DOI: ADD DOINUMBER HERE

SNOS: automatic optimal observations scheduling for sensor networks

Niccolò Faraco[⋆†], *Giovanni Purpura*[⋆], *Pierluigi Di Lizia*[⋆], *Mauro Massari*[⋆],
Moreno Peroni[◇], *Alessandro Panico*[◇], *Andrea Cecchini*[◇], *Francesco Del Prete*[◇]
[⋆]*Politecnico di Milano*
Via G. La Masa 34, 20156, Milan, Italy
[◇]*Aeronautica Militare*
Address
niccolo.faraco@polimi.it · giovanni.purpura@polimi.it · pierluigi.dilizia@polimi.it · mauro.massari@polimi.it
[†]Corresponding author

Abstract

Space Surveillance and Tracking operations are vital for the subsistence of the spacecrafts population. A software, called SNOS (Sensors Network Optimal Scheduler), has been developed in order to automate the observations planning on a sensors network. The tool generates a user-specified number of schedules, optimized in terms of catalog coverage and expected quality of the measurements. SNOS can manage optical sensors as well as mono- or bi-static radar and laser sensors and can arrange both tracking and survey tasks. The SNOS software is currently successfully employed to run the network managed by the Italian Space Operation Centre (ISOC).

1. Introduction

Space Surveillance and Tracking (SST) is growing more and more important in the Space operations and mission analysis field, given how much the crowded environment building up near Earth can hamper them. Italy is involved in SST operations, collaborating with partners both within and outside the EU. The Italian Space Operation Centre (ISOC) has recently upgraded its systems to the ISOC 2.0 Suite, an integrated web-based platform providing multiple functions and services in the SST/SSA domain. The present work describes its sensor tasking module, called SNOS (Sensors Network Optimal Scheduler), developed thanks to a collaboration involving the Italian Air Force, the Leonardo Company and Politecnico di Milano. After the definition of the software architecture, its prototype version has been developed in Python and then translated to C++ language to grant the highest performances in the operational environment.
Given the list of available sensors together with their characteristics, the time window in which the observations must be scheduled and the catalogue of objects to be observed, SNOS is able to generate a user specified number of schedules, optimized both in terms of catalogue coverage and expected quality of the measurements. The results are provided to the sensor operation authority as a JSON file conveying all the information needed for the correct execution of the task and as a Tracking Data Message (TDM), which is the Consultative Committee for Space Data Systems (CCSDS) standard for this kind of operations. SNOS can handle optical sensors as well as mono- or bi-static radar and laser sensors.
The process starts from the analysis of Two-Line Elements (TLE) or Orbit Ephemeris Message (OEM) representing the state of the catalogued objects to identify the visible passes, i.e. the portions of their orbits which are visible from any of the sensors of the network. These are subdivided based on the priority assigned to the related object and conflicting situations are identified, such as the ones in which multiple passes from different objects are simultaneously visible by the same sensor. Candidate solutions are then gradually built by selecting passes (or portions of them) with a heuristic optimization approach.
The generated schedules are assigned a score based on the catalogue coverage and the expected quality of the measurements, which weights factors such as the elevation of the object, the distance from the observer, and the duration of the pass. The scheduler also grants the possibility to modify an already ongoing schedule to allocate tasks for unforeseen necessities highlighted by the sensor authority, while still retaining the optimality of the solution.
Finally, SNOS can compute the necessary sensor pointing sequence for different kind of tasks, namely tracking tasks and various survey modes of practical interest, such as the one for the surveillance of the GEO belt and the ones designed for the identification of new debris through the inspection of areas along the orbit of an object which is expected

to have suffered fragmentation. The SNOS software is currently successfully employed to run the network managed by the ISOC center.

The present work will be organized as follows. Section 2 will introduce the current state of the software suite of which the present work is part of and the collaboration framework the software has been developed in. Section 3 will then illustrate in detail the back-end software architecture and its capabilities. The web interface that's exposed to the final user will be briefly shown too. Finally, the software validation and the first operative results obtained will be discussed in Section 4.

2. Italian SST Operations Centre

The Italian SST Operations Centre, lately referred as ISOC, was established in 2014 by the Italian Air Force to operate in the framework of the European SST Consortium (EU-SST). The system was the Italian front-end aimed at capitalizing the measurements collected by the national sensors and providing services to the European community. Current operations are lead by the the Air and Space Operations Command whereas Flight Test Wing is in charge of research and developments.

The SW Suite was originally developed to support Space Surveillance and Tracking tasks only, but the system evolution has a broader application scenario, nested in the national and international framework of space security. The services that are publicly available for the EU-SST are:

- Re-entry (RE): main responsible for uncontrolled re-entry predictions of large and dangerous objects and all the rocket bodies, suitable to reach the lower layers of the atmosphere.
- Fragmentation (FG): prime responsible for the analysis of the consequences of in-orbit fragmentation, as consequence of satellite break-ups or collisions.
- Conjunction Analysis (CA): cold redundant operational center for the analysis of the collision probability and geometry for conjunction events.

