



FINAL REPORT

NEOPOP

Near Earth Object Population
Observation Program

<u>Project title:</u>	Synthetic Generation of a NEO Population
<u>Revision:</u>	2.0
<u>Document ID:</u>	SGNEOP_FR
<u>Date:</u>	September 07, 2022
<u>ESA/ESTEC Study Manager:</u>	Raffaella Franco
<u>Prepared by:</u>	Johannes Gelhaus (ILR) Gerhard Hahn (DLR) Sven Müller (ILR)
<u>Main-Contractor:</u>	Technische Universität Braunschweig (TUBS) Institute of Aerospace Systems (ILR) Hermann-Blenk-Str. 23 38108 Braunschweig D-Germany

The work described in this report was done under ESA contract. Responsibility for the contents resides in the authors or organizations that prepared it.

The copyright of this document is vested in the European Space Agency. This document may only be reproduced in whole or in part, stored in a retrieval system, transmitted in any form, or by any means e.g. electronically, mechanically or by photocopying, or otherwise, with the prior permission of the Agency.

ESA's Near Earth Object Population Observation Program (NEOPOP) was developed by the Institute of Aerospace Systems at the Technische Universität of Braunschweig and the German Aerospace Center (DLR) under ESA contract. DLR was supported by Observatoire de la Côte d'Azur (OCA) and many additional experts. This report was compiled at the Institute of Aerospace Systems with contributions from the study team.

Study Team: **ILR/TUBS**
Johannes Gelhaus
Sven Müller



DLR
Gerhard Hahn
Stephan Hellmich



supported by OCA
Alessandro Morbidelli
Mikael Granvik (Univ. of Helsinki)
Patrick Michel
Robert Jedicke (Univ. of Hawaii)
Bryce Bolin (Univ. of Hawaii)
Bill Bottke (SwRI, Boulder)
David Nesvorny (SwRI, Boulder)
David Vokrouhlicky (Univ. of Prague)
Ed Beshore (LPL, Univ. of Arizona)



History of Changes:

Date	Version	Author	Change Description
23.03.2014	0.1	J. Gelhaus	<ul style="list-style-type: none"> • First draft version of the FR for AR
21.05.2014	1.0	J. Gelhaus	<ul style="list-style-type: none"> • Final version for acceptance
20.06.2014	1.1	J. Gelhaus G. Hahn	<ul style="list-style-type: none"> • Final version for second acceptance
04.12.2014	1.2	S. Müller	<ul style="list-style-type: none"> • Version for third acceptance • Added “Further Improvements” chapter • Named source regions of new NEO model (cp. RID 93) • Added description of generated plots (cp. RID 58) • Documented that Population Generation generates source distributions and explained that Population Analysis can filter by object source, thus "selecting" sources (cp. RID 47) • A little bit of formatting
08.12.2014	1.3	S. Müller	<ul style="list-style-type: none"> • Version for third acceptance • Improved “Further Improvements” chapter • Accepted all tracked changes in Word document
23.03.2015	1.4	S. Müller	<ul style="list-style-type: none"> • Minor adjustments
10.07.2015	1.5	C. Kepschull, S. Müller	<ul style="list-style-type: none"> • Minor adjustments and more precise wording • Addressed RID 105 and 106 referring to section 3.1.3.5: The naming of physical quantities regarding the object signal computation was not correct. Also, $S_{obj} = I_{rec} \cdot \sin i / E_{rec}$ was not correct - Instead of E_{rec}, E_s is now used.
10.07.2022	2.0	V. Raposo Pulido	<p>Correction of typographical errors</p> <p>The equations describing the background signal of stars (3-118) and due to zodiacal light (3-133) for VIS mode are not correct. They have been updated with the reference [14]</p> <p>The equations describing the density of the circumsolar ring (3-137) and the density of each dust band (3-139) are not correct. They have been updated with the reference [21]</p>

25.08.2022	2.0	V. Raposo Pulido	<ul style="list-style-type: none">• The equation describing the background signal of stars (3-129) for TIR mode has been updated to be consistent with the code
------------	-----	---------------------	---

Table of Content

1	Introduction	11
1.1	Scope of the Software	11
1.2	Terms, definitions and abbreviated terms	11
2	Population Generator	12
2.1	Population Generation	12
2.1.1	The “old” model for generating a NEO population	12
2.1.2	The “new” model for generating a NEO population	13
2.1.3	Generating a fictitious NEO population	14
2.1.4	The Physical Properties File	14
2.2	Population Analysis	15
2.2.1	Object propagation	17
2.2.2	Close Approach Analysis & Distance Computation	18
3	Observation Simulator	20
3.1	Observation Simulation	20
3.1.1	Pre-Filtering	21
3.1.1.1	Compute the filter limits	21
3.1.2	Crossing Analysis	23
3.1.2.1	Identification of crossings	23
3.1.2.1.1	Getting the Sensor State	23
3.1.2.1.2	Getting the Object State	25
3.1.2.1.3	Getting the Orientation of the Line of Sight	25
3.1.2.1.4	Make the Crossing Decision	26
3.1.2.1.5	Find the next time step	27
3.1.2.1.6	Results of the “Identification of crossings”	31
3.1.2.2	Computation of crossing geometry	32
3.1.3	Detection Analysis - Optical Performance Model	34
3.1.3.1	Initializing the sensor model	34
3.1.3.1.1	Airglow initialization	35
3.1.3.1.2	Atmospheric Extinction	35
3.1.3.1.3	Instrumental Spectral Response	36
3.1.3.1.4	Starlight spectral distribution for a 0mag star	37
3.1.3.1.5	Sun spectral irradiance data	37

3.1.3.1.6	Extragalactic background light	37
3.1.3.1.7	Star class properties	38
3.1.3.1.8	Interstellar extinction	39
3.1.3.1.9	Patched Stellar Catalogue	39
3.1.3.1.10	Zodiacal light data	39
3.1.3.1.11	Scattered Sunlight data	39
3.1.3.1.12	Scattered moonlight data	39
3.1.3.1.13	Lunar Phase factor	39
3.1.3.2	Frames	39
3.1.3.3	Basic integrals	40
3.1.3.3.1	Raw signal to compute the objects signal	40
3.1.3.3.2	Raw signal of a 0mag star	40
3.1.3.3.3	Raw airglow signal	41
3.1.3.3.4	Raw signal of the planets	41
3.1.3.3.5	Raw signal of galaxies	41
3.1.3.4	Trace and Pixel Dwell Times	41
3.1.3.5	The objects signal	44
3.1.3.5.1	Received objects signal (VIS)	44
3.1.3.5.2	Received objects signal (TIR)	47
3.1.3.6	The background signal	48
3.1.3.6.1	Background signal of Planets	48
3.1.3.6.2	Background signal of stars	49
3.1.3.6.3	Background signal of galaxies	53
3.1.3.6.4	Background signal due to airglow	53
3.1.3.6.5	Background signal due to zodiacal light	54
3.1.3.6.6	Background signal due to the interstellar medium	57
3.1.3.6.7	Background signal of atmospherically scattered light	59
3.1.3.6.8	Background signal due to scattered Sunlight	59
3.1.3.6.9	Background signal due to scattered Moonlight	59
3.1.3.7	Detection decision	60
3.1.4	Detection Analysis - Radar Performance Model	61
3.1.5	Special Functionalities	63
3.1.5.1	Observation Strategy	63

3.1.5.2	Sky Plots	64
3.1.5.3	Generation of Measurements	64
3.2	Observation Analysis	65
4	Common Routines	66
4.1	From Local Horizon into Topocentric Equatorial	66
4.1.1	Ground Based	66
4.1.2	Space Based	66
4.2	From Topocentric Equatorial into Local Horizon	68
4.2.1	Ground Based	68
4.2.2	Space Based	68
4.3	Tropospheric Refraction	70
4.4	Planck's law	70
4.5	Tri-linear interpolation	72
5	Further Improvements	73
6	Help method & suggestions	74
6.1	Feedback and Bug-Report	74
7	References	75

List of Tables

1	Introduction	11
1.1	Scope of the Software	11
1.2	Terms, definitions and abbreviated terms	11
2	Population Generator	12
2.1	Population Generation	12
2.1.1	The "old" model for generating a NEO population	12
2.1.2	The "new" model for generating a NEO population	13
2.1.3	Generating a fictitious NEO population	14
2.1.4	The Physical Properties File	14
2.2	Population Analysis	15
2.2.1	Object propagation	17
2.2.2	Close Approach Analysis & Distance Computation	18
3	Observation Simulator	20

3.1	Observation Simulation	20
3.1.1	Pre-Filtering	21
3.1.1.1	Compute the filter limits	21
3.1.2	Crossing Analysis	23
3.1.2.1	Identification of crossings	23
3.1.2.1.1	Getting the Sensor State	23
3.1.2.1.2	Getting the Object State	25
3.1.2.1.3	Getting the Orientation of the Line of Sight	25
3.1.2.1.4	Make the Crossing Decision	26
3.1.2.1.5	Find the next time step	27
3.1.2.1.6	Results of the “Identification of crossings”	31
3.1.2.2	Computation of crossing geometry	32
3.1.3	Detection Analysis - Optical Performance Model	34
3.1.3.1	Initializing the sensor model	34
3.1.3.1.1	Airglow initialization	35
3.1.3.1.2	Atmospheric Extinction	35
3.1.3.1.3	Instrumental Spectral Response	36
3.1.3.1.4	Starlight spectral distribution for a 0mag star	37
3.1.3.1.5	Sun spectral irradiance data	37
3.1.3.1.6	Extragalactic background light	37
3.1.3.1.7	Star class properties	38
3.1.3.1.8	Interstellar extinction	39
3.1.3.1.9	Patched Stellar Catalogue	39
3.1.3.1.10	Zodiacal light data	39
3.1.3.1.11	Scattered Sunlight data	39
3.1.3.1.12	Scattered moonlight data	39
3.1.3.1.13	Lunar Phase factor	39
3.1.3.2	Frames	39
3.1.3.3	Basic integrals	40
3.1.3.3.1	Raw signal to compute the objects signal	40
3.1.3.3.2	Raw signal of a 0mag star	40
3.1.3.3.3	Raw airglow signal	41
3.1.3.3.4	Raw signal of the planets	41

3.1.3.3.5	Raw signal of galaxies	41
3.1.3.4	Trace and Pixel Dwell Times	41
3.1.3.5	The objects signal	44
3.1.3.5.1	Received objects signal (VIS)	44
3.1.3.5.2	Received objects signal (TIR)	47
3.1.3.6	The background signal	48
3.1.3.6.1	Background signal of Planets	48
3.1.3.6.2	Background signal of stars	49
3.1.3.6.3	Background signal of galaxies	53
3.1.3.6.4	Background signal due to airglow	53
3.1.3.6.5	Background signal due to zodiacal light	54
3.1.3.6.6	Background signal due to the interstellar medium	57
3.1.3.6.7	Background signal of atmospherically scattered light	59
3.1.3.6.8	Background signal due to scattered Sunlight	59
3.1.3.6.9	Background signal due to scattered Moonlight	59
3.1.3.7	Detection decision	60
3.1.4	Detection Analysis - Radar Performance Model	61
3.1.5	Special Functionalities	63
3.1.5.1	Observation Strategy	63
3.1.5.2	Sky Plots	64
3.1.5.3	Generation of Measurements	64
3.2	Observation Analysis	65
4	Common Routines	66
4.1	From Local Horizon into Topocentric Equatorial	66
4.1.1	Ground Based	66
4.1.2	Space Based	66
4.2	From Topocentric Equatorial into Local Horizon	68
4.2.1	Ground Based	68
4.2.2	Space Based	68
4.3	Tropospheric Refraction	70
4.4	Planck's law	70
4.5	Tri-linear interpolation	72
5	Further Improvements	73

6	Help method & suggestions	74
6.1	Feedback and Bug-Report	74
7	References	75

1 Introduction

1.1 Scope of the Software

The scope of this document is the description of the technical content of the NEOPOP software tool, the “Near Earth Object Population Observation Program”. The focus will be on giving a general overview of the main algorithms used for the software. For further detailed background information on the algorithms and methods, which are used by the software, the user shall be referred to the individual references given in the particular context.

1.2 Terms, definitions and abbreviated terms

CLI	Command line interface
CSS	Catalina Sky Survey
DLR	German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt)
ESA	European Space Agency
GUI	Graphical User Interface
ILR	Institute of Aerospace Systems (Institut für Luft und Raumfahrtssysteme)
MASTER	Meteoroid and Space Debris Terrestrial Environment Reference
NEO	Near Earth Object
NEOPOP	NEO Population Observation Program
OCA	Observatoire de la Côte d’Azur
PROOF	Program for Radar and Optical Observation Forecasting
RID	Review Item Description
SSA	Space Situational Awareness
TBD	To Be Defined / To Be Discussed
TUBS	Technische Universität Braunschweig
WP/WPD	Work Package / Work Package Description

2 Population Generator

The first component of NEOPOP is the “*Population Generator*”. This component includes the two modules “*Population Generation*” and “*Population Analysis*”. While the first one can be used to generate a NEO population in different ways, the second one gives the capability to display parameters distributions of the whole or a subset of a given population.

2.1 Population Generation

2.1.1 The “old” model for generating a NEO population

The old model, also called Bottke Model [27], is based on observations prior to 2000, and is calibrated on a rather small sample of about 120 NEOs discovered by the Spacewatch Survey. It should be used only for comparison of the methods, their improvements, and for discussion about technical details. The seven source regions used in the old model are:

- 3:1 mean-motion with Jupiter - '31all.res'
- Intermediate Mars crosser - 'imc_update_exttarget.res'
- Jupiter Family comets - 'comet_jfc_only.res'
- ν_6 secular resonance - 'nu6all.res'
- Hungaria region - 'hung.res'
- Phocaea region - 'pho.res'
- Outer Belt region - 'ob_superext.res'

These files are used by the software to generate a certain number of objects out of the given source regions. This is in line with the last version of the old model.

The old model was calibrated to $H=22$. When selecting a larger value for H the predictions might be not accurate for the range above $H=22$. The model simply uses the same approach to generate objects beyond $H=22$ as it is used for the validated range. Thus, it is highly recommended to use this outdated model only for comparisons, to re-generated results, which are based on this model and have been published in the past!

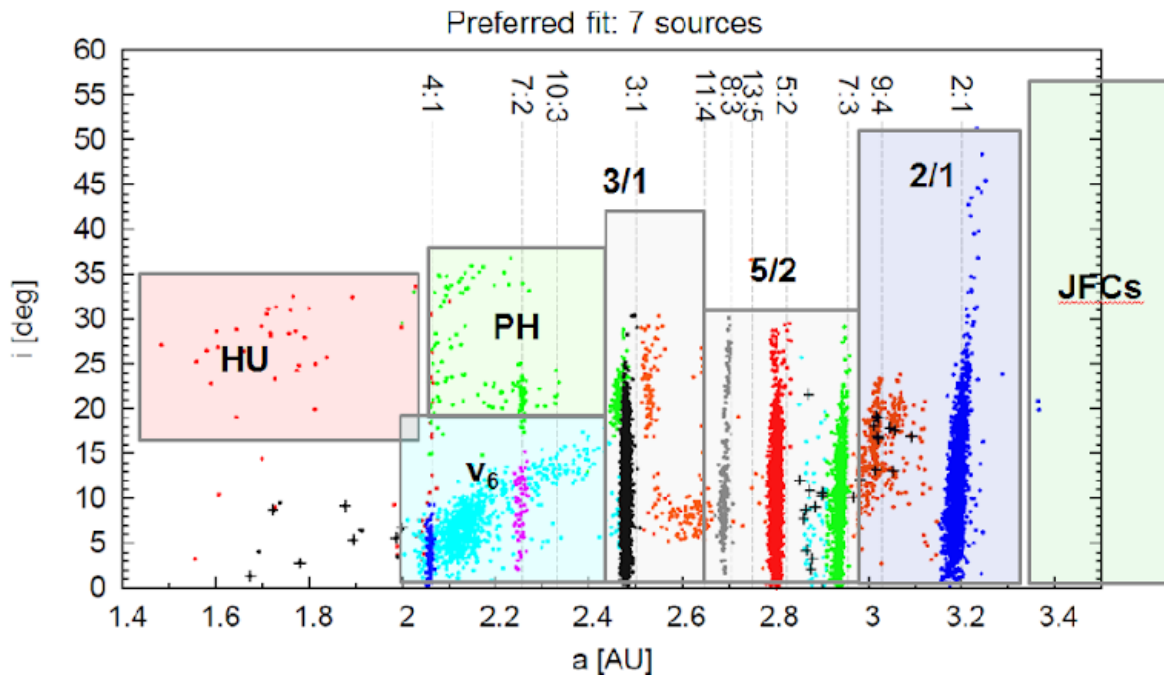


Figure 1: The seven source regions of the new NEO model: Hungaria, Phocea, v_6 , Jupiter Family Comets and Jupiter 3:1, 5:2 and 2:1 resonances

2.1.2 The “new” model for generating a NEO population

The new model is based on the Catalina Sky Survey observations in the years 2006-2011, and comprises more than 4000 NEO discoveries or redetections. The details of the procedures to arrive at this model are described in a separate Technical Note. It is calibrated from an absolute magnitude of $H=15$ up to $H=25$, and allows extrapolation to fainter H using two different parameter sets. One based on bolide observations by satellites, and one by user defined slope parameters and a break point in H . This model also includes an albedo model, which is based on observations taken from the WISE satellite data. It allows predictions of the total number of NEOs up to a chosen H value, and gives numbers for the four subgroups (Atira, Aten, Apollo and Amor) and the seven different source regions (cp. Figure 1):

- Hungaria
- Phocea
- v_6
- Jupiter Family Comets
- Jupiter 3:1 resonance
- Jupiter 5:2 resonance
- Jupiter 2:1 resonance

The generated populations can be obtained in three different output formats: those used by the NEODyS/AstDyS group, the orbit format used by the IAU Minor Planet Center in their MPCORB database, and finally the output in the DES format, used by the Panstarrs group. There is an upper limit of 999999 orbits that can maximally be generated. If the model predicts

more objects, the user can give a scaling factor, which will be taken into account in the generation process.

The new model is calibrated between $15 < H < 25$ magnitudes. At the bright end, $H < 15$, the population is supposed to be complete. A file “*known_NEOs_up_to_H15.dat*” contains the orbital elements a, e, i and H of all known NEOs. This file is based on data from the MPC (<http://www.minorplanetcenter.net/iau/ECS/MPCAT/current/unusual.txt>) and considers all known and listed NEOs below $H=15$. A similar file based on the data from NEODyS (<http://newton.dm.unipi.it/~neodys2/neodys.cat>) could contain a slightly different number of known NEOs since the H values are determined independently by the NEODyS group). In addition, the actual number of NEOs with H brighter than 15.0 might change when new objects are discovered. In such a case the user can simply change the content of the file.

If the limiting absolute magnitude H_{lim} exceeds the calibrated range of the model ($H_{lim} > 25$) an extrapolation is made

$$pobj = totp * 10^{(\alpha_{extr}(H_{lim}-25))} - totp$$

Where α_{extr} has either a fixed value of 0.6434, which comes from satellite observations of bolides [31], or it can be user defined through the input file. In this latter case it was chosen to give a broken power law with two coefficients and a break point in H , where the slope is changing. Then the evaluation of the total number of objects is done in two steps

$$pobj = totp * 10^{(\alpha_{extr1}(H_{lim}-H_{break}))} - totp$$

$$pobj = totp * 10^{(\alpha_{extr2}(H_{break}-25))} - totp$$

2.1.3 Generating a fictitious NEO population

There is an option to generate a fictitious population of NEO orbits, selectable by the four groups mentioned above. In addition a sample of potentially hazardous objects (PHOs), according to the IAU criterion $MOID < 0.05 AU$, and $H = 22$. For these latter, an additional selection criterion can be set by a minimum approach distance to the Earth within the coming 10 years. The output formats for the population are again NEODyS, MPC and DES.

The generation of fictitious PHOs has as selection criterion the $MOID < 0.05 AU$, for the chosen orbit an 2-body ephemeris is calculated to check whether there will be close approaches to the Earth within the user-selected distance.

2.1.4 The Physical Properties File

For the use by the analysis component of this software package, a file containing some physical parameters for the generated orbits is generated: a Physical Properties File (PPF). These are the diameter and its error range, based on the H value and an albedo. The albedo is selected by the model, where there are three categories (p between 0 and 0.1, between 0.1 and 0.3, and above 0.3). These albedos together with an assumed error of $p/3$ are written to this file. In addition, the MOID and the intrinsic collision probability according to Öpik’s method are given.

