Software User Manual

NEOPOP

Near Earth Object Population Observation Program

Project title:	**Synthetic Generation of a NEO Population**
Revision:	2.0
Document ID:	SGNEOP_SUM
Date:	September 07, 2022
ESA/ESTEC Study Manager:	
Prepared by:	Johannes Gelhaus (IRAS) Sven Müller (IRAS) André Horstmann (IRAS) Philipp Maier (ESTEC)
Main-Contractor:	Technische Universität Braunschweig (TUBS) Institute of Space Systems (IRAS) Hermann-Blenk-Str. 23 38108 Brauschweig 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.

ESA's Near Earth Object Population Observation Program (NEOPOP) was developed by the Institute of Space 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:

IRAS/TUBS
Johannes Gelhaus
Sven Müller
André Horstmann

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. Prague)
Ed Beshore (LPL, Univ. of Arizona)

History of Changes:

Date	Version	Author	Change Description
13.01.2014	0.1	J. Gelhaus	● First nearly complete version of the SUM
23.03.2014	0.2	S. Müller	● Complete update of the document
21.05.2014	1.0	S. Müller	● Release for acceptance (AR2) ● Explaining use of Run-ID ● Removed part about number of integrations for radars, since the respective variable has been removed from software. Cp. RID 27. ● Processed Review Item Descriptions (RIDs) 16-18, 21, 23, 26, 40, 49-51, 66, 70, 72-77, 79
20.06.2014	1.1	S. Müller	● Added chapter about configuration and input file parameters ● Improved status flag entry in Observation Simulation results output file description table (cp. RID 16). ● Added reference to Final Report for details about new model extrapolation function (cp. RID 17). ● Corrected month in Final Report reference ● Corrected description of input parameter “Transmitted Power” (Radar Sensors) (cp. RID 29). ● Corrected that the radar model contains a continuous radar, not a pulsed one (cp. RID 30). ● Extended description of gnuplot usage. Explained where the plot data can be found (cp. RID 62). ● Added another note that the old NEO model should be used with care. ● Changed naming of the old NEO model consistently to something like “old NEO model by Bottke” vs. “NEO model by Granvik & Morbidelli” ● Added note about Observation Analysis 3D plots plotting only detections ● Added description of plot lines (cp. RID 87).
11.11.2014	1.2	S. Müller	● Described the usage of sources (cp. RID 35)

Date	Version	Author	Change Description
			● Clarified that source regions and source region alphas of the old NEO model are only accessible via CLT input file (cp. RID 48). ● Updated GUI screenshots (cp. RID 53). ● Clarified that skyplots show the situation at observation start (cp. RID 88). ● Added note about MPC epoch format (cp. RID 89). ● Added listing of possible (abbreviated) values of GRP attribute in PPF (existed already for SRC, cp. RID 92). ● Mentioned where the NEOPOP CLT can be found (cp. RID 94). ● Better explained how radar loss values are understood (cp. RID 98). ● Clarified that min. H-value is indeed used as filter for old NEO model (cp. RID 100). ● Noted that population file header lines must always begin with "!" (cp. RID 101). ● Explained Solar System Plots (cp. RID 57). ● Added "Reported Problems" section (cp. RID 102). ● Created a chapter dedicated to the CLT and explained input and output files as well as their formats (cp. RIDs 59 and 61). ● Reworked Operations Environment chapter (cp. RID 99).
04.12.2014	1.3	S. Müller P. Maier	● Version for third acceptance ● Created tutorial (cp. RID 56) ● Minor adjustments ● Added description of workspace reset ● Mentioned that the propagator used is a simple two-body one named SPICE PROP2B ● Updated all configuration, input and output file description tables
08.12.2014	1.4	S. Müller	● Version for third acceptance ● Accepted all tracked changes in Word document
23.03.2015	1.5	S. Müller C. Kebschull, P. Maier	● Noted that population file must have a defined number of header lines in order to be readable.

Date	Version	Author	Change Description
			● Added “Crash on Project Load” section to a new “Troubleshooting” section in “Help Method & Suggestions” chapter. (cp. SPR 30) ● Space-based location settings description: Noted that sun-synchronous orbits can be defined (cp. RID 33). ● Population Analysis module description: Added table for explanation of Solar System Plot orientation (cp. RID 104). ● Activated repetition of table caption row (in case tables extend over more than one page, cp. RID 71). ● Updated screenshots (cp. RID 54). ● Made several minor corrections. ● Added explanation of Java-check and Java-virtual-machine-architecture-selection steps of installer ● Added some minor changes by Philipp Maier ● Overworked help chapter o Removed “Known Problems” that have been solved o Merged “Known Problems” and “Troubleshooting” sections o Merged general chapter text and feedback section ● Rewrote troubleshooting section about reading external population files to reflect new comment line convention. ● Mentioned Population Analysis epoch limits and the reason for that in section about Population Analysis module ● Suggested deactivating the Pre-Filter for big FOVs in sections about Optical and Radar Settings (cp. SPR 258). ● Updated model name to Granvik, Morbidelli, Bottke and collaborators in two occasions ● Some minor typos & missing internal references

Date	Version	Author	Change Description
			● Added warning that NEOPOP will accept osd-files also if the format of the ASCII file is different than required ● Took out the reference to a template for the submission of errors ● Added table with andomization intervals for MC runs.
07.05.2019	1.6	A. Horstmann	● Adding statement when upgrading NEOPOP from a previous version to section 1.2
04.07.2019	1.7	A. Horstmann	● Indicating Java 1.8 as a requirement
15.10.2019	1.8	A. Horstmann	● Fixing Java version
29.06.2021	1.9	G. Principe	● Removed 32-bit and 64-bit specifications in both Processor and OS Architecture Requirements as too generic. Only x86 and x86-64 to represent the requirement
01.04.2022	1.10	G. Principe	● Updated to version 1.5 of the software
05.07.2022	2.0	A. De Riz	● Updated OSD file example ● Improved explanation of OSD file creation ● Added TCA and TCA range definition ● Improved information about “Flag for coordinate system” ● Correction of typographical errors

Table of Contents

1 Introduction 13
1.1 Scope of this Document 13
1.2 Terms and Definitions 13
1.3 Definitions of Coordinate Systems 17
1.4 Abbreviations 18
2 Operations Environment 20
2.1 System Specification 20
2.2 List of Tools and Libraries 20
3 Installation 22
4 External View of the Software 23
4.1 Installation Folder 23
4.1.1 Data Folders 23
4.2 GUI Configuration Folders 24
5 Tutorial 25
5.1 Tutorial Case 25
5.2 NEO Population Generation and Analysis 25
5.3 Observation Simulation and Analysis 32
6 Basic Usage 37
6.1 Starting NEOPOP 37
6.2 Workspaces, Projects and Runs 37
6.3 GUI Overview and Perspectives 39
6.4 Input Perspective 39
6.5 Output Perspective 40
6.6 Main Toolbar 41
6.6.1 “New” Button 42
6.6.2 “Open” Button 43
6.6.3 “Save” and “Save As” Buttons 43
6.6.4 “PDF” Button 44
6.6.5 “Export” Button 44
6.6.6 “Import” Button 45
6.6.7 “Run” Button 46
6.6.8 “Replot All” Button 47

6.6.9 “Reset” Button 47
6.6.10 “Remove” Button 48
6.6.11 “Settings” Button 48
6.6.12 “Help”, “Plot Help”, “About” and “Quit” Buttons 50
6.6.13 “Switch Perspective” Button 50
6.7 gnuplot Toolbar 51
6.7.1 “Replot Single” Button 51
6.7.2 “Undo gnuplot File” Button 51
6.8 Working with gnuplot 51
6.8.1 The Basics of gnuplot 51
6.8.2 Combination of Results from Different Projects 53
6.9 “Browse” and “Edit” Buttons 54
6.10 Model File Update 54
6.11 Error Conditions & Recover Runs 55
7 Applications and Modules 56
7.1 “Population Generator” Application 56
7.1.1 “Population Generation” Module 56
7.1.1.1 Generating a Synthetic NEO Population 57
7.1.1.2 Generating a Fictitious NEO Population 59
7.1.1.3 Generate a “Physical Properties File“ only 59
7.1.1.4 Results of This Module 59
7.1.2 “Population Analysis” Module 62
7.2 “Observation Simulator” Application 67
7.2.1 “Observation Simulation” Module 67
7.2.1.1 Basic Settings 67
7.2.1.2 Population Settings 70
7.2.1.3 Sensor System Settings 71
7.2.1.4 Ground-based Sensor Location Settings 74
7.2.1.5 Space-based Sensor Location Settings 76
7.2.1.6 Optical Sensor Settings 78
7.2.1.7 Radar Sensor Settings 82
7.2.1.8 Output Files 84
7.2.1.8.1 The OBSSIM results output file (*.res) 84

7.2.1.8.2 The OBSSIM MC summary output file (*_MC00.res) 86
7.2.1.8.3 The OBSSIM summary output file (*.sum) 86
7.2.2 “Observation Analysis” Module 86
8 Using the Command-Line Tool 89
8.1 Project Folder Content 89
8.2 Output Folder Content 92
8.3 Executing gnuplot 95
9 Configuration and Input Files Specification 96
9.1 neopop.cfg 96
9.2 obssim.cfg 96
9.3 obssim.g_b 97
9.4 obssim.inp 98
9.5 obssim.net 98
9.6 obssim.opt 99
9.7 obssim.plt 100
9.8 obssim.rad 100
9.9 obssim.s_b 101
9.10 obssim.src 102
9.11 popgen.cfg 103
9.12 popgen.fil 104
9.13 popgen.inp 106
9.14 popgen.plt 108
10 Troubleshooting and Feedback 109
10.1 Crash on Project Load 110
10.2 Invalid Value Error on Software Run 110
10.3 Problems with External Population Files 110
10.4 GUI does not Start 110
10.5 Forgetting about Global Options 111
11 References 112

List of Tables

1 Introduction 13

1.1 Scope of this Document 13
1.2 Terms and Definitions 13
1.3 Definitions of Coordinate Systems 17
1.4 Abbreviations 18
2 Operations Environment 20
2.1 System Specification 20
2.2 List of Tools and Libraries 20
3 Installation 22
4 External View of the Software 23
4.1 Installation Folder 23
4.1.1 Data Folders 23
4.2 GUI Configuration Folders 24
5 Tutorial 25
5.1 Tutorial Case 25
5.2 NEO Population Generation and Analysis 25
5.3 Observation Simulation and Analysis 32
6 Basic Usage 37
6.1 Starting NEOPOP 37
6.2 Workspaces, Projects and Runs 37
6.3 GUI Overview and Perspectives 39
6.4 Input Perspective 39
6.5 Output Perspective 40
6.6 Main Toolbar 41
6.6.1 “New” Button 42
6.6.2 “Open” Button 43
6.6.3 “Save” and “Save As” Buttons 43
6.6.4 “PDF” Button 44
6.6.5 “Export” Button 44
6.6.6 “Import” Button 45
6.6.7 “Run” Button 46
6.6.8 “Replot All” Button 47
6.6.9 “Reset” Button 47
6.6.10 “Remove” Button 48

6.6.11 “Settings” Button 48
6.6.12 “Help”, “Plot Help”, “About” and “Quit” Buttons 50
6.6.13 “Switch Perspective” Button 50
6.7 gnuplot Toolbar 51
6.7.1 “Replot Single” Button 51
6.7.2 “Undo gnuplot File” Button 51
6.8 Working with gnuplot 51
6.8.1 The Basics of gnuplot 51
6.8.2 Combination of Results from Different Projects 53
6.9 “Browse” and “Edit” Buttons 54
6.10 Model File Update 54
6.11 Error Conditions & Recover Runs 55
7 Applications and Modules 56
7.1 “Population Generator” Application 56
7.1.1 “Population Generation” Module 56
7.1.1.1 Generating a Synthetic NEO Population 57
7.1.1.2 Generating a Fictitious NEO Population 59
7.1.1.3 Generate a “Physical Properties File“ only 59
7.1.1.4 Results of This Module 59
7.1.2 “Population Analysis” Module 62
7.2 “Observation Simulator” Application 67
7.2.1 “Observation Simulation” Module 67
7.2.1.1 Basic Settings 67
7.2.1.2 Population Settings 70
7.2.1.3 Sensor System Settings 71
7.2.1.4 Ground-based Sensor Location Settings 74
7.2.1.5 Space-based Sensor Location Settings 76
7.2.1.6 Optical Sensor Settings 78
7.2.1.7 Radar Sensor Settings 82
7.2.1.8 Output Files 84
7.2.1.8.1 The OBSSIM results output file (*.res) 84
7.2.1.8.2 The OBSSIM MC summary output file (*_MC00.res) 86
7.2.1.8.3 The OBSSIM summary output file (*.sum) 86

7.2.2 “Observation Analysis” Module 86
8 Using the Command-Line Tool 89
8.1 Project Folder Content 89
8.2 Output Folder Content 92
8.3 Executing gnuplot 95
9 Configuration and Input Files Specification 96
9.1 neopop.cfg 96
9.2 obssim.cfg 96
9.3 obssim.g_b 97
9.4 obssim.inp 98
9.5 obssim.net 98
9.6 obssim.opt 99
9.7 obssim.plt 100
9.8 obssim.rad 100
9.9 obssim.s_b 101
9.10 obssim.src 102
9.11 popgen.cfg 103
9.12 popgen.fil 104
9.13 popgen.inp 106
9.14 popgen.plt 108
10 Troubleshooting and Feedback 109
10.1 Crash on Project Load 110
10.2 Invalid Value Error on Software Run 110
10.3 Problems with External Population Files 110
10.4 GUI does not Start 110
10.5 Forgetting about Global Options 111
11 References 112

1 Introduction

1.1 Scope of this Document

The scope of this document is the description of the software NEOPOP, the “Near Earth Object Population Observation Program”. NEOPOP consists of the main tool in the form of a console program and a GUI using this tool. The document you are reading is meant mainly for GUI users, but is worth reading for console users, too.

This document neither explains the architecture of NEOPOP nor the algorithms used. For such detailed background information the user should consult the SGNEOP Final Report [2].

1.2 Terms and Definitions

The following terms are used within the documentation and shall be explained shortly as a reference index. You may skip this part when reading for the first time.

Application
One of the two tool parts which are the Population Generator and the Observation Simulator

Data File
File used by NEOPOP that normally doesn’t change and thus doesn’t need to be touched by the user. Data files normally reside in the `data` sub-subfolders of `<installation folder>/default`.

gnuplot
Plotting program used by NEOPOP

gnuplot Driver File
Files that define plots in a manner that is understood by gnuplot so that it can create plot pictures based on their content

gnuplot Toolbar
Toolbar that is shown in the Output Perspective above the gnuplot driver file content when a plot is selected

Ground(-based) (Sensor) Location
Special type of a sensor location with a fixed place on the ground of the Earth

GUI
See “NEOPOP GUI”

Input
See “Tool Input”

Input File
File used by the tool to determine what to do. Input files are part of the tool's input and normally reside in the `input` sub-sub folders of projects. Most of the input files contain settings.

Input Perspective
Perspective through which the tool's input can be changed

Location
See "Sensor Location"

Main Toolbar
Toolbar located at the top of the GUI. It provides all the main functions like running the tool

Module
One of the four parts of the tool's applications, namely the Population Generation, the Population Analysis, the Observation Simulation and the Observation Analysis

NEO
Near-Earth Object, objects whose orbits come close to Earth's orbit

NEOPOP
NEO Population Observation Program, the software this manual is about. It consists of the tool and the GUI.

NEOPOP GUI
NEOPOP Graphical User Interface. It is one of the two parts of NEOPOP; a program providing windows, dialogs, buttons, text fields etc. for user-intuitive usage of the NEOPOP tool.

NEOPOP Tool
NEOPOP console program. It is the main part of NEOPOP and consists of the Population Generator and the Observation Simulator.

Network
See "Sensor Network"

Observation Analysis
One of the two modules of the Observation Simulator. It creates gnuplot driver files from Observation Simulation output and certain settings.

Observation Simulation
One of the two modules of the Observation Simulator. It simulates the observation of NEOs by a sensor network.

Observation Simulator

One of the two applications of the tool. It consists of the modules Observation Simulation and Observation Analysis.

Optical Sensor
A special type of sensor that catches visible light to observe NEOs

Output
See “Tool Output”

Output Category
The tool’s output is divided into different categories depending on the plot settings used to create the output. A special category type is the summary category. All categories of a run can be accessed through the right sidebar of the GUI’s output perspective.

Output Perspective
One of the two perspectives. The output perspective allows GUI users to view and adjust the tool’s output.

Output Subject
The Observation Analysis creates output that can not only be divided into categories, but also subjects. Subjects are the sensor network and the sensor systems.

Perspective
The GUI yields two different “views”: input and output perspective. Users can switch between them by using the “Switch Perspective” button.

Population Analysis
One of the two modules belonging to the Population Generator. It creates gnuplot driver files from Population Generation output.

Population Generation
One of the two modules belonging to the Population Generator. It creates NEO population files.

Population Generator
One of the two applications of the tool. It consists of the modules Population Generation and Population Analysis.

Project
A project (folder) contains input and output files. The idea is to create a new project for each topic that NEOPOP is used for.

Radar Sensor

A special type of sensor transmitting and receiving radio waves to observe NEOs

Run
The process of starting the tool in the context of a project to create output from input. A project may contain only one input, but output of different runs identified by their respective run id.

Run ID
Identify runs. Output files are prefixed with it. After changing the Run ID inside the GUI you can get access to the output of former runs for example.

Sensor
Observes NEOs as part of a sensor system

Sensor Location
Location of a sensor which can be a ground- or space-based location

Sensor Network
Summarizes all sensor systems

Sensor System
A sensor system is a sensor at a certain sensor location. It can be bi-static meaning that it consists of two sensors at two locations.

Settings
A part of the tool's input that defines what it should do when launched

Sidebar
A sidebar resides at the GUI's left or right side. There exist the settings sidebar (in the input perspective), the output subject and output category sidebars (in the output perspective)

Space(-based) (Sensor) Location
A special type of sensor location, an orbit for example

SPICE
Software collection used by the tool

System
See "Sensor System"

Tool
See "NEOPOP Tool"

Tool Input

Everything the tool needs in order to successfully create output, namely input files and data files

Tool Output
When launched the tool creates output that normally can be found in the "output" sub-subfolders of projects.

Toolbar

A bar containing buttons. The GUI has two toolbars: the main toolbar and the gnuplot toolbar

Workspace
A workspace is a folder which has to be defined by the user during the first launch of the GUI. It is meant to contain the user's projects.

When updating from a previous NEOPOP version, please ensure a clean workspace environment. In particular, please make sure to not reuse the workspace folder of a previous NEOPOP version, as they are not compatible and will cause NEOPOP to fail. An empty workspace folder has to be used in order to run NEOPOP.

1.3 Definitions of Coordinate Systems

The following table provides the definitions of coordinate systems used in NEOPOP.

For positions:

Coord. Sys.	Center	XY-plane	+X direction	+Z direction
Heliocentric space-based location	Sun	Ecliptic	Vernal equinox	Ecliptic north
Geocentric space-based location	Earth	Earth equator at J2000 epoch	Vernal equinox	Geographic Earth North

For orientations from ground-based locations:

Coord. Sys.	Center	XY-plane	+X direction	+Z direction
Local horizon*	Sensor	Horizontal plane at sensor location	North	Zenith
Topocentric ecliptical**	Sensor	Ecliptic	Vernal equinox	Ecliptic north
Topocentric equatorial***	Sensor	Earth equator	Vernal equinox	Earth north

* Azimuth is measured from the north, increasing to the east (negative y-direction)

** Ecliptic longitude measured from the vernal equinox increasing in positive y-direction; ecliptic latitude measured from xy-plane increasing towards north ecliptic pole

*** Right ascension measured from the vernal equinox, increasing to the east (positive y-direction)

For orientations from space-based locations:

Coord. Sys.	Center	XY-plane	+X direction	+Z direction
Geocentric				
Local horizon*	Sensor	Horizontal plane at sensor location, shifted in zenith direction to sensor	Velocity vector projected onto XY-plane, azimuth increases clockwise.	Zenith
Topocentric ecliptical**	Sensor	Ecliptic, shifted to include sensor	Vernal equinox	Ecliptic north
Topocentric equatorial***	Sensor	Earth equator, shifted to include sensor	Vernal equinox	Earth north
Heliocentric				
Local horizon*	Sensor	(Sun) horizontal plane at sensor location, shifted in zenith direction to sensor	Velocity vector, projected onto XY-plane, azimuth increases clockwise.	Zenith
Topocentric ecliptical**	Sensor	Ecliptic, shifted to include sensor	Vernal equinox	Ecliptic north
Topocentric equatorial***	Sensor	Earth equator, shifted to include sensor	Vernal equinox	Earth north

* Azimuth is measured from the x-axis, increasing in negative y-direction

** Ecliptic longitude measured from the vernal equinox increasing in positive y-direction; ecliptic latitude measured from xy-plane increasing towards north ecliptic pole

*** Right ascension measured from the vernal equinox, increasing in positive y-direction

1.4 Abbreviations

The abbreviations used in this document are:

CLI	Command line interface
CSS	Catalina Sky Survey
DLR	German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt)
ESA	European Space Agency
FOV	Field of view
GUI	Graphical User Interface
ILR	Institute of Aerospace Systems (Institut für Luft und Raumfahrtsysteme)
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 Operations Environment

This chapter gives a brief overview of what hardware and software NEOPOP runs on. Section 2.1 defines what system you need to run NEOPOP, section 2.2 explains the tools and libraries NEOPOP uses internally.

