# NEXT-GENERATION SPACE OBJECT RADAR TRACKING SIMULATOR: SORTS++

**D. Kastinen**[(1,2)], **J. Vierinen**[(3)], **J. Kero**[(1)], **S. Hesselbach**[(4)], **T. Grydeland**[(5)], **and H. Krag**[(6)]

[(1)]*Swedish Institute of Space Physics (IRF), Box 812, SE-98128 Kiruna, Email: daniel.kastinen@irf.se*
[(2)]*Umeå University, Department of Physics, SE-90187 Umeå*
[(3)]*University of Tromsø (UiT), Hansine Hansens veg 18, 9019 Tromsø, Norway*
[(4)]*Braunschweig University of Technology (TU-BS), Universittsplatz 2, 38106 Braunschweig, Germany*
[(5)]*Northern Research Institute (Norut), Postboks 6434, 9294 Tromsø, Norway*
[(6)]*ESA ESOC Space Debris Office, Robert-Bosch-Str. 5, 64293 Darmstadt, Germany*

## ABSTRACT

We describe the software toolbox SORTS++ which is the next-generation Space Object Radar Tracking Simulator (SORTS). This toolbox is able to simulate general Space Surveillance and Tracking (SST) applications from catalogue maintenance to cold-start performance for radar systems or collections of radar systems. We have applied this toolbox to the EISCAT_3D radar system, currently under construction, to optimize observation strategies and predict the SST capabilities of the system. We present the preliminary results from this ongoing study and illustrate the versatility and capability of the developed toolbox.

Keywords: EISCAT_3D; SST; Simulation; Radar.

## 1. INTRODUCTION

Earth is surrounded by a cloud of space debris ranging from dead satellites and rocket stages to fragments of material and flecks of paint. According to estimations by the European Space Agency (ESA) the total mass is more than 7500 tonnes. Fragmentation events are assumed to have generated a population of objects larger than 1 cm numbering on the order of 750,000. With the growing man-made population of objects in orbit of the Earth, so increases the need to monitor these objects to minimise the risk for collisions [Kli06]. Collisions may damage valuable infrastructure in space and cause an exponential growth of the number of objects, a catastrophic chain re-action of fragmentation events. Even a small screw travelling at high velocity can damage a satellite or disturb a scientific measurement. Therefore, it is desirable to track the debris population, as well as active satellites and other hardware in space, so that the active infrastructure can be maintained and an exponential growth of space debris is avoided.

Fortunately, there are many reliable instruments for detecting objects in space from the ground, one of them being radars. These radar systems, given the appropriate frequencies to penetrate beyond the ionosphere of the Earth, can transmit radio waves that scatter off the space debris thereby generating echos. These echos can then be coherently integrated revealing both ranges and range-rates between the object and the receiver station. If there are three or more stations sufficiently separated on the ground simultaneously measuring the echoes of a transmitted radio signal, the instantaneous position and velocity vector of an object can be obtained. Given this state vector, an orbit can be determined and using propagator software potential collisions can be predicted and avoided.

Many radar systems are designed for particular purposes, such as ionospheric measurements, and thus it is not known if they are capable of significantly contributing to the monitoring of objects in space. Thus we have developed a software toolbox to simulate the performance of arbitrary radar systems, using arbitrary measurement protocols, discovering and tracking an arbitrary population of space objects in orbit of the Earth. This toolbox is called Space Object Radar Tracking Simulator (SORTS++) and is based on the software developed in [VMK+17]. The software is written in Python but uses wrapped C and FORTRAN code. We will here give a short overview of the software layout, its capabilities, some methodology and preliminary first results from a case study of the EISCAT_3D radar system [MAA+15].

## 2. SOFTWARE OVERVIEW

The software is built in a modular fashion where modules consists of classes, functions and libraries of instances. These modules are divided into categories as described by the coming subsections. The lists below name the particular modules of SORTS++ and shortly describes their use.