It is also possible to read in external populations, and generate a corresponding PPF. The formats available are the NEODyS/AstDyS format [29], the MPCorb format [28] of the MPC, and the DES format [30] developed and used by the Panstarrs survey.

2.2 Population Analysis

The module “Population Analysis” can be used to get access to the data contained in the population files, which for example have been generated with the module “Population Generation”. The main task of the analysis module is to display the parameters contained in the whole population or a subset thereof. There are many options available to filter the given population which can be categorized as follows:

- Filtering by object group
- Filtering by object source
- Filtering by given parameter (e. g. set limits of semi-major axis: $0.5 < a < 1.0$)
- Filtering by computing the objects’ distance to Earth or Sun for a defined epoch
- Filtering by close approach analysis for a defined time period and maximum distance.

In all cases an object is skipped from being considered for the analysis if at least one of the filter criteria is violated. This means that for example an object having a semi-major axis of $a = 3.2$ is not considered for any further filter methods if the semi-major axis is limited to $0.5 < a < 1.0$. The fast filter steps (checking boundaries of given parameters) are conducted first, before the remaining subset of objects is given to the more time consuming filter steps (e.g. close approach analysis) as shown in the flow chart of this module in Figure 2 .

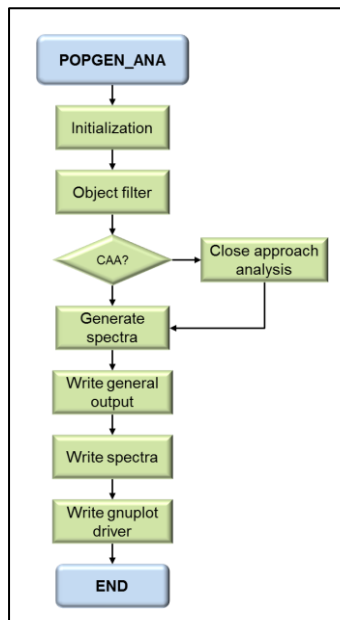


Figure 2: Flow chart of the Population Analysis (main routine)

After all filter criteria have been skipped, those objects not to be considered the histogram files are prepared and written. Those histogram files (or spectra) are the basis for the plots,

which are finally generated by the GUI or manually by the user, when running the software via the command line interface. The user can setup the plots to be generated by an upper and lower limit and by a logarithmic or linear scaling of the axis. For spectra also the number of steps within the defined limits or the class within those limits can be defined. The upper limit is automatically adjusted if the range between the defined limits is not an even multiple of the class width. Once the spectra have been generated and written the gnuplot driver files are created. All plots can easily be generated by loading a single file when executing gnuplot.

NOTE: The defined limits are only used for the histogram and/or plot generation. This means that the total number of objects within the plots is not necessarily the same in each plot. Also not all objects remaining after the filter steps are automatically contributing to each plot. Objects might be ignored, because their parameters are outside the plot limitations.

NOTE: If the number of classes is defined for a histogram the class width is computed internally. Also if the class width is given the number of classes is computed internally. If the range between the defined min. and max. limits is not an even multiple of the defined class width, the upper limit is adjusted.

The different plots for which NEOPOP creates gnuplot files can be viewed in : 2D, 3D, scatter and solar system overview plots can be created. 2D plots exist in differential, cumulative and reverse-cumulative variants, solar system plots in polar and side view variants. The solar system plots show the NEOs of an analyzed population as green dots in the solar system with Jupiter's orbit as boundary. They come in polar and side view variants.

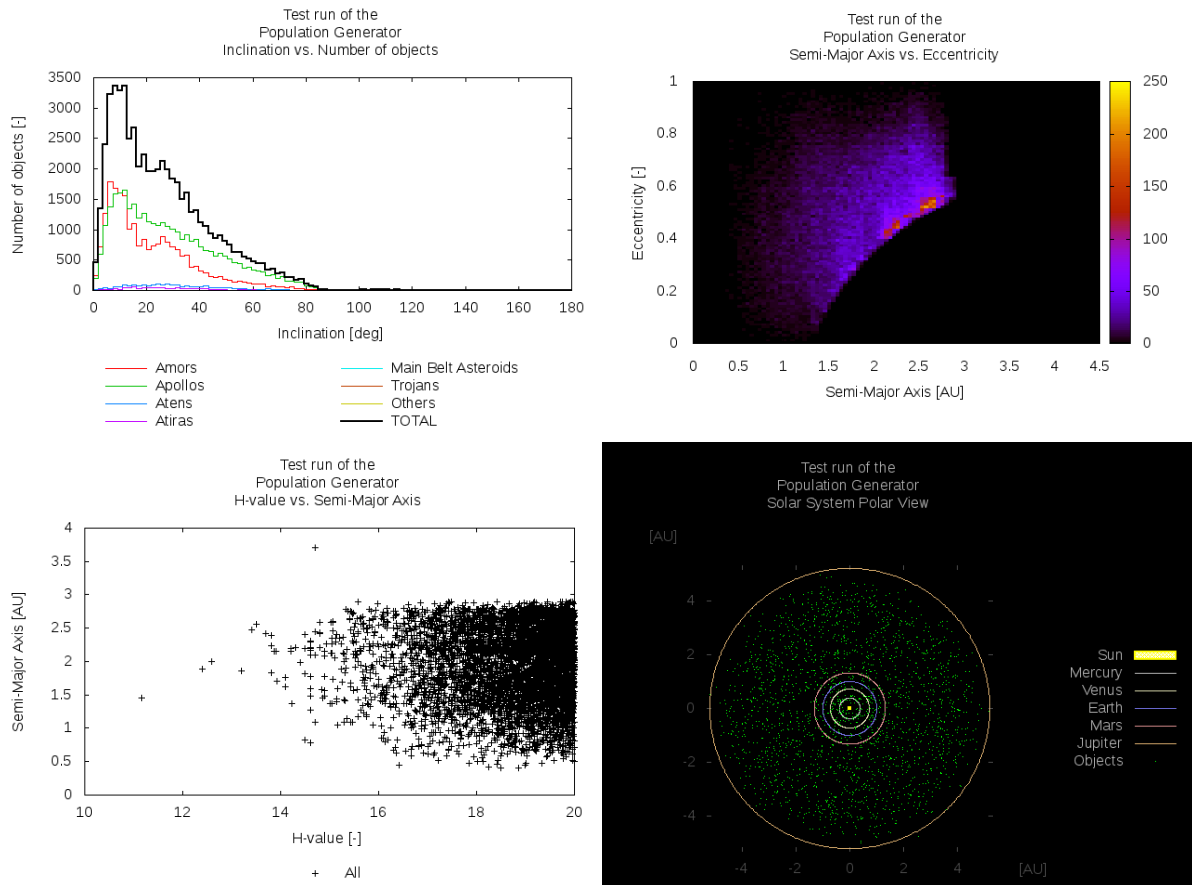


Figure 3: Sample plots of Population Analysis module. From top-left to bottom-right: 2D, 3D, scatter and solar system plot.

2.2.1 Object propagation

Within the filtering there are two steps that require the propagation of the objects orbit:

- Analysis data at a given “*Analysis Epoch*”
- Performing a “*Close Approach Analysis*”

For both options the methods given in the SPICE software library are used [02]. Out of this package the routines *CONICS* and *OSCELT* are used to compute the objects orbital elements and/or the object state for a given epoch. Both routines make use of a two body propagator called *PROP2B* also included in the SPICE library. All other perturbations are not considered (e.g. planets, asymmetric gravitational fields, etc.).

NOTE: All objects are propagated to the analysis epoch (if activated) during the initialization before the filtering starts. Thus, object parameters are related to the analysis epoch when entering the filter steps.

On the website <http://naif.jpl.nasa.gov/naif/documentation.html> (access: March 08th, 2014) a detailed documentation of the SPICE library and even the source code in different programming languages can be found.

2.2.2 Close Approach Analysis & Distance Computation

For the close approach analysis and the distance computation relative to the Earth also its position is required. Therefore also the SPICE library (see [02] and [03]) is used. It provides a routine called *SPKEZR* that allows an easy computation of planetary data. It returns the state (position and velocity) of a target body relative to an observing body, optionally corrected for light time (planetary aberration) and stellar aberration. The usage of this routine requires that a data file (called SPICE kernel) is loaded at program start. The data file can be changed by the user of the *population analysis* by a manipulation of the configuration file (default name: *popgen.cfg*). These SPICE kernels can be found at the following website: http://naif.jpl.nasa.gov/pub/naif/generic_kernels/ [04].

To increase the computation time there is a constant time step of 30 days used for the close approach analysis. For objects coming closer to the Earth the time step is steadily reduced with a decreasing distance between the object and Earth. This time step is computed based on the following equation:

$$dt = \left[\frac{\sin\left(\frac{r}{r_{thresh}} * \pi\right) - \frac{\pi}{2}}{2} + 1 \right] \cdot (dt_{max} - dt_{min}) + dt_{min} \quad \text{Eq. 2-1}$$

The value r_{thresh} is a range threshold. Objects having a distance r beyond this range are propagated with the constant time step. The values dt_{max} and dt_{min} are the maximum and the minimum time step. The minimum time step has been set to one hour and the range threshold used internally is 2AU.

NOTE: There is no iterative process going on to find the exact min. distance of a close approach. This effect is relevant only but especially for very close encounters, where the relative velocities are the largest.

An exemplary curve progression is given in Figure 4 for a range threshold of 3AU and a variation of the min. and max. time step. The closer an object comes to Earth the smaller the time step will be to compute the next distance for.

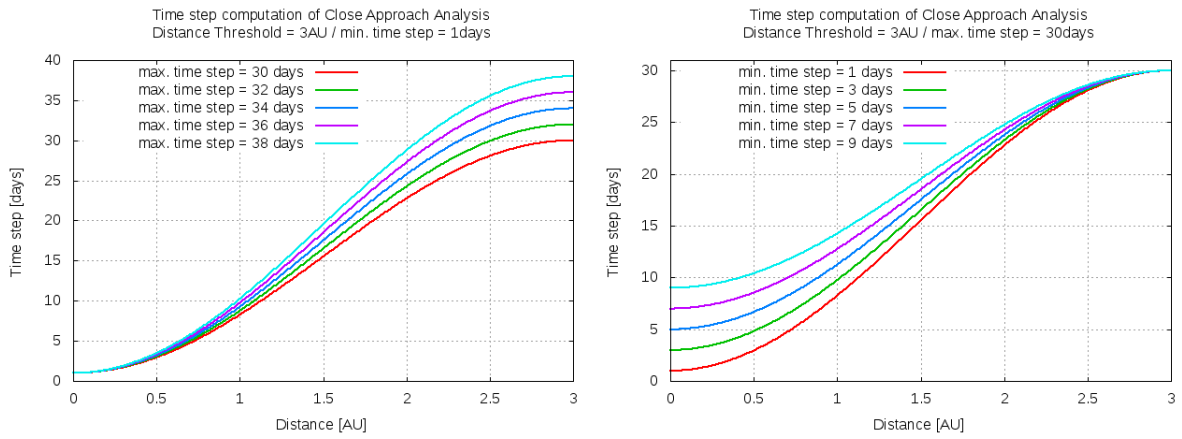


Figure 4: Exemplary curve progression for time step computation with a variation of the max. time steps (left) and a variation of the min. time step (right)

3 Observation Simulator

The “*Observation Simulator*” is the second component of the NEOPOP software tool. The tasks of this component are covered by two modules. The first module, the “Observation Simulation”, can be used to simulate a sensor observing Near Earth Objects. The second module may be used to analyze the results of the simulation and generate the appropriate histogram tables and plots.

3.1 Observation Simulation

The module “Observation Simulation” is the most complex part of the NEOPOP software tool. Its main task is to simulate the behavior of real sensor systems. This is done especially for optical sensors, which are typically used for the search of asteroids and near Earth objects as done for the Catalina Sky Survey [05]. The NEOPOP software shall support the planning of observation strategies and how to build up a sensor network. Due to the fact that this task is done by optical sensors the included radar model is designed in a simple way not covering imaging, etc. as done in reality.

The upper level flow chart of the observation simulation is shown in Figure 5. The first actions are related to the initialization. Out of databases for sensor locations, orbits and types the data relevant for the defined network is assembled. Several instances of the data are used providing on the one hand the real data as given in the input files (modules: *M_raw*) and the data to be used internally for the computation (modules: *M_use*). This step-wise setup process is required to have the initial input data available as the fall back case for the loops over Monte Carlo runs, the defined sensor systems and system specific observation strategies.

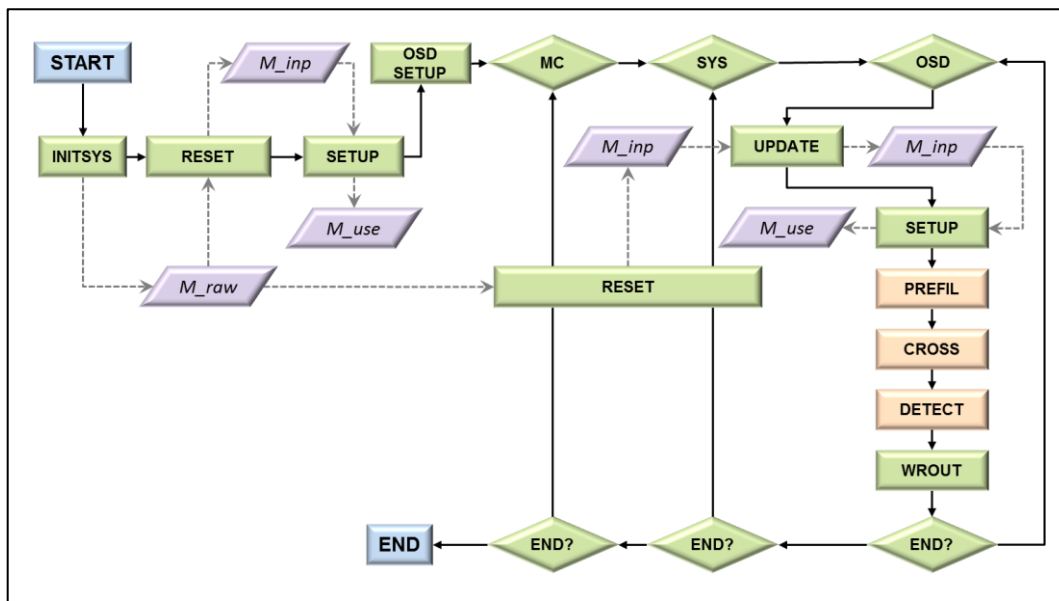


Figure 5: Top level flow chart of the module "Observation Simulation"

The population is read during the initialization and propagated to the observation start after entering the loop over the Monte Carlo runs. This loop is executed once when being in the

deterministic mode. For the statistical analysis, which makes only sense for more than one Monte Carlo run, the population is slightly modified instead of propagated for each of the following Monte Carlo runs.

There are three general steps performed for each sensor system:

- Pre-Filtering
- Crossing Analysis:
 - Identification of crossings
 - Computation of crossing geometry
- Detection Analysis
 - Optical Performance Model
 - Radar Performance Model

These steps will be explained in the following sections.

3.1.1 Pre-Filtering

The task of the Pre-Filtering is to identify those objects that will never be able to enter the field of view of the sensor. The filtering is based on simple geometric constraints that will exclude objects from being able to enter the sensors field of view. For a relatively small number of steps in time the sensors location and its orientation is computed. The line of sight and some values relative to the Sun are computed. In the two dimensional plane of Figure 6 the volume covered by the field of view during the observation is represented by the green area. For each step in time the following parameters are computed:

- r_{min}, r_{max} : Minimum and maximum distance between the points of minimum or maximum range of the line of sight
- $\alpha_{min}, \alpha_{max}$: Minimum and maximum ecliptic longitude related to the point of minimum or maximum range of the line of sight
- δ_{min} : Minimum ecliptic latitude to the point of minimum or maximum range of the line of sight.

3.1.1.1 Compute the filter limits

The time step computation is related to the sensor location or its orbit if the sensor is not ground based. Also the observation duration has an impact on the time steps used within the pre-filter. A detailed description of the time steps used in the pre-filtering is given in Table 1. Out of the values computed for each time step finally the global minimum and maximum values are used to filter the initial population. This is done by the following criteria:

- The minimum range between Sun and the observed volume is used to skip objects having an aphelion below this range.

$$a \cdot (1 + \varepsilon) \leq r_{min} \quad \text{Eq. 3-1}$$

- The maximum range between Sun and the observed volume is used to skip objects having a perihelion above the maximum range.

$$a \cdot (1 - \varepsilon) \leq r_{max} \quad \text{Eq. 3-2}$$

- The minimum ecliptic latitude is used to skip objects having an inclination below this minimum ecliptic latitude.

$$i \leq \delta_{min} \quad \text{Eq. 3-3}$$

- Finally the shortest distance to the minimum or maximum ecliptic longitude is used to skip objects being for example on the other side of the Sun. As a conservative approach the angular velocity in the heliocentric system is computed assuming a circular orbit with a radius of the objects perihelion distance.

$$\Delta t \cdot \sqrt{\frac{\mu_H}{(a \cdot (1 - \varepsilon))^3}} \leq \text{MIN}(\Delta\alpha_{min}, \Delta\alpha_{max}) \quad \text{Eq. 3-4}$$

This filter method is only applied for relative short observation durations. The cut-off durations depend on the sensor placement and its orientation. For example this filter is applied for ground based sensors oriented in the local horizon frame (Azimuth & Elevation) only if the observation duration is less than 6 hours, while for an orientation in one of the inertial reference frames the observation duration can be up to 90 days. These different cut-off durations are linked to the potential range of ecliptic longitude covered during the observation time. The faster a sensor system is in scanning a certain angular range the shorter is the cut-off duration.

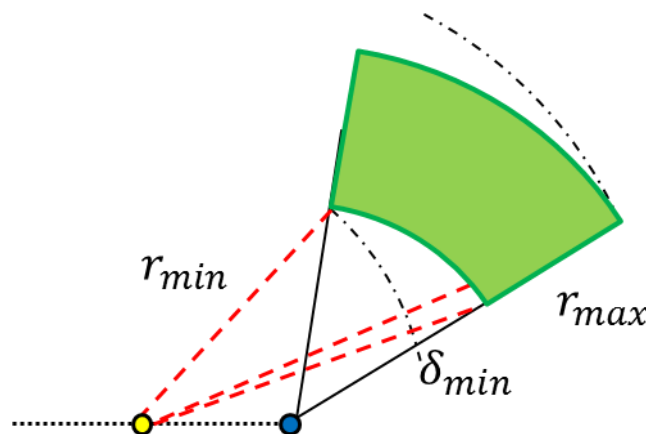


Figure 6: "Volume" covered by the sensors field of view during the observation

Sensor Placement	Observation Duration	Time Step Computation
Ground based	$0 < \Delta t \leq 3 \text{ Days}$	One step every hour
	$3 \text{ Days} < \Delta t \leq 2 \text{ Month}$	For start, mid and end of observation 25 steps during 24 hours
	$2 \text{ Month} < \Delta t < 10 \text{ Years}$	In steps of one month and at the end of observation 25 steps during 24 hours
Space Based (Geocentric)	$0 < \Delta t \leq 3 \text{ Days}$	One step every $T_U/24$
	$3 \text{ Days} < \Delta t \leq 2 \text{ Month}$	For start, mid and end of observation 25 steps during one revolution
	$2 \text{ Month} < \Delta t < 10 \text{ Years}$	In steps of one month and at the end of observation 25 steps per revolution
Space Based (Heliocentric)	$0 < \Delta t < 10 \text{ Years}$	One step every 24 hours

Table 1: Time step computation within the pre-filter

3.1.2 Crossing Analysis

The objects made available to the crossing analysis will be investigated, whether they are crossing the field of view or not. The goal is to identify objects crossing the field of view and providing a crossing geometry to analyze in the following (see chapter 3.1.3), whether the crossing object can be detected or not.

The crossing analysis itself is splitted into two steps. The first one is identifying if an object enters the field of view (see chapter 3.1.2.1) while the crossing geometry is computed in a second step (see chapter 3.1.2.2).