2.1 System Specification

In order to ensure that NEOPOP runs correctly on your system, your operating system must meet the following requirements:

Processor Architecture	32-bit (x86 or x86-64
Processor	Min. AMD or Intel 2,8 GHz
RAM	Min. 2 GiB
Disk Space	Min. 2 GiB
Operating System	Either: ● Windows 10 ● Linux with GNU C Library (glibc) 2.x where x ≥ 26. This applies, for example, to o openSUSE 15.0+ o Ubuntu 17.10+ o Debian 9+
Operating System Architecture	x86-64
Java Runtime Environment	Java Standard Edition Runtime Environment 1.8 which is also known as Java 8 **(has to be available in PATH)**
gnuplot	v4.6+ ● On Windows: must be installed separately ● On Linux: must be installed separately
PDF Viewer	like Adobe Reader or Okular, optional
Text Editor	like Notepad++ or gedit, optional

Note that, however, in all cases, later versions of Windows or Linux should work.

PDF Viewer and text editor are not required to start the GUI and run the tool. Without them you can't open PDF and text files through the GUI though. You can define the programs of your choice at any time after installation via the settings button in the toolbar (section 6.6.11). On Linux systems, popular choices for PDF readers include "acroread", "okular" or "evince"; popular graphical text editors are "kate" or "gedit". When setting them in the GUI, they have to be prefixed with their full path (usually `/usr/bin/` on Linux).

2.2 List of Tools and Libraries

In the following a table is provided stating all tools and libraries that NEOPOP uses:

Tool/Library	Version	Purpose	Licensing	Distribution site
SPICE Toolkit for Fortran	N0064 June 11, 2010	Part of NEOPOP used for calculation purposes, error handling and more	Free of licensing	http://naif.jpl.nasa.gov/naif
Cairo library	1.0.2	Part of Linux versions of GUI. It's a graphics toolkit used by Eclipse launcher.	Mozilla Public License Version 1.1 ("MPL")	http://www.cairographics.org/
pixman library	0.1.6	Used in Cairo Library	Specific licenses; see NEOPOP license agreement for details	http://www.cairographics.org/snapshots
gnuplot tool	4.6+	Used by GUI to create plot picture files	Gnuplot Copyright	http://www.gnuplot.info/
GNU C Library (glibc, only on Linux)	2.5	Major system library	GNU Lesser General Public License	https://www.gnu.org/software/libc/

3 Installation

In this chapter the installation of NEOPOP on your system is described.

Download the ZIP file for Linux (64-bit Linux, 64-bit Java 1.8) and unzip it into an installation folder of your choice. This will create a new folder "NEOPOP_v2.0 " in that folder. In there, you'll find "start.sh", a bash script that you can execute in order to start NEOPOP GUI.

Download the ZIP file for Windows (64-bit Windows, 64-bit Java 1.8) and unzip it into an installation folder of your choice. This will create a new folder "NEOPOP_v2.0 " in that folder. In there, you'll find "start.bat", a batch file that you can execute in order to start NEOPOP GUI.

When updating from a previous NEOPOP version, please ensure a clean workspace environment. In particular, please make sure to not reuse the workspace folder of a previous NEOPOP version, as they are not compatible and will cause NEOPOP to fail. An empty workspace folder has to be used in order to run NEOPOP.

Please check if you need to consult a system administrator with installation privileges to install NEOPOP in a way that is appropriate for your IT configuration. An example for this would be a Windows computer used at a workplace where an administrator would like NEOPOP to be installed in `C:\Program Files` which standard users normally can't do.

4 External View of the Software

This chapter describes some folders and files NEOPOP uses after a successful installation.

4.1 Installation Folder

The installation folder contains all data required by NEOPOP. It contains four sub-folders:

- `default`: contains the default project, i. e. default input and data files as well as the NEOPOP tool's executable. This folder serves as a template: console users may copy it and start from there, the GUI does the same when creating a new project (see section 6.6.1).
- `docu`: contains the documentation of NEOPOP and gnuplot as well as the license agreement
- `gnuplot`: contains the plotting program gnuplot
- `gui`: contains the GUI

There are also some files located in the installation folder:

- `start<ext>`: the platform-dependent script to launch NEOPOP GUI
- `uninstall<ext>`: the platform-dependent script to uninstall NEOPOP

Depending on the used platform `<ext>` translates to `.sh` (Linux), no extension (also Linux) or `.bat` (Windows).

4.1.1 Data Folders

For executing the NEOPOP tool several data files are required that are stored in subfolders of the "default" folder described above. In this context a file is named a "data file" if the content typically remains unchanged (e.g. the NEO model data files). Because typically these data files do not change and some of them are quite large they are not stored in each project folder but once in the installation directory.

The following data files are required by the Population Generator component of the tool and stored in `<install dir>/01-POPGEN/data`:

Filename	***Meaning of the data file***	
`known_NEOs_up_to_H15.dat`	Contains known NEOs up to H=15	
`naif0010.tls`	Files needed by the SPICE software collection used by the NEOPOP tool	
`de430.bsp`		
`pck00010.tpc`		
`31all.res`	3:1 resonance objects	Source region data files of the old NEO model by Bottke
`imc_update_exttarget.res`	Intermediate Mars-crossing population	
`comet_jfc_only.res`	Jupiter Family Comets	
`nu6all.res`	nu6 resonance objects	
`hung.res`	Hungarian Asteroids	

Filename	*Meaning of the data file*	
pho.res	Potentially Hazardous Objects	
ob_superext.res	Outer Belt population	
Gmb_model.dat	Model file of the (new) NEO model by Granvik, Morbidelli, Bottke and collaborators (low-resolution variant)	

The following data files are required by the Observation Simulator component of the tool and stored in <install dir>/02-OBSSIM/data:

Filename	*Meaning of the data file*
de430.bsp	Files needed by the SPICE software collection used by the NEOPOP tool
pck00010.tpc	
naif0010.tls	
obscodes.dat	Unformatted observatory code list
optical.air	Airglow spectral distribution
optical.atm	Atmospheric extinction data
optical.ssz	Starlight spectral distribution for 0mag
optical.sun	Sun spectral irradiance data
optical.pat	Patched stellar catalogue
optical.zod	Zodiacal light data
optical.ssl	Scattered sunlight data
optical.sml	Scattered moonlight data
optical.lpf	Lunar phase factor data
optical.ebl	Extragalactic Background Light
optical.vbe	V-band extinction
optical.str	Star Class Catalogue – VIS
optical.stc	Star Class Catalogue – TIR

4.2 GUI Configuration Folders

As it is the case for nearly any software NEOPOP GUI needs to store configuration files, e. g. to know which project has been used when the GUI was closed the last time in order to open that project at the next launch of the GUI again. Such configuration data is stored in two places. One location is in the user's personal folder – <personal folder>/neopop-gui/conf – that the GUI creates upon its first launch by a user. The other location is .metadata that is generated in each workspace.

NOTE:
The configuration files are not intended to be changed by the user!

5 Tutorial

This chapter is intended to explain the basic workflow of using NEOPOP. The aim is not to discuss every detail, but to understand basic functionalities and to get familiar with NEOPOP in general.
Inputs are thus kept simple and computation times short, so that this tutorial can be completed within about 45 min. For all the other functionalities and the advanced use of NEOPOP, the rest of this manual as well as the Final Report (especially for theoretical background information) should be consulted.

5.1 Tutorial Case

The task to be performed during this tutorial shall be to determine which fraction of NEOs up to a diameter of around 1 km are known already, how dangerous they are, and how well they can be observed by a specific telescope.
For this purpose, firstly a NEO population is generated using the latest NEO population model implemented in NEOPOP and subsequently analyzed. In a second step, an observation of exactly this population using ESA's 1-m telescope on Tenerife will be simulated to evaluate how successful a specific observation strategy would be.

5.2 NEO Population Generation and Analysis

Launching NEOPOP

After the installation of NEOPOP (as described in section **Error! Reference source not found.**), the Graphical User Interface can be started by using the created desktop shortcut or by using the `start.bat` file in the installation directory (default directory under Windows: `C:\Program Files\NEOPOP-<version>`; under Linux: `/opt/NEOPOP-<version>`).
When launching NEOPOP for the first time, the software will ask for a workspace folder. This is where all the projects containing your NEOPOP input and output are stored. Usually you only need one workspace. Note that this folder is by default not located in the NEOPOP installation folder, but in the user's personal folder. Within the workspace folder, the software additionally creates a project folder that is named "NEOPOP" by default.

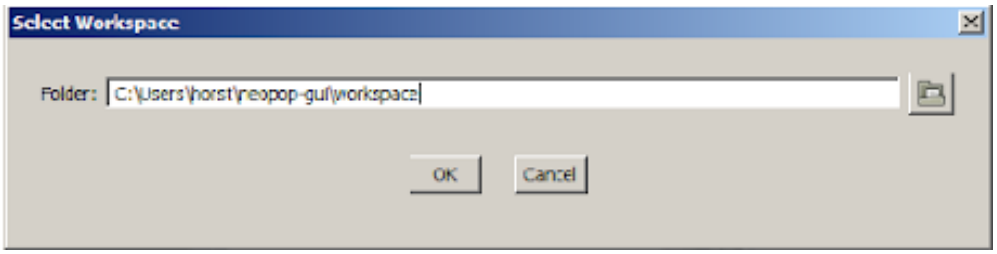

Figure 1: Workspace selection dialogue

Analysing the predicted NEO population

Once the workspace folder is selected and the main GUI window opens , the user can furthermore select a Run ID in the menu on the left. Results of each run within a project are all written to the same project folder within the workspace folder, but with the Run ID as a filename prefix. To prevent results of different simulation runs within one project being overwritten, the Run IDs can be changed. We will call this run “OGS-1”, for the telescope to be used.

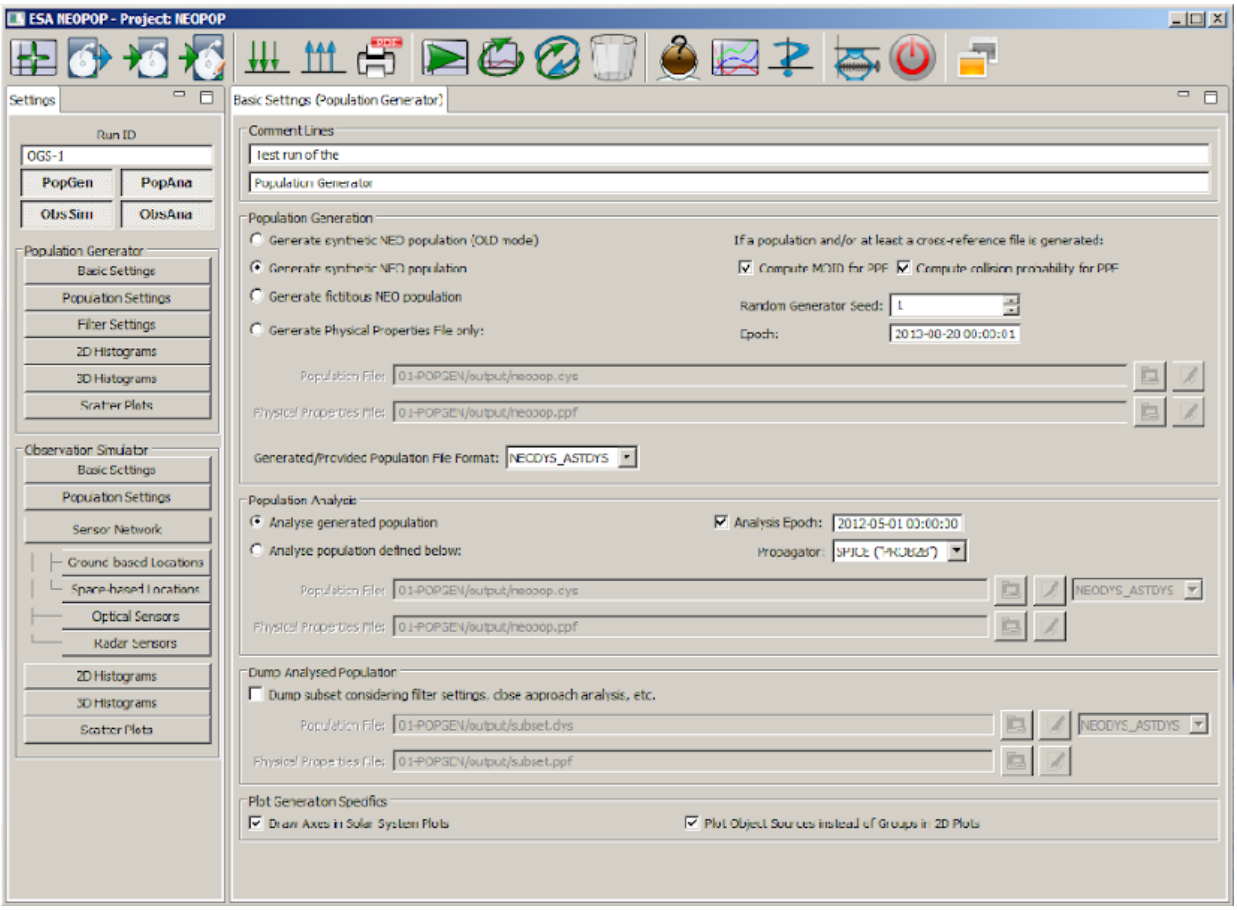

Figure 2: NEOPOP GUI with the "Basic Settings" tab of the population generation module

Right underneath the Run ID input field, the four modules of NEOPOP are listed: the NEO Population Generation (PopGen), the Population Analysis (PopAna), the Observation Simulation (ObsSim), and the Observation Analysis (ObsAna) modules. NEOPOP does not come with a fixed asteroid population, but lets the user generate a population mainly based upon the epoch of interest and the asteroid size range of interest. This also means, though, that before doing an analysis, a population needs to be created. Thus, to generate the predicted NEO population and analyze it, the Population Generation and Population Analysis modules are necessary and should be activated.

The basic settings necessary for generating the population can be made in the “Basic Settings” and “Population Settings” tabs. Firstly, the population model to be used has to be chosen. We will leave this at the latest NEO population model of 2015, which is selected by default. Since the potential minimum distance to Earth (expressed as the Minimum Orbit Intersection

Distance [MOID]) and the collision probability are of interest to assess the risk posed by objects, we activate the computation of these two parameters. Since the most threatening objects, with the potential for a global catastrophe upon impact, are those with a diameter of 1 km or more (equivalent to an absolute magnitude of about 18 or brighter), we will only regard these for now. For this purpose, the “Max. H-value” setting in the “Population Settings” tab should be set to 18.

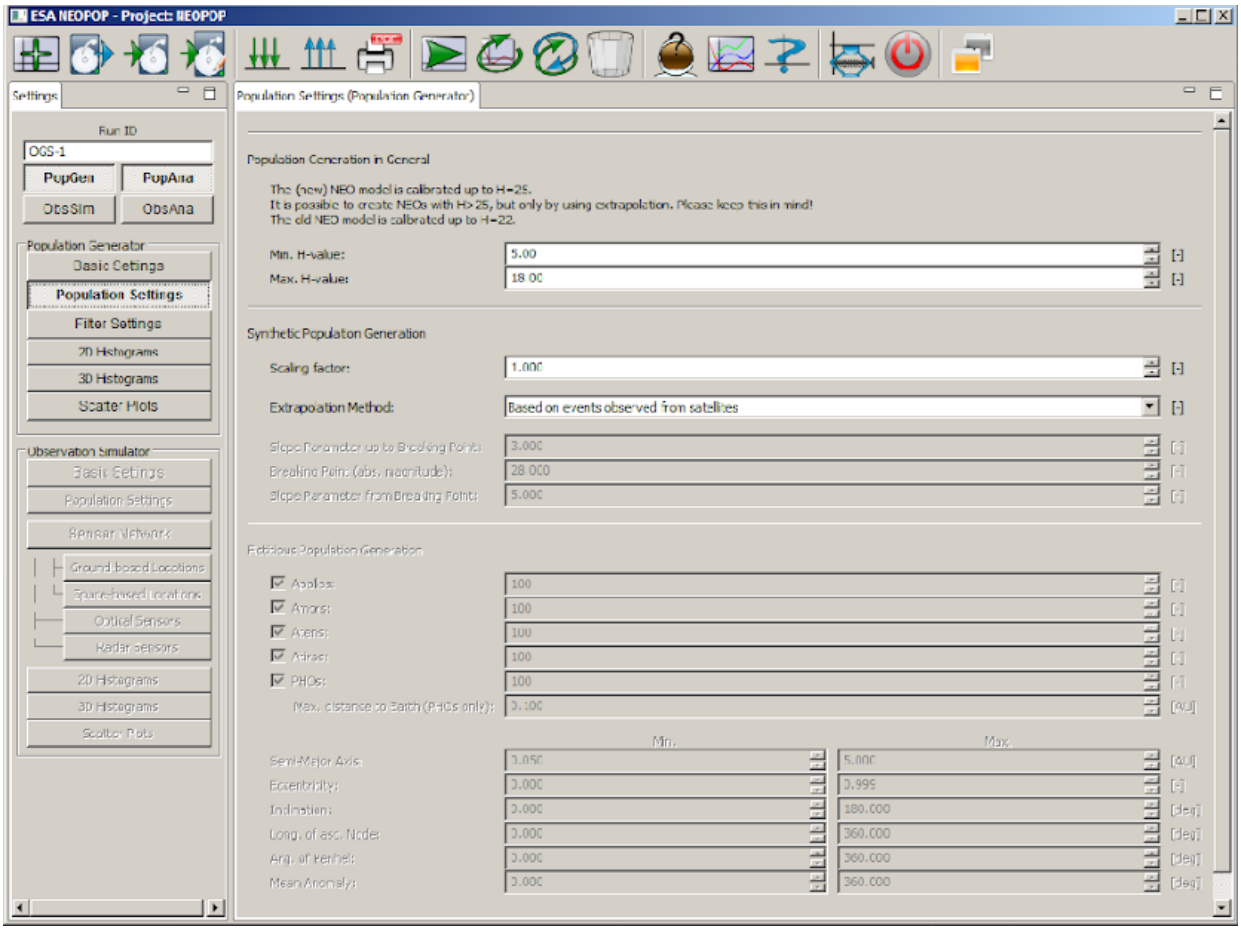

Figure 3: "Population Settings" tab of the population generation module

These settings represent everything that we would like to change in regard to the NEO population. Theoretically, we could now run only the PopGen module to create the population. However, we would also like to visualize the characteristics of the population, such as size and orbital parameters of the objects.
NEOPOP offers the possibility to directly create plots showing different characteristics of a population. We will use this function and choose the following settings for the plots:

- In the “2D Histograms” tab:

Active Axis Type	**Min.**	**Max.**	**Number of Classes**
Inclination	0.000	180.000	100.000

Active Axis Type	Min.	Max.	Number of Classes
H-Value	8.000	18.000	20.000
MOID	0.000	0.1	40.000
Diameter	0	3.5	50.000

Also, deactivate the use of logarithmic axes (“Log.” checkbox) for all plots.

- In the “Scatter Plots” tab:

Active Axis Types	Min.	Max.
Plot 1		
MOID	0.000	0.100
Diameter	0.000	3.500
Plot 2		
MOID	0.000	0.100
Inclination	0.000	180.000

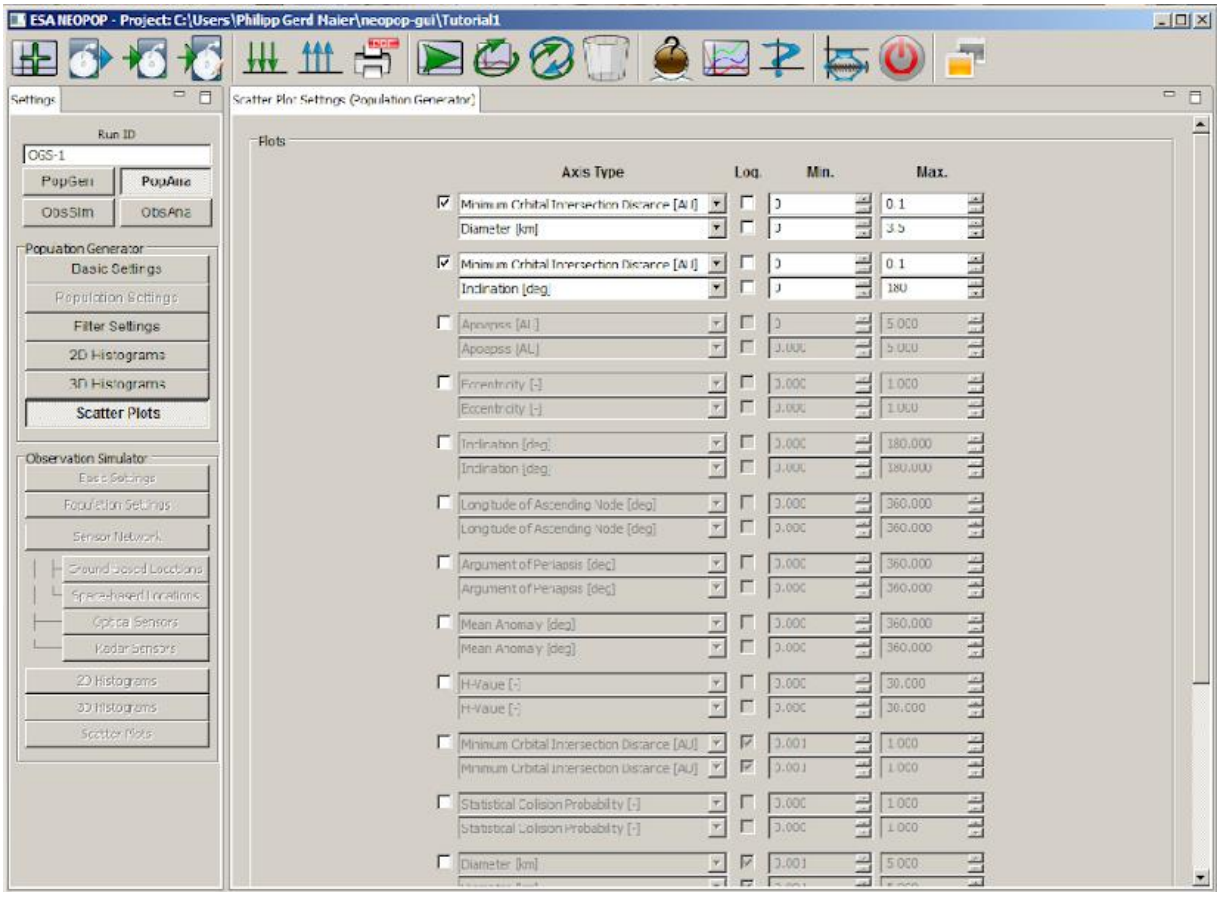

Figure 4: Settings to be made in the "Scatter Plot" tab

Clicking on the “Run NEOPOP” button in the main toolbar starts the analysis. After the run has finished, the resulting plots can be evaluated by switching to the output perspective via the

"Switch Perspective" button on the far right of the main toolbar. This perspective should be empty until "PopAna" is selected for output in the left menu and one of the plots is selected in the right menu.
If the HVAL plot is selected under "2D" in the right menu, it should look something like in figure 5. As expected, it should show a steep increase in the number of objects with decreasing brightness (i.e. size). NEOPOP also allows importing other data into the generated plots. It would, for example, be interesting to compare the number of objects predicted by the model to the number of already known objects. For this purpose, the gnuplot file displayed underneath the plot can be edited to load additional data.