## 2.1. Simulation handler module

- [**simulation**] Main simulation handler in the form of a class using the capabilities of the entire toolbox.

- [**sorts_config**] Lists all propagators and sets which is used system-wide (unless manually changed on module and script level)

## 2.2. Class modules

- [**radar_config**] This module is used to define a radar network configuration.

- [**antenna**] Defines an antennas, or entire radar systems, radiation pattern in the form of a class.

- [**base_propagator**] A parent class used for interfacing a propagator.

- [**population**] Defines a population of space objects in the form of a class.

- [**radar_scans**] Defines a radar observation scheme in the form of a class.

- [**space_object**] Defines a space object in orbit around the Earth (governed by a propagator) in the form of a class.

## 2.3. Function modules

- [**debris**] Radar signal to noise calculations for hard targets.

- [**keplerian**] Keplerian orbit calculations.

- [**ccsds_write**] Consultative Committee for Space Data Systems (CCSDS) Tracking Data Message (TDM) file writer.

- [**coord**] Collection of common coordinate transformations.

- [**dpt_tools**] Functions from Daniels-Python-tools package (plotting and coordinate transformations).

- [**population_filter**] Investigate what fraction of objects can be detected with a radar system.

- [**ray_trace**] Ionospheric radio propagation effects studied using ray-tracing.

- [**simulate_scan**] Simulate discovery of objects with user-defined scan pattern and custom radar system.

- [**simulate_tracking**] Functions for simulating the tracking of an object in space.

- [**simulate_tracklet**] Given scheduled observations of an object, simulate the generated tracklet-data.

- [**simulate_scaning_snr**] Functions for single object propagation and SNR examination.

- [**logging_setup**] Sets up a logging framework that can be imported and used anywhere.

- [**orbit_accuracy**] Linearized error determination for orbital elements.

- [**orbital_estimation**] Estimates a space objects state vector from a set of ranges and range-rates.

- [**plothelp**] Functions for making plots quicker.

- [**lgeom**] Collection of simple geometric functions.

## 2.4. Instance libraries

- [**radar_library**] A collection of radar_config.radar_system instances, such as EISCAT_3D and EISCAT UHF.

- [**antenna_library**] A collection of functions that return common instances of the antenna.beam_pattern class, such as Cassegrain or Antenna array.

- [**radar_scan_library**] A collection of radar_scans.radar_scan instances, such as fence scans or ionospheric grids.

- [**scheduler_library**] Collection of classes and functions related to constructing a radar system scheduler.

## 2.5. Integrator interfaces modules

- [**keplerian_opi**] NEPTUNE-OPI interface with SORTS++.

- [**keplerian_sgp4**] SGP4 interface with SORTS++.

- [**keplerian_simple**] Keplerian propagator interface with SORTS++.

## 2.6. Methodology

The base class structure that all code manipulates is very general and can simulate most system or schemes in existence or in a planning phase. The major restrictions in simulations lie in the specific implementation in the **simulation** class and can be generalized with ongoing updates without framework changes. In the **simulation** class it is assumed that a radar schedule consists of three types of time-slices: slices for SST purposes, slices for other purposes with data access and slices for other purposes without data access. A "user" controls their time slice, i.e. what pulse encoding is used, what inter-pulse period, what pointing direction, etc is used. In a time slice controlled by an SST user, each transmitted radar pulse sequence can be split between tracking and scanning. Changing the respective fraction of the pulse sequence used to track versus scan changes the corresponding SNR for that measurement. The actual echoes are

then coherently integrated over some time to finally yield a tracklet-point: range and range-rate. This methodology is illustrated in Figure 1.

An example setup of a simulation with the **simulation** class is given in Code-listing 1.