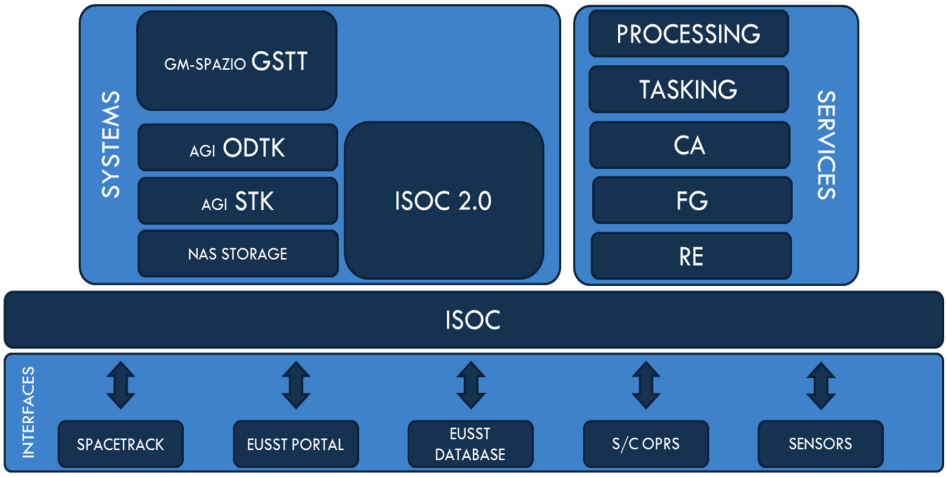


Figure 1: ISOC Architecture

ISOC Suite is a complex system that is used to support the above-mentioned services, whose high level architecture is represented in Fig. 1. The main inputs of the suite are provided by national sensors, consortium observations, European database (DCED) along with available public sources. The inner part of the system is based on both commercial off the shelf (COTS) and proprietary software. The system output are the services shown in the right part of Fig. 1. A functional part of the entire system is the planning generation and tasking optimization process, that could be assured by the suite described in this document. It is carried out for each sensor connected to the ISOC Suite. The piece of SW presented in the following Section 3 can handle lists of observable objects with the respective priorities and optimize the final operational scheduling.

3. Sensors Network Optimal Scheduler

The present section illustrates, in broad terms, the architecture of the SNOS (Sensors Network Optimal Scheduler) software and some of its peculiar capabilities, along with the types of tasks that can be scheduled for the sensors and the schedule optimization process.

3.1 General architecture

SNOS can be executed in two different modes, namely the *static mode* and the *dynamic mode*. The former can be used to produce a new schedule from scratch, trying to allocate the available time into a set of tasks to be performed by the available sensors in order to maximize the efficiency of the network both in terms of coverage of the satellites catalog and expected quality of the measurements to be performed. The dynamic mode, instead, allows the user to modify a schedule whose execution has already started, granting the possibility to accommodate last minute necessities, such as the observation of an unforeseen event or the temporary unavailability of a sensor due to changing weather conditions. In doing so, the algorithm still tries to retain the optimality of the solution.

The minimum set of inputs includes the time window on which the scheduling has to be performed, the list of the available sensors in that timeframe and the catalogue of objects of interest to be observed. In the case of dynamic mode, the list of accepted tasks for the currently on going schedule has to specified too.
The main part of the software is ideally subdivided into two major steps. The first one is the identification of the observation opportunities, i.e. the portions of the orbits of the satellites that could be observed by any of the tracking sensors in the input list (see Section 3.2). Contextually, the pointing coordinates to be commanded to the survey sensors in order for them to correctly perform the tasks requested by the operator are computed too. Then, the results are fed to the optimization algorithm that creates a user-defined number of schedules, trying to maximize the catalog coverage and the expected quality of the observations. In doing so, the algorithm also takes care of the feasibility of the schedule, assuring that no overlapping tasks are assigned to the same sensor and that the time between subsequent tasks is enough for the sensor to reposition itself.
The results are provided to the end user both as a JSON file and as a Tracking Data Message (TDM), which is the Consultative Committee for Space Data Systems (CCSDS) standard format used to express the series of coordinates the sensor has to be pointed to in order to correctly accomplish its task.

A graphical representation of the process is provided in Fig. 2.

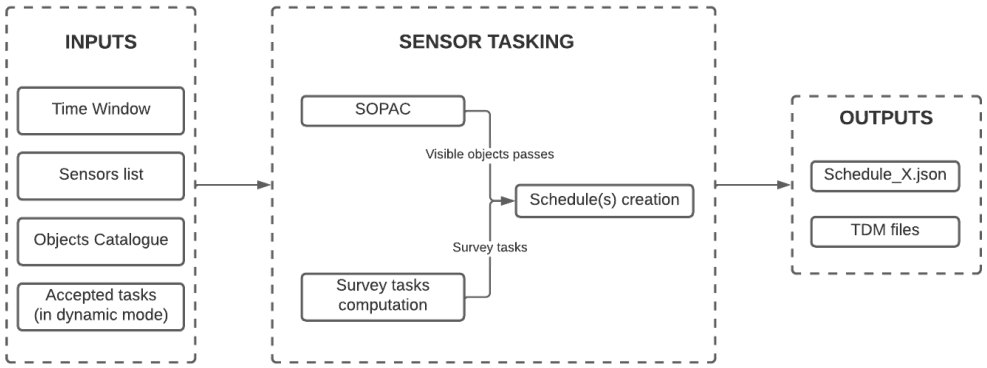


Figure 2: General high-level architecture for the SNOS software.