3.1.2.1 Identification of crossings

For all objects the identification of a potential crossing starts at the observation start. Internally the time is handled in ephemeris seconds after J2000 (ET). Thus, the observation start, which is defined by date and time in UTC, was initially converted into seconds (ET_0) using the time transformation routines *TTRANS* and *UTC2ET* out of the SPICE library. The time steps Δt between two steps in the crossing analysis are computed depending on the objects position relative to the line of sight respectively to its offset from the field of view. For the first step $\Delta t = 0$ and the crossing analysis is made for the starting time ET_0 .

For each step in time the state of the sensor and the object as well as the orientation of the line of sight have to be determined. Based on this the crossing decision can be made.

3.1.2.1.1 Getting the Sensor State

For **ground based** sensors it is a simple task to compute the state vector $geo2sen_{J2000}(ET)$ related to the current epoch ET . This state gives the position and velocity of the sensor in an

Earth centered reference frame. Also here the SPICE library can be used to compute the sensor state from the initially Earth fixed and time independent state vector sen_{IAU_Earth} .

$$geo2sen_{J2000}(ET) = M(ET) \cdot geo2sen_{IAU_Earth} \quad \text{Eq. 3-5}$$

The conversion of Eq. 3-5 is made by the routine *MXVG* with the help of the time dependent transformation matrix $M(ET)$ gathered from the routine *SXFORM*.

For a sensor in a **space based geocentric orbit** the orbital elements are converted into a state vector with the help of the routine *CONICS*. The orbit propagation is based on simple perturbation theory considering the movement of the line of nodes as well as the line of apsides.

$$a(ET) = a_0 \quad \text{Eq. 3-6}$$

$$\varepsilon(ET) = \varepsilon_0 \quad \text{Eq. 3-7}$$

$$i(ET) = i_0 \quad \text{Eq. 3-8}$$

$$\Omega(ET) = \Omega_0 - 9.96^\circ \cdot \left(\frac{R_E}{a}\right)^{3.5} \cdot \frac{\cos(i)}{(1 - \varepsilon^2)^2} \cdot (ET - ET_0) \quad \text{Eq. 3-9}$$

$$\omega(ET) = \omega_0 + 4.98^\circ \cdot \left(\frac{R_E}{a}\right)^{3.5} \cdot \frac{5 \cdot (i) - 1}{(1 - \varepsilon^2)^2} \cdot (ET - ET_0) \quad \text{Eq. 3-10}$$

$$M(ET) = M_0 + \frac{2\pi}{T_U} \cdot (ET - ET_0) \quad \text{Eq. 3-11}$$

For both ground based and geocentric space based sensors the Earth centered state vector in the *J2000* reference frame is now. For the later crossing decision the heliocentric state is of interest, which can be obtained by adding the state vector of the Earth $sun2geo_{J2000}(ET)$ and the sensor state $geo2sen_{J2000}(ET)$. The Earth state can be computed using appropriate SPICE kernels in combination with the routine *SPKEZR*.

$$sun2sen_{J2000}(ET) = sun2geo_{J2000}(ET) + geo2sen_{J2000}(ET) \quad \text{Eq. 3-12}$$

For a sensor in a **space based heliocentric orbit** the state $sun2sen_{ECLIPJ2000}(ET)$ from Sun to the sensor can directly be obtained by propagating the initial sensor state with the SPICE routine *PROP2B*. The orbital elements related to this epoch are converted out of the state vector with the help of the routine *OSCELT*.

Finally the state $sun2sen_{J2000}(ET)$ pointing from the Sun to the sensor, its equivalent in the ecliptic reference frame $sun2sen_{ECLIPJ2000}(ET)$ and also the orbital elements for space based sensors are stored for the later crossing decision.

3.1.2.1.2 Getting the Object State

Also the objects state vector has to be computed for the current epoch ET . This is done in the same way as also space based heliocentric sensors are handled. First the initial object state related to the epoch ET_0 is used to compute the state $sun2obj_{ECLIPJ2000}(ET)$ from Sun to the object. Therefore the two body propagator $PROP2B$ of the SPICE library is used. Again $OSCELT$ is used to obtain the orbital elements and the transformation from the $ECLIPJ2000$ into the $J2000$ reference frame is made with $MXVG$ and the appropriate transformation matrix $M(ET)$ obtained from $SXFORM$:

$$sun2obj_{J2000}(ET) = M(ET) \cdot sun2obj_{ECLIPJ2000}(ET) \quad \text{Eq. 3-13}$$

3.1.2.1.3 Getting the Orientation of the Line of Sight

The third and last information that has to be computed before a crossing decision can be made is a vector along the line of sight. To compute this vector in the reference frame $J2000$ the right ascension and the declination of the line of sight have to be known. These are given already if the sensor is oriented in the **topocentric equatorial reference frame**. For an orientation in the **topocentric ecliptical reference frame**, where the line of sight is defined by a set of angles in ecliptic longitude and ecliptic latitude, a vector along the line of sight has to be computed with the help of the SPICE routine $LATREC$ first and is then converted from the $ECLIPJ2000$ into the $J2000$ reference frame, so that right ascension α and declination δ can be computed by using $RECLAT$.

$$LATREC(ecl. long., ecl. lat.) \Rightarrow los_{ECLIPJ2000} \quad \text{Eq. 3-14}$$

$$los_{J2000}(ET) = M(ET) \cdot los_{ECLIPJ2000}(ET) \quad \text{Eq. 3-15}$$

$$RECLAT(los_{J2000}(ET)) \Rightarrow \alpha(ET), \delta(ET) \quad \text{Eq. 3-16}$$

For both orientations right ascension α and declination δ can be converted into the local horizon reference frame so that also azimuth A and elevation h are known. See chapter 4.2 for the details on the conversion from the topocentric equatorial into the local horizon reference frame.

If the line of sight is oriented in the local horizon by a given azimuth A and elevation h the algorithms explained in chapter 4.1 can be used to obtain the related right ascension α and declination δ .

For sensors not being oriented in the topocentric ecliptical reference frame the vector along the line of sight los_{J2000} has finally to be computed.

3.1.2.1.4 Make the Crossing Decision

At the beginning of the crossing the state $sen2obj_{J2000}$ from the sensor to the object is being computed. Therefore the states $sun2sen_{J2000}$ (see section 3.1.2.1.1) and $sun2obj_{J2000}$ (see section 3.1.2.1.2) are combined:

$$sen2obj_{J2000} = sun2obj_{J2000} - sun2sen_{J2000} \quad \text{Eq. 3-17}$$

NOTE: If selected the tropospheric refraction is considered in the deterministic mode by a method described in [07] and shown in chapter 0 after the $sen2obj_{J2000}$ has been computed. A slightly modified state vector is returned.

For the position vector $\vec{r}_{sen2obj}$ of the state $sen2obj_{J2000}$ the offset angle relative to the line of sight can be computed. Therefore the SPICE function $VSEP$ is used:

$$\gamma_{offset} = VSEP(\vec{r}_{los,J2000}, \vec{r}_{sen2obj}) \quad \text{Eq. 3-18}$$

The state $sen2obj_{J2000}$ from the sensor to the object is then being transformed into the so-called “cone”-system. This system is fixed to the line of sight of a sensor and centered into the sensor itself. Two times the state $sen2obj_{J2000}$ is being rotated before the state $sen2obj_{CONE}$ is built .

The first rotation is made around the z -axis of the inertial topocentric equatorial reference frame. The rotation angle is the right ascension α_{los} in which the sensor is oriented. The second rotation rotates the temporary state vector of the first rotation around the rotated y -axis. The rotation angle is the declination δ_{los} and the sense of rotation is negative.

Thus the former x -axis is oriented along the line of sight, the former y -axis is in the topocentric equatorial plane and the former z -axis is still perpendicular to both the y - and the x -axis. After the state $sen2obj_{J2000}$ has been transformed into the “cone”-system it is a short way to decide whether the object is within the field of view or outside. In a first step the x -component is checked. A negative value indicates that the object is behind the sensor and it cannot be inside the field of view. For positive x -values it is checked whether the offset angle γ_{offset} is lower than half the field of view, which would be the case if the object is inside the field of view:

$$\gamma_{offset} \leq 0.5 \cdot FOV \quad \text{Eq. 3-19}$$

Besides this check it is also check, whether the objects range r_{obj} is within the given range limitations. The range limitation is something that can be used to select or exclude objects from a given population. Thus the object is considered to be within the field of view only if also this criterion is fulfilled:

$$r_{min} < r_{obj} < r_{max} \quad \text{Eq. 3-20}$$

3.1.2.1.5 Find the next time step

After each step in the crossing analysis the next time step has to be computed for which the next crossing analysis shall be made. The time step computation has to make sure on the one hand that the time step is large if possible (e.g. because the object is on the opposite side of the sensor) or that the time step is short if necessary (e.g. if the object comes close to the field of view). The time step computation is made individually depending on the sensor placement and the reference frame selected for defining the orientation of the line of sight.

Ground based sensor: LOS defined in the Local Horizon

The line of sight of a sensor that is ground based and oriented in the local horizon reference frame is moving relatively fast through the solar system, while searching for potentially crossing objects. For large observation durations the line of sight forms a torus around the Earth rotation axis (ERA). The offset angle relative to this torus is used to compute the time step for this combination of sensor placement and orientation. The closer the object comes to the torus the smaller the computed time step will be. For large offset angles γ_{offset} relative to the line of sight the following steps are performed to compute the next time step. Therefore the offset must be larger than $0.55 \cdot FOV$:

The opening angle of the torus is computed with the help of the declination δ_{los} :

$$\gamma_{era,torus} = 90^\circ - \delta_{los} \quad \text{Eq. 3-21}$$

To determine the offset angle $\gamma_{era,obj}$ between the objects position and the Earth rotation axis the position vector $\vec{r}_{geo2obj}$ is computed with the help of object and Earth position vectors, which have been determined before (see 3.1.2.1.1 and 3.1.2.1.2). The Earth rotation axis \vec{r}_{era} is defined as z-axis of the J2000 reference frame:

$$\vec{r}_{geo2obj} = \vec{r}_{sun2obj} - \vec{r}_{sun2geo} \quad \text{Eq. 3-22}$$

$$\gamma_{era,obj} = VSEP(\vec{r}_{geo2obj}, \vec{r}_{era}) \quad \text{Eq. 3-23}$$

The offset between object and torus $\gamma_{torus,obj}$ is the absolute difference between the two offset angles $\gamma_{era,torus}$ and $\gamma_{era,obj}$ and is a threshold value for a large or short increase of time:

$$\gamma_{torus,obj} = ABS(\gamma_{era,torus} - \gamma_{era,obj}) \quad \text{Eq. 3-24}$$

If the offset angle is larger than half the field of view ($\gamma_{torus,obj} > 0.5 \cdot FOV$) the time step Δt is defined between 1 and 90 days:

$$\Delta t = \text{MAX} \left(1, 45 \cdot \left(\sin \sin \left((\gamma_{\text{torus,obj}} - 0.5 \cdot \text{FOV}) - \frac{\pi}{2} \right) + 1 \right) \right) \text{days} \quad \text{Eq. 3-25}$$

Else if the object is already closer to the torus than half the field of view ($\gamma_{\text{torus,obj}} \leq 0.5 \cdot \text{FOV}$) the time step is defined between 1 and 240 minutes:

$$\Delta t = \text{MAX} \left(1, 240 \cdot \left(\sin \sin \left(\gamma_{\text{offset}} - \frac{\pi}{2} \right) + 1 \right) \right) \text{minutes} \quad \text{Eq. 3-26}$$

If the object is already very close to the line of sight ($\gamma_{\text{offset}} < 0.55 \cdot \text{FOV}$) the time steps are computed considering also the range limitations:

The distance Δr between the field of view and the object can be approximated to

$$\Delta r = r_{\text{min}} - r_{\text{obj}} \quad \text{Eq. 3-27}$$

for an object below the minimum range and to

$$\Delta r = r_{\text{obj}} - r_{\text{max}} \quad \text{Eq. 3-28}$$

for an object beyond the maximum range. For the case where $r_{\text{min}} < r_{\text{obj}} < r_{\text{max}}$ the time step is set to $\Delta t = 60\text{s}$. This would be the case if an object is besides the field of view within the thin region considered because the threshold for the shorter time steps starts at $0.55 \cdot \text{FOV}$ already.

For the other cases the time until the object enters the field of view is approximated by the x -component of the velocity in the “cone”-system:

$$dt_{\text{inFOV}} = \Delta r / \dot{x}_{\text{obj, CONE}} \quad \text{Eq. 3-29}$$

and only if

$$dt_{\text{inFOV}} < \left(\frac{\text{FOV}}{360^\circ} \right) \cdot 86400 \text{ seconds} \quad \text{Eq. 3-30}$$

the time step is set to

$$\Delta t = 0.5 \cdot dt_{\text{inFOV}} \quad \text{Eq. 3-31}$$

else a large step is performed by

$$\Delta t = 0.95 \text{ days} \quad \text{Eq. 3-32}$$

This large time step of 22.8 hours is also performed after the object has been found within the “cone”.

Ground based sensor: LOS defined in an inertial fixed reference frame

A much simpler computation of the time step Δt is made for ground based sensor being oriented in an inertial reference frame. In those cases the offset between the line of sight and the object changes relatively slow. Only the motion of Earth on its orbit around the Sun and the objects movement are impacting the change of the offset angle γ_{offset} .

Thus there are only two different approaches to determine the next time step. Again a *sin*-based function is used to decrease the time step with decreasing offset angle:

$$\Delta t = MAX \left(24 \text{ hours}, \left(\sin \left(MIN \left(\frac{\pi}{2}, 2 \cdot \gamma_{offset} \right) - \frac{\pi}{2} \right) + 1 \right) \cdot 30 \text{ days} \right) \quad \text{Eq. 3-33}$$

This gives a time step definition with a lower cut of at 24 hours ($\gamma_{offset} \sim 7.5^\circ$) and an upper limit of 30 days ($\gamma_{offset} > 90^\circ$). For objects that have been found within the field of view the set to two days until the object has left the field of view.

Geocentric space based sensor: LOS defined in the Local Horizon

Similar to the approach chosen for the ground based sensor and an orientation in the local horizon system also the line of sight is moving very fast for a space based sensor being oriented that way. The main difference is that for one revolution T_U is used instead of one day and that the torus is built around the normal of the orbital plane. Again the threshold value is $0.55 \cdot FOV$. If the offset angle γ_{offset}

The opening angle of the torus with respect to the normal of the orbital plane (NOP) is computed with the help of the azimuth A and the elevation h :

$$\gamma_{nop,torus} = \left(\frac{\sqrt{(A) \cdot (h) + (h)}}{\sin \sin (h) \cdot \cos \cos (h)} \right) \quad \text{Eq. 3-34}$$

To determine the offset angle $\gamma_{nop,obj}$ between the objects position and the normal of the orbital plane the position vector $\vec{r}_{geo2obj}$ is computed (see Eq. 3-22). The rotation axis \vec{r}_{nop} is defined as axis perpendicular to the orbital plane and can be computed bay:

$$\vec{r}_{nop} = \vec{r}_{sen,J2000} \times \vec{v}_{sen,J2000} \quad \text{Eq. 3-35}$$

so that the offset between object and rotation axis is

$$\gamma_{nop,obj} = VSEP(\vec{r}_{geo2obj}, \vec{r}_{nop}) \quad \text{Eq. 3-36}$$

The offset between object and torus $\gamma_{torus,obj}$ is the absolute difference between the two offset angles $\gamma_{nop,torus}$ and $\gamma_{nop,obj}$ and is used as a threshold value for a large or short increase of time:

$$\gamma_{torus,obj} = ABS(\gamma_{nop,torus} - \gamma_{nop,obj}) \quad \text{Eq. 3-37}$$

If the offset angle is larger than half the field of view ($\gamma_{torus,obj} > 0.5 \cdot FOV$) the time step Δt is defined between 1 and 90 days:

$$\Delta t = MAX\left(1, 45 \cdot \left(\sin \sin \left(\left(\gamma_{torus,obj} - 0.5 \cdot FOV\right) - \frac{\pi}{2}\right) + 1\right)\right) \text{days} \quad \text{Eq. 3-38}$$

else if the object is already closer to the torus than half the field of view ($\gamma_{torus,obj} \leq 0.5 \cdot FOV$) the time step is defined between 1 minute and $T_U/3$:

$$\Delta t = MAX\left(1, \frac{T_U}{6} \cdot \left(\sin \sin \left(\gamma_{offset} - \frac{\pi}{2}\right) + 1\right)\right) \text{minutes} \quad \text{Eq. 3-39}$$

If the object is already very close to the line of sight ($\gamma_{offset} < 0.55 \cdot FOV$) the time steps are computed considering also the range limitations:

The distance Δr between the field of view and the object can be approximated (similar as in Eq. 3-27 and Eq. 3-28) to

$$\Delta r = r_{min} - r_{obj} \quad \text{Eq. 3-40}$$

for an object below the minimum range and to

$$\Delta r = r_{obj} - r_{max} \quad \text{Eq. 3-41}$$

for an object beyond the maximum range. For the case where $r_{min} < r_{obj} < r_{max}$ the time step is set to $\Delta t = 60s$. This would be the case if an object is besides the field of view within the thin region considered because the threshold for the shorter time steps starts at $0.55 \cdot FOV$ already.

For other cases the time until the object enters the field of view is approximated by the x -component of the velocity in the “cone”-system:

$$dt_{inFOV} = \Delta r / \dot{x}_{obj, CONE} \quad \text{Eq. 3-42}$$

and only if

$$dt_{inFOV} < \left(\frac{FOV}{360^\circ}\right) \cdot T_U \quad \text{Eq. 3-43}$$

the time step is set to

$$\Delta t = 0.5 \cdot dt_{inFOV} \quad \text{Eq. 3-44}$$

else a large step is performed by

$$\Delta t = 0.95 \cdot T_U \quad \text{Eq. 3-45}$$

This large time step is also performed after the object has been found within the “cone”.

Geocentric space based sensor: LOS defined in an inertial fixed reference frame

Again the inertial fixed orientation of the line of sight eases the computation of the time step. For all cases where the object is not within the field of view the time step Δt is computed by:

$$\Delta t = MAX \left(1 \text{ day}, \left(\sin \sin \left(MIN \left(\frac{\pi}{2}, 2 \cdot \gamma_{offset} \right) - \frac{\pi}{2} \right) + 1 \right) \cdot 30 \text{ days} \right) \quad \text{Eq. 3-46}$$

If the object has been found within the field of view then the time step is set to two days.

Heliocentric space based sensor

Only for heliocentric space based sensors the chosen reference frame for the orientation of the line of sight has not such a strong impact on the movement of the line of sight than it has for some cases discussed before. Similar to the already explained time step computation of inertial oriented sensors the time steps are computed in two different ways for a sensor placed in a heliocentric orbit. For objects being inside the field of view the next time step is set to:

$$\Delta t = 1 \text{ day} \quad \text{Eq. 3-47}$$

The other case where the object is not inside the field of view the time step is based on a *sin*-function with a lower cut-off value of one day and an upper limit for the time step of a tenth of the orbit period:

$$\Delta t = MAX \left(1 \text{ day}, \left(\sin \sin \left(MIN \left(\frac{\pi}{2}, 2 \cdot \gamma_{offset} \right) - \frac{\pi}{2} \right) + 1 \right) \cdot \frac{T_U}{10} \right) \quad \text{Eq. 3-48}$$

3.1.2.1.6 Results of the “Identification of crossings”

The identification of crossings is made for all available objects in a loop. For each crossing of an object there are some parameters stored and used as input for the computation of the crossing geometry. Besides the total number of crossings (n_{crs}) for each crossing (i_{crs}) the

time (ET_{crs}) at which the crossing occurred is stored together with the time step (Δt_{crs}) and the offset angle ($\gamma_{offset,crs}$) between the line of sight and the object.

3.1.2.2 Computation of crossing geometry

After all objects have been simulated over the complete observation period for each crossing of an object the crossing geometry is computed. The crossing geometry is a set of state vectors and line of sight orientations valid for an also stored epoch. The process of computing the crossing geometry is used to generate some output results (e.g. the time of closest approach, the path offset, etc.) but its main reason for being computed is to provide the approximated path information to the detection analysis immediately following – if activated – within the loop over the crossings.

Basically the same method is used to compute the crossing geometry as it was used before to identify the crossings. The sensor and object states as well as the orientation of the line of sight is made by the same algorithms, which have been explained in section 3.1.2.1.1, 3.1.2.1.2 and 3.1.2.1.3. The only difference is the input for these algorithms which is at least only the epoch (ET) for which the states, etc. are requested.