Once this simulation has completed the data needs to be analyzed by a scheduler that examines what measurements the radar can actually perform while fulfilling all requirements. For example, due to the scientific experiment scheduled at any particular time, the requirements could be:

- Ionospheric measurements will run for $\approx 80\%$ of radar time

- Ionospheric time slices are 8 s

- SST can access ionospheric measurements data

- SST activities will run for $\approx 20\%$ of radar time

- SST time slices are 0.2 s

Then given 10 seconds of time the scheduler can solve the observation problem in many different ways. One way is to:

- give SST 10 time slices

- then give Ionospheric measurements 1 time slice

- objects above SNR threshold during Ionospheric measurements are discovered

- discovered objects inside the field of view can get maximum 10 follow-up observations (tracklet-points)

- which object that gets the follow up observation is determined by how close it is to its peak SNR and how many tracklet-points it has

In our simulations we have utilized a similar setup for a real-time scheduler with a queue system and a cost function dependant on several parameters.

## 3. PRELIMINARY RESULTS

The simulations presented here were performed as part of an EISCAT 3D Performance Analysis project to find out what quality of orbit data can we expect from EISCAT 3D measurements and how well the radar can be used to maintain a catalogue of objects.

It must be noted that according to the EISCAT agreement [Ass15], "... the Association may contribute to the international task of tracking objects in space (natural or man-made). For this activity, an agreed list of objects shall be
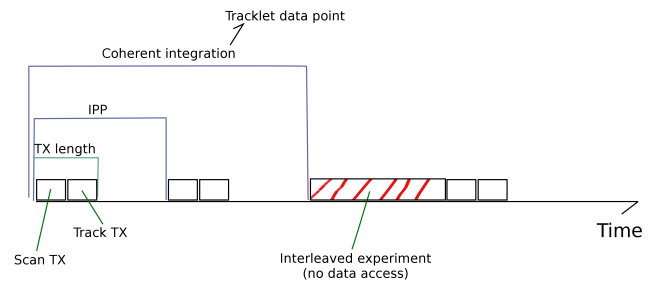


Figure 1. Illustration of the operational mode of a radar system. TX refers to transmission of radio-waves and IPP is the interpulse period, i.e. the time between two consecutively transmitted pulse sequences. Track TX is used for tracking of objects that are known (or recently discovered) and scan is used for discovery of unknown objects. Changing the respective size of Track TX vs Scan TX changes the corresponding SNR for that measurement. The interleaved experiment in this example is marked "no data access" meaning that the data cannot be used for SST purposes.

maintained and the Association shall only conduct tracking of objects from this list."

The simulated cold-start catalogue buildup is a way to evaluate the EISCAT 3D performance, but not allowed according to the current EISCAT agreement and not part of any currently planned activities. The technical implementation status and timeline of the EISCAT 3D project as well as the organisational structure of the EISCAT Scientific Association is further detailed in these proceedings by Kero et al., *EISCAT3D: the next generation international atmo-sphere and geospace research radar*.

The simulation project is still ongoing the results we report here are preliminary. A more detailed description of the contribution to Space Surveillance and Tracking (SST) purposes of the current EISCAT system is given in these proceedings by Vierinen et al., *2018 beampark observations of space debris with the EISCAT radars*.

### 3.1. EISCAT_3D

The target of the case study is the new EISCAT_3D system [MAA+15], a multi-static phased array high power large aperture radar under construction in Northern Fenno-Scandinavia. The radar will be commissioned in the end of 2021 and consist of one transmit and receive site in Skibotn, Norway and two receive-only sites in Kaiseniemi, Sweden and Karesuvanto, Finland. Each site will nominally be equipped with a 75 metre wide array of 109 antenna subgroups, each having 91 antennas, or a total of 9919 antennas (to be decided). The Skibotn transmitter site will be equipped with 5 MW transmitter power.