3.1.1 Supported sensor types

The list of available sensors can be populated both with optical as well as radar and laser sensors. SNOS is, in fact, capable of computing observation opportunities and design operation schedules for both type of sensors. Moreover, both monostatic and bistatic radars are supported, the difference being that in the latter case the transmitting and receiving stations are situated in different locations.
SNOS also supports a peculiar kind of sensors, called *meridian crossings sensors*, which lack the ability of being maneuvered along the azimuthal direction and are therefore limited to move only along the local meridian. An example is the BIRALES[3] sensor operated by the Italian Airforce in Medicina (BO), Italy.
The possibility of tasking space-borne sensors has been implemented too in the prototypal version of the software and

will be exposed to the final user in later versions of the web-based application, once such kind of sensors become more widely available and used.

Different kind of properties can be specified for each sensor in order to identify both their scientific capabilities (and hence whether they are able to observe an object under certain circumstances or not) and their technological capabilities like, for example, the velocity at which they can move and slew from one pointing position to another one.

As far as the operative modes are concerned, by the customers' request, a sensor can be designated to only perform tracking or survey tasks (not both) for the entirety of the scheduling time window. Nonetheless, when in tracking mode, a sensor still preserves the ability to perform simple survey tasks when not already busy with a tracking task.

3.1.2 Supported observation modes

The sensors can be operated either in tracking mode or in a number of different survey modes. In either case (a priori for the tracking sensors and with a later check for the sensors in survey mode), the operative and physical limits of the sensors are respected and the visibility of the observed object, if expected in the operative mode at hand, is taken into account.
In particular, the various modes operate as explained in the following paragraphs.

3.1.2.1 Tracking mode

In tracking mode, an object is followed along its trajectory on the celestial sphere trying to keep it at the center of the FOV of the sensor. For this reason, it is useless to define a FOV for a sensor in this mode, since it is not taken into account during computations.
A special case is the one of the Meridian Crossing (MX) sensors in tracking mode. Such sensors lack, in fact, the ability (at least in one of the stations) to move in azimuthal direction, they can only move along the local meridian. Therefore, when dealing with such kind of sensors, a real tracking can not be performed. The visible passes are instead identified computing the epoch and position at which a certain satellite crosses the local meridian, from which the epochs at which the object enters and exits the FOV of the sensor are computed. In this case, the FOV has to be specified in its height and width (while the pointing is not necessary) and the time span between the entering and exit of the object from the FOV constitutes the pass that is then submitted to the schedule optimization process.

3.1.2.2 Survey modes

Differently from what is done for the tracking tasks, the survey tasks are typically not linked to the necessity to observe a specific space object, but rather used to investigate a certain portion of the celestial sphere looking for satellites and objects that cross it.
Depending on the objective of the observation and on the orbital regime of the objects to be investigated, different kinds of survey modes are made available.

Fixed point survey
The observation of a single point on the celestial sphere, fixed with respect to the observer, is scheduled in this mode. Such pointing is required by the user and expressed as desired azimuth and elevation angles with respect to the receiving station. In the case of a bistatic radar, the corresponding pointing of the TX is computed, given the desired altitude (with respect to the surface of the spherical model of the Earth with radius 6378 km) of the beams crossing point. Such pointing directions are hold for the time necessary for the measurements to be executed. The operator can require the observation of multiple fixed points with the same sensor during the same planning time window at different epochs. In such case, it is necessary to specify the slew motion velocities in azimuth and elevation of the sensor, since those are used to check that the time between the end of an observation and the beginning of the following one is sufficient to reorient the sensor. If not, the two tasks are said to be in conflict and the software either reports this to the operator as a warning or automatically solves the conflict (if the relevant parameter is provided) resizing the second task so that it no longer overlaps with the first one.
Note that, since MX sensors can not move in azimuth, the fixed point one is the only kind of survey (except for the ad-hoc version of the orbital survey explained below) they are able to perform, independently on whether they are in tracking mode or completely designated to the survey mode.

Parking survey
Sensors in tracking mode and meridian crossing ones are allowed to perform a fixed point survey task, if requested by the operator, in those time spans in which no objects passes are being observed. This mode is conceptually identical

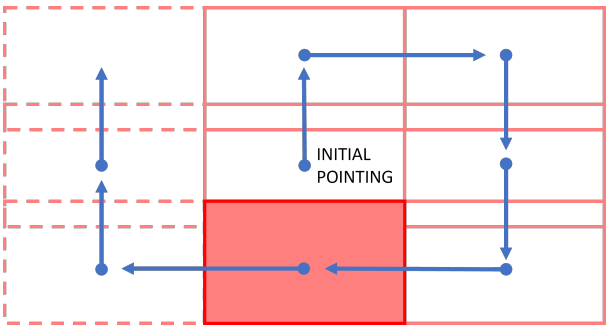


Figure 3: Schematic representation of the spiral survey mode.