Starting with the epoch ET_{crs} and the last time step Δt_{crs} used during the identification of the crossing an iterative process is initiated to identify the epochs ET_{in} and ET_{out} at which the object enters respectively leaves the field of view. Because the crossing geometry is later interpolated within the detection analysis at least the first and last step are slightly outside the field of view (for objects being within the field of view during the whole observation period at least the first and last step are slightly outside the observation time). The threshold values, which are used for the identification of the start and end epoch, are:

$$ET_{thresh} = 50 \text{ seconds} \quad \text{Eq. 3-49}$$

$$r_{thresh} = 1,000 \text{ km} \quad \text{Eq. 3-50}$$

$$\gamma_{thresh} = 1' = 0.016 \text{ degree} \quad \text{Eq. 3-51}$$

The time step Δt used to compute the individual steps of the crossing geometry is constant during the whole crossing and is computed to:

$$\Delta t = \frac{ET_{out} - ET_{in}}{(n_{FOVsteps} - 1)} \quad \text{Eq. 3-52}$$

For each time step of the crossing the following values are stored:

Epoch	ET
State from Sun to sensor (J2000)	$sun2sen_{J2000}$
State from Sun to object (J2000)	$sun2obj_{J2000}$

State from Sun to Earth (J2000)	$sun2geo_{J2000}$
State from sensor to object ("cone"-system)	$sen2obj_{CONE}$
Orientation of the line of sight	$\alpha_{los}, \delta_{los}, A_{los}, h_{los}$

Table 2: Crossing geometry

As a direct result of building the crossing geometry the following parameters are generated and later dumped to the output files:

Time of closest approach (TCA):

$$TCA = ET(i_{step}) \quad \text{Eq. 3-53}$$

where

$$r_{sen2obj}(i_{step}) = MIN(r_{sen2obj}(i))$$

Range at TCA:

$$r_{sen2obj}(i_{TCA}) = \sqrt{x_{sen2obj}^2(i_{TCA}) + y_{sen2obj}^2(i_{TCA}) + z_{sen2obj}^2(i_{TCA})} \quad \text{Eq. 3-54}$$

Range Rate at TCA:

$$\dot{r}_{sen2obj}(TCA) = \frac{r_{sen2obj}(i_{TCA} + 1) - r_{sen2obj}(i_{TCA} - 1)}{2 \cdot \Delta t} \quad \text{Eq. 3-55}$$

FOV dwell time:

$$t_{dwell} = ET(i_{out}) - ET(i_{in}) \quad \text{Eq. 3-56}$$

Min. path offset:

$$\gamma_{offset,min} = MIN(\gamma_{offset}(i)) \quad \text{Eq. 3-57}$$

Mean angular velocity:

$$\dot{\omega} = \frac{\sum_{i=2}^{n_{FOVsteps}} \left(\arccos \left(\frac{\vec{r}_{sen2obj, CONE}(i-1) \cdot \vec{r}_{sen2obj, CONE}(i)}{\sqrt{|\vec{r}_{sen2obj, CONE}(i-1)|} \cdot \sqrt{|\vec{r}_{sen2obj, CONE}(i)|}} \right) \right)}{t_{dwell}} \quad \text{Eq. 3-58}$$

Mean phase angle:

$$\theta_{mean} = \frac{\sum_{i=1}^{n_{FOVsteps}} \theta(i)}{n_{FOVsteps}} \quad \text{Eq. 3-59}$$

3.1.3 Detection Analysis - Optical Performance Model

For the simulation of telescopes an optical performance model has been implemented. This model covers both the visual (VIS) and the thermal infrared (TIR) spectrum of light. It can be selected whether the sensor shall operate in which mode the sensor shall be operated. While the VIS mode can be applied to all sensor placements (ground based and space based) the TIR mode is only applicable for space based sensors, because the Earth atmosphere would filter out the incoming light in this spectrum before it could enter the ground based optical system.

The data required to run the sensor model is in general:

- Initialization flag
- General information (ground/space based, VIS/TIR mode, observation start, etc.)
- Path and file name to data files (e.g. star catalogue)
- The crossing geometry (see Table 2)
- Sensor parameters (location, FOV, integration time, CCD settings, etc.)
- Object parameters (diameter, albedo, H-value)

The results of the sensor model are:

- Detection flag, Status message
- Mean objects irradiation
- Mean objects apparent magnitude
- Mean background magnitude
- Max. signal to noise ratio

3.1.3.1 Initializing the sensor model

When the sensor model is used the first time for a defined sensor system an initialization process is executed. The process covers setting up some parameters for the sensor model as well as reading the required data files.

One of the first steps is setting up the range in wavelength, which depends on the chosen sensor mode (VIS or TIR). For a sensor operating in the visual spectrum of light the considered wavelengths are in the range of $1nm < \lambda < 1,000nm$ while the limit $780nm < \lambda < 1,000\mu m$ is used for sensors operating in the thermal infrared.

The main task of the initialization is reading the data files and computing a spectrum in steps of the defined wavelength ranges for each set of provided data. This data describes on the one hand the sensor itself by defining for example the quantum efficiency of the CCD as well as the system and filter transmission. On the other hand the environment is described by data files so that the contribution of different background sources can be computed by the sensor model.

3.1.3.1.1 Airglow initialization

The data file with the default name “*optical.air*” contains the airglow contribution for a given wavelength. This data has been taken from [08].

Data type	Unit in data file	Unit internally used
Wavelength	μm	m
Airglow contribution $AIR(\lambda)$	$\frac{W}{(cm^2 \cdot \mu m \cdot sr)}$	$\frac{W}{(m^2 \cdot m \cdot sr)}$

Table 3: Airglow contribution

For the given table of airglow contribution (see Figure 7) for each step of the considered wavelength spectrum the corresponding value is interpolated.

3.1.3.1.2 Atmospheric Extinction

The data file with the default name “*optical.atm*” contains the atmospheric extinction data for a given wavelength. For internal use also here the wavelength is converted into [m] while the data itself has no unit. The atmospheric extinction depends on the elevation of the sensors line of sight. During the initialization the atmospheric extinction is computed for a zenith oriented sensor:

$$ATM_{zenith}(\lambda) = e^{-EXT(\lambda)} \quad \text{Eq. 3-60}$$

Later the impact of the elevation is considered by:

$$ATM(h, \lambda) = (ATMEXT_{zenith}(\lambda))^{\frac{1}{\cos \cos(90^\circ - h)}} \quad \text{Eq. 3-61}$$

For each step in the wavelength range the required and used value of the extinction factor EXT is interpolated.

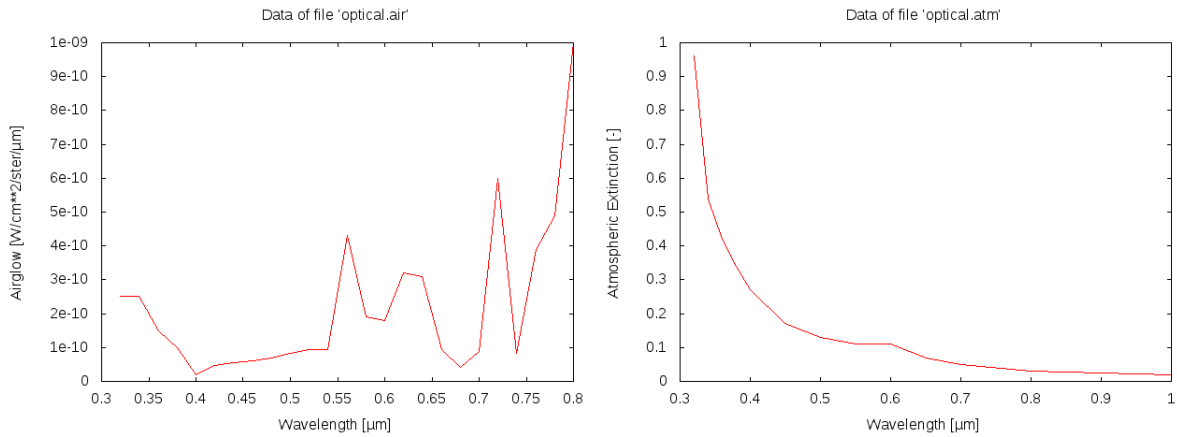


Figure 7: Airglow contribution (left) and atmospheric extinction data (right)

3.1.3.1.3 Instrumental Spectral Response

The instrumental spectral response is computed using the data of the file called “*optical.isr*” by default. For a given wavelength this file contains three factors contributing to the instrumental spectral response:

- Quantum efficiency of the CCD: QE
- Optical system transmission: OT
- Optical filter transmission: OF

The instrumental spectral response ISR is computed for each given wavelength $\lambda[m]$:

$$ISR(\lambda) = QE(\lambda) \cdot OT(\lambda) \cdot OF(\lambda) \quad \text{Eq. 3-62}$$

The ISR is valid for the range $0 \leq ISR \leq 1$. The lower the ISR the stronger is the sensor system decreasing the signal finally detected by the CCD.

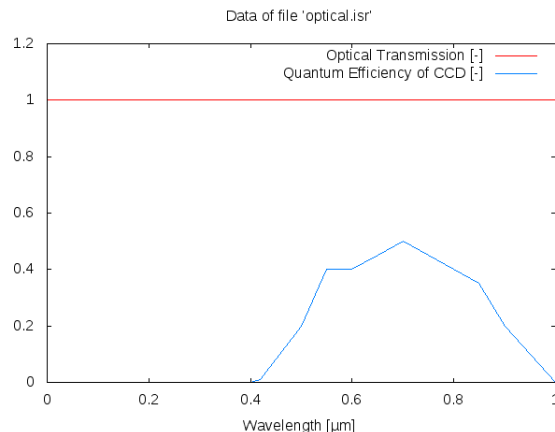


Figure 8: Quantum efficiency and optical transmission (VIS)

3.1.3.1.4 Starlight spectral distribution for a 0mag star

For the visible spectrum of light a data file (default name: *optical.ssz*) of the starlight spectral contribution is given [08]. The wavelength and data is again converted into *SI*-units in the way shown in Table 4 and interpolated for the required steps over the wavelength range for the VIS-mode.

Data type	Unit in data file	Unit internally used
Wavelength	μm	m
SSZ - data	$\frac{W}{(cm^2 \cdot \mu m)}$	$\frac{W}{(m^2 \cdot m)}$

Table 4: Starlight spectral distribution for a 0mag star

For sensor operating in the TIR mode no data file is given. Following [10] the starlight spectral contribution of a zero magnitude star is computed by:

$$SSZ(\lambda) = 1.57 \cdot 10^{-6} \cdot B_v(\lambda, T_{ref}) \quad \text{Eq. 3-63}$$

where $B_v(T_{ref})$ is the result of Planck's law (see 4.4) for the given wavelength λ and the temperature of a reference star, which is set to $T_{ref} = 10.000K$ in accordance with [08].

3.1.3.1.5 Sun spectral irradiance data

In accordance with e.g. the airglow data also the Sun spectral irradiance data is handled. The given wavelength and data is converted into *SI*-units (compare Table 3) and interpolated for the requested wavelength.

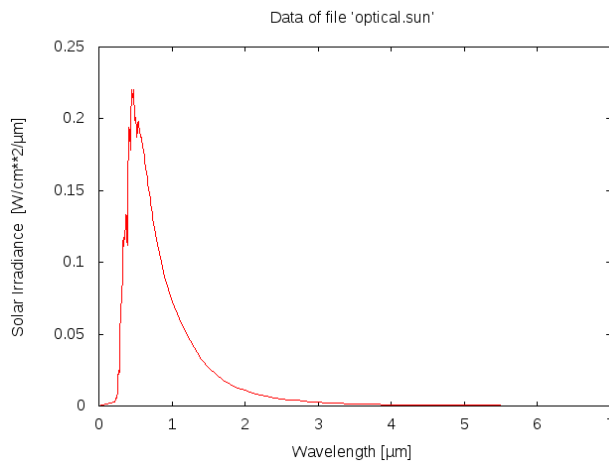


Figure 9: Sun spectral irradiance (VIS)

3.1.3.1.6 Extragalactic background light

The data for extragalactic background light (given in file: "*optical.ebl*") is only required for sensors operating in the TIR mode. Also for this data type the given data is converted into *SI*-units. In addition this data is especially converted into from a frequency into a wavelength dependency:

Data type	Unit in data file	Unit internally used
Wavelength	μm	m
Extragalactic Background Light $EBL(\lambda)$	Jr/sr	$\frac{W}{(m^2 \cdot m \cdot sr)}$

Table 5: Extragalactic Background Light

Therefore the data given in Jansky ($1 Jr = 1 \cdot 10^{-26} Ws/m^2$) is multiplied by the factor c/λ^2 .

3.1.3.1.7 Star class properties

Modeling the infrared stars a statistical model is applied [13]. In addition to the parameters taken from “Table 2: Source Table” of [13] some parameters are computed for each star class during the initialization. Parameters per star class taken from [13]:

Absolute Magnitude at $25\mu m$	M_{25}	[mag]
Scale height	h_z	[pc]
Absolute star density	ρ_0	[pc^{-3}]
Effective Temperature	T_{Eff}	[K]
Scaling factors for disk	f_{DI}	[$-$]
Scaling factors for arms	f_{AR}	[$-$]
Scaling factors for ring	f_{RI}	[$-$]
Scaling factors for bulge	f_{BU}	[$-$]

Additional parameters:

Effective Wavelength	λ_{Eff}	[m]
Scaling factor for flux computation	f_{scale}	[$-$]
Absolute Magnitude	M	[mag]

The additional parameters are computed by:

$$\lambda_{Eff} = \frac{\sum_{i=1}^{n_{int}} ISR(\lambda_i) \cdot f_{\lambda}(\lambda_i, T_{Eff}) \cdot \lambda_i}{\sum_{i=1}^{n_{int}} ISR(\lambda_i) \cdot f_{\lambda}(\lambda_i, T_{Eff})} \quad \text{Eq. 3-64}$$

$$f_{scale} = \frac{\sum_{i=1}^{n_{int}} ISR(\lambda_i) \cdot f_{\lambda}(\lambda_i, T_{Eff}) \cdot \Delta t_{int}}{\sum_{i=1}^{n_{int}} f_{\lambda}(\lambda_i, T_{Eff}) \cdot \Delta t_{int}} \quad \text{Eq. 3-65}$$

$$M = \frac{\sum_{i=1}^{n_{int}} f_{\lambda}(\lambda_i, T_{Eff}) \cdot \Delta t_{int}}{flux_{M=0}} \quad \text{Eq. 3-66}$$

with

$$f_{\lambda}(\lambda_i, T_{Eff}) = \pi \cdot B_{\nu}(\lambda_i, T_{Eff}) \cdot \frac{1}{1.41} \cdot \frac{c}{\lambda_{25\mu m}} \cdot 10^{-0.4 \cdot M_{25} - 2.444} \quad \text{Eq. 3-67}$$

3.1.3.1.8 Interstellar extinction

The interstellar extinction results from the light absorbing effect of the interstellar medium. The used model is based on [12] and initialized together with the initialization of the star classes to compute the spectral extinction of the near Earth region $A_{\lambda,0}$. The impact of the interstellar extinction is relevant for the TIR mode only.

$$A_{\lambda,0} = 0.282 \cdot \frac{A_{\lambda_{Eff}}}{A_J} \cdot A_{V0} \quad \text{Eq. 3-68}$$

where $\frac{A_{\lambda_{Eff}}}{A_J}$ is interpolated out of the given data file (default name: “*optical.vbe*”) for the effective wavelength λ_{Eff} .

3.1.3.1.9 Patched Stellar Catalogue

For faint stars in the spectrum of visible light a so-called patched stellar catalogue is used. This catalogue contains the number of stars per sky field and magnitude class for sky fields of $1^\circ \times 1^\circ$ and for 18 magnitude classes. Please consult [14] details on the patched stellar catalogue.

3.1.3.1.10 Zodiacal light data

The data to compute the background signal of the zodiacal light has been taken from [08] and is given in $[S_{10}/deg^2]$ [14]. The data is given for bins of $5^\circ \times 5^\circ$ and is interpolated into a mesh of $1^\circ \times 1^\circ$ during the initialization.

3.1.3.1.11 Scattered Sunlight data

The data for scattered sunlight has been taken from [15]. During the initialization the data is read only and converted from $\frac{W}{(m^2 \cdot nm \cdot arcsec^2)}$ into $\frac{W}{(m^2 \cdot m \cdot arcsec^2)}$.

3.1.3.1.12 Scattered moonlight data

The data to be used for the computation of the scattered moonlight is considered for ground based sensors operating in the spectrum of visible light. The given data taken from [08] depends on the lunar zenith Z_{moon} angle and is given for 0° , 30° , 60° and 75° . For a constant step size of 15° during the use of the model the values for lunar zenith angles of 15° and 45° are interpolated during the initialization.

3.1.3.1.13 Lunar Phase factor

The data lunar phase of a lunar phase factor giving the relation between scattered moonlight of a full moon and the lunar phases are taken from [08].

3.1.3.2 Frames

The core of the optical performance model is a loop over the frames that can be extracted out of the crossing geometry. One frame is create for each CCD exposure time and contains a subset of the crossing geometry. A frame has at least two data sets for the start and end of the exposure time. A dataset contains the same parameters as listed in Table 2 for the crossing

geometry but only for the time steps within the exposure time. If necessary the data for start and end time are interpolated.

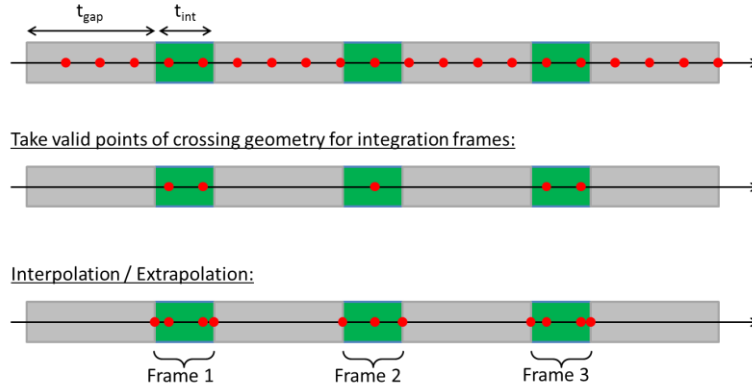


Figure 10: Extracting frames from the crossing geometry in the OPM

3.1.3.3 Basic integrals

Fundamental integrals are computed at the beginning. These values are used during the optical performance model several times and have thus to be computed only once. The integration is made in steps, which are equal to those for which the data has been interpolated during the initialization (see chapter 3.1.3.1). Each integral represents a raw signal received by the sensor, which is finally used to compute the real objects of background signal of one of the sources.

3.1.3.3.1 Raw signal to compute the objects signal

For sensors operating in the visible range of light the integral $sint$ [1/s] is computed by:

$$sint = \frac{\pi}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot SUN(\lambda) \cdot ATM(h, \lambda) \cdot \lambda \, d\lambda \quad \text{Eq. 3-69}$$

while for sensor operating in the thermal infrared the integral is:

$$sint = \frac{\pi^2}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot B_v(\lambda, T_{Sun}) \cdot \lambda \, d\lambda \quad \text{Eq. 3-70}$$

Note that the data of $ATM(\lambda)$ is not required within the TIR mode, because only space based sensors are allowed there and that $ATM(h, \lambda) = 1$ for space based sensors in the VIS mode.

3.1.3.3.2 Raw signal of a 0mag star

As a raw signal of a 0mag star the following integral $bint$ [1/s] is computed. This value is used by the VIS mode only:

$$bint = \frac{\pi}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot SSZ(\lambda) \cdot ATM(h, \lambda) \cdot \lambda \, d\lambda \quad \text{Eq. 3-71}$$

For this integral the interpolated spectral data $SSZ(\lambda)$ is used (see 3.1.3.1.5). Again $ATM(h, \lambda) = 1$ for space based sensors.

3.1.3.3.3 Raw airglow signal

Another integral required for ground based sensors only is the integral $aint \left[\frac{1}{(s \cdot sr)} \right]$ giving the raw signal caused by airglow.

$$aint = \frac{\pi}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot AIR(\lambda) \cdot ATM(h, \lambda) \cdot \lambda \, d\lambda \quad \text{Eq. 3-72}$$

3.1.3.3.4 Raw signal of the planets

Because the infrared signal of planets is computed in a different way than for the signal within visible light the integral $pint [1/s]$ is computed for the TIR mode for each planet i_{pl} only:

$$pint(i_{pl}) = \frac{\pi^2}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot B_v(\lambda, T_{Eff}(i_{pl})) \cdot \lambda \, d\lambda \quad \text{Eq. 3-73}$$

The effective temperature of the planets can be taken from [20].