Loading additional data into a plot

Additional data to be plotted must be provided in a text file. For this reason, copy the NEODyS data in table 1 (as of May 2014) to an empty text file and save it to the PopGen Output Folder in your workspace (in our case in Windows: C:\Users*Username*\neopop-gui\workspace\NEOPOP\01-POPGEN\output, in Linux: /home/neopop-gui/workspace/NEOPOP/01-POPGEN/output).

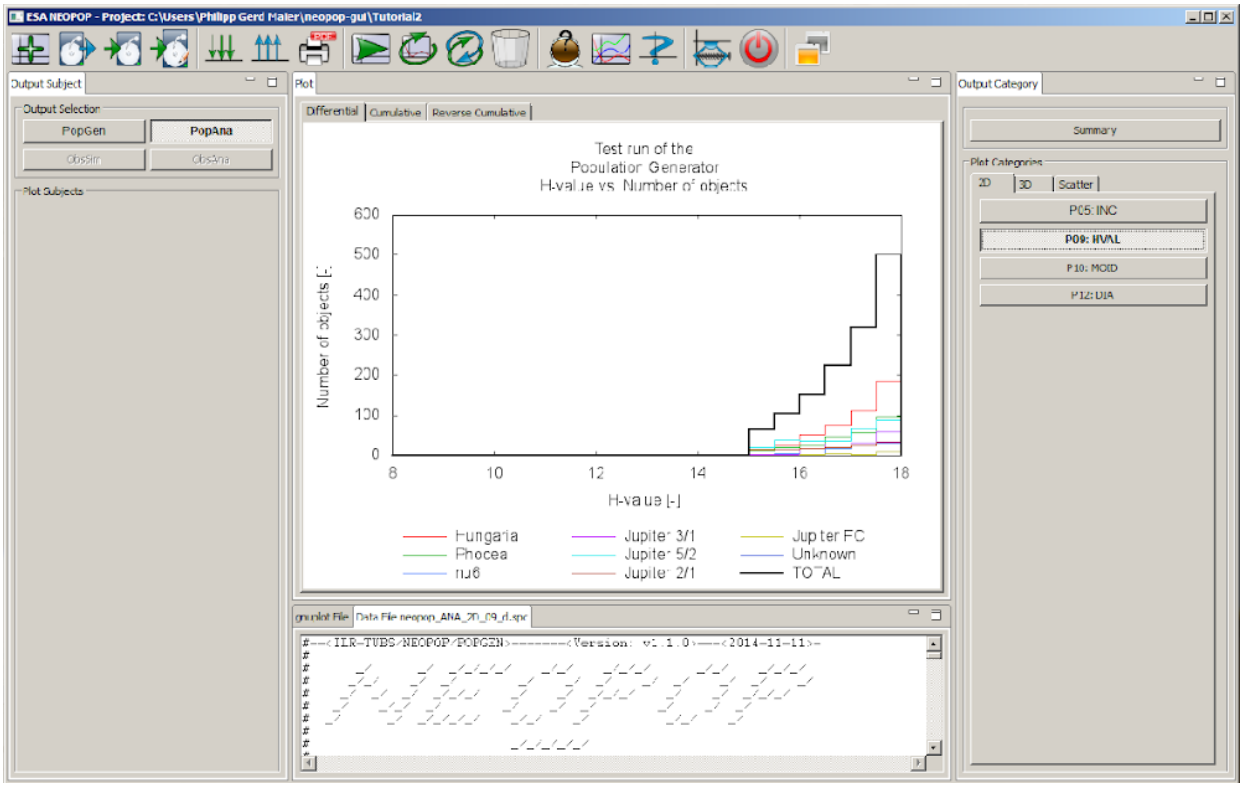

Figure 5: Output perspective of NEOPOP, showing the distribution of objects over absolute magnitude

The left column in the table contains the data bin centers, the right column holds the numbers of objects per respective bin of 0.5 absolute magnitudes. Save the file under the name "neopop.txt".

8.25	0
8.75	0
9.25	1
9.75	0
10.25	0
10.75	1
11.25	0
11.75	0
12.25	1
12.75	2
13.25	4
13.75	7
14.25	11
14.75	32
15.25	54
15.75	84
16.25	134
16.75	164
17.25	262
17.75	325

Table 1: Number of NEOs per bin of 0.5 absolute magnitues according to NEODyS (May 2014)

To plot the data in NEOPOP, it has to be added to the list of data to plot in the gnuplot File underneath the plot. To do so, end the last existing line in the editor with ",\" and add the following line underneath it (see also figure 6):

"neodys.txt" with histeps lt 2 lw 2 t "NEODyS"

The number after "lt" hereby sets the color of the line, the number after "lw" the line width, and the string after "t" the title that will be shown for the new line.

Figure 6: Plot and gnuplot File editor in the NEOPOP output perspective

If you now click the green “replot” button right above the gnuplot File editor widget, the additional data will be plotted. The result nicely shows how many large objects are still undetected.
Using the other created plots, the characteristics of these objects can be further examined and the questions investigated of why they have not been detected yet or how they can be systematically searched for in the future. Notice also that the last two plots in the plot category “Scatter” show the position of all objects in the solar system at the analysis epoch.

Consider now we would like to only explore the properties of objects with a certain characteristic, e.g. a diameter larger than 200 m. To do so, it is not necessary to regenerate the population. The Population Analysis module allows to separately change the analysis settings. For this purpose, change back to the input perspective via the “Switch Perspective” button. There, firstly de-activate the PopGen module by clicking on its module button on top of the left settings menu (see figure 7). Now change to the “Filter Settings” tab and activate the desired filter, in our case an object diameter between 0.5 and 1000 km. Running NEOPOP

now will apply those filter settings and show only properties of the filtered objects in the output perspective. This can be easily verified by having a look at the 2D plot of object diameters.

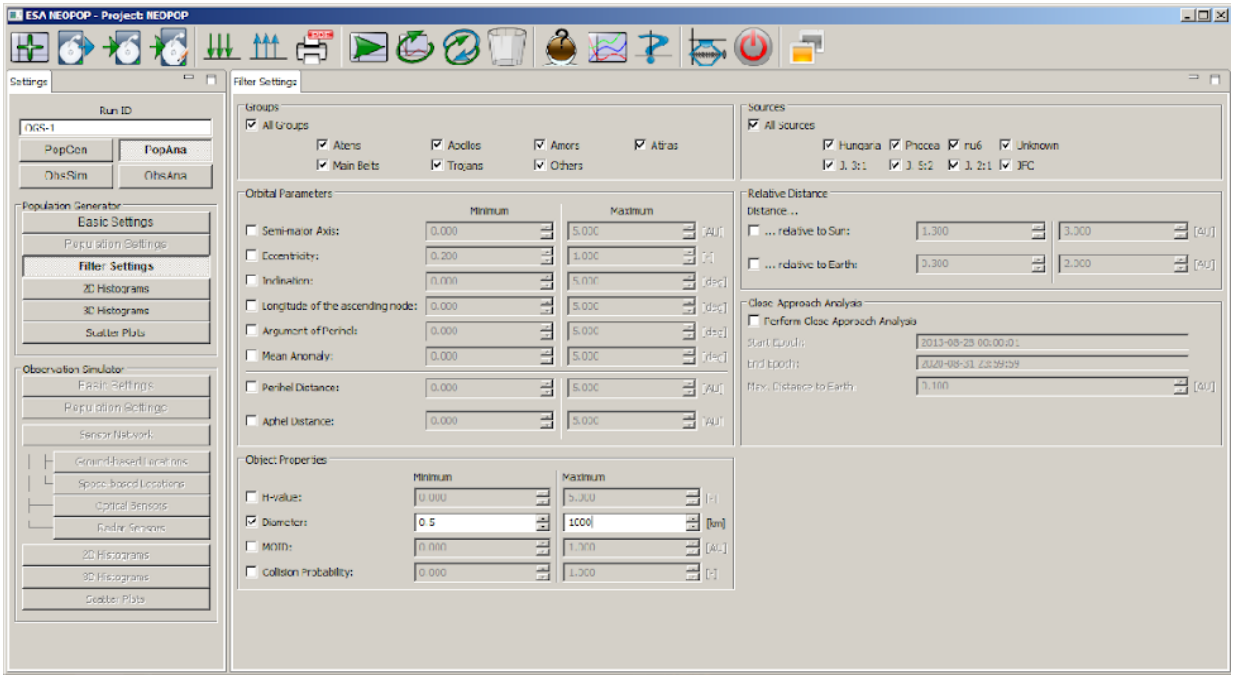

Figure 7: Changing filter settings to run only the Population Analysis module

The tutorial at this point will move on to simulating an observation of the created NEO population using ESA's Optical Ground Station (OGS) on Tenerife.

5.3 Observation Simulation and Analysis

Simulating an observation of NEOs

One of the main strengths of NEOPOP is the possibility to simulate observations of NEOs to evaluate the effectiveness of instruments or observation strategies. In this tutorial, the number of large asteroids (i.e. with an absolute magnitude brighter than 18) that could be observed by ESA's Optical Ground Station on Tenerife during continuous observations over one year shall be assessed. The approach in the software is similar to the one used to analyse a NEO population: using the module ObsSim, the observation is simulated, using ObsAna, the results are analysed and displayed graphically. You can therefore deselect the PopGen and PopAna, and select the ObsSim and ObsAna buttons in the left menu, once you have changed back to the input perspective of NEOPOP.

In the first step, the observation duration and population to be observed need to be defined. To do so, change the "Duration of Observation" in the "Basic Settings" tab to 1 year. Make sure that, under "General Switches", both "Perform Crossing Analysis" and "Perform Detection Analysis" are switched on:

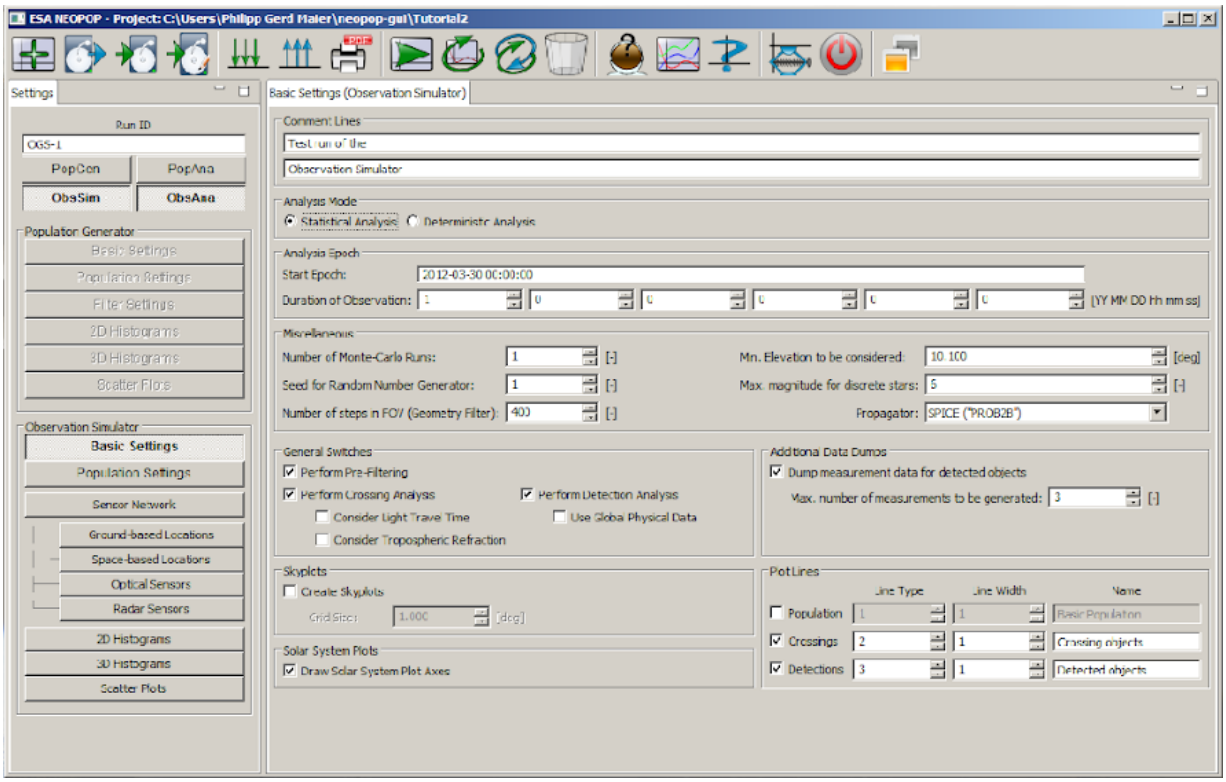

Figure 8: Settings in the "Basic Settings" tab of the observation simulation module

In the “Population Settings” tab, make sure that “Use Population Generation Output” is selected (underneath “Population No.1).

Defining a sensor system to be used

In the sensor systems tab, add a new sensor system and call it “OGS”. Make sure it is defined as ground based and optical. In the “Observed Volume” area, set the line-of-sight direction in the local horizon coordinate system to an elevation of 90 deg. To decrease the necessary computation time, limit the maximum range of the observed volume to 10 AU. Note that you currently can only choose “703” and “G96” (the two Observatories used for the Catalina Sky Survey) as “Location” and “Sensor”, since the location of the OGS and its technical details have not been defined yet:

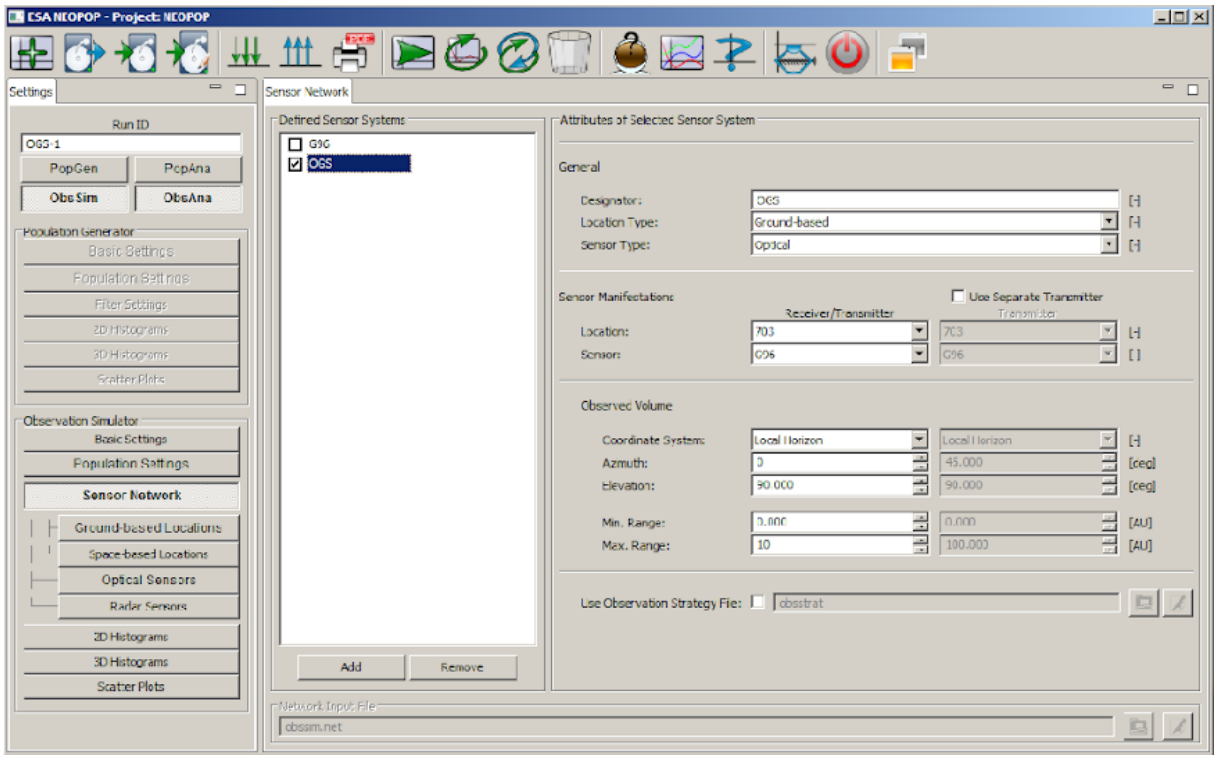

Figure 9: Adding the OGS as a sensor system to NEOPOP

Adding elements to the database of locations and sensors

To define these details, open the “Ground-based Locations” tab and define a new ground based location using its Observatory Code. The observatory code of the OGS is “J03”.

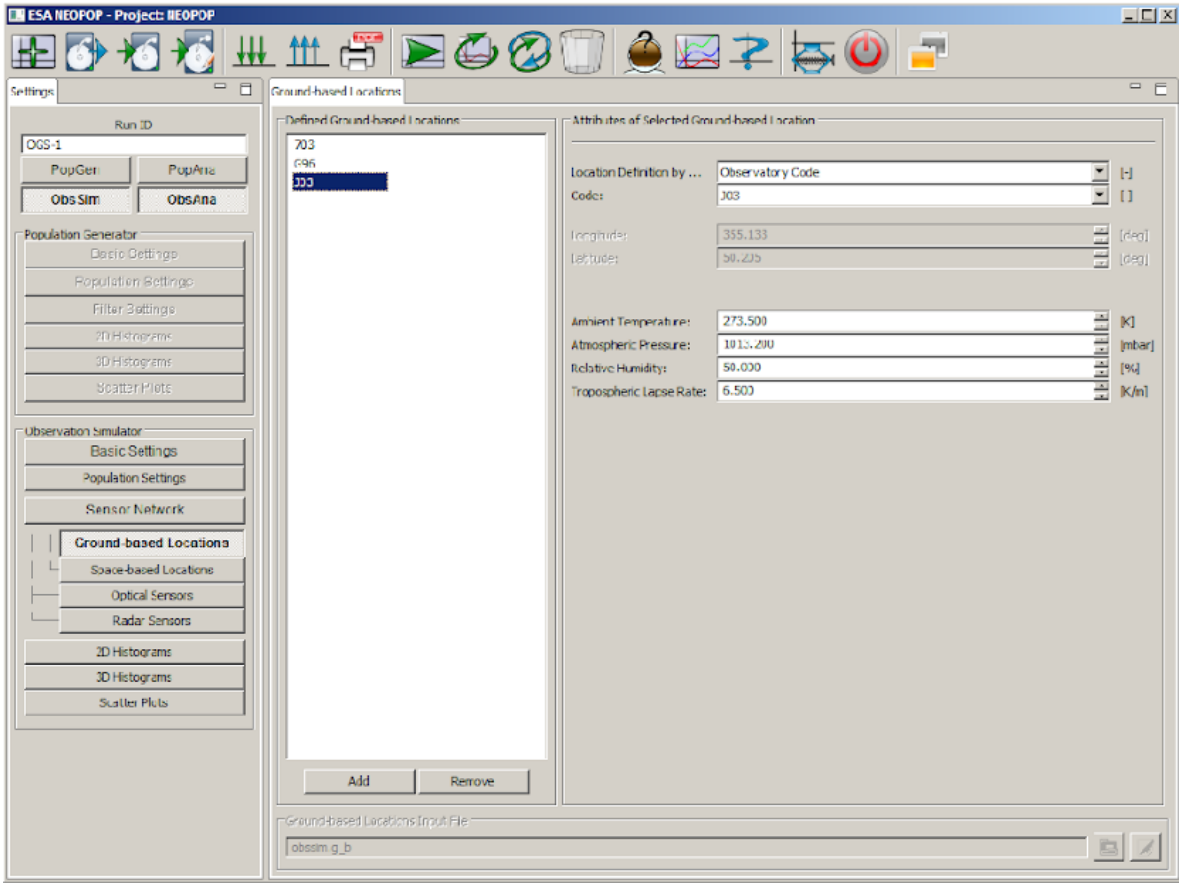

Figure 10: Settings to add the location of the OGS to the locations database of NEOPOP

In the “Optical Sensors” tab, also add a new sensor called “OGS” with the parameters in figure 11:

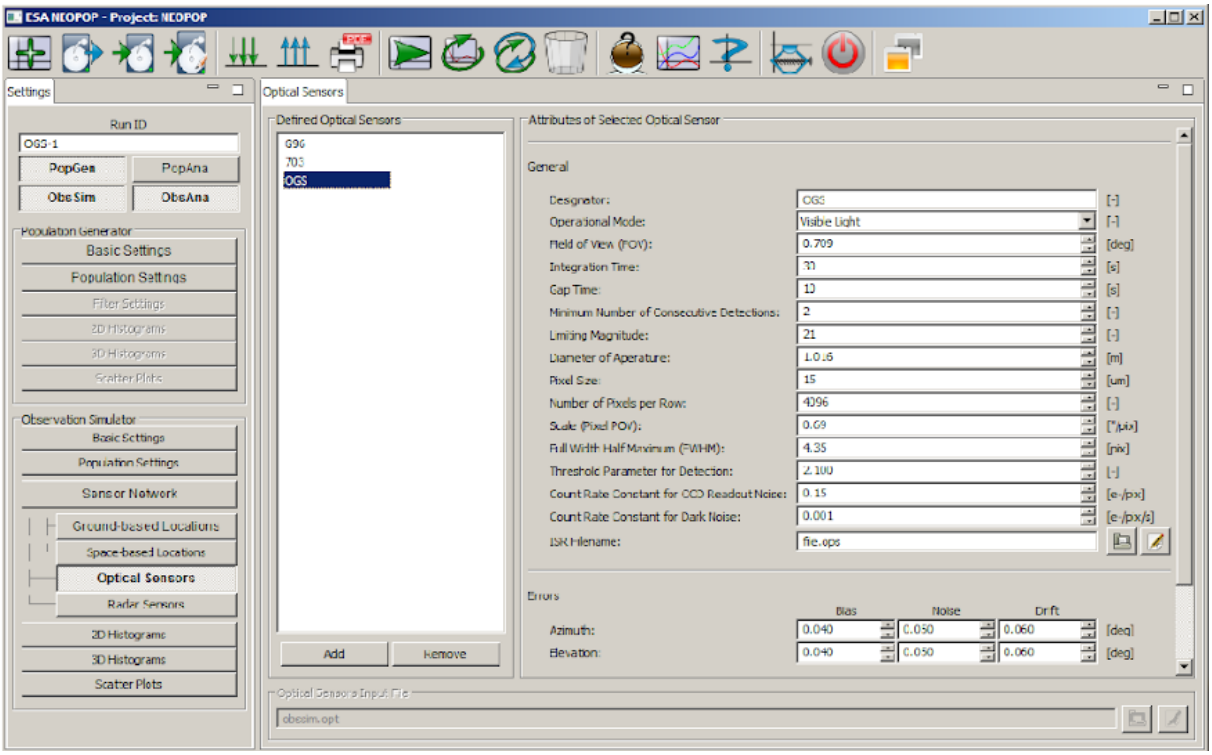

Figure 11: Settings to add the OGS as an optical sensor to NEOPOP

Now you can go back to the "Sensor Network" tab and choose "J03" as location and "OGS" as sensor for the "OGS" sensor system. Make sure to deselect the other sensor systems, unless you would like to simulate them as well.