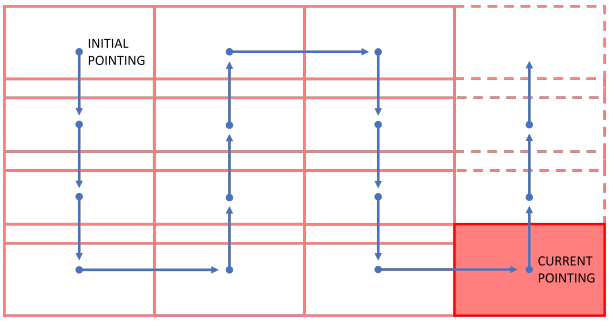


Figure 4: Schematic representation of the GEO survey mode

to the fixed one, the only difference is that all of the sensor's free time is allocated to the observation of the specified point, rather than keeping the position for a specified time.

Spiral survey
This operative mode can be performed by optical sensors and monostatic radars only. It consists of a number of consecutive discrete pointings, laid down in a spiral pattern realized specifying the initial point (center of the spiral) and the number and duration of the measurements to be performed for each pointing, from which the time spent in each position is derived. The sensor stays still for the time necessary to the measurements and it is then moved of an angle equal to a fraction of the width of the FOV in that direction, so that a certain overlap is granted between subsequent pointings and the coverage of the area of interest is improved. A graphical representation of this survey mode is provided in Fig. 3. Just like in the case of the fixed point survey, it is possible to specify multiple initial epochs and pointings. The software will verify if they are compatible with each other, assuring that the time necessary to perform a spiral and reorient the sensor for the next one is smaller of the one between the two initial epochs. If not, a warning message reports the list of conflicting tasks and the operator is asked to manually select a subset of such tasks to be performed.

GEO survey
In this mode, given a range of longitudes and one of latitudes across the equator, the sensor sweeps the area defined by these limits along vertical contiguous stripes travelled in alternating directions. Just like in the spiral survey, this is not done in a continuous way, but keeping a fixed position on each pointing for the time necessary to perform the required measurements. As in the previous case, it is therefore necessary to specify the dimensions of the FOV. Figure 4 provides a graphical representation of this survey mode.

3.1.2.3 Orbital survey modes

The survey modes explained above are of general interest, but they are not the only ones that SNOS can design and schedule. In particular, when orbital survey modes are prescribed, the neighborhood of a satellite orbit is analyzed, looking for possible objects of interest. This is particularly useful in the case of satellite fragmentations, for example, to try to identify and locate new objects generated by the event. The orbital survey can be performed in three different ways, depending on the requirements from the sensor operator and/or the kind of sensor.

Simil-tracking mode
The tracking of the point of maximum elevation of the orbit is performed (see Fig. 5) and the specification of the FOV is therefore not required. The point of maximum elevation is identified through a golden section search algorithm on

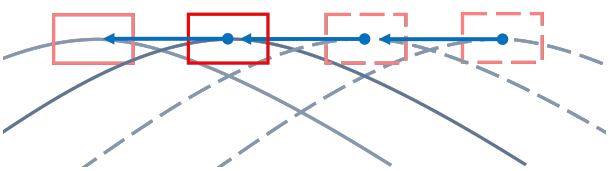


Figure 5: Schematic representation of tracking-like orbital survey mode.

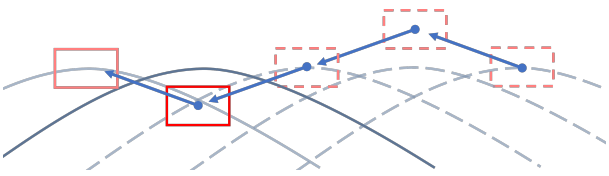


Figure 6: Schematic representation of multi-shot orbital survey mode.

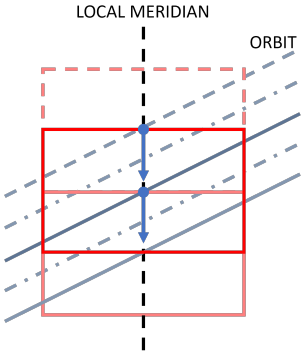


Figure 7: Schematic representation of survey mode for MX sensors.

the nominal orbit at the initial time of the task (perturbations on the orbit are negligible in such a short timeframe). This mode is best suited for radars and laser sensors.

Multi-shot mode
Best suited for optical sensors, this mode performs multiple shots (in a number specified by the user) in discrete pointings along the direction transversal to the locus of the maximum elevation points, following the maximum elevation point in azimuth and moving by a fraction of the FOV height in elevation after each pointing (see Fig. 6). The FOV dimensions have therefore to be specified.

MX mode
For sensors in meridian crossing, the pointing is kept still until the crossing point between the orbit and the local meridian is within the FOV of the sensor. Then, the sensor is moved to center the crossing point in the FOV again (see Fig. 7). The FOV dimensions need to be specified.