The radar specifications that were used are listed in Ta-

*Listing 1. Example setup of a simulation using SORTS++. Specifically this example simulates a catalogue cold-start using the EISCAT_3D system measuring the MASTER population by scanning on a dedicated SST mode using a North-South fence scan.*

```python
1   #!/usr/bin/env python
2   import sys
3
4   import numpy as n
5   from mpi4py import MPI
6
7   # SORTS++ core imports
8   import population as p
9   import simulation as s
10
11  #SORTS++ Libraries
12  import radar_library as rl
13  import radar_scan_library as rslib
14  import scheduler_library as sch
15  import antenna_library as alib
16
17  #Create the Radar system instance from library
18  e3d = rl.eiscat_3d()
19
20  #Configure radar system
21  e3d.set_FOV(max_on_axis=25.0, horizon_elevation=30.0)
22  e3d.set_SNR_limits(min_total_SNRdb=10.0, min_pair_SNRdb=0.0)
23  e3d.set_TX_bandwith(bandwith = 1.0e6)
24
25  #Set radar systems radiation pattern instance from library
26  e3d.set_beam('TX', alib.e3d_array_beam_stage1() )
27  e3d.set_beam('RX', alib.e3d_array_beam() )
28
29  #Create SST observation schema instance from library
30  #North-South random fence scan
31  e3d_scan = rslib.ns_fence_rng_model(min_el = 30.0, angle_step = 2.0, dwell_time = 0.2)
32
33  #Set this as the radar systems observation schema
34  e3d_scan.set_radar_location(e3d)
35  e3d.set_scan(e3d_scan)
36
37  #Load the population in the standard population class instance format
38  #Here a pre-defined instance for for MASTER2009 is used
39  #As the MASTER model is a statistical model a seed is required for consistency.
40  pop = p.filtered_master_catalog_factor(seed=12345)
41
42  #Setup the simulation instance
43  sim = s.simulation( \
44      radar = e3d,\
45      population = pop,\
46      sim_root = '/sst_cold_start'
47  )
48
49  #Set the simulation observation parameter restrictions
50  sim.calc_observation_params(
51    duty_cycle=0.25, #EISCAT_3D duty cycle in fraction
52    SST_fraction=1.0, #Radar time allocated to SST activities
53    tracking_fraction=0.2, #Fraction of the IPP TX length used for tracking vs scanning
54    SST_time_slice=0.2, #Coherent integration length in seconds
55  )
56
57  #Run simulation and scan for detections for 24 hours
58  sim.run_scan(24.0)
```

ble 1. When doing discovery of unknown objects the side-lobes of the antenna radiation pattern become significant contributors to coherent radar echos in the signal and can therefor not be ignored. Thus for all simulations the full gain pattern was calculated with the assumption that the single antenna gain pattern is proportional to the cosine of the zenith angle and that the signal mixing is described by complex summation. The resulting gain pattern for pointing the antenna array at zenith is illustrated in Figure 2.

### 3.2. Simulation setup

The simulation used the SGP4 propagator [SJPSMV17] to govern the dynamics of the simulation. To simulate a correction for ionospheric effects, ray-tracing and the International Reference Ionosphere (IRI) 2009 model [BR08] were used, this introduced additional errors to the simulated measurements. The signal was coherently integrated for 0.2 seconds to create a tracklet point or to discover a new object. The simulations are paralellized using Message Passing Interface (MPI) and were run on a 64-core server with 128 GB of RAM.

### 3.3. Input population

To perform a simulation of detections or tracking of space objects one needs a realistic population distribution to detect or track. We used the Meteoroid and Space Debris Terrestrial Environment Reference (MASTER) 2009 catalogue [FGW$^+$09]. This is a statistical model that is used to assess the debris or meteoroid flux imparted on a spacecraft on an arbitrary earth orbit. The model covers a size regime for debris and micro-meteoroids between 1 $\mu$m and 100 m.

The MASTER model is based on taking known populations and all known, historic debris-generation events and thereby, with high spatial resolution, calculate an object population. More than 290 in orbit fragmentation events, 2000 solid rocket motor firings, 16 reactor core ejections have been used. The included meteoroids are derived from the ESA Interplanetary Meteoroid Model (IMEM).