3.1.3.3.5 Raw signal of galaxies

Also the raw signal of galaxies $gint \left[\frac{1}{(s \cdot sr)} \right]$ is computed for sensors operating in the TIR mode only. Considering the extragalactic background light the integral is computed by:

$$gint = \frac{\pi}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot EBL(\lambda) \cdot \lambda \, d\lambda \quad \text{Eq. 3-74}$$

3.1.3.4 Trace and Pixel Dwell Times

For a given crossing geometry of a frame the trace on the CCD and the pixel dwell times have to be computed for getting finally the signal per pixel. As shown in Figure 11 the given path of the object during the exposure has to be mapped on the CCD. The path is mirrored at the focal point which has a distance to the CCD computed by:

$$f = \left(\frac{FOV \cdot L_{pix}}{2 \cdot FOV_{pix} \cdot TAN\left(\frac{FOV}{2}\right)} \right) \quad \text{Eq. 3-75}$$

where FOV is the field of view of the sensor, L_{pix} is the edge length of a single pixel of the CCD and FOV_{pix} is the field of view per pixel. All three parameters are required input to the optical performance model and have to be defined by the user.

The image on the CCD is given in CCD coordinates x_{CCD}, y_{CCD} . The origin of this coordinate system is chosen in a way that positive coordinates are used only. While looking along the line of sight of the sensor one could say that the origin of the image is on the bottom left side.

For each step of a given frame the CCD-coordinates are computed. Therefor the following steps are performed:

Compute the angle between the line of sight and the direction of the object:

$$Z_{CCD,obj} = ACOS \left(\frac{x_{CONE}}{\sqrt{x_{CONE}^2 + y_{CONE}^2 + z_{CONE}^2}} \right) \quad \text{Eq. 3-76}$$

With the help of this angle the length from center of the CCD to the point of the object on the CCD can be computed by:

$$dist = f \cdot \tan \tan (Z_{CCD,obj}) \quad \text{Eq. 3-77}$$

The angle A_{CCD} between the y_{cone} -axis and $dist$ can be computed by

$$A_{CCD} = atan \left(\frac{y_{CONE}}{z_{CONE}} \right) \quad \text{Eq. 3-78}$$

so that the coordinates in the CCD coordinate system are

$$x_{CCD} = offset \cdot dist \cdot \cos (A_{CCD}) \quad \text{Eq. 3-79}$$

$$y_{CCD} = offset \cdot dist \cdot \sin (A_{CCD}) \quad \text{Eq. 3-80}$$

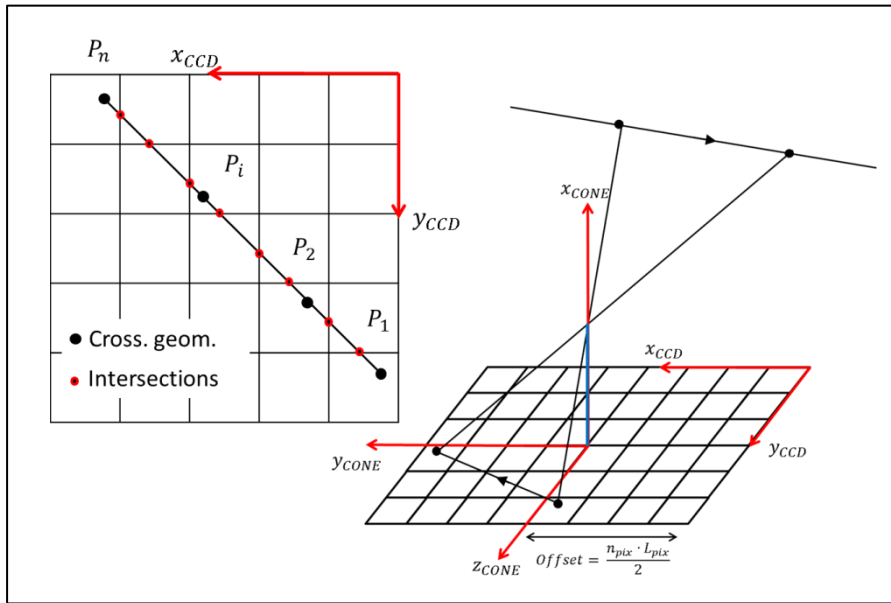


Figure 11: Projection of the object's motion on the CCD.

For each point P_i given by the crossing geometry of the frame the CCD coordinates $x_{CCD}(P_i)$ and $y_{CCD}(P_i)$ are known. As it is demonstrated in Figure 11 these points are not necessarily sufficient for the dwell time computation. For this the entry and exit points and the related time have to be known by computing the trace.

On the left side of Figure 11 the trace of the object on the CCD is illustrated. The large black dots are given by the crossing geometry of the frame and have the CCD coordinates as computed above. The trace between two nearby points is assumed to be linear. Here the impact of the number of steps per field of view as defined by the user becomes visible. The more steps the better is the approximation of a curved object path.

Of interest for the pixel dwell time computation are the red dots at the edges of the pixels touched by the objects image. Because the line between two black dots is assumed to be linear these points can be computed easily:

The vector \vec{v}_{trace} between two nearby points of the crossing geometry can be computed by:

$$\vec{v}_{trace} = \vec{v}_i + k \cdot (\vec{v}_{i+1} - \vec{v}_i) \quad \text{Eq. 3-81}$$

Thus each point on this trace section with the coordinates x_{trace} and y_{trace} can be computed by:

$$x_{trace}(k) = x_i + k \cdot (x_{i+1} - x_i) \quad y_{trace}(k) = y_i + k \cdot (y_{i+1} - y_i) \quad \text{Eq. 3-82}$$

This leads to two equations for k which have to be used to compute on the one hand intersection points with the edge of a pixel being parallel to the x_{CCD} -axis and on the other hand to the intersection with the y_{CCD} -axis.

$$\text{in } x - \text{direction: } k_x = \frac{x_{edge} - x_i}{x_{i+1} - x_i} \quad \text{in } y - \text{direction: } k_y = \frac{y_{edge} - y_i}{y_{i+1} - y_i} \quad \text{Eq. 3-83}$$

The coordinates x_{edge} and y_{edge} are multiples of the pixel length. The missing coordinates are $x_{trace}(k_y)$ and $y_{trace}(k_x)$. Only for $0 < k < 1$ the found intersections are valid. Finally not the intersection points but the mean coordinates are stored for j affected pixel:

$$x_{mean,j} = \frac{x_j + x_{j+1}}{2} \quad y_{mean,j} = \frac{y_j + y_{j+1}}{2} \quad \text{Eq. 3-84}$$

The dwell time for each pixel is computed assuming that the objects velocity is constant of the trace section:

$$t_{dwell,j} = \frac{t_{i+1} - t_i}{\sqrt{(x_i - x_1)^2 + (y_i - y_1)^2}} \cdot \left((x_j - x_{j+1})^2 + (y_j - y_{j+1})^2 \right) \quad \text{Eq. 3-85}$$

For j affected pixel there are $(j + 1)$ intersections within the frame of i steps in the crossing geometry.

After having determined the geometrical conditions and the dwell time the object's signal can be computed for each pixel.

3.1.3.5 The objects signal

Supported by the geometrical conditions, the mean positions of the objects trace per pixel and the related pixel dwell times can be computed. Therefore the irradiance originating from the object is computed first.

3.1.3.5.1 Received objects signal (VIS)

For a sensor operating in the visible light spectrum the basic irradiance I_0 , which the sensor receives from the object, is computed by assuming the behavior of a Lambertian sphere:

$$I_0 = I_{reflect}(\alpha, I_{obj}, R_{obj}, r_{sen2obj}) \cdot p(\theta_{pha}) \quad \text{Eq. 3-86}$$

The average irradiance originating from the object at the distance of the sensor (assuming the object radiates uniformly into an entire sphere) $I_{reflect}$ is a function of the object's albedo α , its radius R_{obj} , the solar irradiance at the object I_{obj} and of the distance between sensor and object $r_{sen2obj}$. A phase function $p(\theta_{pha})$ is applied to fully consider the geometrical conditions between object, Sun and sensor with the help of the phase angle θ_{pha} .

In detail the combination of average irradiance originating from the object

$$I_{reflect}(\alpha, I_{obj}, r_{obj}, r_{sen2obj}) = I_{obj} \cdot \alpha \cdot \frac{\pi \cdot R_{obj}^2}{4\pi \cdot r_{sen2obj}^2} \quad \text{Eq. 3-87}$$

and the phase function

$$p(\theta_{pha}) = \frac{8}{3\pi} \cdot (\sin \sin (\theta_{pha}) + (\pi - \theta_{pha}) * \cos \cos (\theta_{pha})) \quad \text{Eq. 3-88}$$

gives the basic irradiance

$$I_0(\theta) = \frac{2}{3\pi} \cdot \alpha \cdot I_{obj} \cdot \frac{R_{obj}^2}{r_{sen2obj}^2} \cdot (\sin \sin (\theta_{pha}) + (\pi - \theta_{pha}) * \cos \cos (\theta_{pha})) \quad \text{Eq. 3-89}$$

which is received by the sensor.

For phase angles $\theta_{pha} > 40^\circ$ the computed basic received irradiance I_0 is used as received irradiance:

$$I_{rec}(\theta_{pha}) = I_0(\theta_{pha}) \quad \text{Eq. 3-90}$$

For phase angles $\theta_{pha} < 40^\circ$ the opposition effect is considered. This effect considers the brightening of the rough object's surface, when it is illuminated from behind the sensor. Therefore the computed basic irradiance is scaled:

$$I_{rec}(\theta_{pha}) = I_0(\theta_{pha}) \cdot \frac{I_{ref}(\theta_{pha})}{I_{ref}(40^\circ)} \quad \text{Eq. 3-91}$$

The reference irradiance is once computed for the current phase angle and once for the reference point at $\theta_{pha} = 40^\circ$. In the following θ is used to explain how $I_{ref}(\theta)$ is determined. Starting with the apparent magnitude m_{obj} of an object that can be computed in relation to a reference of $m_{ref} = 0$ by the following equation:

$$m_{obj} - m_{ref} = m_{obj} = -2.5 \cdot \frac{F_{obj}}{F_{ref}} \quad \text{Eq. 3-92}$$

where F is the flux coming from the object (F_{obj}) and the reference (F_{ref}). As reference the incoming flux from a zero magnitude star is used that is computed by using the basic integrals (see section 3.1.3.3)

$$F_{ref} = \frac{b_{int} \cdot E_s}{s_{int}} \quad \text{Eq. 3-93}$$

with E_s being the solar constant at Earth's distance from the Sun. The wanted object's irradiance is identical with the object's flux so that Eq. 3-92 can be written in the form:

$$F_{obj} = 10^{\left(-\frac{m_{obj}}{2.5} + I_{ref}\right)} \quad \text{Eq. 3-94}$$

The missing object's apparent magnitude m_{obj} is computed using an approach for solar system bodies:

$$m_{obj} = H(1, \theta) + 2.5 \cdot \left(\frac{r_{sen2obj}^2 \cdot r_{sun2obj}^2}{p'(\theta) \cdot AU^4} \right) \quad \text{Eq. 3-95}$$

with the phase function

$$p'(\theta) = \frac{2}{3} \cdot \left(\left(1 - \frac{\theta}{\pi} \right) (\theta) + \frac{1}{\pi} \cdot \sin \sin (\theta) \right) \quad \text{Eq. 3-96}$$

As described in [16] the opposition effect is described by computing the absolute magnitude as a function of the phase angle:

$$H(1, \theta) = H(1, 0) - \frac{a}{1 + \theta} + b \cdot \theta \quad \text{Eq. 3-97}$$

where a characterizes the amplitude of the opposition effect and b describing the linear part of the magnitude-phase relations (as described in [17]).

The value a depends on the asteroid type. As shown in Table II of [16] the different asteroid types have different albedo values. Thus, the albedo is used here to determine the value a :

$\alpha \leq 0.05$	$a = 0.110$
$\alpha > 0.05 \ \& \ \alpha \leq 0.08$	$a = 0.160$
$\alpha > 0.13 \ \& \ \alpha \leq 0.17$	$a = 0.340$
$\alpha > 0.17 \ \& \ \alpha \leq 0.25$	$a = 0.350$
$\alpha > 0.48 \ \& \ \alpha \leq 0.54$	$a = 0.230$
<i>else</i>	$a = 0.238$

Table 6: Determining the amplitude of the opposition effect

The value b is a function of the albedo, too. For phase angles $\theta_{pha} < 5^\circ$ the value is computed in accordance with [17] while for larger phase angles $\theta_{pha} > 5^\circ$ the value is computed in accordance with [16]:

$$\theta_{pha} < 5^\circ: b = 0.0125 - 0.01045 \cdot \ln \ln \alpha \quad \text{Eq. 3-98}$$

$$\theta_{pha} > 5^\circ: b = 0.013 - 0.024 \cdot \alpha \quad \text{Eq. 3-99}$$

Finally the object's signal per pixel is:

$$S_{obj}(\theta_{pha}) = I_{rec}(\theta_{pha}) \cdot \frac{shint}{E_s} \quad \text{Eq. 3-100}$$

where the ratio of the received object signal (I_{rec}) in relation to the maximum possible signal (E_s - solar constant), is multiplied with $shint$, the factor that converts the signal ratio into the

responds in the CCD, based on the integration time and the quantum efficiency (QE) of the sensor. The resulting unit of S_{obj} is $[e^-/s]$.

3.1.3.5.2 Received objects signal (TIR)

A sensor operating in the infrared would receive a signal from the object that consists out of two components. One the one hand the object reflects a part of the incoming signal it receives. On the other hand the object emits itself a signal in the infrared:

$$S_{obj}(\theta_{pha}) = S_{reflected}(\theta_{pha}) + S_{emitted} \quad \text{Eq. 3-101}$$

The reflected contribution can be calculated by:

$$S_{reflected}(\theta_{pha}) = \frac{1}{4} \cdot sint \cdot \left(\frac{R_{sun} \cdot R_{obj}}{r_{sun2obj} \cdot r_{sen2obj}} \right)^2 \cdot (1 - A) \cdot p(\theta_{pha}) \quad \text{Eq. 3-102}$$

where $p(\theta_{pha})$ is the phase function as already defined in Eq. 3-88 . The bolometric bond albedo A is in accordance with [18] equal to (G is set to 0.15 by default):

$$A = (0.290 + 0.684 \cdot G) \cdot \alpha \quad \text{Eq. 3-103}$$

The emitted contribution can be calculated by:

$$S_{emitted} = \epsilon \cdot oint \cdot \left(\frac{R_{sun}}{r_{sun2obj}} \right)^2 \quad \text{Eq. 3-104}$$

In accordance with [18] the emissivity $\epsilon = 0.9$. Similar to the basic integral $oint$ is the integrated raw signal of the object, which depends on the objects temperature T_{obj} which on his part depends on the phase angle θ_{pha} and is thus not pre-computable.

$$oint = \frac{\pi^2}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot B_v(\lambda, T_{obj}) \cdot \lambda \, d\lambda \quad \text{Eq. 3-105}$$

The objects temperature is computed by the standard thermal model [18]:

$$T_{obj} = \left(\frac{(1 - A) \cdot E_{rec}}{\sqrt[4]{\eta \cdot \epsilon \cdot k_b}} \right) \cdot \cos \cos(\theta_{pha}) \quad \text{Eq. 3-106}$$

with a so-called “beaming parameter” $\eta = 0.756$ [18] and the Stefan-Boltzmann constant k_b .

3.1.3.6 The background signal

The background signal is a conglomerate of different sources. Each of the following subchapters is dealing with one of these sources. Some sources are relevant only in the visual light spectrum, some in the thermal infrared only. And the impact of a source might vary also if the source is relevant for both spectral regimes.

For both modes (VIS & TIR) the sum of the background signals of all continuous sources is computed. Those being not relevant – e.g. airglow for space based sensors – contain no signal ($S = 0$). The total background signal of continuous sources is thus:

$$S_{bkg,cont.} = S_{ISM} + S_{Stars,V>V_{max}} + S_{Airglow} + S_{Zodiacal} + S_{Galaxis} + S_{sc.atm.} + S_{sc.moon} + S_{sc.sun} \quad \text{Eq. 3-107}$$

In addition the contribution of planets is handled discrete because of the limited number of planets. Also stars with a magnitude $V < V_{max}$ are handled as discrete sources and are not part of the signal $S_{Stars,V>V_{max}}$.

Table 7 gives an overview over all background sources described in the following sections:

Background Signal	Section / Page
Planets	3.1.3.6.1 on page
Stars	3.1.3.6.2 on page
Galaxies	3.1.3.6.3 on page
Airglow	3.1.3.6.4 on page
Zodiacal Light	3.1.3.6.5 on page
Interstellar Medium	3.1.3.6.6 on page
Atmospherically Scattered Light	3.1.3.6.7 on page
Atmospherically Scattered Sunlight	3.1.3.6.8 on page
Atmospherically Scattered Moonlight	3.1.3.6.9 on page

Table 7: Where you will find the background sources

3.1.3.6.1 Background signal of Planets

As done for the objects signal also the signal of each planet is handled in a discrete way. This means that the trace of the planet on the CCD is approximated by its position at start and end of each frame and also the pixel dwell time is computed for each affected pixel.

In the **VIS mode**, the signal of each star is computed by:

$$S_{planet} = b_{int} \cdot 10^{-0.4 \cdot m_{pl}} \quad \text{Eq. 3-108}$$

The apparent magnitude m_{pl} of the planet is calculated as described in [19]. Therefor the phase angle θ and the range from the sensor to the planet are required:

For Mercury:

$$m_{pl} = -0.36 + 5r_{sen2pl} + 3.8 \left(\frac{\theta}{100^\circ} \right) - 2.73 \left(\frac{\theta}{100^\circ} \right)^2 + 2 \left(\frac{\theta}{100^\circ} \right)^3 \quad \text{Eq. 3-109}$$

For Venus:

$$m_{pl} = -4.29 + 5r_{sen2pl} + 0.09 \left(\frac{\theta}{100^\circ} \right) - 2.39 \left(\frac{\theta}{100^\circ} \right)^2 - 0.65 \left(\frac{\theta}{100^\circ} \right)^3 \quad \text{Eq. 3-110}$$

For Mars:

$$m_{pl} = -1.52 + 5r_{sen2pl} + 0.016 \cdot \theta \quad \text{Eq. 3-111}$$

For Jupiter:

$$m_{pl} = -9.25 + 5r_{sen2pl} + 0.005 \cdot \theta \quad \text{Eq. 3-112}$$

For Saturn:

$$m_{pl} = -7.19 + 5r_{sen2pl} + 0.044 \cdot \theta \quad \text{Eq. 3-113}$$

For Uranus:

$$m_{pl} = -7.19 + 5r_{sen2pl} + 0.028 \cdot \theta \quad \text{Eq. 3-114}$$

For Neptune:

$$m_{pl} = -6.87 + 5r_{sen2pl} + 0.01 \cdot \theta \quad \text{Eq. 3-115}$$

For Pluto:

$$m_{pl} = -1.01 + 5r_{sen2pl} + 0.041 \cdot \theta \quad \text{Eq. 3-116}$$

For Earth and Moon no signal will be computed because the software ignores frames where at least one of those two bodies is in the field of view.

In the **TIR mode** the basically integrated value $pint$ (see chapter 3.1.3.3.4) is used together with the geometric conditions of the planets radius and its distance from the sensor:

$$S_{planet} = pint(i_{pl}) \cdot \left(\frac{R_{pl}}{r_{sen2pl}} \right)^2 \quad \text{Eq. 3-117}$$

3.1.3.6.2 Background signal of stars

For the *faint stars* in the **VIS mode** a so-called patched stellar catalogue is used [14]. The number of stars has been summed up in bins of $1^\circ \times 1^\circ$ and for 25 magnitude classes. Only for the area on the CCD (or field of view) of the objects trace the background signal is computed and not for the whole CCD (or field of view). To consider this area the number of stars taken from the patched stellar catalogue is scaled with the solid angle Ψ_{trace} .

$$N_{stars,\psi} = N_{stars,1 \times 1} \cdot \frac{\Psi_{trace}}{\Psi_{bin}}$$

The signal of the stars fainter than the defined maximum magnitude of stars to be considered as discrete point sources is:

$$S_{stars, V > V_{max}} = t_{int} \cdot bint \cdot (FOV_{pix})^2 \cdot \sum_{k=k(V_{max}+1)}^{25} N_{stars, \psi}(k) \cdot 10^{-0.4 \cdot (V(k))} \quad \text{Eq. 3-118}$$

where k is the magnitude class and V the visual magnitude. The unit of S_{stars} is $[e - /s]$.