3.2 SOPAC (Space Objects PAss Calculator)

Once the inputs have been defined and the relevant Python classes instantiated, the software has to identify all of the possible observation opportunities on the network, i.e. finding when and for how long any of the objects in the catalogue can be seen by any of the available sensors in the network during the prescribed scheduling time window. Such occurrences are referred to as *visible passes*. Developed as a library for the implementation of the SENSIT software suite,[4] the SOPAC (Space Objects PAss Calculator) sub-module is entrusted with such task. Within SOPAC, NAIF's SPICE Toolkit is used to perform core computations related to space missions geometry.[1,2]

The evaluation is only done for the sensors which are set in tracking mode, since survey tasks are uncorrelated to the instantaneous motion of the satellite and can therefore be designed asynchronously with respect to the tracking ones, based on the user requirements.

Along with additional data, such as the NORAD number and estimated values of the brightness and radar cross section of the satellite, the latest available Two-Line Element (TLE) is provided too as part of the object definition. This information is used by SOPAC in order to propagate the trajectory of the objects in the catalogue for the entire duration of the scheduling window. To do so, the SGP4 simplified perturbations model is used. The reason behind this design decision is twofold:

- TLEs are typically the most readily available information on the orbital state of a satellite and their representation is specific to the simplified perturbations model;
- The evaluation of the state of a satellite using SGP4 is computationally much more efficient than the propagation with a dedicated dynamic model and integrator, hence allowing a greater scalability of the application for larger catalogues of objects.

The downside to this choice is that the evaluation with the SGP4 model is reasonably accurate only within a few days from the TLE epoch. This, therefore, limits the maximum duration of the scheduling time window. However, in an operative context, the scheduling is anyway only performed for the few upcoming days, so the limits of the method are deemed irrelevant.

The algorithm employs the bisection method in order to identify when the object at hand becomes visible from a certain sensor during its motion along the orbit. In doing so, the visibility of the spacecraft is evaluated based on

several factors, both based on the capabilities of the sensor, its type, its coordinates and altitude, and on the features of the object to be observed. As an example, among the others, the elevation of the object above the horizon, its distance from the stations, its magnitude and illumination conditions, its slant range and slant range velocity, the signal loss due to atmospheric extinction and its topocentric coordinates are evaluated and checked against the admissible values for the sensor.
Some parameters are specific to the type of sensor at hand, for example, the brightness of the object is accounted for only in the case of optical sensors, while the radar cross section is considered for radar sensors only. Similarly, the angular separation from the moon is only evaluated for optical sensors. The separation from the Sun direction, instead, is also accounted for when using radar sensors, being the star a relevant source of radiation in the radio frequencies range which would cause a poor quality of the measurements.

Specific forbidden regions can also be specified for each sensor to avoid, for example, areas of the celestial sphere that are never observable due to physical limitations of the mounting of the sensor or obstructions like the presence of mountains.

3.3 Heuristic optimization process

Once all the passes of the objects in the catalog over the sensors in the list of available ones have been identified, the survey tasks are built too and they are fed (along with any task scheduled during a previous run of the software, if in dynamic mode), to the core portion of SNOS which generates a user-specified number of admissible schedules. The term *admissible* indicates that the schedules are actually feasible, i.e. they don't contain any task which is in conflict with the others. A conflict is identified whenever two tasks can not be both performed, e.g. when two passes are both observable from the same sensor at the same time or when the time interval between them is not sufficient for the sensor to move from the last pointing of the first task to the first one of the second pass.

During the input definition, a priority level is assigned to each of the objects to be observed by tracking sensors. The schedule is procedurally generated by cycling through the various priority levels in descending order. In other terms, given a group of passes which are linked to targets with equal priority, the conflicts among them and any other task already scheduled are identified and solved. The chosen tasks are then added to the schedule before moving to the next group of tasks with lower priority.
The solution of the conflicts, in particular, is handled as follows. The passes are subdivided into groups in which any of the passes is conflicting with at least one of the other passes in the same group. The passes are then split into smaller pieces of specified time duration, called *subpasses*, the conflicts among the subpasses are identified and a true-false table is built, which conveys this information. The solution is then built by iteratively choosing random subpasses among the ones which are not in conflict with the ones which have already been chosen in the previous iterations.
This ensures the feasibility of the set of passes, but not the optimality of the solution. To overcome the problem, more than one possible solution is generated and a score is assigned to each of them as the summation of the scores evaluated for each of the passes that compose them. This score is based on the expected quality of the observation and the catalog coverage, weighting the radar signal loss or magnitude of the objects, the elevation over the horizon, the distance from the transmitting and receiving stations, the duration of the pass and whether or not the object is already scheduled to be observed by another sensor.
At this point, all of the groups of conflicting passes for the given priority level are solved, the set of tasks corresponding to the highest scoring solution for each of the conflicting pools is added to the final schedule along with the tasks that were initially free from conflicts and the analysis moves onto the next priority level. The method hence somehow resembles the first step of a genetic algorithm.
Once all the tracking tasks have been analyzed, the survey tasks are fit within the proposed schedules and resized as needed so that there are no conflict among them and the tracking tasks.