Plotting the results

As during the generation of a NEO population, you can choose which plots NEOPOP shall generate. You can alter the settings in the "2D Histograms", "3D Histograms", and "Scatter Plots" tabs, as for the population analysis. To evaluate the OGS observations, however, we can leave them on their default values. The computation is started again by clicking the "Run NEOPOP" button in the main toolbar.

After the computation run has finished, you can change to the output perspective again and view the results by selecting "ObsAna" in the left menu, "OGS" right below it, and the plot you wish to have displayed in the right menu. Using the various plots, a possible observational bias may be spotted and it may be examined whether the OGS would likely be able to detect the objects with a small orbital intersection distance.

More information on the individual objects that either cross the sensors field of view during the observation duration or actually can be detected, can be found in the "OGS-1.res" text file in the ObsSim output directory (in our case in Windows: C:\Users*Username*\neopop-gui\workspace\NEOPOP\02-OBSSIM\output, in Linux: /home/neopop-gui/workspace/NEOPOP/02-OBSSIM/output).

6 Basic Usage

This chapter describes everything the user needs for basic usage of NEOPOP. It covers basic the first startup, basic concepts and all the main functions. After reading this users should be able to know how to use NEOPOP.

6.1 Starting NEOPOP

NEOPOP can be operated using its Java-based Graphical User Interface (GUI) providing a wide field of functionalities to the user. Executing NEOPOP via the command line is also possible, though possibly much less comfortable than via the GUI.

Perform one of the following actions to start NEOPOP GUI:

Windows

- Double-click the “NEOPOP (32-bit)” or “NEOPOP (64-bit)” link on your desktop
- Open the “NEOPOP (32-bit)” or “NEOPOP (64-bit)” entry in the Start=>Programs menu and click on “NEOPOP”
- Navigate to NEOPOP’s installation directory and double-click/execute `start.bat`

Linux

- Double-click the “NEOPOP (32-bit)” or “NEOPOP (64-bit)” link on your desktop
- Navigate to NEOPOP’s installation directory and click/execute `start.sh`

The Command-line Tool executable is located in `<Installation directory>/default`. Note that this is also the place of the default NEOPOP project. Doing something in that folder may have unexpected consequences. It is recommended to copy the default folder and to work in that copy.

6.2 Workspaces, Projects and Runs

When starting the GUI for the first time you will be asked to select a workspace (Figure 12). Workspaces basically are folders on your system in which projects containing your NEOPOP input and output files are stored. That way all your files can be found in one place.

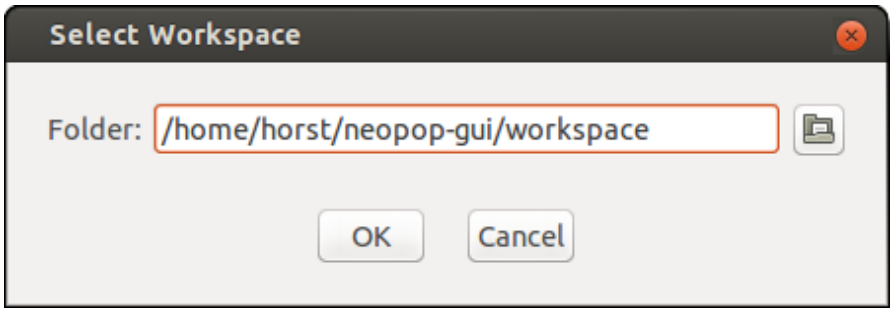

Figure 12: Workspace selection dialog

Usually you only need one workspace. That's why NEOPOP GUI will only ask you to select your workspace folder when you start it for the first time. After that you can still change your workspace by using the Settings Button (see section 6.6.11).

Your workspace can be located anywhere on the system but should not be located in the installation folder of NEOPOP. Generally, it should be a new folder to prevent conflicts with other data. Additionally, it should be a folder you have full read and write access to so the default one proposed by the GUI usually is a good choice. That folder is `<your personal folder>/neopop-gui/workspace` (the exact location depends on your operating system; on Linux it may be `/home/<name of user>/neopop-gui/workspace` for example).

NOTE:
Do not locate the workspace in an already-existing directory like the NEOPOP installation directory because, in a worst case, data might get lost during a later uninstallation process!

As stated above, workspaces contain projects. When a workspace is opened for the first time a project named "NEOPOP" is created like it would have been created by the user explicitly (cp. section 6.6.1) and opened by default (location: `<workspace>/NEOPOP`). Otherwise, the previously-opened project is opened again.

A project (folder) is meant to cover one topic for which NEOPOP is used. It contains all input files needed by the tool to generate output files (which are created in subfolders of the project folder). Since data files usually don't change and are big they are not stored in project folders created through the GUI. Instead the input files reference the default data files in the GUI installation folder (cp. section 4.1). The tool's executable file also resides there. For a better structure it is possible to create projects in subfolders of the workspace to combine several projects under a common topic.

You can "run" a project, i. e. launch the NEOPOP tool with the settings that you have made. The tool will then create output files and prefix them with the "Run-ID". That way you can change your Run-ID to try different settings and view the old run's output at a later time by providing the old Run-ID and using for example the "Replot"-button (see section 6.6.8).

NOTE:

When using the same Run-ID twice the old output files are overwritten!

6.3 GUI Overview and Perspectives

NEOPOP GUI's look after successfully starting it for the first time can be seen in Figure 13 . By clicking on the rightmost button of the main toolbar (the "Switch Perspective" button, see section 6.6.13) you can switch between two view types called "perspectives": The "input perspective" (see section 6.4) and the "output perspective" (see section 6.5). The figure mentioned shows the GUI in the input perspective. The main toolbar (see section 6.7) does not belong to one of the perspectives but to the window itself.

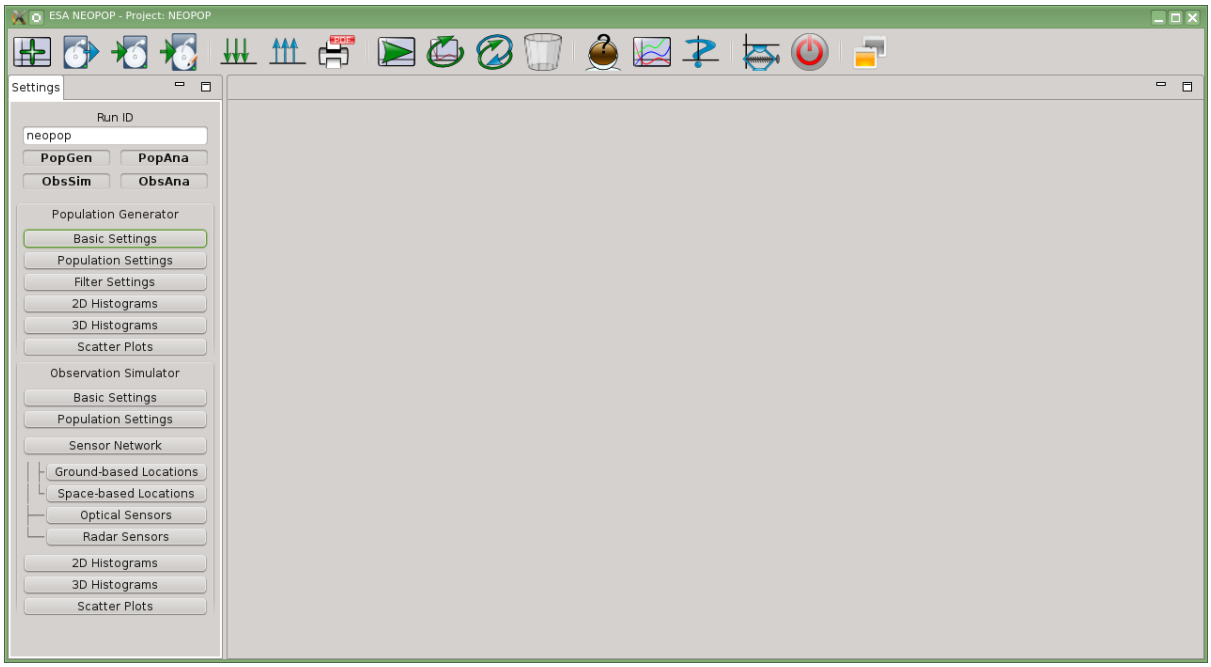

Figure 13: The GUI look after first launch showing the input perspective

6.4 Input Perspective

The GUI's input perspective (cp. Figure 14) allows you to manipulate the input data for the NEOPOP tool mainly by providing access to its settings via the sidebar on the left which is called the "settings overview". When clicking on one of the buttons below the captions "Population Generator" or "Observation Simulator" the respective settings are shown in the main area right of the settings overview. At the top you can specify the Run-ID of the current run (cp. section 6.1) and below that you can toggle (activate or deactivate) the tool's modules (cp. section 6.10).

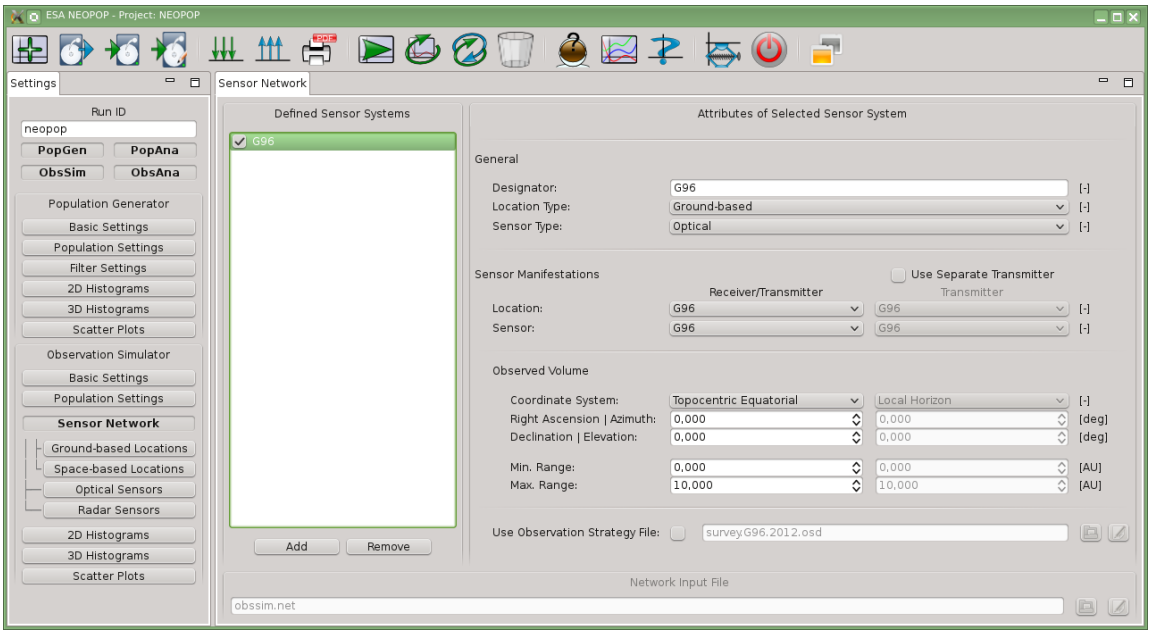

Figure 14: The GUI's input perspective

6.5 Output Perspective

The GUI's output perspective is shown in Figure 15 . Users can view and change the NEOPOP tool's output or – to be more specific – the current run's output. By changing the Run-ID in the input perspective the GUI tries to load the respective run's output in the output perspective.

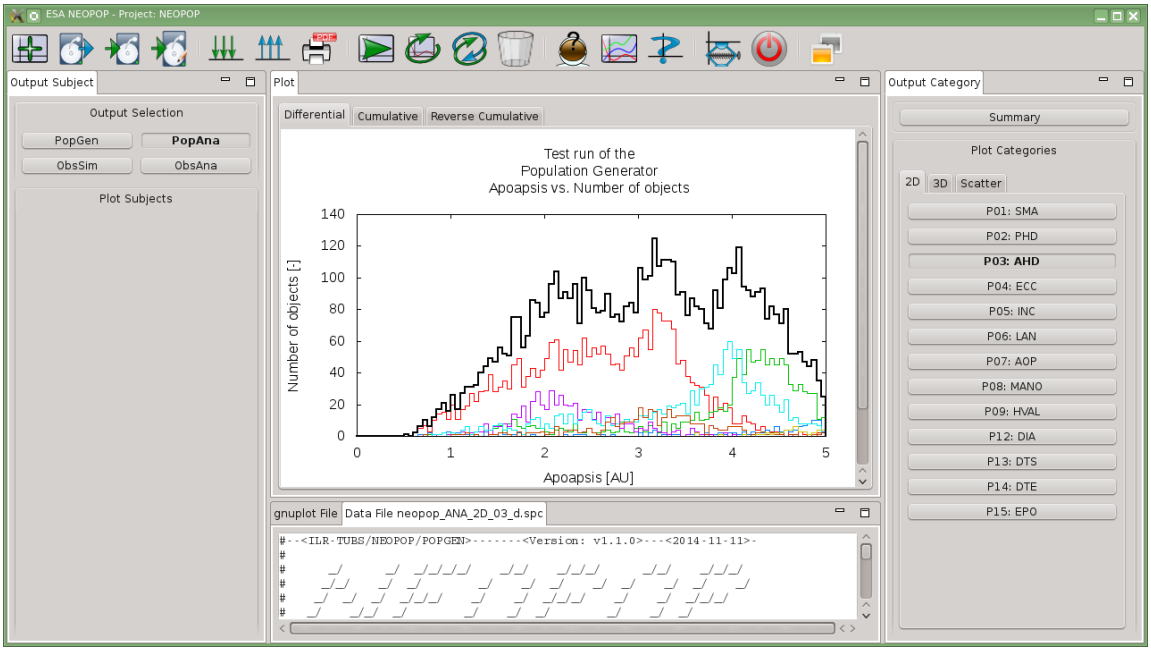

Figure 15: The GUI's output perspective

On the left side of the output perspective there is the “output subject” sidebar where you can select the output of what module (NEOPOP tool part, cp. section 6.10) you want to view. Depending on what output there is for the specific module the output subject sidebar and the “output category” sidebar, which is on the right, are filled.

If you select any of the modules the output category is filled with the summary category on the top right. By clicking on it the middle part of the perspective shows the summary giving you an overview of what the module did.

If you select the Population Analysis or Observation Analysis module the output category sidebar will be filled with the plot categories that have been selected in the plot settings of the input perspective when the tool was run. Also, it may contain sky plot entries if the corresponding settings had been made. You can then select one of the plot categories and view the corresponding plots, gnuplot driver files and data files.

If you select the Observation Analysis output the output subject sidebar is filled with the sensor systems that have been selected in the sensor system settings of the input perspective when the tool was run. An additional output subject is added for all the systems together (the network). You can then click on the subjects to view the corresponding plots, gnuplot driver files and data files.

Clicking on a plot category opens the corresponding plot (in case of the Observation Analysis module you have to additionally select an output subject) in the top-middle of the perspective. Below that the plot’s gnuplot driver file and the data files used are opened. In case of 2D plots you can switch between the differential, cumulative and reverse-cumulative plots and their gnuplot driver and data files.

When viewing a gnuplot driver file the gnuplot toolbar is shown.

6.6 Main Toolbar

The main toolbar at the top of the GUI includes several buttons which execute all main functions. The functions of these buttons will be described in the following sections. An overview is given in Table 2.

Description	Button / Label	Section
Create a new project	“New”	6.6.1
Open an existing project	“Open”	6.6.2
Save a project	“Save” & “Save As”	6.6.3
Dump results into PDF	“PDF”	6.6.4
Export a project	“Export”	6.6.5
Import a project	“Import”	6.6.6
Run NEOPOP tool	“Run”	6.6.7
Regenerate the plots	“Replot”	6.6.8
Reset input settings	“Reset”	6.6.9
Remove a project	“Remove”	6.6.10

Description	Button / Label	Section
Edit general Settings	“Settings”	6.6.11
Open this document	“Help”	6.6.12
Open gnuplot documentation	“Plot Help”	6.6.12
General info about the GUI	“About”	6.6.12
Quit NEOPOP GUI	“Quit”	6.6.12
Switch between input and output perspective (“view”)	“Switch Perspective”	6.6.13

Table 2: Overview of the main toolbar buttons

6.6.1 “New” Button

The “New” button (see Figure 16) has to be used for creating a new project. A pop-up window will appear after the button has been pressed (see Figure 17). The user may close the dialog at any time by pressing the “Cancel” button. The dialog shows the absolute path to the workspace. The user has the chance to browse through the whole file system or define the absolute path to the new project folder. Thus, or by writing the absolute path directly into the dialog it is possible to create the new project folder. After pressing the button “OK” a folder will be created and the pop-up window disappears. The new project will be opened, while the current project will be closed. The content of the project folder (input files) is taken from the default folder of the NEOPOP installation folder.

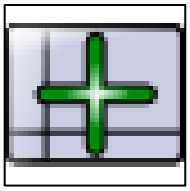

Figure 16: "New Project" button

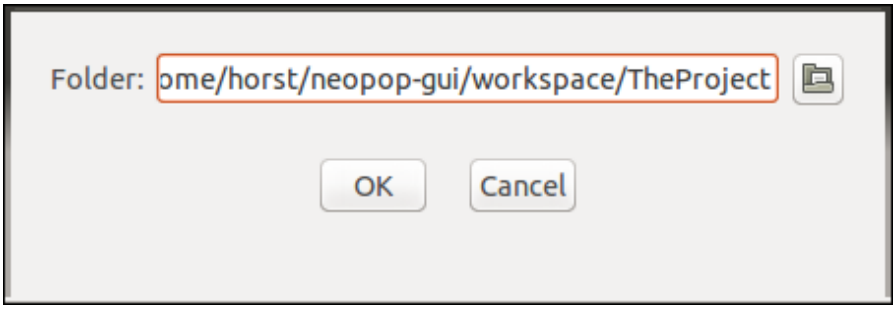

Figure 17: "New Project" dialog

6.6.2 “Open” Button

The “Open” button (see Figure 18) has to be used for getting access to an already-existing project. A pop-up window will appear after the button has been pressed (see Figure 19). The user may close the pop-up window at any time by pressing the “Cancel” button. This dialog shows by default the absolute path to the workspace. The user is able to browse through the whole file system for defining the project to be opened. After pressing the button “OK” the project will be opened and the pop-up window disappears. The new project will be loaded, while the current project will be closed.

Figure 18: "Open Project" button

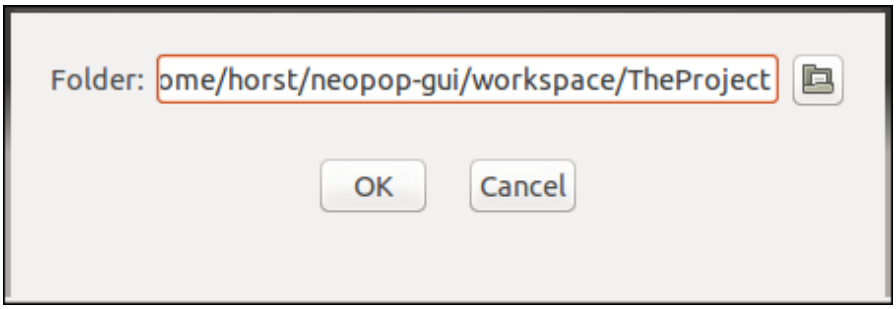

Figure 19: "Open Project" dialog

6.6.3 “Save” and “Save As” Buttons

The “Save” button (see Figure 20) simply stores the user input (that is already visible in the GUI) into the input files of the currently opened project. The “Save As” button is a combination of the functionality of the “New” and “Save” button. However, it stores all user input. A pop-up window will appear after the button has been pressed (see Figure 21). The user may close the pop-up window at any time by pressing the “Cancel” button. This dialog shows by default the absolute path to the workspace. The user is able to browse through the whole file system for defining the new project folder, where the current project shall be saved to. After pressing the button “OK” a folder of the defined name will be created and the dialog disappears.