For *brighter stars* with a magnitude below the maximum magnitude of stars to be considered as discrete point sources are handled in the same way as it is done for the objects signal. The star signal is computed for each affected pixel along the trace on the CCD. By using the basically integrated value $bint$ the signal per pixel and second is computed by:

$$S_{stars, V < V_{max}} = bint \cdot 10^{-0.4 \cdot (V_{star})} \quad \text{Eq. 3-119}$$

For the **TIR mode** the consideration of stars is performed by using a star class catalogue taken from [13]. For each of the 87 listed star classes a star density is given. By scaling factors f the contribution of this density to stars of the different galaxy regions (Bulge, Arms, Ring, Disk) is computed. The signal is computed by integrating along the line of sight in 534 integration steps.

The length of the integration interval $lenint$ along the line of sight is determined first for each integration step j :

Integration step	Length of the integration interval
$j \leq 50$	$lenint = 2 \text{ pc}$
$50 < j \leq 100$	$lenint = 4 \text{ pc}$
$100 < j \leq 150$	$lenint = 8 \text{ pc}$
$150 < j \leq 200$	$lenint = 16 \text{ pc}$
$200 < j \leq 250$	$lenint = 32 \text{ pc}$
$250 < j \leq 300$	$lenint = 64 \text{ pc}$
$300 < j \leq 534$	$lenint = 80 \text{ pc}$

Table 8: Length of the integration interval along the line of sight

The volume of each integration interval ($j > 1$) has the form of a truncated cone. Some parameters are necessary for the determination of the star density within the considered volume of each integration step:

Parameter	Explanation	Unit
r_{glc}	Distance between Sensor the galactic center	$[pc]$
$V_{tr.cone}$	Volume of the truncated cone	$[pc^3]$
r_{com}	Range from the sensor to the "center of mass (COM)" of the truncated cone	$[pc]$
r_a	Range from the galactic center to COM	$[pc]$

r_b	Range from the galactic center to the projection of the COM onto the galactic plane	[pc]
r_c	Height of the COM above the galactic plane	[pc]
θ_p	Angle of from the x-axis of the galactic coordinate system (centered in the galactic center) to the projection of the COM onto the galactic plane.	[deg]

The model is based on the star densities ρ_{stars} , which have to be determined for each star class i_{stc} before the contribution of the stars can be considered as background signal.

$$\rho_{stars} = \rho_{disc} + \rho_{bulge} + \rho_{ring} + \sum_{iarms=1}^4 \rho_{iarms} \quad \text{Eq. 3-120}$$

The overall star is the sum of the individual sources. The “halo”, which is also described in [13], has not been considered as source for the implemented model:

Star density of the galactic disk:

As described in [13] the star density within the galactic disk can be computed by:

$$\rho_{disc} = \rho_0 \cdot f_{disc} \cdot e^{\left(-\frac{(r_b - r_{glc})}{h} - \frac{r_c}{h_z}\right)} \quad \text{Eq. 3-121}$$

Therein the so far unknown h is the radial scale length, which is adopted to be $3.5kpc$.

Star density of the galactic bulge:

The star density for the galactic bulge is supported by:

$$X = \frac{\sqrt{(r_b^2 + k_1^2 \cdot r_c^2)}}{r_{bulge}}$$

where – in accordance with [13] – k_1 is the bulge axis ratio, which is set to 1.6, and r_{bulge} is the bulge radius, which is set to $2.0kpc$. Thus the star density can be computed by:

$$\rho_{bulge} = \rho_0 \cdot f_{bulge} \cdot f_{corr,bulge} \cdot X^{-1.8} \cdot e^{-X^3} \quad \text{Eq. 3-122}$$

For this model a correction factor $f_{corr,bulge} = 3.6$ has been defined.

Star density of the molecular ring:

As third component the contribution of the molecular ring is determined. Again a correction factor is used and has been set to $f_{corr,ring} = 25.0$. Another parameter σ_{ring} corresponds to a half-width for the Gaussian ring: $\sigma_{ring} = 544kpc$.

$$\rho_{ring} = \rho_0 \cdot f_{ring} \cdot f_{corr,ring} \cdot e^{\left(-\frac{(r_b-r_{ring})}{2\sigma_{ring}} - \frac{r_c}{h_z}\right)} \quad \text{Eq. 3-123}$$

where the radius r_{ring} of the molecular ring is $3825kpc$.

Star density of the spiral arms:

At least the star density for each of the four spiral arms is computed. The same equation as for the computation of the disks star density is used with the correction factor $f_{corr,arm} = 5.0$:

$$\rho_{arm} = \rho_0 \cdot f_{arm} \cdot f_{corr,arm} \cdot e^{\left(-\frac{(r_b-r_{glc})}{h} - \frac{r_c}{h_z}\right)} \quad \text{Eq. 3-124}$$

The contribution of a spiral arm is only considered if the volume of the interpolation step along the line of sight comes in touch with one of the arms.

To have a more realistic distribution of the magnitude computed out of the star density ρ_{stars} in the following the density is split up in five subclasses (i_{sub}) approximating a Gaussian distribution. For each in only a fraction of the whole star density is considered:

Difference of absolute magnitude ΔM	Fraction of Density δ_{frac}
$\Delta M = -1.939$	$\delta_{frac} = 0.0668$
$\Delta M = -0.921$	$\delta_{frac} = 0.2417$
$\Delta M = 0.000$	$\delta_{frac} = 0.3830$
$\Delta M = 0.921$	$\delta_{frac} = 0.2417$
$\Delta M = 1.939$	$\delta_{frac} = 0.0668$

Table 9: Subclasses per star class

Thus the number of stars within the volume of the current integration step is:

$$n_{stars} = \rho_{stars} \cdot \delta_{frac} \cdot V_{tr.cone} \quad \text{Eq. 3-125}$$

The resulting apparent magnitude of the an exemplary star of this class is determined with the absolute magnitude M , which has been determined during the initialization for each star class (see section 3.1.3.1.7):

$$m_{star}(i_{stc}, i_{sub}) = (M(i_{stc}) - \Delta M(i_{sub})) + 5(r_{com}) + f_{ext} \quad \text{Eq. 3-126}$$

By adding f_{ext} the extinction of the star light is considered for each integration step j :

$$f_{ext}(j) = A_{\lambda,0}(j) \cdot \sum_{i=1}^j e^{\left(-\frac{r_b - r_{glc}}{h} - \frac{r_c}{h_{zd}}\right)} \quad \text{Eq. 3-127}$$

where the scale height of the absorption $h_{zd} = 100pc$. Based on the now known apparent magnitude m_{star} the flux F incoming from this star can be determined with the already know equation:

$$F_{star} = F_{mag0} \cdot 10^{-0.4 \cdot m_{star}} \quad \text{Eq. 3-128}$$

so that finally the signal of all stars can be determined as incoming electrons per second with the unit $[e^-/s]$:

$$S_{stars} = t_{int} \cdot \frac{(FOV_{pix})^2}{(2\pi + 1 - \cos(\frac{FOV}{2}))^2} \cdot n_{stars} \cdot \left[\frac{\pi}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \cdot (F_{star} \cdot f_{scale}) \cdot \lambda_{Eff} \right] \quad \text{Eq. 3-129}$$

3.1.3.6.3 Background signal of galaxies

For the background signal of galaxies to be considered in the **VIS mode** a model taken from [08] is used. This model is based on the number of observed galaxies in the magnitude range down to magnitude $m = 22$. The signal of galaxies is computed independent from the orientation and thus assumed to be uniformly distributed:

$$S_{galaxies} = \frac{1}{4\pi} \cdot (FOV_{pix})^2 \cdot b_{int} \cdot t_{int} \cdot N_{22} \cdot \sum_{m=6}^{22} \left(\left(\frac{1}{4} \right)^{22-m} \cdot 10^{-0.4 \cdot m} \right) \quad \text{Eq. 3-130}$$

with the constant number of galaxies of magnitude $m = 22$ is $N_{22} = 7.5 \cdot 10^8$.

Within the **TIR mode** the basically computed value g_{int} (see section 3.1.3.3.5) is used to compute the contribution of the galaxies to the background sources:

$$S_{galaxies} = (FOV_{pix})^2 \cdot g_{int} \cdot t_{int} \quad \text{Eq. 3-131}$$

3.1.3.6.4 Background signal due to airglow

The contribution of airglow as background source has only to be considered for the VIS mode, because only sensors operating in the spectrum of visible light can be ground based so that the airglow becomes relevant. As described in [08] for most systems the impact of the airglow depends on

$$F(h) = \frac{1}{\cos \cos(90^\circ - h)}$$

so that the contribution to the background signal is

$$S_{galaxies} = (FOV_{pix})^2 \cdot a_{int} \cdot t_{int} \cdot \frac{1}{\cos \cos (90^\circ - h)} \quad \text{Eq. 3-132}$$

3.1.3.6.5 Background signal due to zodiacal light

The zodiacal light contributes to the background signal in both the VIS and the TIR mode. For the **VIS mode** the zodiacal light is given for a grid in ecliptic coordinates. The signal is computed by

$$S_{zodiacal} = t_{int} \cdot (FOV_{pix})^2 \cdot b_{int} \cdot 10^{-4} \cdot ZOD \quad \text{Eq. 3-133}$$

where $ZOD [S_{10}/deg^2]$ is taken from the interpolated table as explained in 3.1.3.1.10.

In the **TIR mode** the zodiacal light contribution is based on the modeled interplanetary dust clouds [21]. Similar as done for the background signal generated by stars (see section 3.1.3.6.2) again in integration along the line of sight is performed. This time the dust density of three types of dust clouds is computed:

- Smooth Cloud
- Circumsolar Ring
- Bust Bands

The integration along the line of sight is performed in steps of ending at least at , which is about the distance to Jupiter. The heliocentric ecliptic coordinates (X, Y, Z) of the current point of interest are:

$$\begin{aligned} X &= r_{sun2sen} \cdot \cos \cos (\lambda_{sen}) + s \cdot \cos \cos (\beta_{LOS}) \cdot \cos \cos (\lambda_{LOS}) \\ Y &= r_{sun2sen} \cdot \sin \sin (\lambda_{sen}) + s \cdot \cos \cos (\beta_{LOS}) \cdot \sin \sin (\lambda_{LOS}) \\ Z &= s \cdot \sin \sin (\beta_{LOS}) \end{aligned} \quad \text{Eq. 3-134}$$

where $r_{sun2sen}$ is the heliocentric distance of the sensor, λ_{sen} is the heliocentric longitude of the sensor and $\lambda_{LOS}, \beta_{LOS}$ give the orientation of the line of sight in ecliptic coordinates. The distance between the Sun and the point of interest s is: $R = \sqrt{X^2 + Y^2 + Z^2}$.

An offset of the dust cloud is considered and the coordinates (X, Y, Z) are corrected by the offset values (X_0, Y_0, Z_0) :

$$\begin{aligned} X' &= X - X_0 = X - 0.01190AU \\ Y' &= Y - Y_0 = Y - 0.00548AU \\ Z' &= Z - Z_0 = Z + 0.00215AU \end{aligned}$$

which are used to determine also a corrected distance R_c .

Smooth Dust Cloud:

The symmetry plane of the smooth cloud is allowed to be tilted. The height above the tilted midplane is thus:

$$Z_c = X' \cdot \sin \sin (\Omega) \cdot \sin \sin (i) - Y' \cdot \cos \cos (\Omega) \cdot \sin \sin (i) + Z' \cdot \cos \cos (i) \quad \text{Eq. 3-135}$$

where $\Omega = 77.7^\circ$ is the ascending node and $i = 2.03^\circ$ is the inclination of the smooth cloud. Some further parameters are required to compute the dust density of the smooth cloud. These values are taken from Table 1 of [21]:

Description	Parameter	Value	Unit
Density at 1AU	n_0	$1.13 \cdot 10^{-7}$	[AU ⁻¹]
Radial power-law exponent	α	1.34	[-]
Vertical shape parameter	β	4.14	[-]
Vertical power-law exponent	γ	0.942	[-]
Widening parameter	μ	0.189	[-]

Table 10: Density and geometrical parameters of the smooth cloud

The vertical distribution is given by the function $f(\zeta)$, where ζ is the ratio of height Z_c and distance R_c :

$$f(\zeta) = e^{-\beta \cdot \gamma \cdot g(\zeta)} \quad \text{with } \zeta = \left| \frac{Z_c}{R_c} \right|$$

where $g(\zeta) = \frac{\zeta^2}{2 \cdot \mu}$ (for $\zeta < \mu$) or $g(\zeta) = \zeta - \frac{\mu}{2}$ (for $\zeta \geq \mu$). The resulting dust density of the smooth cloud is thus:

$$n_{cloud} = n_0 \cdot R_c^{-\alpha} \cdot f(\zeta) \quad \text{Eq. 3-136}$$

Circumsolar Ring:

The circumsolar ring is created by the Earth gravity, which temporarily traps migrating dust particles into resonance orbits near 1AU [21]. The model also considers the “trailing blob” but also mentions that disregarding the blob would cause only a small error. Thus the density of the circumsolar ring is computed by:

$$n_{ring} = n_{sr} \cdot e^{\left[-\frac{(R-R_{sr})^2}{2\sigma_{r,sr}^2} - \frac{|Z_R|}{\sigma_{z,sr}} \right]} \quad \text{Eq. 3-137}$$

with several parameters given in Table 11, the radial distance R and the vertical distance Z_R given by

$$Z_R = X' \cdot \sin \sin (\Omega) \cdot \sin \sin (i) - Y' \cdot \cos \cos (\Omega) \cdot \sin \sin (i) + Z' \cdot \cos \cos (i) \quad \text{Eq. 3-138}$$

where $\Omega = 22.3^\circ$ is the ascending node and $i = 0.49^\circ$ is the inclination of the circumsolar ring.

Description	Parameter	Value	Unit
Density at 1AU	n_{sr}	$1.83 \cdot 10^{-8}$	[AU ⁻¹]
Radius of peak density	R_{sr}	1.03	[AU]
Radial dispersion	$\sigma_{r,sr}$	0.025	[AU]

Vertical dispersion	$\sigma_{z,sr}$	0.054	[AU]
---------------------	-----------------	-------	------

Table 11: Density and geometrical parameters of the circumsolar ring

Dust Bands:

The dust bands, which have been discovered in the IRAS and DIRBE data, are believed to be the result of asteroid collisions. In the model there are three dust bands include, which are located at $\beta = \pm 1.4^\circ$, $\beta = \pm 10^\circ$ and $\beta = \pm 15^\circ$. The bands are centered on the Sun and have an inclination i_{b_i} and an ascending node Ω_{b_i} . Those and some other parameters are given in

Description	Parameter	Value of band			Unit
		1	2	3	
Density at 3AU	n_{3b_i}	$5.59 \cdot 10^{-10}$	$1.99 \cdot 10^{-9}$	$1.44 \cdot 10^{-10}$	[AU ⁻¹]
Shape parameter	δ_{ζ,b_i}	8.79	1.99	15.0	[deg]
Shape parameter	ν_{b_i}	0.10	0.90	0.05	[–]
Shape parameter	p_{b_i}	4.0	4.0	4.0	[–]
Inner radial cutoff	δ_{R,b_i}	1.5	0.94	1.5	[deg]
Ascending node	Ω_{b_i}	80.0	30.3	80.0	[deg]
Inclination	i_{b_i}	0.56	1.2	0.8	[deg]

Table 12: Density and geometrical parameters of the three dust bands

Based on these parameters the density of each dust band is computed. Again the radial distance R and the vertical distribution $\zeta_{b_i} = \left| \frac{z_{b_i}}{R_c} \right|$ are used:

$$n_{b_i} = \frac{3 \cdot n_{3b_i}}{R} \cdot \exp \left(- \left(\frac{\zeta_{b_i}}{\delta_{\zeta,b_i}} \right)^6 \right) \cdot \left[\nu_{b_i} + \left(\frac{\zeta_{b_i}}{\delta_{\zeta,b_i}} \right)^{p_{b_i}} \right] \cdot \left[1 - \exp \left(- \left(\frac{R}{\delta_{R,b_i}} \right)^{20} \right) \right] \quad \text{Eq. 3-139}$$

Dust Temperature:

The dust temperature is required to perform the integration over the bandwidth as done for the basic integrals also. In accordance with [21] the dust temperature T can be computed as a function of the distance R :

$$T = T(s) = T(R) = T_0 \cdot R^{-\delta} \quad \text{Eq. 3-140}$$

with a derived value $\delta = 0.0467$.

Integrated signal of the zodiacal light:

The final signal of the zodiacal light includes two integrations. The first one is performed along the line of sight, as introduced already, where for each step the dust density is computed. The second integration is performed over the bandwidth of the TIR mode.

$$S_{zodiacal} = \frac{\pi}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \cdot t_{int} \cdot \int_{s=0}^{5.2AU} n_{dust}(s) \cdot \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot B_{\lambda}(\lambda, T) \cdot \lambda d\lambda ds \quad \text{Eq. 3-141}$$

where B_λ is the Planck radiation for a given wavelength λ and the dust temperature T .

3.1.3.6.6 Background signal due to the interstellar medium

Equivalent to the interplanetary dust also the interstellar medium has a contribution to the background signal in the TIR mode. Data of the DIRBE and IRAS missions have been used to generate a model, which is used to compute the contribution of the interstellar medium to the background signal [22].

The contribution of the ISM depends on the galactic coordinates (longitude l , latitude b) in which the line of sight is oriented.

$$S_{ISM} = \frac{\pi}{4} \cdot \frac{D_{apert}^2}{h \cdot c} \cdot t_{int} \cdot (FOV_{pix})^2 \cdot \int_{\lambda_{min}}^{\lambda_{max}} ISR(\lambda) \cdot ISM(l, b) \cdot \lambda d\lambda \quad \text{Eq. 3-142}$$

where the intensity $ISM(l, b)$ is modeled as described below:

$$ISM(l, b) = \begin{cases} I_v(l) \cdot \exp \exp \left[-\frac{1}{5} \cdot b^2 \right] + a_2 \text{ for } |b| \leq 5^\circ \\ -\frac{a_2}{5} \cdot |b| + 2a_2 \text{ for } 5^\circ < |b| \leq 10^\circ \\ 0 \text{ for } |b| > 10^\circ \end{cases} \quad \text{Eq. 3-143}$$

This shows that the interstellar medium is only considered if the line of sight is less than 10° apart from the galactic plane. The intensity factor $I_v(l)$ considers also the impact of the wavelength λ :

$$I_v(l) = I_{v0}(l) \cdot \left[1 + \left(\frac{\lambda}{60} \right)^2 \right] \quad \text{Eq. 3-144}$$

with

$$I_{v0}(l) = \begin{cases} a_1 \cdot \left[\frac{3}{4(2 + X^2)} + \frac{1}{10} \cos \cos \left(\frac{9}{10} \cdot X + \pi \right) \right] \text{ for } |X| \leq 4 \\ a_1 \cdot \frac{1}{(2 + X^2)} \text{ for } 4 < |X| \leq 20 \end{cases} \quad \text{Eq. 3-145}$$

where $X = X(l) = l/9$ and the galactic latitude has to be $-180^\circ \leq l \leq 180^\circ$. Finally the scaling factors a_1 and a_2 are related to the wavelength λ and can be determined by interpolation between the scaling factor of Table 13, which are given for the IRAS wavelength:

Wavelength λ [μm]	a_1 [MJy/sr]	a_2 [MJy/sr]
3.5	9	0.018
4.9	14	0.028
12	250	0.5
25	210	0.42
60	862	1.72

100	5388	10.78
-----	------	-------

Table 13: Scaling factors for IRAS wavelength

3.1.3.6.7 Background signal of atmospherically scattered light

As all atmospherically related background sources these are required for the **VIS mode only** because infrared sensors are not locatable on ground. A model taken from [08] and also used in [14] is applied:

$$S_{sc.atm.} = \frac{(FOV_{pix})^2 \cdot t_{int} \cdot sint \cdot F_{SA}(\lambda = 0.55\mu m)}{SUN(\lambda = 0.55\mu m)} \cdot \frac{1}{\cos \cos(Z_{LOS})} \quad \text{Eq. 3-146}$$

The flux out of the zenith direction is $F_{SA} = 3.5 \cdot 10^{-11} \left[\frac{W}{cm^2 \cdot \mu m \cdot sr} \right]$ and the impact of the elevation is considered by the *secant* of the zenith angle Z_{LOS} . The basic integral *sint* has to be corrected by the Sun irradiation at $0.55\mu m$ to account for the accurate consideration of the scattered light, which is assumed to have an equivalent spectrum as the Sun and is given for this reference wavelength.

