and el elevation angle in radians.

2.2. Pass computation



	SCN	Pass start	Pass stop	Sensor	Pass #	Subpass #	RX Az. start	RX Az. stop	RX El
1	1	2021-01-01T00:26:08.961	2021-01-01T00:33:12.551	DUMMY1	0	0			
2	1	2021-01-01T00:35:17.467	2021-01-01T00:47:17.279	DUMMY3	0	0	2.68283	0.331266	4.42339
3	1	2021-01-01T00:42:14.297	2021-01-01T00:42:57.366	DUMMY2	0	0			
4	1	2021-01-01T02:01:27.115	2021-01-01T02:16:26.561	DUMMY1	1	0			
5	1	2021-01-01T02:14:42.000	2021-01-01T02:27:05.184	DUMMY3	1	0	3.62678	6.08281	2.46801
6	1	2021-01-01T02:22:09.288	2021-01-01T02:23:01.527	DUMMY2	1	0			
7	1	2021-01-01T03:46:11.520	2021-01-01T03:50:16.293	DUMMY1	2	0			
8	1	2021-01-01T03:59:36.063	2021-01-01T03:59:51.351	DUMMY2	2	0			
9	1	2021-01-01T23:47:42.499	2021-01-01T23:56:49.790	DUMMY3	2	0	2.22174	0.658245	9.09167
10	1	2021-01-01T23:51:56.855	2021-01-01T23:52:17.800	DUMMY2	3	0			
11	2	2021-01-01T04:55:39.247	2021-01-01T05:04:03.002	DUMMY4	0	0	0.292449	1.70661	1.18703
12	2	2021-01-01T04:57:49.655	2021-01-01T05:07:54.398	DUMMY3	0	0	0.494154	2.20061	0.09959
13	2	2021-01-01T05:03:09.235	2021-01-01T05:03:34.848	DUMMY2	0	0			
14	2	2021-01-01T06:34:03.489	2021-01-01T06:43:05.988	DUMMY4	1	0	6.22795	2.97755	6.37617

Figure 8. SΞNSIT *GUI: passes tab*

This process computes the observable passes of the objects belonging to the reference population, taking into account the observability conditions set by the user. The computed passes are shown in the GUI (Fig. 8).

The computation is based on the bisection algorithm and takes advantage from several optimizations:

- The states of the stations and of the objects are cached when possible
- The core of the algorithm is written in C++
- The code runs in parallel on separate processes

The algorithm, schematized in Fig. 9, starts from the time window chosen by the user and splits it into segments. At each splitting point, it evaluates the observability condition, giving a preliminary estimation of the passes. At the end, it applies bisection to find the precise start and stop epochs of the passes.

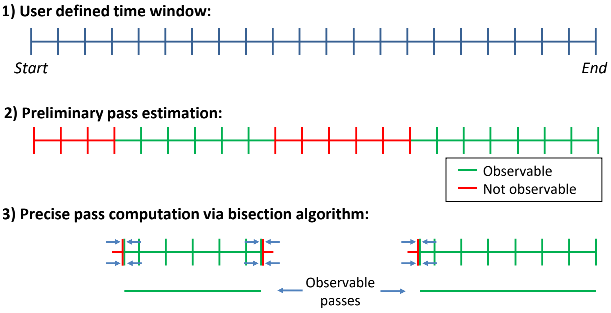
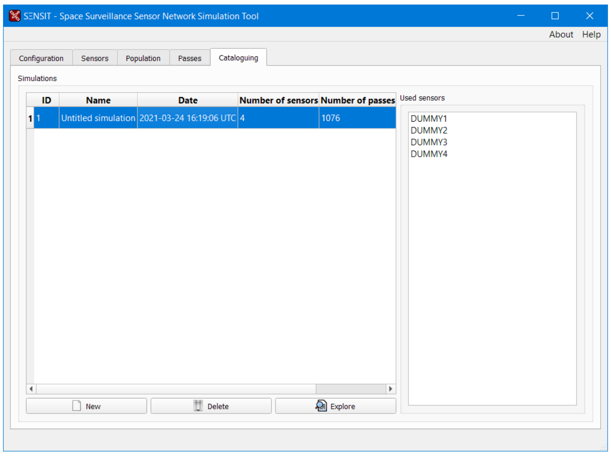

1) User defined time window:
Start
End
2) Preliminary pass estimation:
Observable
Not observable
3) Precise pass computation via bisection algorithm:
Observable passes

Figure 9. SΞNSIT *algorithm for passes computation*

2.3. Catalogue build-up

The third module processes the observable passes in order to build-up and maintain the network catalogue.