The computed schedules are ordered by score and exposed to the user so that he can choose the one to be performed among them and request modifications through the dynamic mode if needed. If in dynamic mode, information about cancelled or modified tasks from the baseline schedule is conveyed to the operator too.

3.4 Results encoding

The results obtained by the back-end analysis is finally encoded in a format that can be used from the web interface to expose the necessary information to the end user (i.e. the sensor operator), so that he can make an informed decision and launch the operative routine.
The information is provided both as a .json file and as a Tracking Data Messages (TDM). The JSON file contains an entry for each of the tasks that have been scheduled, with information on the sensors that have to perform them. Each task entry reports the series of coordinates couples (azimuth and elevation or right ascension and declination) and the

epochs at which the sensor has to be pointed towards them in order to correctly perform the task.
A TDM file is produced for each of the tasks too, conveying the same information.

3.5 Web Interface

The final result of the algorithm, when fully integrated in the ISOC architecture is depicted in Fig. 8. With a very intuitive HMI, the operator can easily set the sensor availability and plan the operational mission, running the algorithm for the sensor tasking and mission planning in back-end.

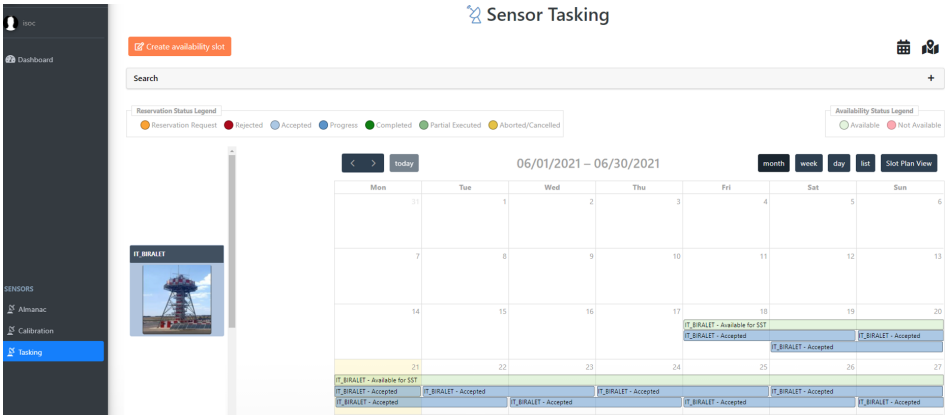


Figure 8: ISOC - Sensor Tasking web interface

4. Software validation and results

The present section reports tests and results obtained through the use of the SNOS software.

4.1 SOPAC validation

As explained in Section 3, SNOS relies on the SOPAC module for the computation of the satellites transits that could possibly be observed by any of the available sensors in the network. Validating this piece of software is therefore of uttermost importance in order ensure that valid schedules are obtained.

Among the performed tests, the one involving the PdM-MiTe sensor[1] is here partially reported. Observations were carried out with the sensor and the related TDMs were produced using the canonical techniques. The epochs reported in the TDMs were then compared, a posteriori, to the transit windows computed by SOPAC for the same objects to verify that the passes identified by the software actually match the ones observed in real life.

For the sake of brevity, Fig. 9 only shows a portion of the validation report that was generated during the test, but analogous performances were obtained in all the other cases. All of the observed transits were actually predicted by SOPAC and, just like what's shown for the specimens in the picture, full coverage of the passes would have been obtained, should the analysis of the observation opportunities for the given time window be entrusted to the SOPAC sub-module.
The accuracy of the pointing directions as computed by SOPAC with respect to the ones reported in the .tdm files was evaluated too. The average difference is in the order of 10^{-4} while the highest absolute value registered was roughly 2.5×10^{-3}, well below the sensible tolerance for the kind of tasks to be performed.

4.2 Typical results

Here we report and analyse an example of the typical schedule structure that can be obtained using SNOS.

Figure 10 reports the results obtained with SNOS in different conditions, demonstrating that the work logic expressed in Section 3 is respected. Figures 10a and 10b report the visible transits, as computed by SOPAC, of three simulated satellites over a fictitious bistatic radar sensor in tracking mode whose transmitting station is located in