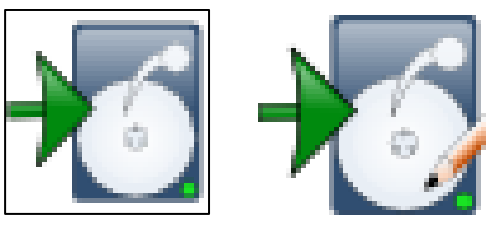

Figure 20: “Save” and "Save As" buttons

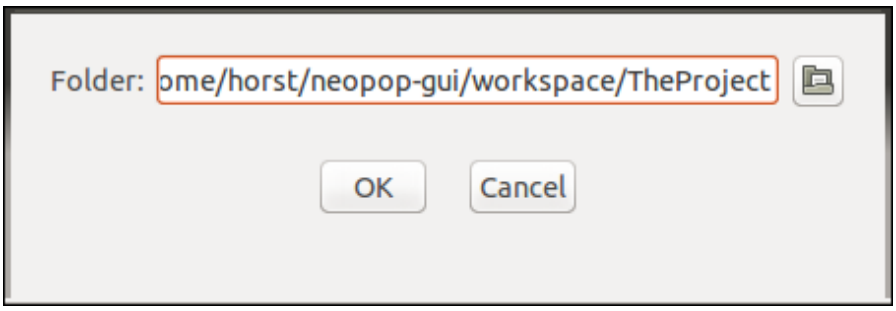

Figure 21: "Save As" dialog

6.6.4 “PDF” Button

The “PDF” button (see Figure 22) has to be used for creating a PDF containing the input and output of the current project. A pop-up window will appear after the button has been pressed (see Figure 23). The user may close the pop-up window at any time by pressing the “Cancel” button. The user is able to browse through the whole file system for defining the absolute path and file name of the PDF. After pressing the button “OK” the pop-up window disappears and the PDF will be created.

Figure 22: "PDF" button

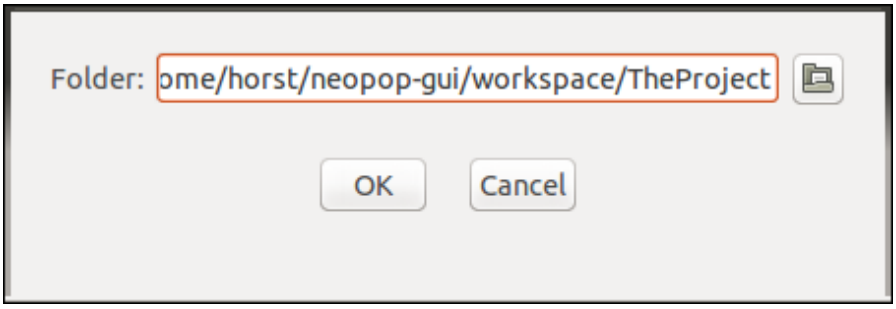

Figure 23: "PDF" dialog

6.6.5 “Export” Button

The “Export” (see Figure 24) can be used to export the content of a project into a single file. It is called “NEOPOP project file and is basically a zip file. This function provides an easy way for the users of the software to share a given project with others. After pressing the button a popup window appears (see Figure 25). In the upper part of the window the project that should be exported has to be selected. The user is able to browse through the whole file system. It is possible to create new folders. A name of the project archive has to be defined

beneath that. Both input and output of NEOPOP are exported. After pressing the button “OK” at the bottom of the pop-up window the window disappears and the project file will be generated. Data files and the executable are not included in the project archive. The user may close the pop-up window at any time by pressing the “Cancel” button.

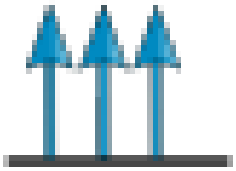

Figure 24: "Export" button

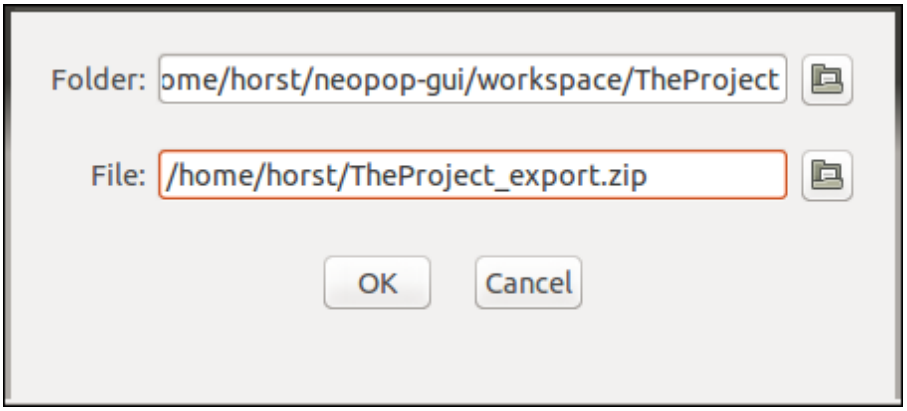

Figure 25: "Export" dialog

6.6.6 “Import” Button

The “Import” button (see Figure 26) is the pendant to the “Export” button. It allows the user to import exported projects from other users into their workspace. After pressing the button “Import” a pop-up window will appear (see Figure 27). The user may close the pop-up window at any time by pressing the “Cancel” button. This window consists of two parts. The upper part is for the selection of the project archive, which was generated by using the export functionality (see 6.6.5). After a project file has been selected the destination has to be defined in the lower part of the pop-up window. Again already the absolute path to the workspace is pre-defined. The user may change the project name. After pressing the button “OK” the project will be imported and the dialog disappears.

NOTE:
Importing a project doesn’t open it. The general idea of the import function is to just get another user’s project into your workspace.

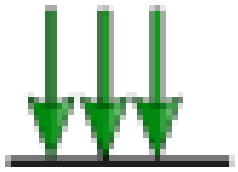

Figure 26: "Import" button

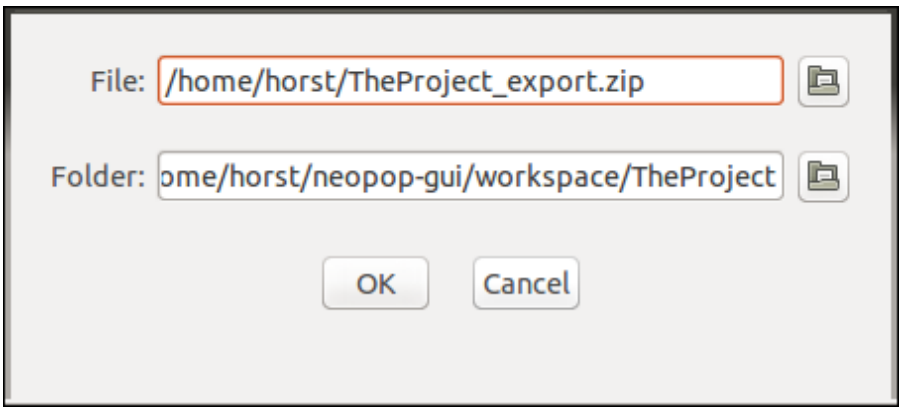

Figure 27: "Import Project" dialog

6.6.7 "Run" Button

The "Run" button (see Figure 28) saves the current project and starts the NEOPOP tool, which then executes the selected modules. Parallel to that the run dialog (Figure 29) is opened. At the top it communicates the execution status via a progress bar. To get detailed information users may view the tool's log output by clicking the corresponding item. In case of errors the error log output may be consulted. The tool's execution can also be canceled . This dialog may only be closed when the tool isn't running (anymore) and the "Replot All" function has been executed after the tool's execution.

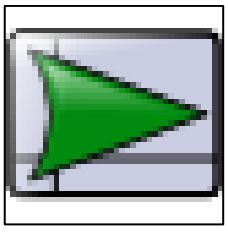

Figure 28: "Run" button

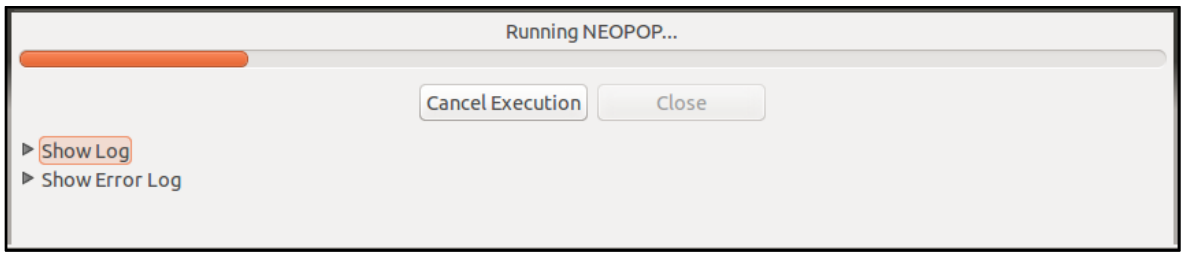

Figure 29: "Run" dialog

6.6.8 "Replot All" Button

Pressing this button leads to a regeneration of all plots of the current run shown in the output perspective. In order to achieve that, gnuplot is run.

Figure 30: "Replot All" button

NOTE:
The plots that were generated with the "run id" that is currently selected are replotted, rather than the plots of your last NEOPOP tool run.

6.6.9 "Reset" Button

Using the "Reset" button (see Figure 31) opens a small pop-up window (see Figure 32) where the user has the option to select to which source of data the input data of the GUI shall be set. The default option is to reset the data to the data stored previously in the input files. The second option is to reset the data to the data of the default files located in the installation directory of NEOPOP. At any time the user has the chance (if the access to the installation path is not restricted) to modify the default values to create his own default scenario.

Figure 31: "Reset" button

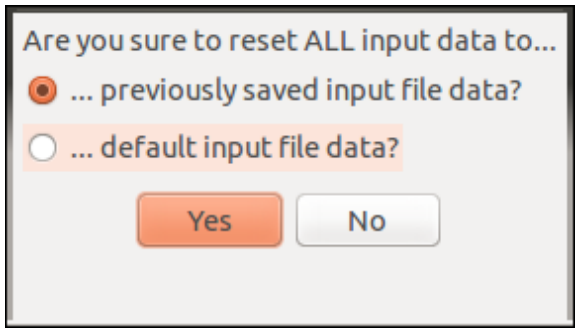

Figure 32: "Reset" dialog

6.6.10 “Remove” Button

The “Remove” button (see Figure 33) can be used to clean up the work space. It opens the “Remove” dialog (Figure 34) where the user can select any project folder. The selected folder will be removed when clicking “OK”.

Figure 33: "Remove" button

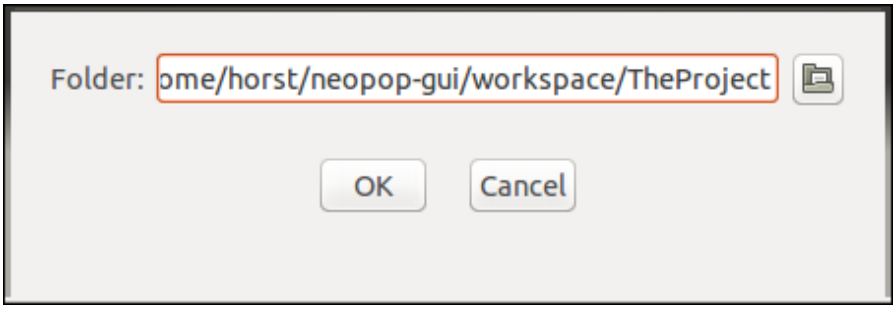

Figure 34: "Remove" dialog

6.6.11 “Settings” Button

The “Settings” button (Figure 35) opens the “Settings” dialog (Figure 36). At the top (“Binaries”) the file paths to the executable files of the programs used by the GUI can be defined. These paths are saved by the GUI globally on a per-user basis in its configuration files.

The NEOPOP tool and gnuplot file paths are set by default so that users are not supposed to change them (unless they know what they are doing). The PDF viewer and text editor paths can be set by the user in order to make the “Help” and “Plot Help” (section 6.6.12) as well as the “Edit File” (section 6.9) buttons work correctly.

Underneath the binaries settings users are able to close or reset their current workspace and to close the GUI directly after that in one go. After the user starts the GUI again (this is not done automatically!) he will be asked for a workspace again.

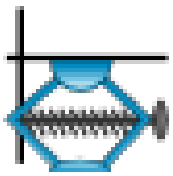

Figure 35: "Settings" button

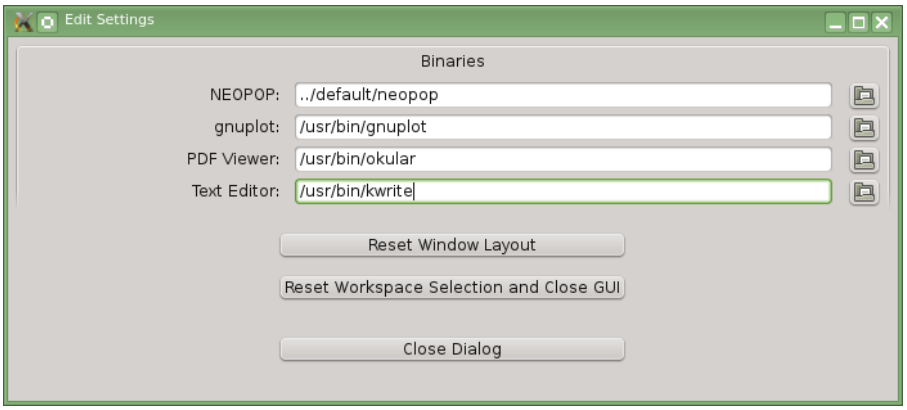

Figure 36: "Settings" dialog

6.6.12 “Help”, “Plot Help”, “About” and “Quit” Buttons

Pressing the “Help” or “Plot Help” button (Figure 37) starts the PDF viewer defined via the “Settings” button (section 6.6.11) to open this software user manual or the gnuplot manual as first help options to the user. The “About” button shows general information about the NEOPOP software and contact information for support. Finally, using the “Quit” button closes NEOPOP GUI.

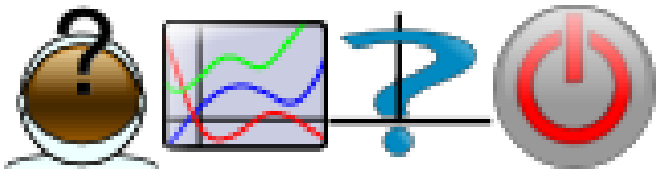

Figure 37: "Help", "Plot Help" "About" and "Quit" buttons

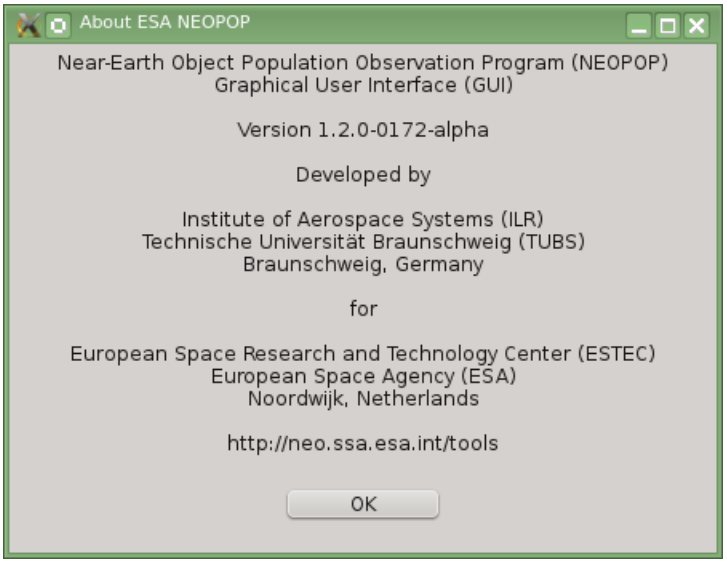

Figure 38: "About" dialog

6.6.13 “Switch Perspective” Button

This button which is shown in Figure 39 switches between the input (see section 6.4) and the output perspective (see section 6.5), i. e. the “view variant” of the GUI where you can access NEOPOP’s input and output respectively.

Figure 39: "Switch Perspective” button

6.7 gnuplot Toolbar

In the output perspective of NEOPOP GUI two additional buttons are available when a gnuplot driver file is shown: the “Replot Single” and “Undo gnuplot File” buttons. These two are explained in this section.

6.7.1 “Replot Single” Button

The “Replot Single” button looks just like the “Replot All” button (Figure 30) – only smaller. It saves the changes that you’ve made to the gnuplot driver file that is currently open and executes gnuplot to recreate the corresponding plot.

6.7.2 “Undo gnuplot File” Button

The “Undo gnuplot File” button (Figure 40) reads the gnupot driver file that is currently open again thus resetting it and undoing your changes.

Figure 40: "Undo gnuplot File” button

6.8 Working with gnuplot

The gnuplot software is used by the GUI to generate diagrams defined by driver files that are created by the NEOPOP command-line tool. Thus, the use of NEOPOP in the command-line version generates the full set of results. Only gnuplot would have to be executed manually in this case to generate the diagrams. An easy way to create all plots at once is to go into the output folders and provide the summary gnuplot file to gnuplot:

NEOPOP Component	Directory to Change to	Summary gnuplot File
Population Generator	`<project>/01-POPGEN/output`	`<run id>_ANA.gnu`
Observation Simulator	`<project>/02-OBSSIM/output`	`<run id>_OBSSIM.gnu`

The gnuplot files reference and use spectra files (*.spc) that are output to the same output folders mentioned above. This is where the actual data for each plot is being stored in.

6.8.1 The Basics of gnuplot

An example for the initial settings of a diagram is shown below:

```
reset
set term png
set output "neopop.png"
set title "NEOPOP - Default Settings\n number of objects vs. inclination"
set xlabel "inclination [deg]"
set ylabel "number of objects [-]\n(cumulative)"
set xrange [*:*]
set yrange [*:*]
set logscale x
```

```
set logscale y
set key below
plot [...]
```

The basic settings are explained below. For more details please consult the gnuplot manual [2], which is also located in the installation directory of NEOPOP (*<installdir>/docu/gnuplot.pdf*) and can be opened via the "Plot Help" button (cp. section 6.6.12) in the GUI.

Output format

The default output format is *set term png*. Modifying this line would for example be required if the diagram shall be generated as Encapsulated Postscript (.eps). In this case the line could be changed to the following command: *set terminal postscript eps*.

File name

The name of the generated file would be *neopop.png* as shown in the example above. Changing the file name should be self-explanatory. Take care of the quotation marks, which have to be present at the beginning and the end of the file name.

Title and labels

Changing the title (*set title ""*) or one of the axis labels (e.g. *set xlabel ""*) should also be self-explanatory. As shown in the example above a *\n* indicates a line break and can be implemented without blanks into the description.

Range of axis

If a figure shall contain the whole set of data that is provided in the data file the ranges of all axis are unlimited, which is indicated by the two asterisk separated by the colon (*set xrange [*:*]*). If the range shall be limited the user can define the limits by replacing the asterisk with numbers (e.g. *set xrange [-300.00:3E5]*).

Logarithmic or linear

An axis can be either linear or logarithmic. In the shown example both the x-axis as well as the y-axis would be of logarithmic scale. Changing the scale of the x-axis to a linear scale would be possible by a leading *#* indicating the line as a comment , by deleting the whole line or by changing the command to *unset logscale x*. Take care that the data is not negative in case of logarithmic scale.

Legend (key)

The legend of the example would be shown below the diagram. Other positions could be "*left*", "*right*", "*top*", "*bottom*", "*outside*". To generate a plot without a legend the user may use the command *unset key*.

After having defined the general settings for the plot the data to be used and the way it is presented has to be defined. This is done via the *plot* command, which is typically the last command in a driver file. Although this command ranges over several lines in the example below it is only a single command. Breaking the lines by using a backslash (\) gives a better overview. Please take care that the backslash is the last sign of a line. Even blanks are not allowed after this sign and would crash gnuplot while generating the plot.

NOTE:
When manually manipulating a gnuplot driver file a very frequent mistake is that a backslash is not the last sign in a line when breaking the *plot* command. Even blanks or comments are not allowed.

An example for a *plot* command over several lines is shown below:

```
plot \
"allnum.cat" u ($01):($2) w steps lt 01 lw 02 t "Amors", \
"allnum.cat" u ($01):($3) w steps lt 02 lw 02 t "Apollos", \
"allnum.cat" u ($01):($4) w steps lt 03 lw 02 t "Atens", \
"allnum.cat" u ($01):($9) w steps lt 03 lw 02 t "Total"
```

The content of the example is very simple and only of explanatory character. All data is contained in a single data file (*allnum.cat*). Each line is representing a single data set that is embedded in the diagram. There are several abbreviations used:

- The *u* is the abbreviation for *using*.
- The *w* is the abbreviation for *with*.
- The *t* is the abbreviation for *title*.
- The *lt* is the abbreviation for *linetype*.
- The *lw* is the abbreviation for *linewidth*.

The data set is defined directly after the *u*. The data set used for this plot will be the first column for the x-axis and the 18th column for the y-axis. The results will be shown in steps as typical for a histogram and with different line types for the data sets, but all of the same line with. The title that is shown in the legend is defined at the end of each line.