	ID	Name	Date	Number of sensors	Number of passes
1	1	Untitled simulation	2021-03-24 16:19:06 UTC	4	1076

Used sensors

DUMMY1
DUMMY2
DUMMY3
DUMMY4

Figure 10. SΞNSIT *GUI: cataloguing tab*

As a default, the cataloguing simulation is performed considering all the defined sensors. Alternatively, the user can select a subset of sensors from the GUI (Fig. 10): in this way, it is possible to conveniently execute different simulations and compare the results.

A schematic representation of the catalogue build-up process is given in Fig. 11.

It is supposed at the beginning of the simulation process to have an empty network catalogue. When a non catalogued object is observable by a sensor in survey mode, Initial Orbit Determination (IOD) is performed (Fig. 12). Provided that the diagonal components of the state covariance in the QSW reference frame are below a user-defined threshold, the object will join the network catalogue. The condition on the covariance is checked throughout the simulation, since position and velocity uncertainties enlarge over time. When a catalogued object is observable, a scheduling algorithm decides whether the pass is actually observed. If this is the case, Refined Orbit Determination is executed, that updates the covariance matrix of the object. In order to simplify the process, the

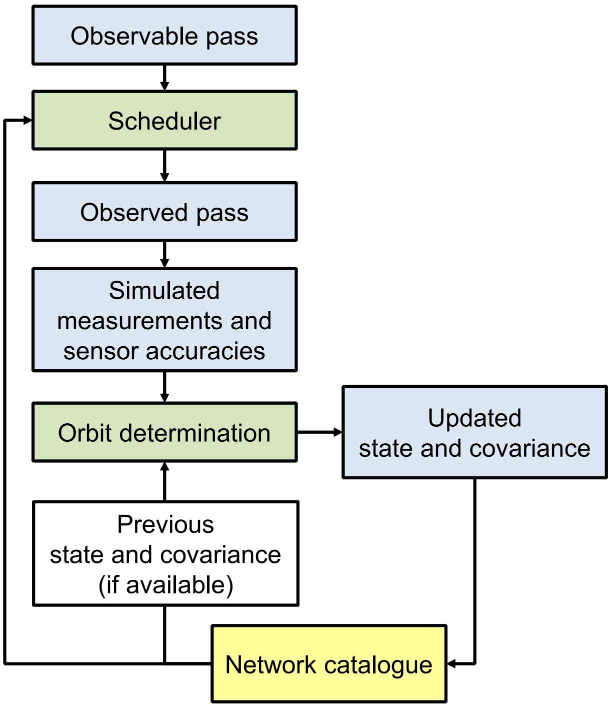

Observable pass
Scheduler
Observed pass
Simulated measurements and sensor accuracies
Orbit determination
Updated state and covariance
Previous state and covariance (if available)
Network catalogue

Figure 11. Schematic representation of the catalogue build-up and update process.

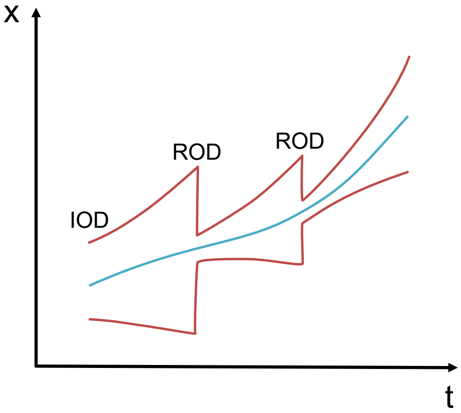


Figure 12. Schematic representation of the IOD and ROD processes, blue line = mean state, red line = uncertainty.

mean state of the object obtained by the orbit determination process is not recorded: instead, it is always evaluated by SGP4 propagation of the elements provided by the user, adding noise when necessary. Concerning the OD sub-module two different pipelines have been proposed: the Estimated covariance OD and the Non-Linear Least Squares OD. The former ensures a reduced computational cost by estimating only the expected outcome of an OD process in terms of covariance. The latter instead performs the whole orbit determination process using the Non-Linear Least Squares optimization algorithm.