3.1.3.6.8 Background signal due to scattered Sunlight

Also the specific scattering of Sunlight is considered for the **VIS mode only**. In [15] a model is described, which has also been used in [14]:

$$S_{sc.sun.} = \frac{(FOV_{pix})^2 \cdot t_{int} \cdot sint \cdot F_{sc.sun}(Z_{sun}, \lambda = 0.55\mu m)}{SUN(\lambda = 0.55\mu m)} \quad \text{Eq. 3-147}$$

Again the given data for the scattered Sunlight are given for the wavelength $\lambda = 0.55\mu m$ so that the basic integral *sint* has to be corrected. The contribution of scattered Sunlight is only considered shortly after sunset or before sunrise. The zenith angles Z_{sun} of the Sun has to be in the range: $90^\circ \leq Z_{sun} \leq 107.5^\circ$. If the zenith angle is lower the Sun is still above the horizon. For larger zenith distances the impact of scattered Sunlight is neglected.

The scattered sunlight $F_{sc.sun}(Z_{sun}, \lambda = 0.55\mu m)$ is a function of the zenith angle of the Sun. The scattered sunlight data is given for some zenith angles only, so that the used values have to be interpolated. The model does not consider the orientation of the line of sight but give the scattered sunlight contribution for the whole sky.

3.1.3.6.9 Background signal due to scattered Moonlight

Similar as for the scattered sunlight the scattered moonlight is required in **VIS mode only**. Also this model is based on the zenith angle of the light source, which of course is the moon. Also the zenith distance of the line of sight Z_{LOS} and the absolute distance between the azimuth A_{Moon} and the azimuth of the line of sight A_{LOS} are of interest. Furthermore a lunar phase factor $f(\theta)$ is interpolated out of the given data set (see section 3.1.3.1.13) and a fraction of the scattered moonlight (F_{RS} , Rayleigh scattered flux) is derived by a tri-linear interpolation based on Z_{LOS} , ΔA and Z_{Moon} out of the data set introduced in 3.1.3.1.12. The total contribution F_{Moon} of the atmospherically scattered moonlight is computed by:

$$F_{Moon}(\lambda) = f(\theta) \cdot [F_{RS}(\lambda) + F_{MS}(\lambda)] \quad \text{Eq. 3-148}$$

While the Rayleigh scattered contribution is directly interpolated out of the given data set, the Mie-scattered aerosol contribution is computed by the following equation:

$$F_{MS}(\lambda) = a_1 \cdot \left[e^{-\frac{\psi}{\psi_1}} + a_2 \cdot e^{-\frac{180-\psi}{\psi_2}} \right] \cdot F_{RS}(\lambda) \quad \text{Eq. 3-149}$$

where ψ is derived from

$$\cos \cos(\psi) = \cos \cos(Z_{Moon}) \cos \cos(Z_{LOS}) + \sin \sin(Z_{Moon}) \sin \sin(Z_{LOS}) \cos \cos(\Delta A) \quad \text{Eq. 3-150}$$

In accordance with [08] the scaling parameters have been determined as mean values of the given intervals or set to the approximated ones: $a_1 = 75$, $a_2 = 0.015$, $\psi_1 = 15^\circ$ and $\psi_2 = 50^\circ$.

Based on the total contribution $F_{Moon}(\lambda)$, which gives the atmospherically scattered moonlight at $\lambda = 0.55\mu m$ the resulting signal $S_{sc.moon}$ is computed. This is done again with the correction of the basic integral $sint$ assuming that the incoming scattered moonlight has the same spectrum than the Sun:

$$S_{sc.moon} = \frac{(FOV_{pix})^2 \cdot t_{int} \cdot sint \cdot F_{Moon}(\lambda = 0.55\mu m)}{SUN(\lambda = 0.55\mu m)} \quad \text{Eq. 3-151}$$

3.1.3.7 Detection decision

After in chapter 3.1.3.5 the objects signal S_{obj} and in 3.1.3.6 the background signal S_{bkg} as sum of several background sources have been explained, the detection decision based on these values can be performed now.

To determine whether the object can be detected or not it is of interest whether the signal of the object exceeds the background signal – or noise – by a certain threshold. For each pixel (i_{pix}) in the area around the objects trace the detection analysis is made. The pixels are addressed by their coordinates x_{pix}, y_{pix} . Besides the noise of the background sources also internal noise of the system needs to be considered. Thus there are two additional noise terms:

- $k_{dark} [e^-/pix/s]$: By this value the current that flows without light is considered. The smaller the dark current the higher is the sensitivity of the used CCD sensor.
- $k_{read} [e^-/pix]$: By this value the loss of signal due to the process of reading out the charge from a single pixel and converting the stored current into a digital signal.

In [14] an object is detected by a single pixel of the sensor, if

$$S_{obj}(x_{pix}, y_{pix}) > T = S_{bkg}(x_{pix}, y_{pix}) + a \cdot \sigma_{S_{bkg}(x_{pix}, y_{pix})} \quad \text{Eq. 3-152}$$

This means that the object is detected within a pixel if the objects signal is larger than the sum of the background signal and a factor of applied Poisson statistics. The factor a is defined by the user and can be used to adjust the detection threshold T . Typically values of a are in the range of 1.2 to 2.5 as mentioned by [14]. The applied Poisson statistics are:

$$\sigma_{S_{bkg}} = \sqrt{S_{bkg}}$$

where the background signal is the sum of all noise terms :

$$S_{bkg} = S_{bkg.cont.} + S_{bkg}(x_{pix}, y_{pix}) + k_{dark} \cdot t_{int} + k_{read}^2$$

The largest signal to noise ratio SNR_{max} of all affected pixels is return as output parameter in accordance [24] (and [23] as mentioned by [14]):

$$SNR_{max} = \frac{S_{obj}(x_{pix}, y_{pix})}{\sqrt{[S_{obj}(x_{pix}, y_{pix}) + S_{bkg}]}}$$

3.1.4 Detection Analysis - Radar Performance Model

For the simulation of radars a simple performance model has been derived and implemented. The model relies on the basic radar equation and a number of uncertainties the user can define.

The data required to run the sensor is similar to the used in optical performance model:

- The crossing geometry (see Table 2)
- Transmitter parameters (Signal power, wave length, losses, etc.)
- Sensor parameters (Min. received power, losses)
- Object parameters (diameter, state vector)

The results of the sensor model are:

- Detection flag, Status message
- Mean received power at the object from the transmitter
- Mean received power at the sensor from the sensor
- Max. detection range

The power reflected at the object is defined as:

$$P_E = \frac{P_S \cdot G \cdot \sigma}{r_{obj}^2 \cdot 4\pi} \quad \text{Eq. 3-153}$$

Where the P_S is defined as the power that is transmitted from the antenna that has the gain G . That power is the reflected from the body with the radar cross section σ . The body has a distance r_{obj} from the transmitter. Due to the spherical signal propagation signal the power decreases with the square of the distance.

The distance r_{obj} is calculated from the available vector pointing toward the object from the radar:

$$r_{obj} = \sqrt{r_x^2 + r_y^2 + r_z^2} \quad \text{Eq. 3-154}$$

The radar cross section σ is derived from the given diameter d of the object. A spherical body is assumed:

$$\sigma = 1/4 \cdot \pi \cdot d^2 \quad \text{Eq. 3-155}$$

The gain G is based on the antenna area A_A which is of round shape:

$$G = \frac{4 \cdot \pi \cdot A_A \cdot A_{eff}}{\lambda^2} \quad \text{Eq. 3-156}$$

where the wave length λ is considered and an efficiency factor A_{eff} .

In order to determine the power at the receiver, which is assumed to be in the same location as the transmitter, the second part of the travel distance has to be taken into account. The reflected power also decreases with the square of the distance:

$$P = \frac{P_S \cdot G^2 \cdot \lambda^2 \cdot \sigma}{r_{obj}^4 \cdot (4\pi)^3 \cdot L_A \cdot L_F \cdot F_D} \quad \text{Eq. 3-157}$$

which results in a signal power relation that is heavily dependent (distance to the power of 4) on the distance of the object and the square of the wave length λ . Adding to it are the losses within the system, where

- L_A are losses due to the propagation through the atmosphere
- L_F are the losses due to fluctuations during the reflection
- L_D are losses due to attenuation on the transmission and receiving path

In the implementation this relation is resolved to the distance in order to better determine whether a given object at the distance r_{obj} can be detected:

$$R_{max} = \sqrt[4]{\frac{P_S \cdot G^2 \cdot \lambda^2 \cdot \sigma}{P_{Emin} \cdot (4\pi)^3 \cdot L_A \cdot L_F \cdot F_D}} \quad \text{Eq. 3-158}$$

The power P_{Emin} is the minimum signal power that can be detected by the sensor. So in order to decide whether an object can be detected by a given radar setup the radar specific maximum range R_{max} achievable by the radar is compared to the distance of the object r_{obj} from the radar:

$$R_{max} - r_{obj} \geq 0 \rightarrow \text{detection} < 0 \rightarrow \text{no detection} \quad \text{Eq. 3-159}$$

A detection is registered when the object distance is within the maximum range of the radar. In this process it is assumed that a single pulse is used to transmit a signal to and being reflected from the target object. An integration time over multiple pulses is not considered. For each crossing step in the field of view it is checked whether the object can or cannot be detected purely based on the relations stated above. A single return signal based on the reflected power in one of the crossing steps is sufficient to change the state to “detected”.

3.1.5 Special Functionalities

3.1.5.1 Observation Strategy

An observation strategy can be defined for each sensor system. Typically this can be used to rebuild observation campaigns where the sensor points into different orientations. The observation strategy is based on the standard input values. Only those, which are listed in the observation strategy, will be replaced. Thus, it is possible to change the orientation as a function of time. A detailed list of the parameters that can be defined within an observation strategy definition file is given in the Software User Manual ([25]).

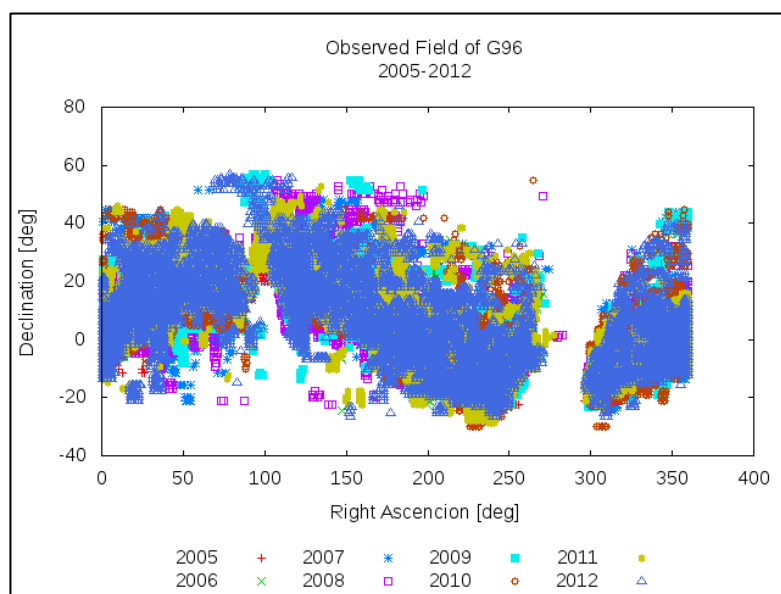


Figure 12: By G96 observed fields for the years 2005 – 2012

Figure 12 shows the orientation of the line of sight of the sensor G96 within the years 2005 to 2012. It is possible by using an observation strategy to perform the observation of all field within a single run of the component observation simulation. Every single dot of the given orientation would represent a single run of the component “Observation Simulation” if no observation strategy would be used.

3.1.5.2 Sky Plots

The option “Sky Plots” makes use of the mode “Observation Strategy” as described in 3.1.5.1. If activated, the software internally generates an observation strategy file by setting up a grid over the sky, where each grid point represents an orientation of the line of sight. In the following a normal execution of the software is performed and a list of crossings and detections is generated. Each entry of this list also contains the orientation of the line of sight, thus it is possible to generate sky plots. The plot generation is performed by the component “Observation Analysis” (see chapter 3.2).

It shall be mentioned that the sky plot option ignores the observation duration as defined by the user. The distribution of objects over the sky is valid only for the start of the observation. Sky plots for ground based sensor will cover an Azimuth range of $0^\circ \leq A < 360^\circ$ and a range in elevation of $0^\circ \leq h \leq 90^\circ$ while for space based sensors the whole sky around the sensor is considered.

3.1.5.3 Generation of Measurements

As an option the user can choose to enable the output of measurement data. In order for this option to be available the deterministic instead of the statistical observation simulation mode has to be chosen. Out of the given number of steps in the field of view measurements can be derived independent from the fact whether a detection occurred or not. The measurement points are equally spaced from the first step in the field of view continuing to the maximum number of measurements specified by the user. A file with the extension *.mes is generated. It contains the ID of the object, the epoch of the measurement, followed by the right ascension and the declination of the object. For radar measurements the range information is given additionally. The coordinates can be chosen to be imprinted with artificial bias, noise and drift errors. These values can be specified in the corresponding error section for the selected sensors. While the bias is a constant offset from the true position, the noise varies around the mean of all measurements. The third error is called the drift. It represents a moving deviation from the true position over time. Figure 13 shows the three errors. The errors are given in radians (RA and DECL) or km for the range.

The drift has been implemented as a function of time:

$$drift = \frac{1}{15} \cdot \left(\frac{1}{3} \sin\left(\frac{t_{init} + t}{7000}\right) - \sin\left(\frac{t_{init} + t}{25450}\right) + \cos\left(\frac{t_{init} + t}{15000}\right) \right) \quad \text{Eq. 3-160}$$

where t_{init} is constant offset and t the time passed from the simulation start in seconds.

The noise is applied using a random number generator, which uses a uniform distribution in the interval of $-\text{noise}/2$ and $\text{noise}/2$.

The three errors are added to each element of the coordinates separately.

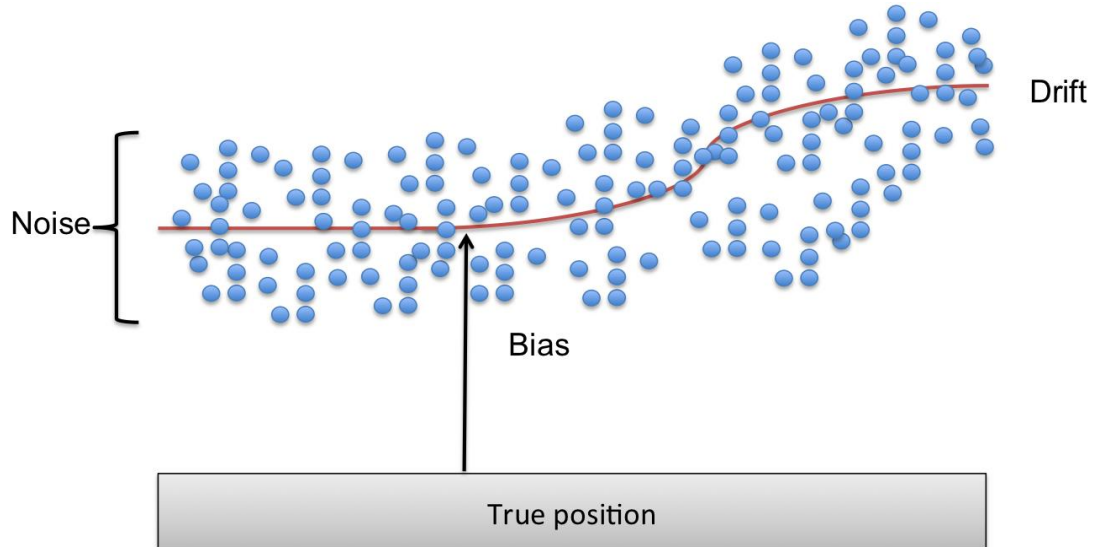


Figure 13: Bias, noise and drift in relation to the true position.

3.2 Observation Analysis

The component “Observation Analysis” follows the “Observation Simulation” and is responsible for the generation of histogram tables and plot driver files, which have to be executed by *gnuplot* to generate a plot. There is no scientific part included in this component as it was the case for the component “Population Analysis” (see section 2.2) because its only task is the visualization of the simulation results. In opposite to the PROOF software [26] the analysis is a separate component in this similar software tool for being able to analyze the results independent from the simulation, which can be very time consuming.

The plots generated by this module are the same as for Population Analysis. However, the 2D plots, by default, show crossings and detections instead of population objects. Sky plots have already been mentioned in section 3.1.5.2 and the solar system plots have green dots for crossings and red ones for detections.

4 Common Routines

This chapter contains some commonly used routines. These routines are not directly connected to the “Population Generator” or the “Observation Simulator” and combined in a separate library similar to the SPICE library.

4.1 From Local Horizon into Topocentric Equatorial

To convert the orientation given in the local horizon system by Azimuth A and Elevation h into Right Ascension α and Declination δ the algorithms described in the following are used.

4.1.1 Ground Based

In addition to the orientation (A, h) also the local sidereal time θ_{LST} and the geodetic latitude Φ_{gd} of the sensor is required for the conversion. The declination is computed first by

$$\delta = ASIN \left(\cos(\Phi_{gd}) \cdot \cos \cos(A) \cdot \cos \cos(h) + \sin \sin(\Phi_{gd}) \cdot \sin(h) \right) \quad \text{Eq. 4-1}$$

For the computation of the right ascension first the values $\sin(\alpha)$ and $\cos(\alpha)$ are computed:

$$\sin \sin(\alpha) = \frac{-\sin \sin(\Phi_{gd}) \cdot \sin \sin(\theta_{LST}) \cdot \cos \cos(A) \cdot \cos \cos(h) + \cos \cos(\delta)}{\cos \cos(\delta)} \quad \text{Eq. 4-2}$$

$$\cos \cos(\alpha) = \frac{-\sin \sin(\Phi_{gd}) \cdot \cos \cos(\theta_{LST}) \cdot \cos \cos(A) \cdot \cos \cos(h) + \cos \cos(\delta)}{\cos(\delta)} \quad \text{Eq. 4-3}$$

The right ascension is then computed to be

$$\alpha = \tan \tan \left(\frac{\sin(\alpha)}{\cos(\alpha)} \right) \quad \text{Eq. 4-4}$$

4.1.2 Space Based

In addition to the orientation (A, h) also the sensor state sen_{J2000} is required for the conversion from the local horizon into the topocentric equatorial reference frame. First the magnitudes of the position and velocity vector given in the sensor state are computed:

$$r = \sqrt{x_{sen}^2 + y_{sen}^2 + z_{sen}^2} \quad \text{Eq. 4-5}$$

$$v = \sqrt{\dot{x}_{sen}^2 + \dot{y}_{sen}^2 + \dot{z}_{sen}^2} \quad \text{Eq. 4-6}$$

An transformation matrix M is used for the conversion. M is of the form

$$M = (M_{11} M_{21} M_{31} M_{12} M_{22} M_{32} M_{13} M_{23} M_{33}) \quad \text{Eq. 4-7}$$

The first column is similar to the axis in flight direction and can be written as

$$(M_{11} M_{12} M_{13}) = \frac{1}{v} \cdot (\dot{x}_{sen} \dot{y}_{sen} \dot{z}_{sen}) \quad \text{Eq. 4-8}$$

The third column – axis perpendicular to the orbital plane – can be computed by a cross product of the velocity and position vector, which is normalized by $|\vec{r} \times \vec{v}|$, and written as:

$$(M_{31} M_{32} M_{33}) = \frac{1}{saux \cdot r \cdot v} \cdot (\dot{y}_{sen} \cdot z_{sen} - y_{sen} \cdot \dot{z}_{sen} \dot{z}_{sen} \cdot x_{sen} - z_{sen} \cdot \dot{x}_{sen} \dot{x}_{sen} \cdot y_{sen} - x_{sen} \cdot \dot{y}_{sen}) \quad \text{Eq. 4-9}$$

with the auxiliary parameter $saux$ as sinus of the angle between \vec{r} and \vec{v} :

$$saux = \sqrt{1 - \left(\frac{x_{sen} \dot{x}_{sen} + y_{sen} \dot{y}_{sen} + z_{sen} \dot{z}_{sen}}{r \cdot v} \right)^2} \quad \text{Eq. 4-10}$$

Finally the second column, which completes the coordinate system as a perpendicular axis to the first and third one, can be written as cross product of both of them:

$$(M_{21} M_{22} M_{23}) = (M_{31} M_{32} M_{33}) \times (M_{11} M_{12} M_{13}) \quad \text{Eq. 4-11}$$

Starting the conversion from the given Azimuth A and Elevation h a vector along this orientation would look like

$$\vec{los}_{local} = (x_{los} y_{los} z_{los}) = (\cos(h) \cdot \cos(A) \sin(h) \cos(h) \cdot \sin(A)) \quad \text{Eq. 4-12}$$

A cross product of the inverse of matrix M and this vector gives than the orientation in J2000:

$$\vec{los}_{J2000} = M^{-1} \times \vec{los}_{local} \quad \text{Eq. 4-13}$$

Now the right ascension α and declination δ can be computed:

$$\alpha = \tan \left(\frac{\vec{los}_{J2000}(2)}{\vec{los}_{J2000}(1)} \right) \quad \text{Eq. 4-14}$$

$$\delta = \text{ATAN} \left(\frac{\vec{los}_{J2000}(3)}{\sqrt{(\vec{los}_{J2000}(1))^2 \cdot (\vec{los}_{J2000}(2))^2}} \right) \quad \text{Eq. 4-15}$$

4.2 From Topocentric Equatorial into Local Horizon

The inverse conversion as described in 4.1 shall be explained in this chapter. Also here two different algorithms are necessary to consider the conversion of right ascension α and declination δ into Azimuth A and Elevation h for both a ground based and a space based origin.