6.8.2 Combination of Results from Different Projects

Let us assume that three different projects (project-1, project-2 and project-3) have been used to generate results and the user wants to combine the total results of the individual plots. The folder structure within the workspace (WS-NEOPOP) would look like the following:

```
WS-NEOPOP
|
project-1
|  |
|  output (Results of the first simulation)
```

```
|
project-2
|  |
|  output (Results of the second simulation)
|
project-3
   |
   output (Results of the current simulation)
```

The user has now to modify the above-shown *plot* command to refer to the other output files which by default all have the same name but are located in different folders as shown above. First of all the user can delete the first three data sets (“Amors”, “Apollos” and “Atens”) and copy the last line for the data set “total” two times. Now three identical data sets are in the plot command of which the first two lines should end with “, \”. The titles can be modified to “Total-1”, “Total-2” and “Total-3” as well as the line types have to be changed.

Referring finally to the other projects would lead to the following plot command:

```
plot \
"../../project-1/output/neopop.spc" u ($01):($9) ...
"../../project-2/output/neopop.spc" u ($01):($9) ...
                        "neopop.spc" u ($01):($9) ...

... w steps lt 01 lw 02 t "Total-1", \
... w steps lt 02 lw 02 t "Total-2", \
... w steps lt 03 lw 02 t "Total-3"
```

6.9 “Browse” and “Edit” Buttons

The “Browse” and “Edit” buttons shown in Figure 41 on the right side are used in various places of the GUI. Their functionality should be self-explanatory. As known from other common software the left button triggers the action to open a pop-up dialog for a file or directory browser. Thus, the user can select which file to use for a given task (e.g. the “*population.cat*” file used in the figure).

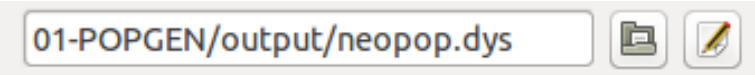

Figure 41: Input field with “Browse” and “Edit” buttons

The second button opens the file shown in the corresponding text field using the text editor defined in the settings dialog (section 6.6.11).

6.10 Model File Update

It is possible to update the model data used by the Population Generation to create populations based on the model by Granvik, Morbidelli, Bottke and collaborators. For this purpose, a new

model file is provided. This file should be saved as "gmb_model.dat" in "<installation folder>/default/01-POPGEN/data/". From then on, all GUI projects will use the updated model file. CLT users have to ensure that the projects that should use the updated model file have the file saved as "gmb_model.dat" in "<project folder>/01-POPGEN/data/". If you change the folder and/or file definitions in the configuration and/or input files, you have to adjust the file and folder names accordingly.

6.11 Error Conditions & Recover Runs

Typically the errors will be reported within the GUI if invalid input has been specified by the user. The error is marked by a sign (see Figure 42), which is shown next to the input field containing the invalid input. If this is the case the invalid data cannot be saved and used as input for NEOPOP.

Figure 42: Error symbol shown in GUI in case of invalid input

If the user is not informed by such an error symbol, because the error is caused by, in general, valid parameters, leading to internal problems, the resulting error is dumped into an error file (*neopop.err*, only existing in case of error appearance). Warnings as for example adjusting a parameter will be reported and dumped into a logfile (neopop.log), which also contains the status messages of the program execution. Both files (especially the logfile) can be consulted easily by the user after the NEOPOP software has been launched by clicking on "Show Log" or "Show Error Log" in the run dialog shown in Figure 57.

Errors occurring during the use of the GUI or the software execution that cannot be solved with the information given should be reported to ESA. To do so please refer to the information at http://neo.ssa.esa.int/neo-population. Attaching an export (see section 6.6.5) of the current project when submitting an error description may greatly help its solution.

7 Applications and Modules

This chapter describes the NEOPOP tool's applications in detail and shows how to use their modules through the GUI.

7.1 "Population Generator" Application

This application can be used, on the one hand, for the generation of a NEO population. On the other hand it can be used to analyze the generated NEO population or any other population of asteroids given in the supported formats. These two functionalities are covered by the modules "Population Generation" and "Population Analysis", which will now be discussed in detail from the point of view of a GUI user. For theoretical details on the modules the Final Report [1] should be consulted.

7.1.1 "Population Generation" Module

For this module only two buttons of the settings overview (cp. section 6.4) below the caption "Population Generator" are relevant. These buttons are called "Basic Settings" and "Population Settings". Within the "Basic Settings" (Figure 43) the user has, first of all, the chance to describe what he is going to do. The comment lines will be copied to each output file that is generated by the software. After that has been done there are three functionalities of using the module "Population Generation":

1. Generate a synthetic population
2. Generate a fictitious population
3. Generate a physical properties file only

Mode 3 only generates physical properties for an existing population, whereas modes 1 and 2 both generate a population for a selectable H-value range. Note, however, that for H-values ≤15 the actually known NEOs are used.

The H-range is specificable by the user:
For values brighter than 15 the H and a,e,i are taken from the real NEO objects file in the selected range. The angular elements are randomly chosen between 0 and 360, as for the synthetic and fictitious population.

When choosing an H larger than 25, extrapolation will be used, either with a fixed slope parameter, or with two user-selected slopes and an H where the slope shall change - to be selected before running the program in Population Settings.

Note: the selected H-range applies to both the synthetic and the fictitious population generation.

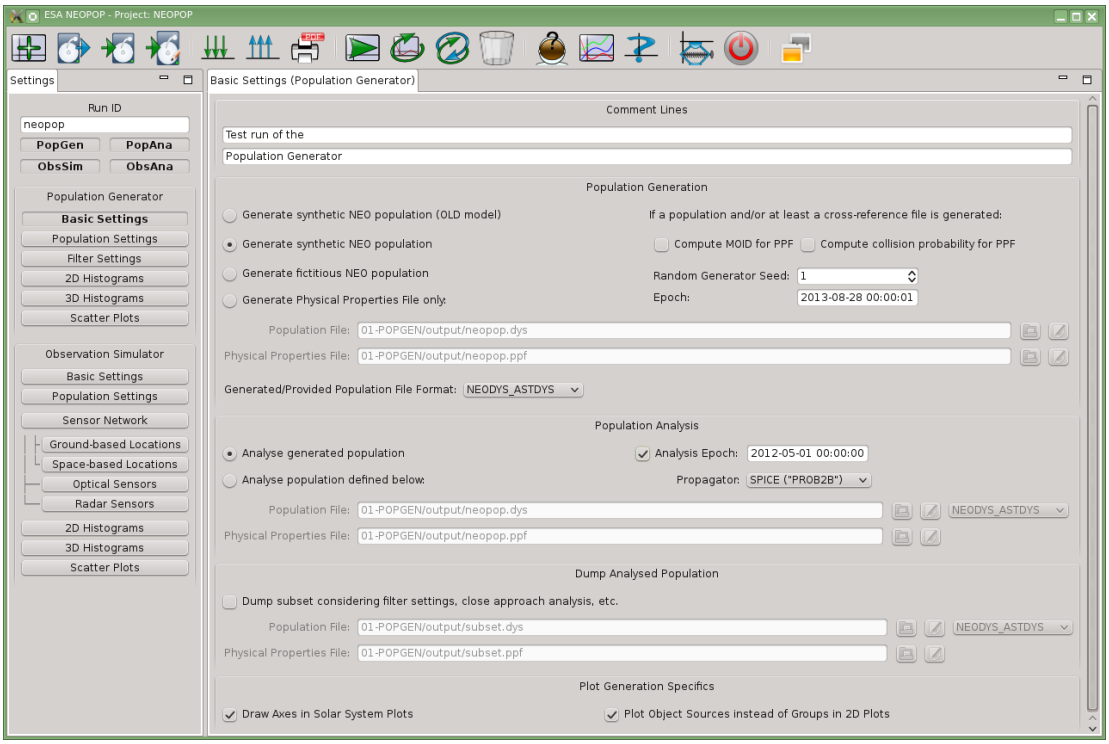

Figure 43: Basic Population Generator settings

Before the details of these functionalities are described there is the option given to the user to compute the minimum orbital intersection distance (MOID) for the generated objects and to select if the statistical collision probability shall also be generated. Finally the user may specify the name of the generated population file and its physical properties file name. The last section of the “Basic Settings” will be inaccessible as long as the module “Population Analysis” is not selected.

7.1.1.1 Generating a Synthetic NEO Population

After the basic settings have been made the “Population Settings” (Figure 44) are next. The corresponding button is only enabled if the module “Population Generation” is active and a population shall be generated. While the button is disabled if the above-mentioned functionality “Generate a physical properties file only” is selected, the access to the individual sections of this sidebar button is controlled via the other two functionalities. In case a synthetic population shall be generated the upper section is accessible. The user can select between two ways of generating a synthetic NEO population. The first one is based on the model generated in 2002 by Bottke et.al. [3, 4]. The second way is to use the most recent model generated by Granvik, Morbidelli, Bottke and collaborators in 2015. It is highly recommended to use the most recent model if you are not sure which one to use!

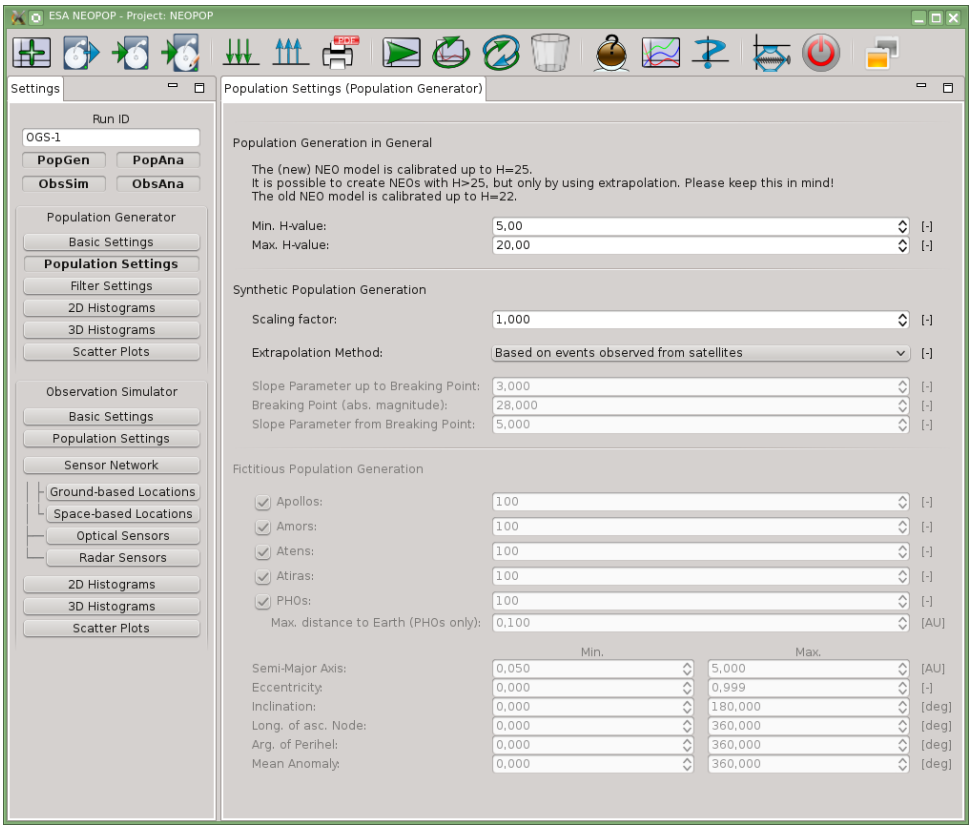

Figure 44: Population settings

Old Model by Bottke

For generating a population based on the Bottke model [3, 4] the user only has to specify a max. H-value (the minimum is used as a filter though). This max. absolute magnitude value controls the number of objects being generated. In addition a scaling factor can be defined, which scales the number of objects being generated. Typically this factor is 1.0 but for some cases (depending on what the user wants to do with the generated population) it is of interest to clone the generated objects. Additionally, there are seven data files required for the population generation process together with the so-called alpha values. Each data file and its alpha value represent a source region (the region in space where the object comes from; alpha values serve as weighting factors between the different source regions.). Although these values and files can be changed, they are only accessible via CLT input file and it is highly recommended not to change the values or the files' content.

Note: The old model is deprecated and it is strongly recommended to not use it anymore.

Model by Granvik & Morbidelli

For a population generation based on the most recent model by Granvik, Morbidelli, Bottke and collaborators users have to define a minimal and maximal H value. In addition the extrapolation method for H values above the model's maximum has to be set which can be based on bolide observations by satellites or "User-defined Slope". In the latter case users provide three parameters to define the slope: A breaking point in H defined by absolute

magnitude, a slope parameter defining interpolation up to breaking point and another one defining interpolation from breaking point on (all without unit). See Final Report [1] for details.

7.1.1.2 Generating a Fictitious NEO Population

The section at the bottom of the "Population Settings" will be enabled only if users want to generate a fictitious population. The NEO population generated when using this functionality is completely randomized and has no relation to the scientifically-generated models which are used when generating a synthetic NEO population. The way the fictitious NEO population is generated is that the user first has to decide on the H value interval (just like when generating a synthetic population). Then he can define a certain number of objects for the given NEO types (e.g. Atens, Apollos, etc.). Additionally, it is possible to define a certain number of objects that will have a close approach to Earth below a user defined distance.

7.1.1.3 Generate a "Physical Properties File" only

A physical properties file (PPF) is generated also when using the two functionalities mentioned before. Thus this option can be used if the user already is providing a population file. In this case the file names defined in the basic settings represent the existing population file and the to-be-generated PPF. The PPF contains additional object data that could not be included in the population file, where already-existing formats are supported. The PPF is used for the analysis of the generated (or existing) population and required especially by the application "Observation Simulator".

A general note about the population file read in for this generation mode: Comment lines must always begin with "!".

7.1.1.4 Results of This Module

The main results of this module are the generated population file together with the physical properties file. In addition there will be a general output file in form of a short description of the population results. This summary is the only content that will be displayed by the GUI.

The different output formats of the population files (MPC, NEODyS, AstDyS) are slightly different, but contain in general the same information (especially consider the note about MPC epoch format though!). A Physical Properties File (PPF) is created in any case in the same fashion. The content of the mentioned output file formats is as follows:

MPC

- *Designation – Des'n*
  max seven digits of designation. This limits the output to 999999 orbital elements
- *Absolute Magnitude – H*
  the magnitude at 1 AU from the Sun and the Earth, at zero solar phase angle
- *Slope Parameter – G*
  the slope parameter in the H,G magnitude system - see Bowell et al. 1989
- *Osculating Epoch – Epoch*
  the date the orbital elements refer to – in compact MPC format, (see http://www.minorplanetcenter.net/iau/info/PackedDates.html)

NOTE: This epoch format does only contain information about a year, a month and a day! With the smallest time step being one day, this thus may lead to slightly different results when running an observation analysis on two populations created with the same settings, but saved in different formats.

- *Mean Anomaly – M*
  Given in degrees
- *Argument of Perihelion – Peri*
  Given in degrees
- *Longitude of Ascending Node – Node*
  Given in degrees
- *Inclination – Incl.*
  Given in degrees
- *Eccentricity – e*
- *Daily motion – n*
  Given in degrees per day
- *Semi-major Axis – a*
  Given in AU

NEODyS/AstDyS

- *Name – Object identifier*
  11 digits
- *Osculating Epoch – Epoch*
  the date the orbital elements refer to – in modified Julian Date (MJD)
- *Semi-major Axis – a*
  Given in AU
- *Eccentricity – e*
- *Inclination – i*
  Given in degrees
- *Longitude of Ascending Node – long. node*
  Given in degrees
- *Argument of Perihelion – arg. Peric.*
  Given in degrees
- *Mean Anomaly – mean anomaly*
  Given in degrees
- *Absolute Magnitude – H*
  the magnitude at 1 AU from the Sun and the Earth, at zero solar phase angle
- *Slope Parameter – G*
  the slope parameter in the H,G magnitude system - see Bowell et al. 1989

DES

- *Object identifier – oid*
  max. 7 digits
- *Type of orbital elements – ETY*
  here KEP is used for keplerian elements

- *Semi-major Axis – a*
  Given in AU
- *Eccentricity – e*
- *Inclination – I*
  Given in degree
- *Longitude of Ascending Node – Omega*
  Given in degree
- *Argument of Perihelion – argperi*
  Given in degree
- *Mean Anomaly – mean_an*
  Given in degree
- *Absolute Magnitude – absmag*
- *Epoch – epoch(MJD)*
- *Index – inx (not used)*
- *N_par – n_par*
  Not used parameter
- *Minimum Orbital Intersection Distance – moid*
  Given in AU

PPF

- *ID, referencing ID in population file – ID*
- *Orbit group (Apollo, Aten, …) – GRP*
  Possible values are:
  'AMO': Amors
  'APO': Apollos
  'ATE': Atens
  'ATI': Atiras
  'MBA': Main Belt Asteroids
  'JTR': Jupiter Trojans
  'OTH': Others
- *Most probable source region according to Granvik, Morbidelli, Bottke and collaborators NEO model – SRC*
  Possible values are:
  "2/1": Jupiter 2:1 resonance objects
  "3/1": Jupiter 3:1 resonance objects
  "5/2": Jupiter 5:2 resonance objects
  "NU6": nu6 secular resonance objects
  "HUN": Hungaria
  "JFC": Jupiter Family Comets
  "PHO": Phocea
  "UNK": Unknown (default)
  Is set during synthetic generation based on the Granvik , Morbidelli, Bottke and collaborators NEO model for objects in the boundaries of the model and with H<15.
  Is set during generation of only a PPF for an external population file for objects in the boundaries of the model.

Source determination is based upon source affiliation probabilities per NEO model a-e-i-H cell. They are listed in the NEO model file which can be found at `<installation folder>/default/01-POPGEN/gmb_model.dat`

- *Scaling Factor (object class contains <scale> elements) – Factor*
- *Minimum Orbital Intersection Distance (MOID) wrt. Earth – MOID*
- *Statistical Collision Probability wrt. Earth – Collprob.*
- *Diameter and its standard deviation – Diameter+Sig*
- *Visual albedo and its standard deviation – p_vis+Sig*

7.1.2 "Population Analysis" Module

This module can be used in combination with the module "Population Generation" or alone. Thus it is possible to analyze a freshly generated population or an already-existing one. The main task of this functionality is to visualize the provided population. Additionally, it is also possible to filter the given data to consider only a subset of the given population. This filtering can be done by individual parameters (e.g. range of semi-major axis) or by mathematical methods to consider for example only those objects having a close approach to Earth within a given time frame.

How to Analyze a Population

Defining input parameters for the population analysis starts already in the "Basic Settings". While these settings are used for both, the Population Generation and the Population Analysis, "Filter Settings", "2D Histograms", "3D Histograms" and "Scatter Plots" are exclusively used by the Population Analysis.

Basic Settings

Within the panel "basic settings" (Figure 43) the user has to specify whether a generated or a defined population shall be analyzed. In case a generated population shall be used the software will make use of the generated file automatically, without a file definition of the user. The generated population has not to be generated in the same simulation run (together with the analysis), because the file name is related to the given run id, which has to be the same as used when generating the population. In case a user-defined population shall be used the filenames for the population and the physical properties file have to be defined.

Furthermore, an epoch for which the population shall be analyzed can be defined. All objects are propagated in that case (with the simple two-body propagator SPICE PROP2b, see Final Report for details [1]) to the given epoch at the very beginning of the analysis. This is of interest for the generated figures where for example the distance relative to earth or sun is displayed for the given analysis epoch or for the epoch of the orbit data as given in the population file (if the analysis epoch is not defined respectively selected to be used). Note that the analysis epoch year is constrained to [1900, 2100]. This is owed to the significant propagation errors with the implemented propagator over longer times.

There is also the option to dump a subset of the analyzed population. If this option is selected, the population subset that passes all filter criteria (explained in the following) and is thus finally

available to be considered for the plot generation, will be dumped into the files as defined by the user.

At the bottom of the Basic Population Generator Settings users can decide specifics about plot generation. See “Defining the Resulting Figures” beneath for details.

Filter Settings

Within the filter settings (Figure 45) the user has the chance to exclude objects from being analyzed. This is of interest if the user for example is interested in the objects being inside the orbit of the inner planets. In the top left part of these settings it is possible to include the asteroid classes of interest. Besides the NEO classes (Atiras, Apollos, Amors, Atens) there are also the asteroid classes included, because the user might provide a population including also these objects (e.g. by a download from the AstDyS homepage[1]).