2.3.1. Catalogue build-up and maintenance process

First of all, the list of observable passes is chronologically sorted. Then, for each transit, if it occurs over a sensor in tracking mode, the following checks are carried out:

- The object to be detected needs a valid covariance computed from a previous IOD or ROD and propagated.
- The re-observation time for an object is to be higher than a user-defined threshold. The underlying idea is to ensure to follow up less frequent objects.
- If time is not sufficient to steer the sensor to the desired spot, the transit will be considered invalid.
- To reduce the tracking sensors workload, if the object will be observed by a survey sensor within a given time, it is not observed by the tracking sensor.

Before moving on, it is worthwhile to explain how the covariance is assessed to be valid or not. Firstly, the rotation matrix $\mathbf{R_{E2Q}}$ from the Earth-Centered Inertial (ECI) reference frame to the QSW frame [7] is computed, where:

- $\vec{\mathbf{q}}$ is the unit vector collinear to the geocentric satellite position (from the planet center to the spacecraft)
- $\vec{\mathbf{w}}$ is the unit vector collinear to the orbital kinetic momentum (normal to the orbital plane)
- $\vec{\mathbf{s}}$ is the unit vector equal to $\vec{\mathbf{w}} \wedge \vec{\mathbf{q}}$

The covariance matrix $\mathbf{C_Q}$ in QSW frame is obtained from the one in ECI frame $\mathbf{C_E}$ as in Eq. 5.

$$\mathbf{C_Q} = \mathbf{R_{E2Q}}\, \mathbf{C_E}\, \mathbf{R_{E2Q}}^{\mathrm{T}} \tag{5}$$

The object enters or remains in the network catalogue if the first three diagonal components $\mathbf{d_Q}$ of $\mathbf{C_Q}$ are lower than a specified threshold $\tilde{\mathbf{d}}_\mathbf{Q}$ as described in Eq. 6.

$$\mathbf{d_{Q_i}} < \tilde{\mathbf{d}}_\mathbf{Q_i} \qquad \text{for} \quad i = 1, 2, 3 \tag{6}$$

$\tilde{\mathbf{d}}_\mathbf{Q}$ has to be defined by the user and may be different according to the altitude of the object.

Next, if the object has already a valid covariance $\mathbf{C_{E,t_{i-1}}}$ at the previous t_{i-1} ROD or IOD instant, it is propagated up to the initial observation epoch ($\mathbf{C_{E,t_i}}$) as shown in Eq. 7:

$$\mathbf{C_{E,t_i}} = \mathbf{J_{kep}}\, \mathbf{C_{E,t_{i-1}}}\, \mathbf{J_{kep}}^{\mathrm{T}} \tag{7}$$

$\mathbf{J_{kep}}$ is the Jacobian obtained by a Keplerian state transition matrix and projects the covariance from t_{i-1} to t_i. If $\mathbf{C_{E,t_i}}$ is valid too, ROD will be performed.

For this stage, as stated in Sec. 2.3, two approaches have been implemented. The first one is the Estimated Covariance OD, that requires:

- $\mathbf{C_{E,t_i}}$: the known state covariance matrix,
- $\mathbf{C_m}$: the covariance matrix of the measurements (sensor dependent),
- $\mathbf{J_{m/s}}$: the Jacobian matrix of the measurements with respect to the propagated states of the object,
- $\mathbf{J_{s/s_0}}$: the Jacobian matrix of the propagated states with respect to the initial state, approximated as the state transition matrix of a Keplerian propagation.

The updated covariance $\mathbf{C_{E,t_{i,new}}}$ is computed as:

$$\begin{aligned} \mathbf{J_{m/s_0}} &= \mathbf{J_{m/s}}\,\mathbf{J_{s/s_0}} \\ \mathbf{C_{E,t_{i,new}}} &= (\mathbf{J_{m/s_0}}^{\mathrm{T}}\,\mathbf{C_m}^{-1}\,\mathbf{J_{m/s_0}} + \mathbf{C_{E,t_i}}^{-1})^{-1} \end{aligned} \tag{8}$$

The other OD formulation relies on a Non-Linear Least Squares optimization [8]. The algorithm has been designed as follows:

- Synthetic measures $\mathbf{m_{sens}}$ are generated with a fixed time step within the observation window, and Gaussian noise is added according to the sensor accuracy.
- The initial state guess $\mathbf{s_0}$ is computed by SPG4 propagation of the elements of the object and by the addition of Gaussian noise, according to the state covariance $\mathbf{C_{E,t_i}}$.
- An iterative procedure takes $\mathbf{s_0}$, propagates it with Keplerian dynamics up to the time instants of the measurements and projects it to the measurement space ($\mathbf{m_{LS}}$). A design matrix $\mathbf{D}$ (Eq. 9) is built using the Jacobian matrix of the measurements with respect to the initial state ($\mathbf{J_{m/s_0}}$) and weights $\mathbf{W}$ (defined from the sensor accuracy). The initial state with its covariance is considered as *a priori information*.
$$\mathbf{D} = \mathbf{W} \odot \mathbf{J_{m/s_0}} \tag{9}$$
- The cost function is the measures residual $\mathbf{res}$:
$$\mathbf{res} = \mathbf{m_{sens}} - \mathbf{m_{LS}} \tag{10}$$
- $\mathbf{res}$ and $\mathbf{D}$ are intended to solve the normal equation, that outputs $\mathbf{C_{E,t_{i,new}}}$ and the $\mathbf{s_0}$ correction factor.
- The routines stops if a maximum number of iterations or convergence is reached.

In the event that the object is out of catalogue and passes over a survey ground station, IOD will be conducted in similar manner as illustrated for ROD. The differences are:

- For the Estimated Covariance OD, the updated covariance is determined as:
$$\mathbf{C_{t_{i,new}}} = (\mathbf{J_{m/s_0}}^{T}\mathbf{C_s}^{-1}\mathbf{J_{m/s_0}})^{-1} \tag{11}$$
- For the Non-Linear Least Squares algorithm, no *a priori information* is considered.

After ROD or IOD, the updated covariances and the corresponding epochs are saved in the program database.

3. RESULTS

The results from the previous steps are stored in the application database file. These are employed to illustrate data as different interactive graphs, designed to maximize the user awareness of the network performance. The representations are both sensor-oriented and population-oriented, to have an organic view of the results of the simulation.

The available visualizations are described hereafter. Then, an analysis of the computational time required by the software is reported.

3.1. Sensor network performance analysis

3.1.1. Pass list

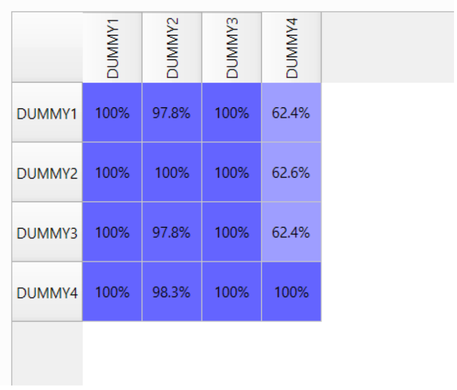
The first view that is shown to the user is a list of the observed passes, with the ID of the object, the epoch, the sensor name, the orbit determination type and the resulting covariance. This allows to analyze in detail the observations performed by each sensor.

3.1.2. Redundancy matrix

The redundancy matrix (Fig. 13) is a table that shows the ratio of objects visible from a given sensor that can also be seen by another. This can help in determining the redundancy of the sensors.

3.1.3. Catalogue population plot

An example of population-focused plot is displayed in Fig. 14. It shows a Cartesian plane, having two orbital parameters as axes. X and Y can be modified by the user and the distribution changes accordingly, allowing to understand the orbital regime they belong to. The color of each point varies according to the number of times it was observable by the sensors. It is possible to select only a subset of sensors, and the color intensity of the points changes accordingly: this lets the user understand the importance of each sensor in the observations.



	DUMMY1	DUMMY2	DUMMY3	DUMMY4
DUMMY1	100%	97.8%	100%	62.4%
DUMMY2	100%	100%	100%	62.6%
DUMMY3	100%	97.8%	100%	62.4%
DUMMY4	100%	98.3%	100%	100%

Figure 13. Redundancy matrix: each cell contains the percentage of objects visible by the sensor on the row that are also observable by the sensor on the column.