4.2.1 Ground Based

The unit vector in the topocentric equatorial reference frame along the given orientation is:

$$\vec{los}_{J2000} = (\cos(\alpha) \cdot \cos(\delta) \sin(\alpha) \cdot \cos(\delta) \sin(\delta)) \quad \text{Eq. 4-16}$$

The conversion into the local horizon reference frame is made with the help of a transformation matrix M , containing also the geodetic latitude Φ_{gd} and the local sidereal time θ_{LST} :

$$M = \begin{pmatrix} \sin(\Phi_{gd}) \cdot \cos(\theta_{LST}) \sin(\Phi_{gd}) \cdot \sin(\theta_{LST}) - \cos(\Phi_{gd}) \\ -\sin(\theta_{LST}) \cos(\theta_{LST}) \ 0 \ \cos(\Phi_{gd}) \\ \cdot \cos(\theta_{LST}) \cos(\Phi_{gd}) \cdot \sin(\theta_{LST}) \sin(\Phi_{gd}) \end{pmatrix} \quad \text{Eq. 4-17}$$

$$\vec{los}_{local} = M \cdot \vec{los}_{J2000} \quad \text{Eq. 4-18}$$

Finally the Azimuth A and Elevation h can be computed:

$$A = \text{atan} \left(\frac{\vec{los}_{local}(2)}{-\vec{los}_{local}(1)} \right) \quad \text{Eq. 4-19}$$

$$h = \text{asin}(\vec{los}_{local}(3)) \quad \text{Eq. 4-20}$$

4.2.2 Space Based

To compute for a space based sensor system the azimuth A and the elevation h out of the given right ascension α and declination δ also some orbital elements are required. First a unit vector along the given orientation is computed:

$$\vec{los}_{J2000} = (\cos(\alpha) \cdot \cos(\delta) \sin(\alpha) \cdot \cos(\delta) \sin(\delta)) \quad \text{Eq. 4-21}$$

With the help of the following transformation matrix M , which considers the orbital elements inclination i , right ascension of the ascending Ω (respectively ecl. longitude of the ascending node), argument of pericenter ω and true anomaly ν (with $u = \omega + \nu$):

$$M = (\vec{M}_1^T \ \vec{M}_2^T \ \vec{M}_3^T) \quad \text{Eq. 4-22}$$

with

$$\vec{M}_1 = (\cos(u) \cdot \cos(\Omega) - \sin(u) \cdot \cos(i) \cdot \sin(\Omega) \cos(u) \cdot \sin(\Omega) + \sin(u) \cdot \cos(i) \cdot \cos(\Omega) \sin(u) \cdot \sin(i)) \quad \text{Eq. 4-23}$$

and

$$\vec{M}_2 = (-\sin(u) \cdot \cos(\Omega) - \cos(u) \cdot \cos(i) \cdot \sin(\Omega) - \sin(u) \cdot \sin(\Omega) + \cos(u) \cdot \cos(i) \cdot \cos(\Omega) \cos(u) \cdot \sin(i)) \quad \text{Eq. 4-24}$$

and

$$\vec{M}_3 = (\sin(i) \cdot \sin(\Omega) - \sin(i) \cdot \cos(\Omega) \cos(i)) \quad \text{Eq. 4-25}$$

so that finally the orientation in the local horizon system can be computed as

$$\vec{los}_{local} = M \cdot \vec{los}_{J2000} \quad \text{Eq. 4-26}$$

to extract azimuth A and elevation h :

$$A = \text{atan} \left(\frac{\vec{los}_{local}(3)}{\vec{los}_{local}(2)} \right) \quad \text{Eq. 4-27}$$

$$h = \text{asin} (\vec{los}_{local}(1)) \quad \text{Eq. 4-28}$$

4.3 Tropospheric Refraction

The model of the tropospheric refraction from [07] and used within the component “Observation Simulation” shall be presented here. Only the refraction of the elevation is considered here in the form:

$$h_{ref} = h - \Delta h$$

where h_{ref} is the corrected elevation, h is the elevation without considering the tropospheric refraction and Δh is the impact of the refraction. The latter is determined by

$$\Delta h = \frac{N}{\tan \tan h} \cdot 10^{-6} \quad [rad]$$

where N is the refractivity of the troposphere. It is the sum of the two components N_1 as dry component and N_2 as wet component:

$$N_1 = 77.624 \cdot \frac{p[hPa]}{T[K]}$$

with the atmospheric pressure p in [hPa] and the local ambient temperature T in [K]

$$N_2 = 371900 \cdot \frac{e[hPa]}{(T[K])^2} - 12.92 \cdot \frac{e[hPa]}{T[K]}$$

with the partial pressure of water vapor e in [hPa] determined as function of the relative humidity $0 \leq f_h \leq 1$:

$$e = 6.10 \cdot f_h \cdot \exp \exp \left(\frac{17.15 \cdot T[^\circ C]}{234.5 + T[^\circ C]} \right)$$

4.4 Planck's law

The Planck law is used to determine the spectral radiance of a body for a given wavelength λ and at a given temperature T of the body. Alternatively the frequency ν can be used instead of the wavelength λ .

Spectral radiance $B_\lambda(T)$ for a given wavelength λ :

$$B_\lambda(T) = \frac{2 \cdot h \cdot c^2}{\lambda^5} \cdot \frac{1}{e^{\frac{h \cdot c}{\lambda \cdot k_B \cdot T}} - 1} \quad \left[\frac{W}{m^2 \cdot m \cdot sr} \right]$$

Spectral radiance $B_\nu(T)$ for a given frequency ν :

$$B_\nu(T) = \frac{2 \cdot h \cdot \nu^3}{c^2} \cdot \frac{1}{e^{\frac{h \cdot \nu}{k_B \cdot T}} - 1} \quad \left[\frac{W \cdot s}{m^2 \cdot sr} \right]$$

4.5 Tri-linear interpolation

The tri-linear interpolation is here explained shortly for the use within the determination of the atmospherically scattered moonlight. The cube shown in Figure 14 is built by the given data points (C_{xyz}). By interpolating between those points the data of the auxiliary points C_{xy} is determined. A second interpolation gives the data at the points C_x and finally the third interpolation gives the requested data at the point C .

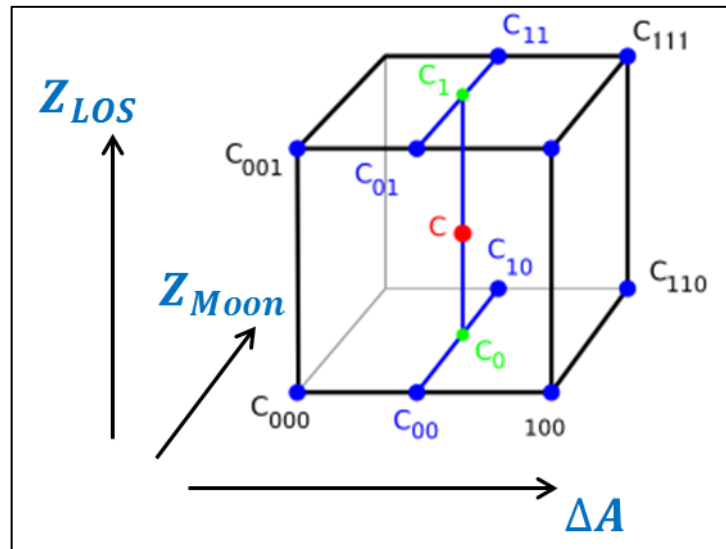


Figure 14: Cube for the tri-linear interpolation

The computation of these three interpolation steps can be done in a single step to obtain the data $V(x, y, z)$ at the point C :

$$\begin{aligned}
 V(x, y, z) = & V(0,0,0) \cdot (1 - x) \cdot (1 - y) \cdot (1 - z) + V(1,0,0) \cdot x \cdot (1 - y) \cdot (1 - z) \\
 & + V(0,1,0) \cdot (1 - x) \cdot y \cdot (1 - z) + V(0,0,1) \cdot (1 - x) \cdot (1 - y) \cdot z \\
 & + V(1,0,1) \cdot x \cdot (1 - y) \cdot z + V(0,1,1) \cdot (1 - x) \cdot y \cdot z + V(1,1,0) \cdot x \\
 & \cdot y \cdot (1 - z) + V(1,1,1) \cdot x \cdot y \cdot z
 \end{aligned}$$

This approach is based on a unit cube (edge length equal one). The requested point C thus has to be given in coordinates for which $0 \leq x + y + z \leq 3$ needs to be valid.

5 Further Improvements

NEOPOP already offers an extensive amount of functionality. Nevertheless, software is never completely “done” which is why some ideas for improvements always exist.

The radar model used, for example, is a simple one based on the basic radar equation. An idea for a more advanced model would be to take into account the coherent integration of radar pulses. This is currently an important field of research and a radar model based on this idea would enable NEOPOP users to simulate radar sensors specifically adjusted for NEO detection. And, since NEOPOP is built modular, the new radar model could relatively easily be integrated into the software. A follow-up project to SGNEOP addressing this matter is already in preparation.

Parallelization would be another point. NEOPOP in general and the Observation Simulation’s Detection Analysis are already prepared to be parallelized. However, NEOPOP makes use of the FORTRAN toolkit named SPICE which has not been implemented with parallelization in mind. One possibility to solve this issue would be to replace SPICE routines with something similar which would be easier to adjust for parallel computing. However, using SPICE has proven to be valuable as, for example, planetary ephemeris data can easily be used. The second possibility would be to adjust SPICE. It has become clear during the project though that an adjustment of SPICE would not fit into the project’s boundaries.

Finally, observation simulations with long observation durations also could be a point of interest for further work. The SGNEOP Requirements Baseline states that the software should allow observations of up to 10 years. Nevertheless, this limit has been raised to 68 years, but these long simulations have not been tested thoroughly. And it should be investigated if even longer observation durations are wanted by users.

6 Help method & suggestions

In general the user should be able to solve any problem that might occur during the operation of the NEOPOP software with the help of

- this the related final report and the software user manual [25]
- the GNUPLOT manual [01] (*<installdir>/docu/gnuplot.pdf*)
- the warning and error messages generated by the software tool.

In Figure 15 the pop-up window as displayed to the user during the execution of NEOPOP is shown. Warning messages will appear in the logfile (*neopop.log*) while errors will be written to the dedicated error file (*neopop.err*). This will only be generated if an error occurred during execution.

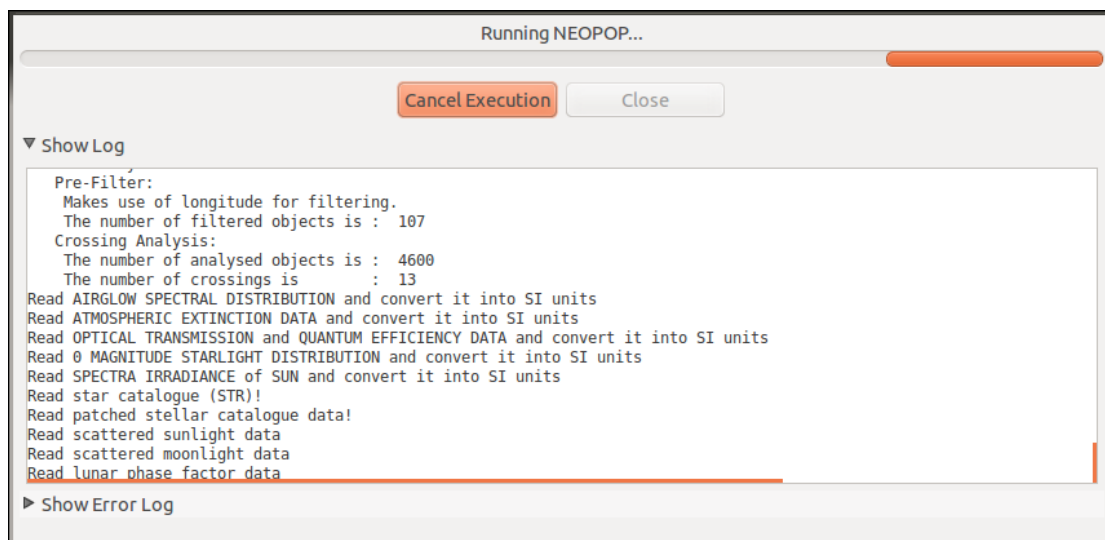


Figure 15: Progress bar when running NEOPOP.

6.1 Feedback and Bug-Report

You can contribute to further improvements of NEOPOP by giving feedback on errors, and by making suggestions for enhancements. Please consultate the following webpage:

<http://neo.ssa.esa.int/tools>

7 References

- [01] T. Williams, C. Kelley, et.al., “*GNU PLOT - An Interactive Plotting Program*”, Version 4.4.4, November 12th, 2011 (Source: http://www.gnuplot.info/docs_4.4/gnuplot.pdf, 09.05.2013)
- [02] C.H. Acton, “*Ancillary Data Services of NASA's Navigation and Ancillary Information Facility*”, Planetary and Space Science, Vol. 44, No. 1, pp. 65-70, 1996
- [03] SPICE toolkit online documentation for different programming languages (Source: <http://naif.jpl.nasa.gov/naif/documentation.html>, March 8th, 2014)
- [04] SPICE generic kernels, (Source: http://naif.jpl.nasa.gov/pub/naif/generic_kernels/, March 8th, 2014)
- [05] The Catalina Sky Survey, (Source <http://www.lpl.arizona.edu/css/index.html>, March 8th, 2014)
- [06] D. A. Vallado, “*Fundamentals of Astrodynamics and Applications*”, Third Edition, Space Technology Library, Microcosm Press (CA) & Springer (NY) , 2007
- [07] O. Montenbruck, E. Gill, “*Satellite Orbits -Models, Methods and Applications*”, Corrected second printing, Springer 2001
- [08] G.M. Daniels, “*A night sky model for satellite search systems*”, Optical Engineering, February 16th, 1977, pages 66-71
- [09] MBB, IFRR TUBS, JENOPTIK, IFA BERLIN, “*Optische Debris – Detektion*”, Durchführbarkeits-Studie 1991, Endbericht, März 1992
- [10] G. Neugebauer, et.al., “*IRAS Explanatory Supplement VI. Flux Reconstruction and Calibration*”, (Source: <http://irsa.ipac.caltech.edu/IRASdocs/exp.sup/ch6/C2a.html>, March 14th, 2014)
- [11] A. Franceschini, et.al., “*Galaxy counts and contributions to the background radiation from 1 μ m to 1000 μ m*”, Astron. Astrophys. Suppl Ser. 89, 285-310 (1991)
- [12] J.S. Mathis, “*Interstellar Dust and Extinction*”, *Annual Review of Astronomy and Astrophysics*, Vol. 28, Palo Alto, California, 1990, pp.37-70
- [13] R.J. Wainscoat, et.al., “*A model of the 8-25 micron point source infrared sky*”, The Astrophysical Journal Supplement Series, 83, November 1992, pp. 111-146
- [14] H. Krag, “*A method for the validation of space debris models and for the analysis and planning of radar and optical surveys*”, ZLR-Forschungsbericht 2003-05, Shaker Verlag Aachen 2003
- [15] D.R. Lobb, “*Study on optical sensors for space debris observation*”, Volume 1 and Volume 2, Final Report, SIRA Research Development Study Report, November 1992
- [16] I.N. Belskaya, V.G. Shevchenko, “*Opposition Effect of Asteroids*”, Icarus 147, pp.94-105, 2000

-
- [17] V.G. Shevchenko, “*Analysis of the asteroid phase dependences of brightness*”, Lunar Planet Sci. XXVII,1086
- [18] A.W. Harris, J.S.V. Lagerros, “*Asteroids III - Asteroids in the thermal infrared*”, The University of Arizona Press, pp. 205–218, 2002
- [19] Nautical Almanac Office, U.S. Naval Observatory, “*Explanatory Supplement to the Astronomical Almanac*”, University Science Books, Mill Valley California, ISBN 0-935702-68-7
- [20] K.R. Lang, “*Astrophysical Data: Planets and Stars*”, Springer New York Inc., 1992
- [21] T. Kelsall, J.L. Weiland, et.al., „The COBE diffuse infrared background experiment search for the cosmic infrared background, II. Model of the interplanetary dust cloud“, The Astrophysical Journal, 508, November 22th, 1998, pp. 44-73
- [22] A. Kopp, “*Infrared Detection of Space Debris*”, Diploma Thesis, Institute of Aerospace Systems, TU Braunschweig, Germany, May 2008
- [23] D. R. Lobb, “*Study on Optical Sensors for Space Debris Observation*”, Final Report, ESTEC Contract No. 9267/90/D/MD 1992
- [24] T. J. Fellers and M. W. Davidson “*Concepts in Digital Imaging Technology – CCD Noise Sources and Signal-to-Noise Ratio*”, National High Magnetic Field Laboratory, The Florida State University, Tallahassee, Florida (Source: <http://micro.magnet.fsu.edu/primer/digitalimaging/concepts/ccdsnr.html>, March 21st, 2014)
- [25] J. Gelhaus, “*Software User Manual of the Near Earth Objects Population Observation Program*”, ESA Project AO/1-7015/11/NL/LvH, April 2012
- [26] J. Gelhaus, “*Final Report – Program for Radar and Optical Observation Forecasting*”, Institute of Aerospace Systems, TU Braunschweig, Germany, May 2011, ESA Contract 21705/08/D/HK
- [27] W.F. Bottke, A. Morbidelli, R. Jedicke, J.-M. Petit, H.F. Levison, P. Michel, T.S. Metcalfe, “*Debiased Orbital and Absolute Magnitude Distribution of the Near-Earth Objects*”, Icarus 156 (399-433), 2002
- [28] Homepage of the Minor Planet Center orbital database, www.minorplanetcenter.net/iau/MPCORB.html
- [29] Homepage of the NEODys database, <http://newton.dm.unipi.it/neodys/>
- [30] Pan-STARRS Data Exchange Standard (DES 2.02) PSDC-MOPS-DES-2.02, http://adams.dm.unipi.it/orbfit/PS_DES_v2.02.pdf
- [31] Brown, P., Spalding, R.E., ReVelle, D.O, Tagliaferri, E., Worden, S.P.\ 2002.\ The flux of small near-Earth objects colliding with the Earth.\ Nature 420, 294-296.

-
- [32] J. Gelhaus, J. Radtke, G. Hahn and S. Müller, “Design Justification File - Synthetic Generation of a NEO Population”, Institute of Aerospace Systems, TU Braunschweig, Germany, 2014, ESA Contract AO/1-7015/11/NL/LvH