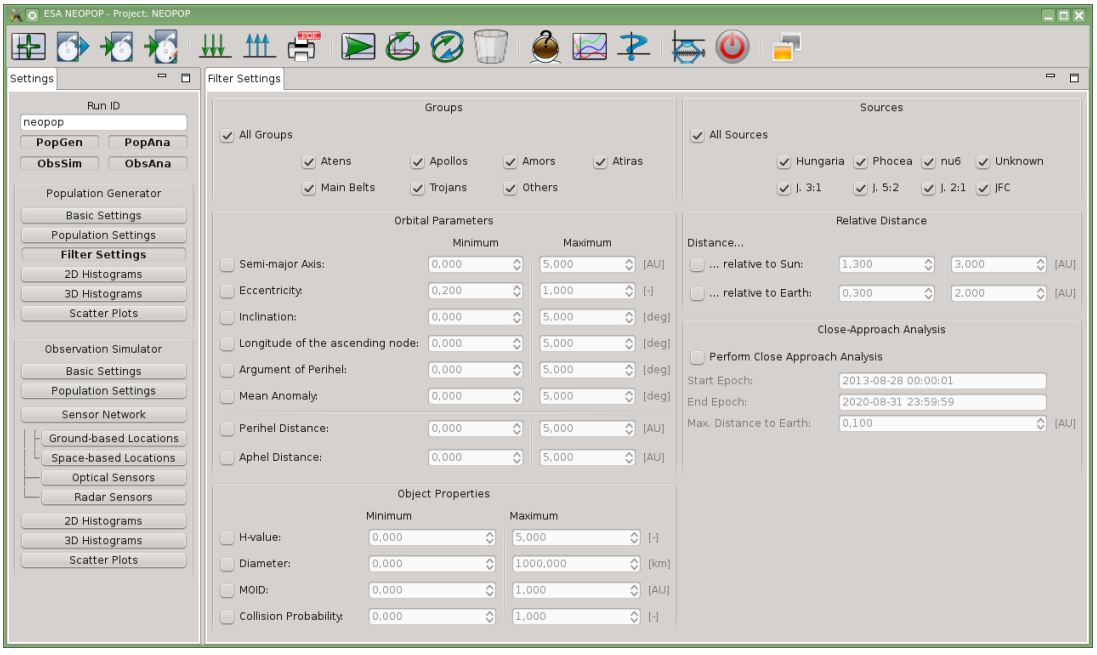

Figure 45: Filter settings

On the top right of the Filter Settings object sources (object region/class of origin before object came into near-Earth space, i.e. Hungaria and Phocea asteroid families, v6-, 3:1-, 5:2-, or 2:1-resonance regions, or Jupiter Family Comets) can be filtered in the same way as object groups. The most probable source region of an object is determined solely based on the distribution of objects from different source regions to a-e-i-H cells according to the Granvik, Morbidelli, Bottke and collaborators NEO model. When importing an external population or generating a population using the old model or fictitious model creation, the most probable source region is also chosen on this basis considering their a-e-i-H properties. The probabilities for each a-e-i-H cell can be found in the NEO model file at `<installation folder>/default/01-POPGEN/gmb_model.dat`. See also section 7.1.1.4.

[1] http://hamilton.dm.unipi.it/astdys/

In the remaining part of this panel, the user can select parameters and specify their ranges. These parameters are checked for each object of the included classes. Objects, where at least one of the selected parameters is outside of the defined range, are skipped from being further analyzed. If the close approach analysis is selected all objects not approaching Earth closer than the given maximum distance within the defined time frame will be excluded. Therefore the objects are propagated (with the simple two-body propagator SPICE PROP2b, see Final Report for details [1]) from the user-defined start and end epoch for close approach analysis. Keep in mind that the implemented close approach analysis is only an approximation and might lead to different results for known objects if compared to published results.

The values for “Relative Distance” refer to the distance between the object and Sun/Earth at the analysis epoch (if defined respectively selected to be used) or at the orbit epoch of each object as given in the population file.

Defining the Resulting Figures

The plots the user wants the Population Analysis to create can be defined via the Population Generator plot settings ("2D Histograms", "3D Histograms", "Scatter Plots", Figure 48) and the Basic Population Generator Settings (Figure 43). The plot settings contain the following parameters for each plot axis:

- Logarithmic / Linear scale:
  The axis can be defined in a logarithmic or linear scale.

- Minimum:
  The given value is used as lower limit for the axis shown in the plot. This means also that data below this value is not represented in the resulting plot.

- Maximum:
  The given value is used as upper limit for the axis shown in the plot. This means also that data above this value is not represented in the resulting plot.

- Class Width / Class Number
  The resolution of the histogram can be defined in two ways. The user may specify either the class width to be used for generating the histogram classes (or bins) or a predefined number of classes. In the latter case the class width would be computed internally by dividing the range between the given minimum and maximum by the given class number. If the class width is predefined the software will compute the number of classes internally with a possible increase of the upper limit (Maximum).

  Because there is only a single value to be defined as class width or as class number the user has to define by a switch the meaning of this value.

For scatter plots the same parameters have to be defined except for the class settings.

The bottom of the Basic Population Generator Settings yields two more options for defining the plots that are generated:

- Axes can be drawn in Solar System Plots or not.
- 2D histogram plots can show a division of either object group (Amors, Apollos, …) or source (Jupiter 2:1 resonance, Hungaria, …)

Population Analysis produces two Solar System Plots by default (Figure 46 and Figure 47). They both show the solar system with Jupiter's orbit being the boundary. One of them shows the solar system from "above" (Polar Solar System Plot) and the other from the side (Solar System Side Plot). Each green dot in the plot represents a NEO which is why this plot type is categorized as a scatter plot in the GUI.

Table 1 gives an overview of the orientation of the Solar System Plots.

	-x	**+y**
Polar Plot	Vernal Equinox	View from ecliptic North Pole
Side Plot	Vernal Equinox	Ecliptic North

Table 3: Orientation of the Solar System Overview Plots

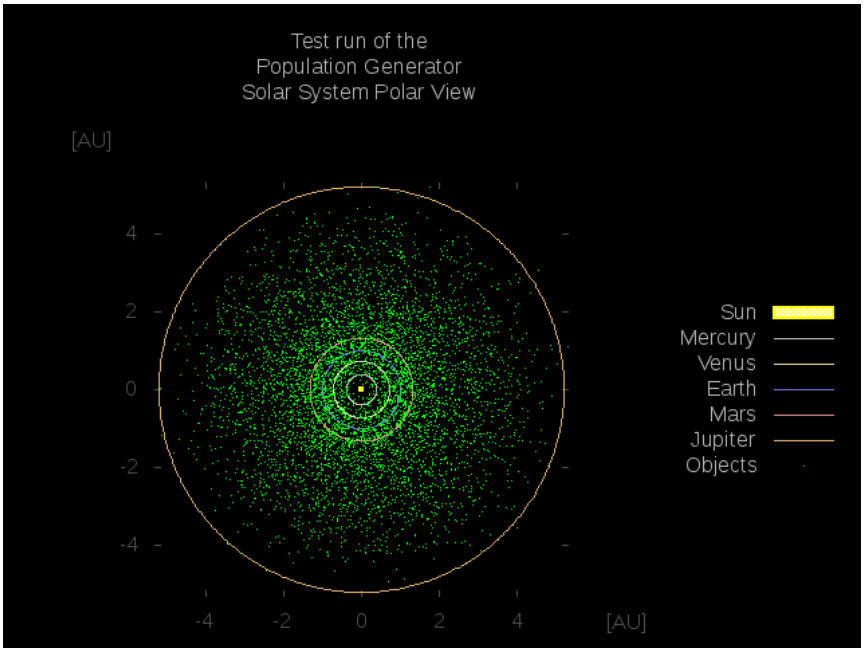

Figure 46: Polar Solar System Plot

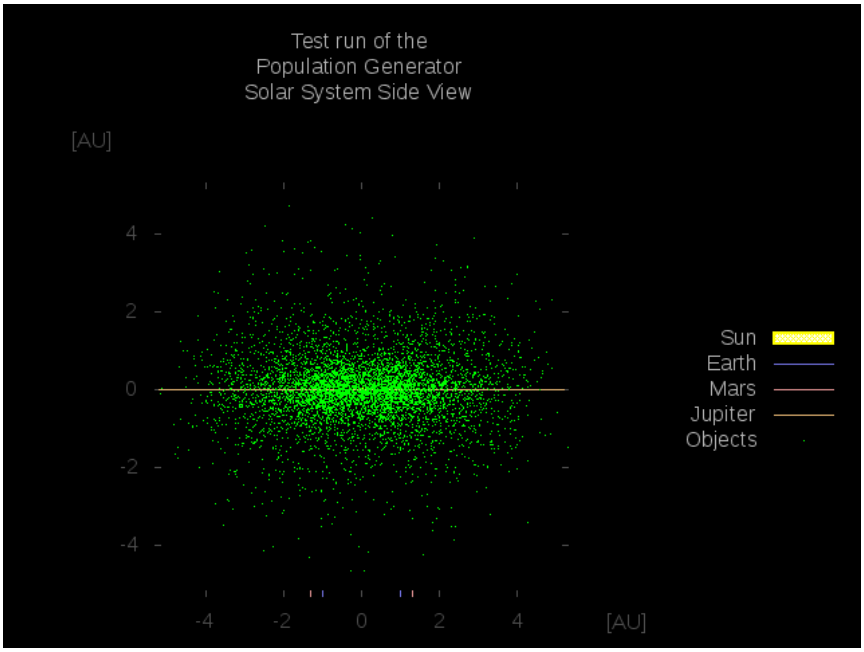

Figure 47: Solar System Side Plot

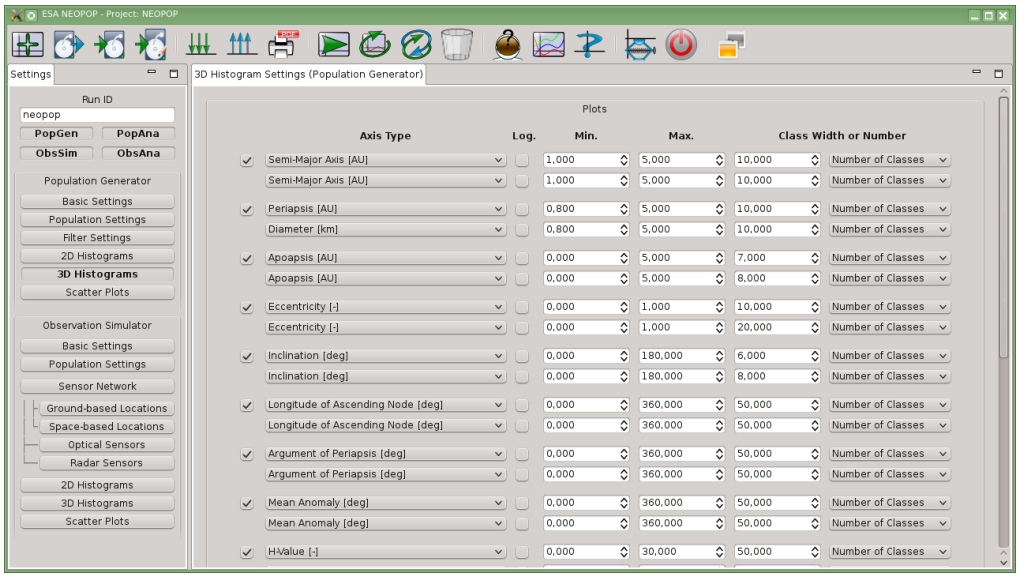

Figure 48: 3D Population Generator histogram settings. For defining 2D Histograms only a single line per plot is required, while for defining scatter plots the class definition is not required.

7.2 “Observation Simulator” Application

This application can be used on the one hand for the simulation of the observation of a NEO or, in general, an asteroid population. On the other hand it can be used to analyze the generated results from the observation simulation part. These two functionalities are covered by the modules “Observation Simulation” and “Observation Analysis”, which will now be discussed in detail from the point of view of a GUI user. For theoretical details on the modules the Final Report [1] should be consulted.

7.2.1 “Observation Simulation” Module

Several buttons of the settings overview (see section 6.4) belong to this module. The first three (“Basic Settings”, “Population Settings” and “Sensor Systems”) are of special interest for the long-term use of this module. The next four sidebars (“Ground Locations”, “Space Locations”, “Optical Sensors” and “Radar Sensors”) have to be considered as a way to maintain a database of locations and sensors, which typically will be done less frequently.

The end output files produced by this module are described in section 7.2.1.8.

7.2.1.1 Basic Settings

Also for this application the first requested input data is the comment lines (cp. Figure 49) which will be written into the header of each generated output file.

After this has been done one of two options has to be selected defining the way the observation simulation is being performed. The first option is the “Statistical Analysis”. This option will perform a statistical analysis of the provided population by the means that Monte-Carlo runs can be performed and the given orbit will slightly be randomized (random selection of the mean

anomaly and slight variation of diameter, albedo, inclination, longitude of ascending node (LAN), and argument of perigee (AOP). See Table 4). The “Deterministic Analysis” instead will take the object’s orbit as given in the input file and no Monte-Carlo simulations are possible. Instead some parts of the simulation will be enabled that will not be considered for statistical analysis (e.g. atmospheric refraction).

Parameter	Interval of randomization
Mean Anomaly	0..360°
Albedo	Corresponding bin mean in PPF ± std. deviation of bin
Inclination	Population file value ± 1°
Diameter	Corresponding bin mean in PPF ± std. deviation of bin
LAN	Population file value ± 1°
AOP	Population file value ± 1°

Table 4 Randomized object parameters during Monte-Carlo runs

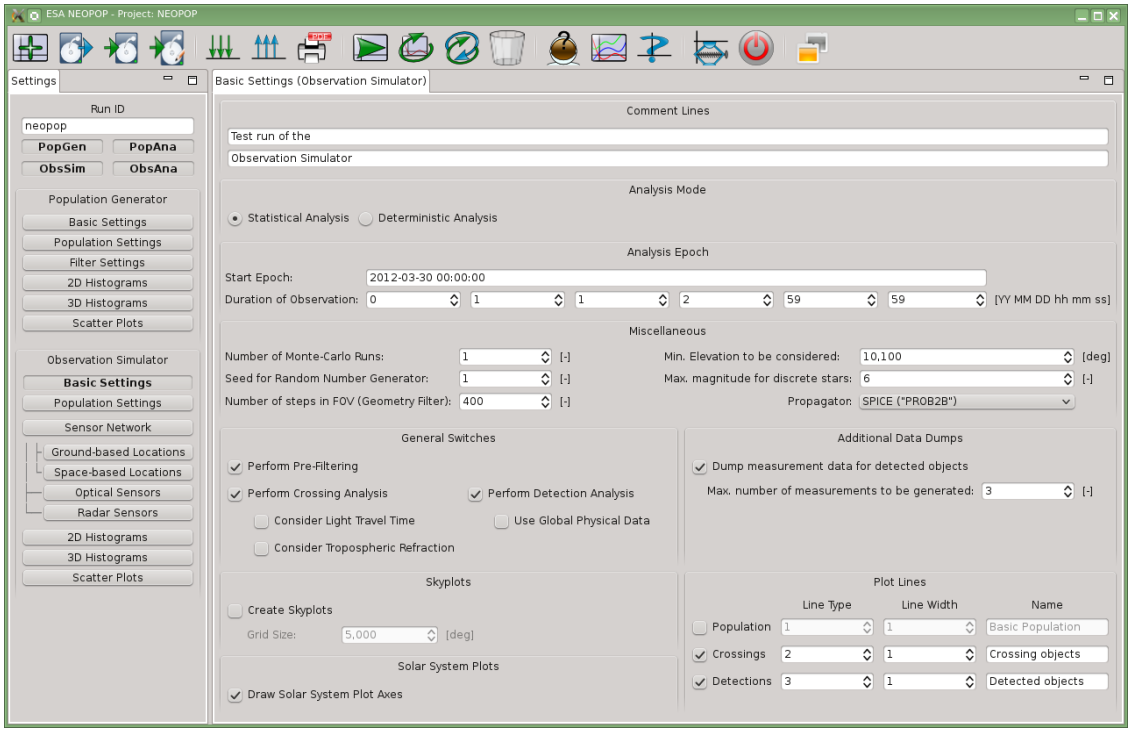

Figure 49: Basic Observation Simulator settings

Setting up the observation parameters starts with the definition of a global start time of the format [YYYY-MM-DD hh:mm:ss]. The length of the observation can be specified by a similar format not covering a date but only the duration in the format [YY MM DD hh mm ss].

Several parameters have to be defined in the following:

“Number of Monte Carlo Runs”:
This value gives the number of Monte-Carlo-Simulations being performed in the “Statistical Analysis”. In case of a “Deterministic Analysis” this value will be inactive.

"Seed for Random Number Generator":
This value is used to initialize the random number generator for the "Deterministic Analysis". For statistical analysis the random number generator will be initialized at the beginning of every Monte-Carlo-Simulation with a different value based on the here defined one.

"Number of steps in FOV":
This value is used when the crossing geometry is computed. It represents how exact the path of the object while being inside the FOV is internally stored as a number of steps.

"Minimum Elevation to be considered":
For objects crossing the FOV the part below the given minimum elevation will be cut off. This is to have an option to consider the surrounding topography (e.g. houses, trees, etc.)

"Max. Magnitude of discrete stars":
For deterministic analysis, this value defines the largest magnitude of a star that is treated as a discrete point source for the background signal computation of telescope simulations. Although this value is not required (and has no changing effect) if only radars are simulated it will not be deactivated in such a case.

After these settings have been made the "General Switches" have to be set. These switches are the top level controllers for the parts of the Observation Simulation that will be executed.

"Perform Pre-Filtering":
When this switch is selected the software will skip all objects from being analyzed that will definitely have no chance to cross the field of view of each defined sensor. For example, in case of a heliocentric sensor with a line of sight in anti-sun direction all objects with an aphelion below the perihelion of the sensor orbit can be excluded. The intention of this filter step is to reduce the computational effort. Besides the orientation of the line-of-sight also a minimum and maximum range is considered for this filter. This switch is optional and does not restrict the user from performing a crossing and detection analysis.

"Perform Crossing Analysis":
When this switch is selected the objects (remaining after the pre-filtering – if selected) will be analyzed to determine whether they are crossing the FOV or not (identification phase). As a result, a list of crossings will be generated containing object, sensor and crossing parameters (e.g. object and sensor state, FOV dwell time, line of sight orientation, etc.), allowing to reconstruct the so-called crossing geometry. Related to this switch there are two sub-switches which are enabled in case of a deterministic analysis only. Thus the user may select to "Consider Light Travel Time" and to "Consider Tropospheric Refraction" for ground based sensors.

"Perform Detection Analysis":
This switch will be enabled only if the switch "Perform Crossing Analysis" is also selected. If a detection analysis shall be performed for each simulated sensor the required performance model will be applied for each crossing. The list of crossings will be completed with parameters related to the detection process (e.g. object magnitude, mean signal to noise ratio, etc.).

"Plot Lines":
Here you can define what lines NEOPOP shall draw into the plots. You can select the base population, crossings and detections. Additionally, you can change the name or label of the lines and adjust line width and line type.

You can also tell NEOPOP to create simulation data needed by the Observation Analysis to create so-called skyplots (see section 7.2.2). With the grid size users can adjust the skyplots' resolution. The data for the skyplots is generated by the "Observation Simulation" while the plots are finally generated by the "Observation Analysis".

Note that, if skyplot generation is activated, NEOPOP doesn't use a provided observation strategy file, but internally generates one. Also note that skyplots don't show crossings and detections for the whole observation duration. You should instead think of them as a snapshot showing the situation at observation start.

Finally it is possible to generate additional data. The user can generate measurements for the detected objects in case of a deterministic analysis. For optical sensor systems the generated measurements are right ascension and declination, while for radar systems in addition the range to the detected object is dumped.

7.2.1.2 Population Settings

The "Population Settings" (Figure 45) allow the user to define the population that shall be used for the simulation. It is possible to define two population files together with the related "Physical Properties Files" (PPF). These two populations are internally combined and handled as a single population, but this option allows to provide the software with the NEO population (as for example created with the application "Population Generator" – see chapter 6.7) and with an additional population containing for example the remaining asteroid population.

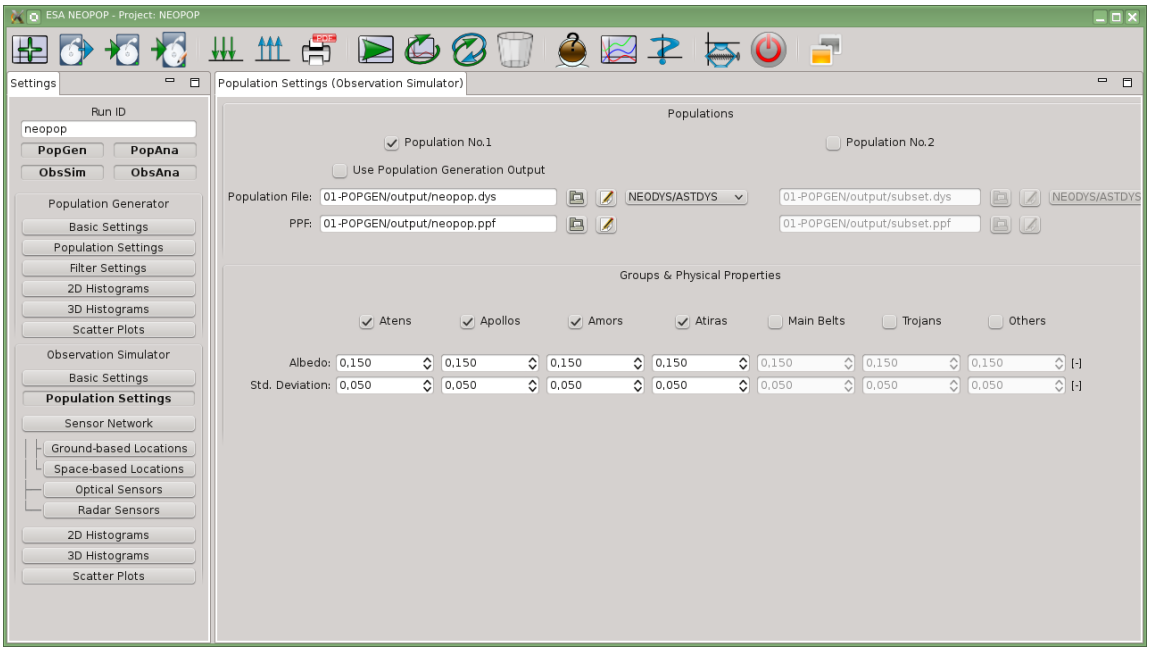

Figure 50: Observation Simulator Population Settings

A general note about the population file(s) read in for observation simulation: Comment lines must always begin with "!".

The second section of this panel allows excluding objects from being simulated although they are included in the provided population file(s). There is an auxiliary switch that can be used to select or deselect all object groups (Note: at least one has to be selected before starting the simulation). For each group it is possible to define some physical properties that will overwrite the properties as defined in the PPF when the switch "Use Global Physical Properties" is selected.