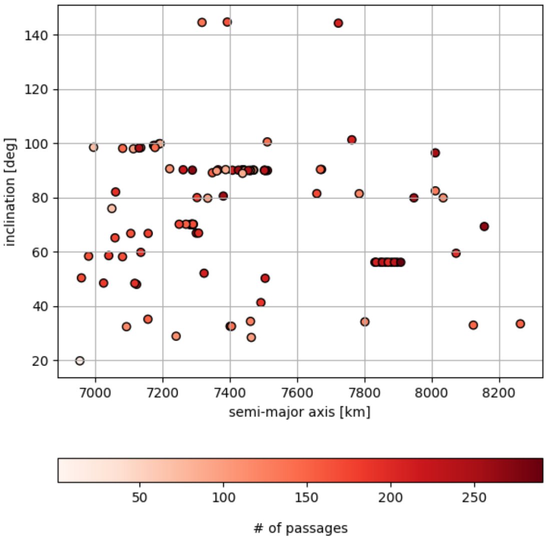


Figure 14. Catalogue population plot: dots represent space objects, their color intensity depicts the number of passes observed by the network.

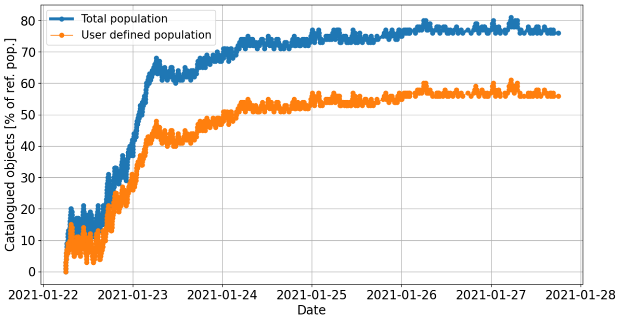


Figure 15. Evolution of catalogued population over time. The blue line represents the total amount of objects inside the catalogue, the orange one depicts a user defined subset of objects belonging to one or more orbital regimes.

3.1.4. Catalogue evolution

A further point of view about the interaction between the simulated population and the network is the catalogue evolution representation (Fig. 15), in terms of percentage of objects belonging to the reference population. The entries are determined by a successful initial orbit determination, while the exits by the covariance exceeding the thresholds. The plot portrays two different population trends: the blue one refers to the total amount of objects belonging to the catalogue, while the orange one depicts the evolution of a subset of objects located in a specific orbital regime selected by the user.

3.1.5. Covariance evolution

Knowing the covariance evolution in time of a specific target is crucial to understand if it is inside or outside the sensor network catalogue within a simulation. To be part of it, a satellite requires an initial orbit determination from a survey ground station where the outcome is the only state covariance matrix. Each object is subjected to an increase in position and velocity uncertainty as time goes by, while acquisitions from sensors shrink it down through refined orbit determination. If the covariance exceeds a predefined limit, the object exits the catalogue and a further initial orbit determination is needed. In Fig. 16 a Cartesian plot describes the trend of a catalogued object covariance over time, using the square root of its trace as metric. The wide oscillations are due to the period of time between two IOD or ROD procedures that decrease uncertainty after every new observation. The orange dots represent initial orbit determinations, while the green ones refined orbit determinations. The user can also choose to see separately the first three diagonal components of the covariance matrix in the QSW reference frame.

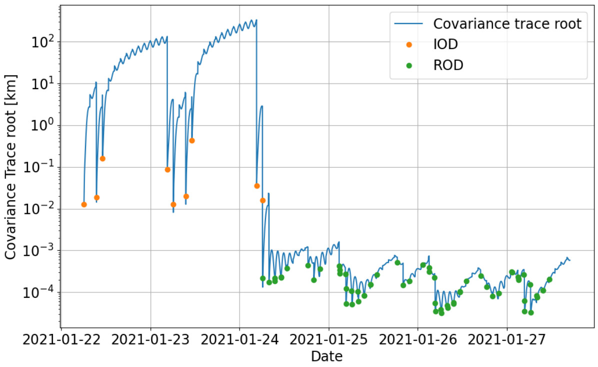


Figure 16. Covariance trend over time: using its trace square root as a figure of merit, the beneficial effect of successful IODs and RODs can be easily noticed by the orange and green dots respectively.