To reduce computation time and to have a reference number we also performed a filtering on the MASTER population before the simulations commenced. The filtering was done as such: propagate the entire master catalogue for $t$ time, find the maximum SNR measurement that can be archived during that time (if pointing the beam straight at the target), if this SNR is above a threshold it is considered detectable and kept in the population of final size $N_t$. We did not consider objects smaller than 1 cm, we used $t = 48$ h and an SNR threshold of 10 dB resulting in $N_{48}$ =73,886 objects. Originally there are 549,014 objects larger than 1 cm in MASTER 2009 so around 13.5% can be considered detectable given a short amount of time. Of course this number has a asymptote at $N_\infty$ but it should be close to $N_{48}$.

### 3.4. Long term catalogue maintenance

The first case study was to examine the performance of EISCAT_3D in maintaining a large catalogue of objects, i.e. updating their orbital elements and with continued measurements keep orbital uncertainties low. We also wanted to examine the maximum possible performance so it was assumed that the radar would run on continues 25% duty cycle and that SST applications was given 100% of this radar time. To perform this simulation we marked all 73,886 objects as known and simulated 7 days of maintenance, during this time 99.98% of objects were maintained and 2,409,287 tracklets were created. However, out of these tracklets only 1,548,307 contained measurement points while 860,980 were empty due to lack of radar time, this is illustrated in figure 4. The scheduler had on average 1,000-2,000 objects simultaneously in the queue requesting additional tracklet points making the maintenance tracklets very sparse. The load on the system is also illustrated by the tracklet-frequency in figure 3. Since the scheduler was set to achieve maximum object coverage. It was not set to perform triage, i.e. prioritize object with the best probability of being maintained and ignore objects that have a low probability of being maintained with the configured radar system. Instead, the tracklet point distribution center around the optimum SNR, as illustrated in figure 5, while keeping the span relatively small, as illustrated in figure 6. This indicates that since the scheduler is overloaded it only hands out very few data points but of the highest possible quality.

### 3.5. Cold-start catalogue buildup: SST mode

The opposite to a full maintenance operation is to have an empty catalogue and fill it with discovered objects, a so called cold start. Again we wished to examine maximum performance and thus it was assumed that the radar ran on continues 25% duty cycle and that SST applications was given 100% of this radar time. The SST observation mode was set to a random North-South fence scan. We simulated 2 days of filling the catalogue and during this time 34,124, or 46.18% of the detectable population, were discovered. The cumulative catalogue buildup is illustrated in figure 7 where the red line indicate the detectable population. Given more time and changes in the observation pattern this should get closer to $\approx$100% but at a decreasing rate. The maximum possible SNR of the follow up tracklets is illustrated in Figure 8.

### 3.6. Cold-start catalogue buildup: piggyback mode

Since we have simulated the cold start catalogue buildup in the best case scenario, total dedication to SST, it is also

Table 1. EISCAT_3D simulated specifications.

| Site | Latitude, Longitude [deg] | Gain [dB] | Frequency [MHz] | Duty cycle | Power [MW] |
|------|---------------------------|-----------|-----------------|------------|------------|
| Skibotn TX | 69.340238°, 20.313166° | 42 dB | 233 | 1-25% | 5 |
| Skibotn RX | 69.340238°, 20.313166° | 45 dB | 233 | - | - |
| Karesuvanto RX | 68.463862°, 22.458859° | 45 dB | 233 | - | - |
| Kaiseniemi RX | 68.148205°, 19.769894° | 45 dB | 233 | - | - |
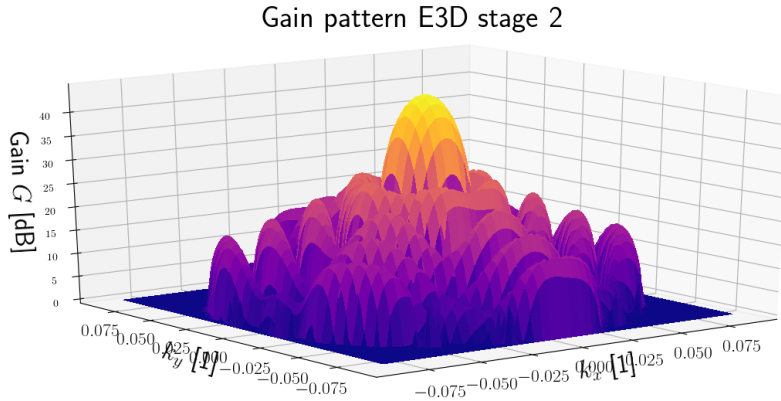