[1]PdM-MiTe is a 350 mm optical telescope located in the military airport base "Mario de Bernardi" in Pratica di Mare and it is used by the personnel of the Flight Test Wing for Space Surveillance and Tracking applications.

```
--------------------   Pass of sat: 48341   ---------------------
  TDM   epochs: 2021-07-15T19:16:31.000 2021-07-15T19:28:46.000
  sopac epochs: 2021-07-15T19:16:30.781 2021-07-15T19:29:42.936
  TDM epochs within sopac pass (+-2s): 50 / 50 - 100.00 %
  TDM time within sopac pass: 735.000 s / 735.000 s - 100.00 %
  Max err ra:  2.4732e-03 deg
  Max err dec: 1.2264e-03 deg
-------------------------------------------------------------

--------------------   Pass of sat: 42778   ---------------------
  TDM   epochs: 2021-07-15T19:33:46.000 2021-07-15T19:35:16.000
  sopac epochs: 2021-07-15T19:28:15.490 2021-07-15T19:35:28.545
  TDM epochs within sopac pass (+-2s): 7 / 7 - 100.00 %
  TDM time within sopac pass: 90.000 s / 90.000 s - 100.00 %
  Max err ra:  7.0402e-04 deg
  Max err dec: 5.0174e-04 deg
-------------------------------------------------------------

--------------------   Pass of sat: 29165   ---------------------
  TDM   epochs: 2021-07-15T20:42:28.000 2021-07-15T21:23:13.000
  sopac epochs: 2021-07-15T19:51:19.124 2021-07-15T22:31:20.017
  TDM epochs within sopac pass (+-2s): 83 / 83 - 100.00 %
  TDM time within sopac pass: 2445.000 s / 2445.000 s - 100.00 %
  Max err ra:  1.0809e-03 deg
  Max err dec: 5.1084e-04 deg
-------------------------------------------------------------
```

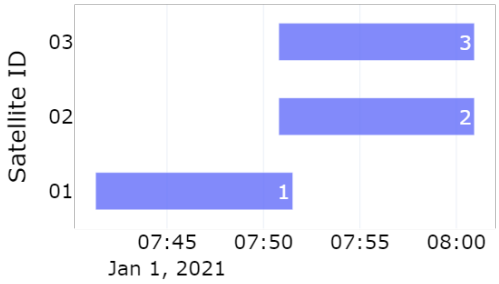
Figure 9: Some of the results from the SOPAC validation.

(N39.60°, E9.45°) and receiving station in (N44.52°, E11.60°). Observation are assumed to be performed on the first of January of 2021. As you can see, the involved time spans, representing the visibility windows of different satellites in the catalogue, overlap and the sensor can not therefore be used to observe all the passes without any proper scheduling. The numbers inside the bars represent the priority assigned by the operator to that particular satellite and this influences the resulting schedule. Figures 10c and 10d,in fact, show instead the scheduled tasks as chosen by the SNOS algorithm in the two different scenarios. First of all, one can notice that the conflicts are solved in both cases, as the tasks don't overlap anymore and a certain time span is allocated between them to allow the sensor to move from one pointing direction to the next one. The entity of this gap depends on the angular separation between the pointing directions as well as the physical capabilities of the sensor, i.e. the maximum allowed slew rates in azimuth and elevation. The additional time needed for operative reasons, such as data download, is also accounted for if specified by the operator.

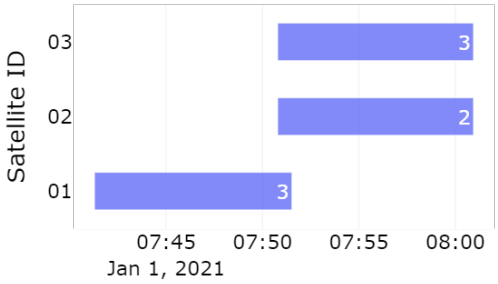
It can be noticed how, even though the visible transits are the same in both cases, different fractions of them are chosen depending on the priority. The pass of the satellite with ID 02 is completely neglected in both cases, as its duration overlaps with the one from the transit of satellite 03, which is assigned an higher priority. Notice, instead, how in the first case (Figs. 10a and 10c) the transit of the satellite with ID 03 is completely observed at the expenses of the pass of satellite 02, which, having lower priority, is only partially observed to allow for the needed technical time between the tasks. In the second case, instead, both tasks are resized to minimize the cost function since they have the same priority.

Finally, Fig. 11 shows a portion of a typical schedule timeline as it is exposed to the sensor operator after an execution of the software in dynamic mode to allow for the addition to the catalogue of a new space object to be tracked. For readability reasons, the picture focuses only on the results obtained for a sensor, but the whole timeline reports the analogous information for all the other sensors in the network too.

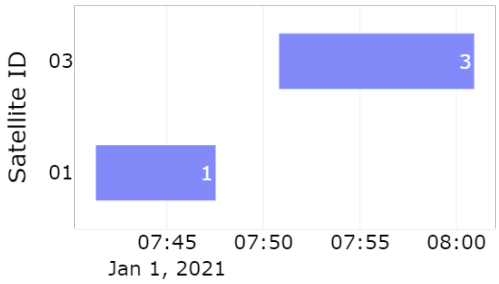
By design choice, information on the tasks of the original schedule that were modified or deleted during the dynamic run is retained and it is also shown in the timeline. Red bars, in fact, represent the tasks that were scheduled in static mode but got deleted in dynamic mode to allow for the introduction of new tasks linked to the introduced modification either to the sensors list or object catalogue. Analogously, purple lines show the tasks that were modified, i.e. a slightly different portion of the same visible transit is chosen with respect to what was done in the previous execution of the scheduling software. Finally, unchanged tasks are shown in blue while green bars represent the newly introduced tasks and the modified versions of the old ones. So, in conclusion, the summation of blue and green tasks represents



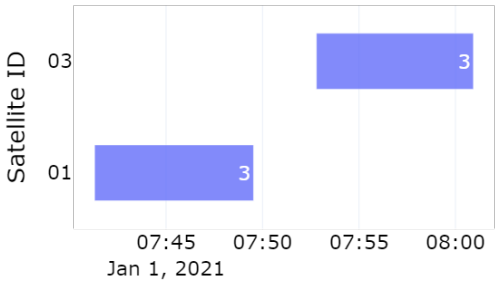
(a) Observable transits computed by SOPAC for three sample satellites, with different priorities, over the same sensor.