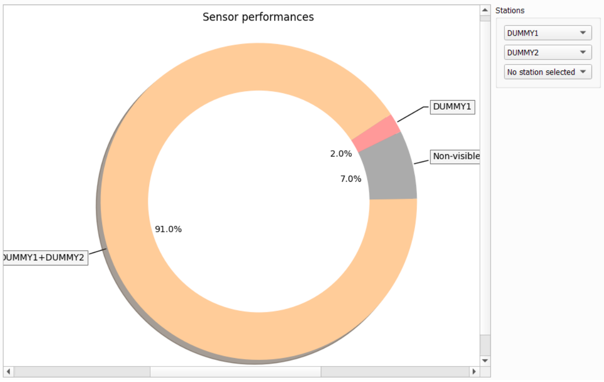


Figure 17. Pie chart plot showing the contribution of a maximum of three selected sensors and their combinations, in terms of ratio between observed objects and simulated population.

3.1.6. Coverage pie chart

A clear view of a sensors subset contributions to the entire network performance can be pictured as a pie chart (Fig. 17): each slice represents the percentage of simulated objects seen only by the corresponding sensor or combination of sensors (without any intersection among them, due to the rule used to determine the contributions). The user can select up to three sensors at the same time.

3.1.7. FoV projection

In order to have an organic view of the network coverage over areas of interest, the plot in Fig. 18 provides a geographical projection of the FoV of survey sensors at different altitudes. The representation depends on the sensor position, the FoV size and shape (rectangular or elliptical), the sensor pointing and the intersection altitude. Coverage areas are colored according to their type (optical or radar) and their size changes according to the intersection altitude, that can be modified by the user in real-time. This plot can also be useful to understand overlaps between the FoVs of survey sensors.

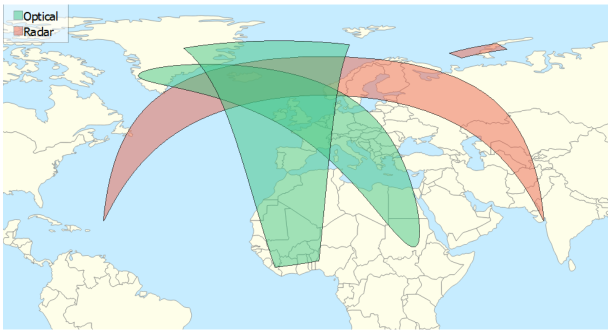


Figure 18. Sensors coverage areas are used to have a quick look at zones covered by survey sensors.

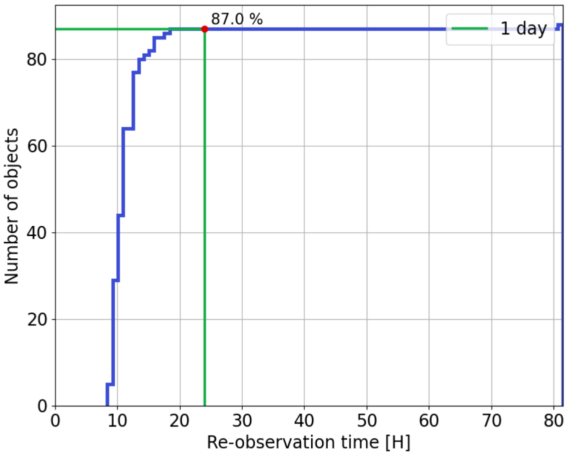


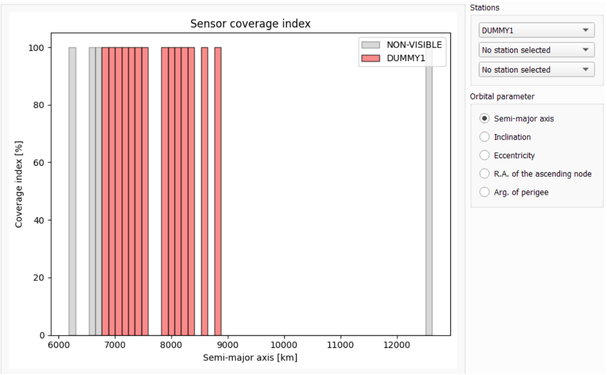
Figure 19. Maximum re-observation times

3.1.8. Maximum re-observation time

It is important to have a sense of what is the maximum re-observation time (i.e., the time elapsing between two consecutive observable passes) for all the objects passing over a ground station. This is instrumental to figure out how a sensor can contribute to the catalogue maintenance, since the lower the re-observation times the more the network can keep up to date the covariance estimates in the catalogue. The graph in Fig. 19 illustrates the cumulative distribution of the maximum re-observation times of all the objects transiting over a user defined sensor or over the entire sensor network. On top of that, a vertical line set at 24 hours splits the distribution in two parts, to highlight the percentage of objects which can always be seen within one day or less.