Figure 2. Calculation of the full array radiation pattern of an EISCAT_3D station. The axis $k_x$ and $k_y$ are the ground projection of the normalized incoming wave vector where $k_x > 0$ points to the East and $k_y > 0$ points to the North. The range of $k_x$ and $k_y$ here corresponds to around 5° zenith angle (or 85° elevation).
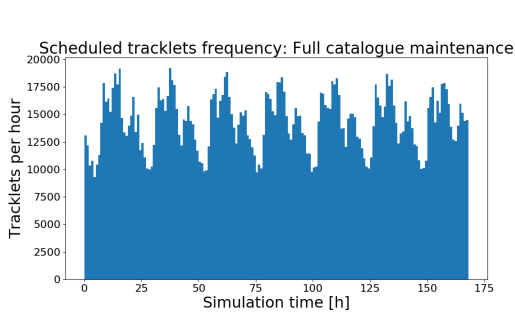


Figure 3. Frequency of tracklet creation during the long term catalogue maintenance. The periodic oscillation in frequency is consistent with groups of objects generated by collision events that are included in the MASTER 2009 statistical model.
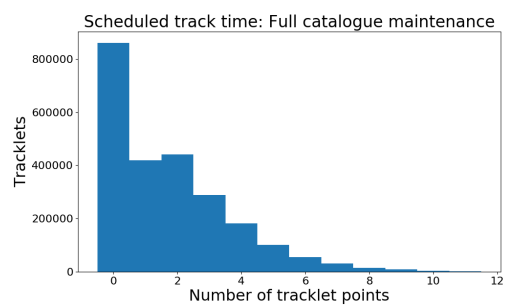


Figure 4. Number of tracklet points for each tracklet generated during the long term catalogue maintenance. A tracklet with 0 points means that the tracklet could have been created but other concurrent objects were prioritized. A tracklet point consists of 0.2 seconds of coherent integration.
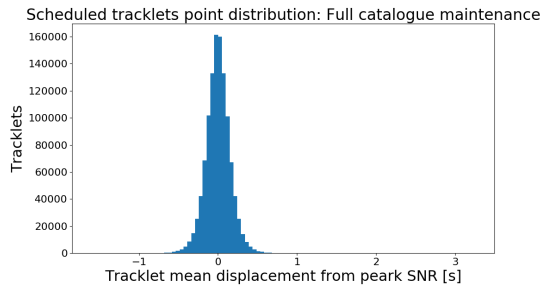
Figure 5. *Distribution of tracklet points in time relative the tracklet peak SNR during the long term catalogue maintenance. The SNR curve for a tracklet is a uni-modal function since the orbital dynamics of objects in Earth orbit is close to Keplerian on the time scales of a pass over a radar field of view.*
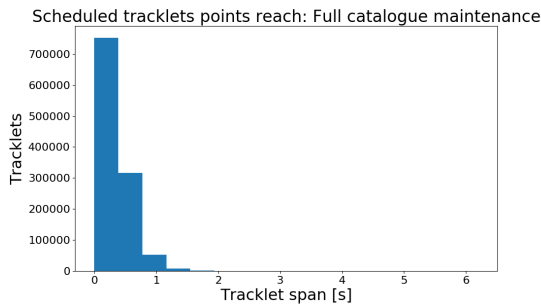