(b) Observable transits computed by SOPAC for three sample satellites, with same priorities, over the same sensor.



(c) Schedule computed by SNOS selecting tasks from the available passes shown above.



(d) Schedule computed by SNOS selecting tasks from the available passes shown above.

Figure 10: Representative figure of the behavior of SOPAC and SNOS.

the current accepted version of the schedule, while the sum of blue, purple, and red tasks represent the old schedule. Obviously this is just a graphical representation of what is detailed in the .json output file introduced in Section 3, but it is useful and intuitive nonetheless.

5. Conclusions

In this work, we presented SNOS, a new software developed by Politecnico di Milano in collaboration with Leonardo company, for the Italian air-force. The software allows for the automatic optimal scheduling of the observations over a network of sensors, handling different types of sensors and tasks to be performed depending on user requirements. It is part of the ISOC2.0 suite and can be used by the operators of the sensors via a simple and intuitive web interface. It was demonstrated that it accurately predicts the observation opportunities and that the created schedules respect the physical limits of the sensors and the desired policy in terms of priority handling of the space objects to be observed. A refined version of the scoring function used for the optimization process and the possibility to use evolutionary algorithms instead of the heuristic approach currently employed will be evaluated as possible future developments for the next iteration of the ISOC software suite.

6. Acknowledgments

The authors would like to thank Aeronautica Militare and Leonardo Company for the opportunity and the support provided during the development of this project.

References

[1] Charles Acton, Nathaniel Bachman, Boris Semenov, and Edward Wright. A look towards the future in the handling of space science mission geometry. *Planetary and Space Science*, 150:9–12, 2018. Enabling Open and Interoperable Access to Planetary Science and Heliophysics Databases and Tools.

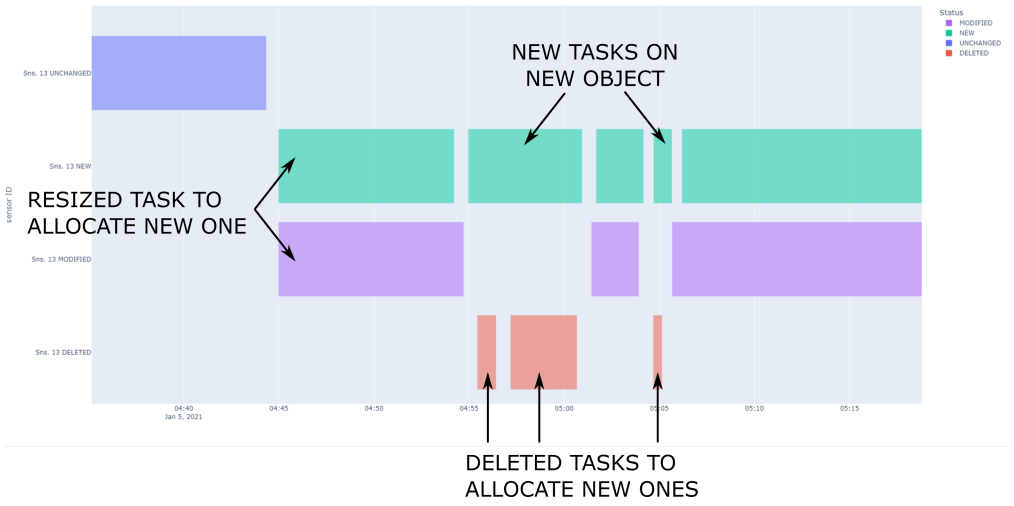


Figure 11: Example schedule generated using SNOS in dynamic mode.

[2] Charles H. Acton. Ancillary data services of nasa's navigation and ancillary information facility. *Planetary and Space Science*, 44(1):65–70, 1996. Planetary data system.

[3] Matteo Losacco, Pierluigi Di Lizia, Mauro Massari, Germano Bianchi, Giuseppe Pupillo, Andrea Mattana, Giovanni Naldi, Claudio Bertolotti, Mauro Roma, Federico Schiaffino, Marco Perini, Luca Lama, Alessio Mazro, Denis Cutaiar, Josef Borg, Walter Villadei, and Marco Reali. The multibeam radar sensor birales: Performance assessment for space surveillance and tracking. In *2019 IEEE Aerospace Conference*, pages 1–13, 2019.

[4] G Purpura, A De Vittori, R Cipollone, P Di Lizia, M Massari, C Colombo, A Di Cecco, and L Salotti. Sensit: a software suite for observation scheduling and performance assessment of sst sensor networks. In *72nd International Astronautical Congress (IAC 2021)*, pages 1–13, 2021.