3.1.9. Coverage histogram

The coverage histogram (Fig. 20) permits to evaluate the coverage of specific sensors with respect to different or-



Stations

DUMMY1
No station selected
No station selected

Orbital parameter

● Semi-major axis
○ Inclination
○ Eccentricity
○ R.A. of the ascending node
○ Arg. of perigee

Figure 20. Coverage histogram

Table 1. Time required by pass computation

	1 sensor	5 sensors	10 sensors
10 objects	3 s	4 s	4 s
100 objects	21 s	23 s	25 s
1000 objects	204 s	223 s	244 s

bital parameters. The user can choose an orbital parameter (for the horizontal axis) and up to three sensors to compare their coverage. Each sensor is described as a bar distribution, whose height represent the percentage of observable objects within that range of the selected orbital parameter.

3.2. Computational time

The goal of SΞNSIT is not merely to provide an accurate sensor network modeling, but also to output the expected results in a reasonable time frame. The analysis of the computational time has been conducted on a PC featuring a 3700x AMD processor, with 8 physical and 16 logical cores, and 16 GB of RAM. Several sensor network simulations have been performed by setting the following parameters:

- Number of input objects: 10, 100, 1000.
- Number of involved sensor: 1, 5, 10
- Simulation time frame: 3 days

Table 1 and Fig. 21 outline the time required for the computation of observable passes (Sec. 2.2). Time clearly scales up when taking into account an increasing number of objects, while it is less affected by the amount of sensors thanks to the implemented optimizations (mainly, the cache of object states).

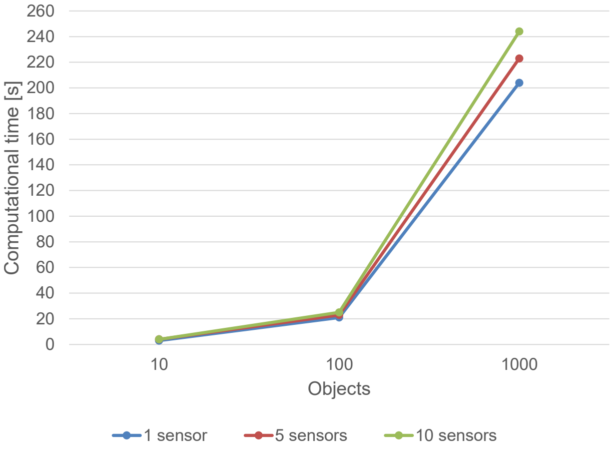


Figure 21. Time required by computation of observable passes

Table 2. Time required by catalogue build-up using estimated OD

	1 sensor	5 sensors	10 sensors
10 objects	1 s	1 s	1 s
100 objects	3 s	3 s	3 s
1000 objects	19 s	30 s	83 s

Table 2 and Fig. 22 report the time needed to perform catalog build-up and maintenance using Estimated Covariance OD (Sec. 2.3). It is similarly influenced by both objects and sensors.

Table 3 and Fig. 23, instead, report the time needed to perform catalog build-up and maintenance if Non-Linear Least Squares OD is used. The reported times are approximately double with respect to the ones required by Estimated Covariance OD.

Summing up the values contained in Tab. 1 and 2, the times reported in Tab. 4 are obtained. It showcases the total time required for a single simulation run, if Estimated Covariance OD is used. It is worth noting that in the worst case scenario, SΞNSIT takes roughly 5 minutes, allowing for fast simulations.