Figure 6. *The total span of a tracklet in time during the long term catalogue maintenance. This is calculated by taking the difference between the Unix time of the first tracklet point and the last tracklet point.*
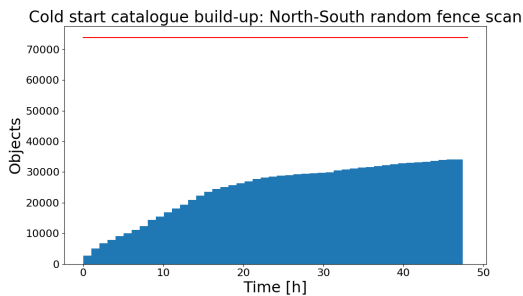


Figure 7. *The cumulative catalogue buildup for a cold-start using a North-South random fence scan. The red line indicates the size of the detectable population.*
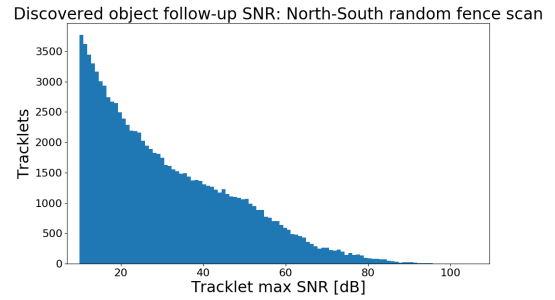


Figure 8. *The distribution of peak SNR for measurements of discovered objects during the cold-start using a North-South random fence scan. The detectability threshold was set to 10 dB as can be seen in the cutoff.*
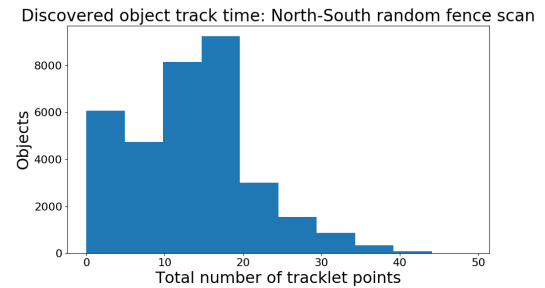


Figure 9. *The total number of follow-up tracklet points a object received after being discovered during the cold-start using a North-South random fence scan.*

desirable to simulate more restrictive scenarios. The one we choose was a piggyback mode where the main observation schema of the radar is grid incoherent scatter measurements of the ionosphere and the radar is running in power saving mode at 1% duty cycle. Here the observation scheme is a 3x3 grid at 300km altitude where corners are separated by 100km in the xy plane. The incoherent integration time assumed was 7.5 seconds, i.e. the time slice for this experiment. We assumed that we are allowed data access to this data so we can coherently integrate the radar echos in the raw data that is otherwise used for incoherent scatter measurements, thereby discovering objects. We also decided that 10 percent of the radar time was allowed for SST following up measurements for orbit refinement of detected objects, but only in-between ionospheric grid point calculations in an interleaved fashion.