Table 3. Time required by catalogue build-up using NLS OD

	1 sensor	5 sensors	10 sensors
10 objects	1 s	1 s	1 s
100 objects	4 s	6 s	12 s
1000 objects	40 s	63 s	191 s

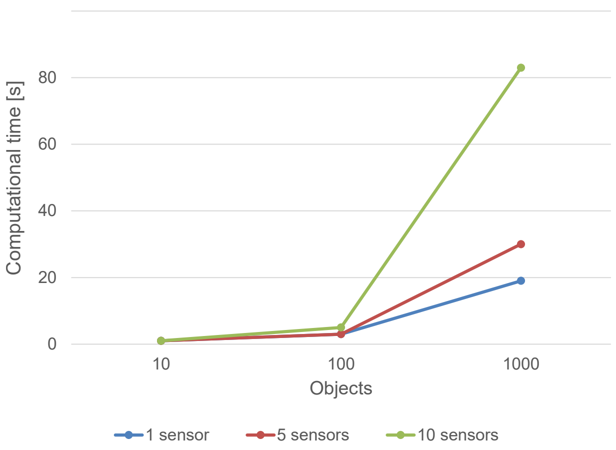


Figure 22. Time required by catalogue build-up using Estimated OD

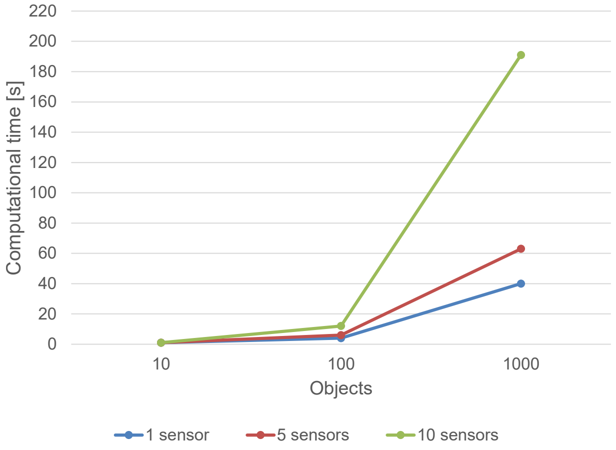


Figure 23. Time required by catalogue build-up using NLS OD

Table 4. Total simulation times using estimated OD

	1 sensor	5 sensors	10 sensors
10 objects	4 s	5 s	5 s
100 objects	24 s	26 s	30 s
1000 objects	223 s	253 s	327 s

4. CONCLUSIONS AND FUTURE WORK

SΞNSIT, the program developed in this work, answers the need of having a software tool that allows to model SST sensor networks and evaluate their performance, in terms of coverage and capability of building and maintaining a catalogue of space objects. Moreover, SΞNSIT allows to perform this tasks in a user friendly way, thanks to its Graphical User Interface, the capability of running through different operative systems and the speed of execution.

Yet, several improvements can still be implemented in SΞNSIT. In particular, the developers are considering to introduce the automatic generation of new sensors according to given criteria (e.g., different locations for a sensor within a given area, or different type of sensors in a fixed position). An improved scheduling algorithm for sensors in tracking mode is also being studied, along with the ability to consider random events that could prevent observability, such as adverse weather conditions or maintenance downtime. Furthermore, the possibility of starting the cataloguing simulation from a non-empty network catalogue will be implemented.

ACKNOWLEDGMENTS

The results of this project have been supported by the agreement of the Italian Space Agency and the National Institute for Astrophysics on Space Debris (Detriti Spaziali – Supporto alle attività IADC e SST 2019-2021, n. 2020-6-HH.0).

REFERENCES

1. A. Rossi, “Population models of space debris,” *Proceedings of the International Astronomical Union*, vol. 2004, no. IAUC197, p. 427–438, 2004.
2. “Space debris - evolution in pictures.” `https://www.esa.int/About_Us/ESOC/Space_debris_-_evolution_in_pictures`.
3. “Debris object evolution.” `https://www.esa.int/ESA_Multimedia/Images/2017/04/Debris_object_evolution`.
4. J. C. Dolado, V. Morand, and C. Yanez, “BAS3E: A framework to Conceive, Design, and Validate Present and Future SST Architectures,” in *First International Orbital Debris Conference*, vol. 2109, p. 6154, Dec. 2019.
5. S. A. V. Natalia Ortiz Gómez, Inés Alonso Gómez, “Architectural description of the spanish space surveillance and tracking system,” ESA Space Debris Office, 2017. 7th European Conference on Space Debris.
6. D. W. E. Green, “Magnitude corrections for atmospheric extinction.” `http://www.icq.eps.harvard.edu/ICQExtinct.html`, July 1992.

7. CCSDS Secretariat, National Aeronautics and Space Administration , “Navigation data— definitions and conventions,” 2019.

8. A. Milani and G. F. Gronchi, *Theory of Orbit Determination*. Cambridge University Press, 2010.