This simulation was run for 2 days and during these days 1,834 objects, or 2.48% of the detectable population, were discovered and maintained as illustrated in Figure 10. It is here seen that the buildup has not yet trailed off meaning that the simulation needs significantly longer integration time to find the break point of less efficiency in discoveries. Compared to the full SST mode fence scan that on average detected 711 objects per hour this mode detected 38. These discoveries are also given a reasonable amount of tracklet points given the low available follow up time as illustrated in figure 12. The advantage of such a observation schema is that low amounts of radar
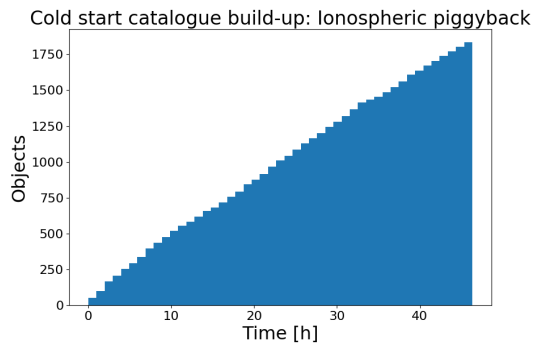
*Figure 10. The cumulative catalogue buildup for a cold-start in a low power ionospheric piggyback mode during a 2 day campaign.*
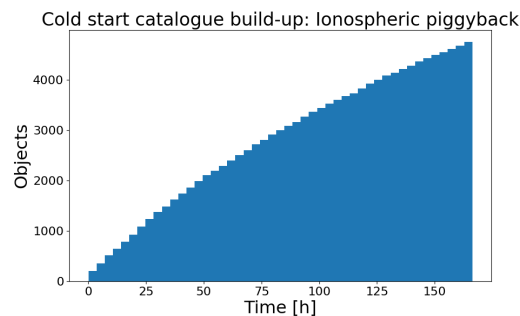


*Figure 11. The cumulative catalogue buildup for a cold-start in a low power ionospheric piggyback mode during a 7 day campaign.*
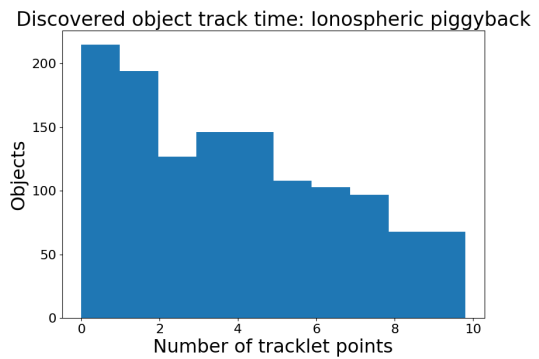


*Figure 12. Number of tracklet points for each tracklet generated during a cold-start in a low power ionospheric piggyback mode during a 2 day campaign. A tracklet with 0 points means that the tracklet could have been created but other concurrent objects were prioritized. A tracklet point consists of 0.2 seconds of coherent integration.*

power and time were utilized and the ionospheric data was re-used for a useful purpose. We extended the simulation to 7 days with a resulting cumulative catalogue illustrated in figure 11, where 4,754 objects, or 6.43% of the detectable population, were discovered showing a slow exponential decline in detections.

## ACKNOWLEDGMENTS

## REFERENCES

Ass15. EISCAT Scientific Association. *EISCAT BlueBook*. EISCAT, 2015.

BR08. Dieter Bilitza and Bodo W Reinisch. International reference ionosphere 2007: improvements and new parameters. *Advances in space research*, 42(4):599–609, 2008.

FGW+09. S. Flegel, J. Gelhaus, C. Wiedemann, P. Vorsmann, M. Oswald, S. Stabroth, H. Klinkrad, and H. Krag. Invited Paper: The MASTER-2009 Space Debris Environment Model. In *Fifth European Conference on Space Debris*, volume 672 of *ESA Special Publication*, page 15, March 2009.

Kli06. Heiner Klinkrad. *Space debris: models and risk analysis*. Springer Science & Business Media, 2006.

MAA+15. Ian McCrea, Anita Aikio, Lucilla Alfonsi, Evgenia Belova, Stephan Buchert, Mark Clilverd, Norbert Engler, Björn Gustavsson, Craig Heinselman, Johan Kero, et al. The science case for the EISCAT_3D radar. *Progress in Earth and Planetary Science*, 2(1):21, 2015.

SJPSMV17. Juan Félix San-Juan, Iván Pérez, Montserrat San-Martín, and Eliseo P Vergara. Hybrid SGP4 orbit propagator. *Acta Astronautica*, 137:254–260, 2017.

VMK+17. Juha Vierinen, Jussi Markkanen, Holger Krag, Jan Siminski, and Alexandru Mancas. Use of EISCAT 3D for observations of space debris. 2017.