7.2.1.3 Sensor System Settings

The "Sensor Systems" button (Figure 51) is probably the most important sidebar button for defining the sensor architecture. The approach that is followed here is that this panel heavily relies on four databases containing possible locations and sensors. These databases can be maintained by accessing the four sidebar buttons below the "Sensor Systems" button (see chapter 7.2.1.4 to 7.2.1.7).

The left side of this panel contains a list of sensors that have been defined already. Sensors can be selected or deselected to consider or exclude them. The right part shows the settings of the selected sensor system. It starts with the "Designator" of the sensor system followed by the general definition of the system being a "Ground-based" or "Space-based" system, an "Optical" or a "Radar" sensor.

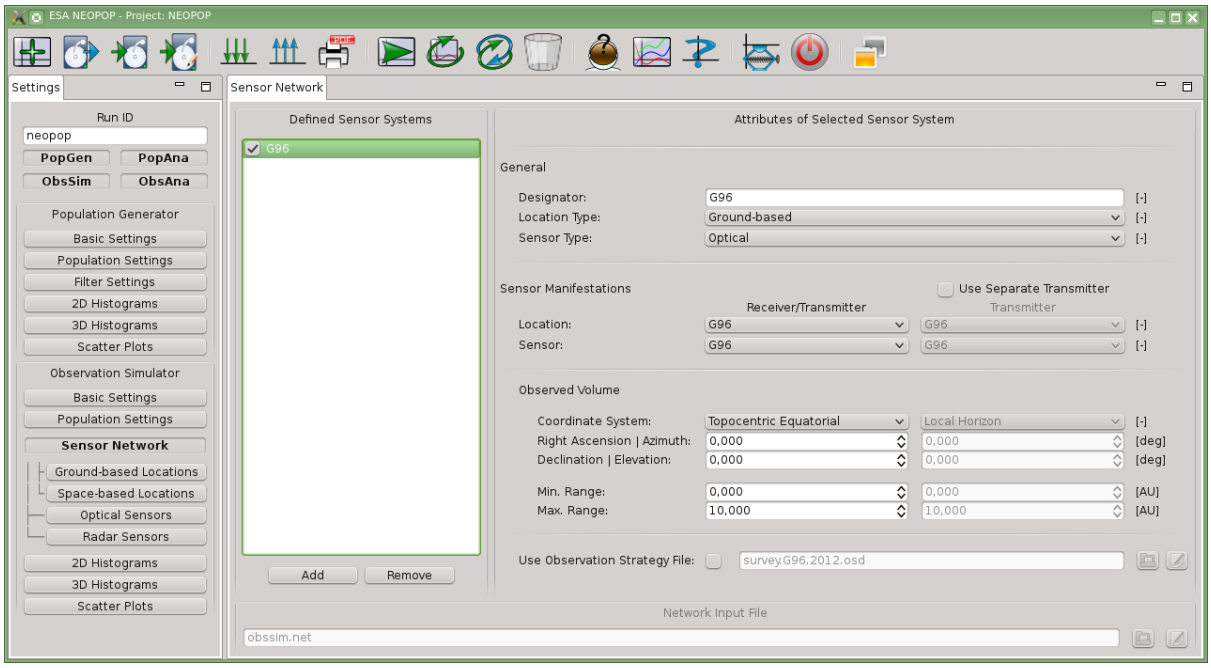

Figure 51: Sensor network settings

Depending on the general system definition (Ground-Based/Space-Based and Optical/Radar) the drop-down menus in the middle of the panel contain the list of elements defined in the related database. For example, in case the user selects “Optical” as sensor type, the drop-down menu for sensors contains only the list of sensors included in the database for “Optical Sensors” (see chapter 7.2.1.6).

After the location and the sensor have been selected, the orientation of the sensor has to be defined. First of all there are three reference frames implemented. Together with two offset angles the orientation is defined (see Table 5).

	“Local Horizon”	**“Topocentric Equatorial”**	**“Topocentric Ecliptical”**
Offset 1	Azimuth	Right Ascension	Ecliptic Longitude
Offset 2	Elevation	Declination	Ecliptic Latitude

Table 5: Sensor Orientation

In addition to the line of sight definition a range limit can be defined. These limitations can for example be used to exclude an object from being considered as a crossing object when they are, on the one hand, inside the field of view but, on the other hand, below the given minimum limitation and vice versa.

Finally an observation strategy can be provided for each sensor. The observation strategy has to be defined in an ASCII file as shown in Table 6 for example. Therein only those parameters that have to be changed at least once have to be defined. All other parameters will be taken from the standard way of defining the input settings.
Please note that if a readable ASCII file is linked that does not follow the required format, NEOPOP will still run. As an example, in Table 6 the correct values to insert in column 101, when defining the reference frame, should be 1, 2 or 3. Nonetheless, if a floating point number

is entered, the code will still run reading just its integer part. In any case, to ensure correct results, please check that the observation strategy file you provide has the correct format.

1	2	3	101	102	103
2013	01	01	2.00	75.11	11.12
2014	02	01	2.00	24.86	15.23
2015	03	02	1.00	33.46	08.90
2016	04	02	1.00	75.77	06.78

Table 6: Exemplary Observation Strategy File

The first non-comment line of the ASCII defines the content of each column by using ID values. The valid values and their related parameters are listed in Table 7:

Related Parameter	Value	
	Transmitter	Receiver
Start of observation (Year)	1	
Start of observation (Month)	2	
Start of observation (Day)	3	
Start of observation (Hour)	4	
Start of observation (Minute)	5	
Start of observation (Second)	6	
Duration of observation (Year)	7	
Duration of observation (Month)	8	
Duration of observation (Day)	9	
Duration of observation (Hour)	10	
Duration of observation (Minute)	11	
Duration of observation (Second)	12	
Flag for coordinate system (1 = Local Horizon, 2 = Topocentric Equatorial, 3 = Topocentric Ecliptical)	101	201
Orientation of LOS (AZ, RA, LON)	102	202
Orientation of LOS (EL, DE, LAT)	103	203
Minimum Range	104	204
Maximum Range	105	205
Atm. Condition (Temperature)	106	206
Atm. Condition (Pressure)	107	207
Atm. Condition (Humidity)	108	208
Atm. Condition (Lapse rate)	109	209
Field of View	110	210
Integration Time	111	211
Gap Time	112	212
Min. number of consec. frames for Det.	113	213
Limiting Magnitude	114	214
Diameter of Aperture	115	215
Pixel Size	116	216
Number of Pixels per Row	117	217
Scale (FOV per pixel)	118	218
FWHM (full width half maximum)	119	219

Related Parameter	Value	
	Transmitter	Receiver
Threshold parameter for detection	120	220
Count rate constant for CCD readout noise	121	221
Count rate constant for dark noise	122	222
Sensor error: Bias – (AZI, RA, LON)	123	223
Sensor error: Noise – (AZI, RA, LON)	124	224
Sensor error: Drift – (AZI, RA, LON)	125	225
Sensor error: Bias – (ELE, DEC, LAT)	126	226
Sensor error: Noise – (ELE, DEC, LAT)	127	227
Sensor error: Drift – (ELE, DEC, LAT)	128	228
Sensor error: Bias – (Range)	129	229
Sensor error: Noise – (Range)	130	230
Sensor error: Drift – (Range)	131	231

Table 7: Valid values and related parameters for the definition of an observation strategy

The network definition makes use of the databases, which are explained in the following four sections. A schematic view in form of a tree structure is given below:

7.2.1.4 Ground-based Sensor Location Settings

By pressing the sidebar button “Ground Locations” the panel for the definition of ground based sensors is shown (Figure 52). This panel is the first of in total four panels, which are used to define the underlying databases. In this case this panel can be used to maintain the database of sensor sites located on ground. There is a dedicated panel following to define the orbits of space based sensors. Also for the two sensor types there are separate panels to maintain the databases of optical and radar sensors.

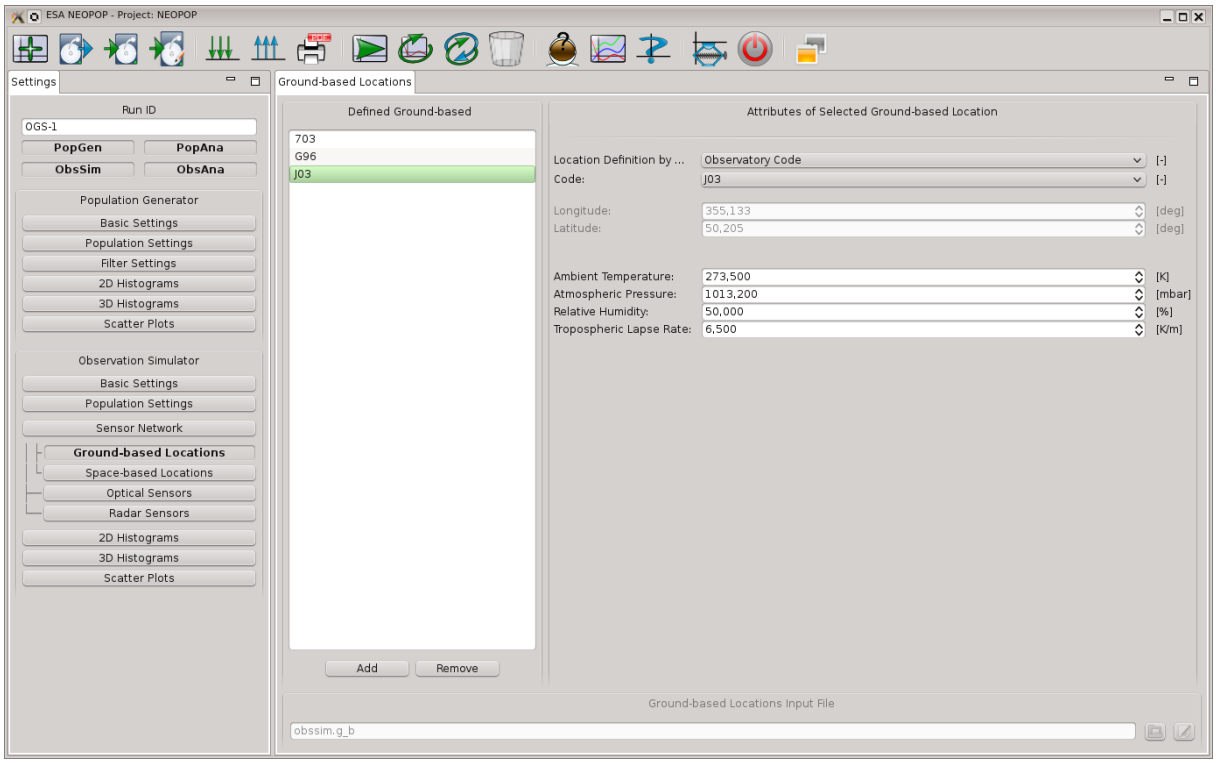

Figure 52: Ground-based Sensor Location Settings

The panel shown gives an overview over the defined ground based sensor sites. The left side of the panel (next to the sidebar) contains a list of all sites that have been defined so far. It is possible to add or remove sensor site from the database by using the buttons “Add” and “Remove”, which are located right below the list of sensors. On the panel’s right site the parameters of the selected sensor site are shown. It is possible to define the location by four different ways:

- Geocentric Coordinates: Longitude (East) / geoc. latitude / geoc. distance
- Geodetic Coordinates: Longitude (East) / geod. latitude / altitude
- Earth fixed state: State vector
- Observatory List: Designator is used to get data of observatory code list.

Before editing the location parameters the user may change the default designator. The designator is used to identify the sensor site. When setting up a sensor system (see section 7.2.1.3), the designators are shown in a dropdown menu to be selected. When a sensor site is selected out of the list of observatory codes the selected code will be used as designator and cannot be changed by the user. The related site definition parameters are shown to the user when using the GUI but cannot be changed. When using the command line version only, the user may define the code only. The software itself looks up the related parameters.

NOTE:
The observatory code list is not updated automatically and updates of NEOPOP may not include an update of this list. If you want to use the most recent version, you can get it from

the Minor Planet Center's website (www.minorplanetcenter.net) and put it into `<installation folder>/02-OBSSIM/data`.

Additionally it is possible to define the atmospheric conditions at the sensor site, which are used by the software in the deterministic analysis mode to compute the atmospheric refraction of the signal. The refraction model makes use of the following parameters:

- Ambient Temperature: Temperature at the sensor site in [K]
- Atmospheric pressure: Air pressure at the sensor site in [mbar]
- Relative Humidity: Relative air humidity at sensor site in [%]
- Tropospheric lapse rate: Decrease of temperature with height in [K/km]

Finally it is also possible to change between different databases for ground based sensor sites. At the bottom of the panel the user may change the file name manually or select a file by browsing through the file system after pressing the related button at the bottom right of the panel. It is also possible to change the content of the database manually. There the button next to the text field can be used. Because this has to be done very accurately it is recommended to use this button only to have a complete overview over the database inside an editor but not to change the file manually.

7.2.1.5 Space-based Sensor Location Settings

By pressing the sidebar button "Space Locations" the panel for the orbit definition of space based sensors is shown (Figure 53). This panel is the second of four panels in total which are used to define the underlying databases. In this case this panel can be used to maintain the database of geocentric and heliocentric orbits. There was a dedicated panel shown before (see section 7.2.1.4) to define the ground-based locations of sensors and there are two panels following (see section 7.2.1.6 and 7.2.1.7) to maintain the databases of optical and radar sensors.

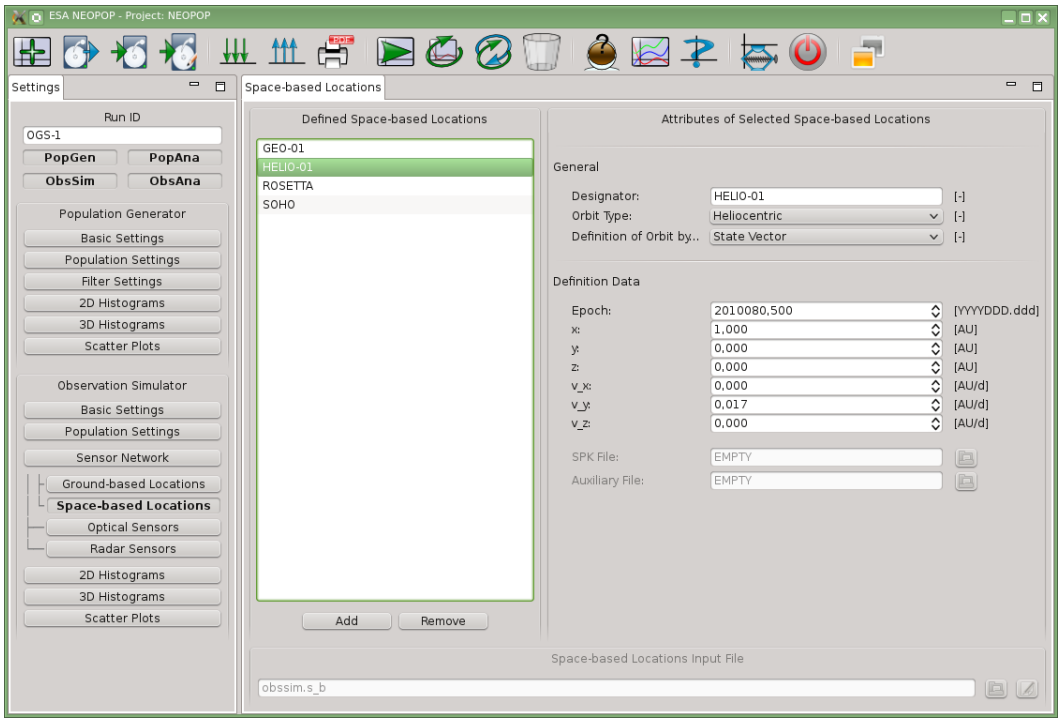

Figure 53: Space-based Sensor Location Settings

Similar to the “Ground Location” settings there is a list on the left side (next to the sidebar) showing a list of already defined space based locations. When selecting one of these the related parameters are shown on the right site of the panel. First of all the user has to specify a designator of the sensor orbit. The designators are used within the panel “Sensor Systems” as content of a drop down menu (see section 7.2.1.3). Below the designator field the type of orbit has to be selected. The user can select between a geocentric and a heliocentric Earth-like orbit. For both it is possible to select between the three types of defining the orbit itself:

- State vector
- Orbital Elements
- SPK Kernel (bsp and tls file)

Geocentric Orbit			
Orbital Elements		**State Vector**	
Semi-major Axis	[km]	x – Position	[km]
Eccentricity	[-]	y – Position	[km]
Inclination	[deg]	z – Position	[km]
Right Ascension of the Asc. Node	[deg]	$\dot{x}$ – Velocity	[km/s]
Argument of Perigee	[deg]	$\dot{y}$ – Velocity	[km/s]
Mean Anomaly	[deg]	$\dot{z}$ – Velocity	[km/s]
Heliocentric Orbit			
Orbital Elements		**State Vector**	
Semi-major Axis	[AU]	x – Position	[AU]
Eccentricity	[-]	y – Position	[AU]
Inclination	[deg]	z – Position	[AU]
Longitude of the Ascending Node	[deg]	$\dot{x}$ – Velocity	[AU/d]

Argument of Perigee	[deg]	$\dot{y}$ – Velocity	[AU/d]
Mean Anomaly	[deg]	$\dot{z}$ – Velocity	[AU/d]

Table 8: List of orbit elements depending on the selected orbit type

The section below the general settings contains the parameters, which are defining the sensor orbit. It is used for the state vector and orbital elements orbit definition. First of all the epoch for which the given data is valid has to be defined. This value has to be defined for all types of orbits in the same format ([YYYYDDD.ddd]). The other six parameters vary with the selected orbit type. This section is disabled for the orbit definition using an SPK kernel. The definition of these files in turn is enabled when the orbit definition 'SPK kernel' is selected.

All defined sensor orbits will be computed internally over time by making use of state vectors or SPK file provided data. The orbit evolution is performed as simple keplerian orbits where only for geocentric sensor the movement of the line of nodes and of the line of apsides is considered by a simple approach (see [1] for details). Because of this, the definition of sun-synchronous orbits is possible. SPK kernel files provide a satellite's position data for a given time interval. They are not propagated. The user has to make sure that the supplied SPK kernel fits to the simulation time frame. Obssim checks whether the supplied SPK kernel does not start before or exactly at the beginning of the simulation time or ends before the simulation time. Additionally the user has to consider whether the supplied SPK kernel is valid for the geocentric or heliocentric system.

7.2.1.6 Optical Sensor Settings

The third sidebar button, which is used for the background databases, is for the "Optical Sensors" (Figure 54). The two sidebar buttons above have been discussed in section 7.2.1.4 (ground based sensor sites) and 7.2.1.5 (space based sensor orbits). There is a fourth sidebar button for the database definition below the "Optical Sensors" which is used for the definition of a radar sensor database.

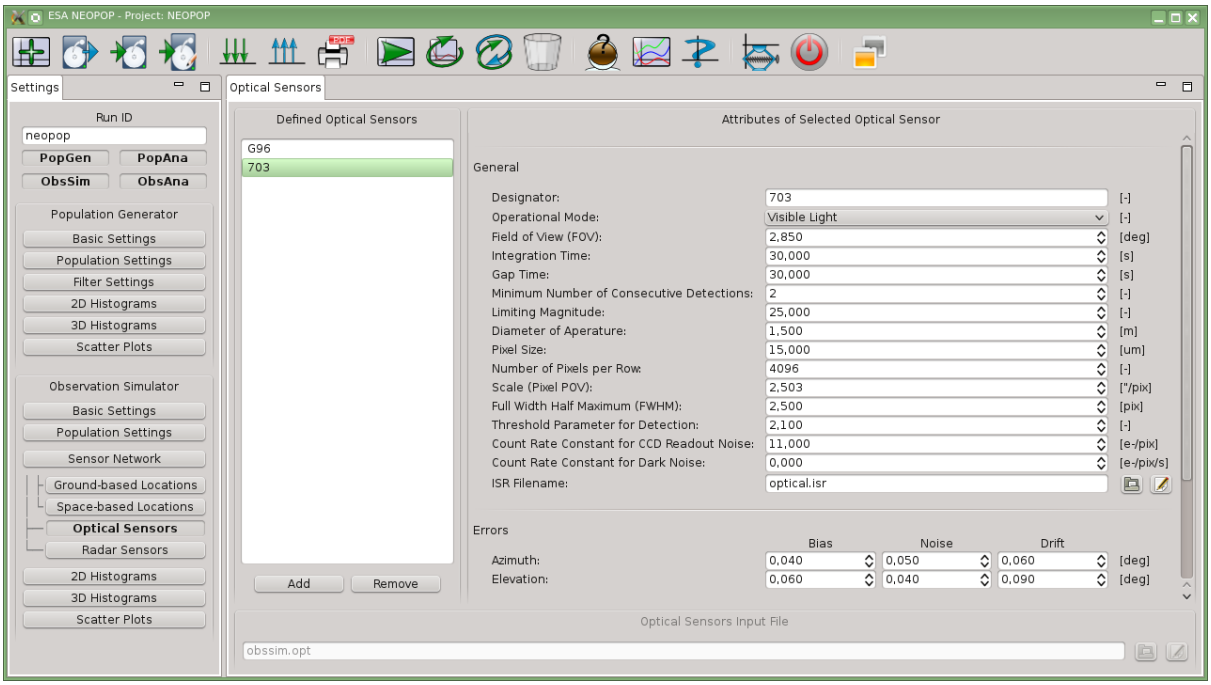

Figure 54: Optical Sensor Settings

As in all panels used for the database settings the panel for the “Optical Sensors” contains a list of all defined optical sensors next to the sidebar, while the parameters of the selected sensor are shown on the right side of the panel. It is possible to add or remove sensors by pressing the buttons below the list of sensors.

The list of related parameters starts also for the optical sensor settings with an input field for the designator. The designators are listed in the dropdown menu of the “Sensor Systems” settings. This allows an easy definition of sensor architectures. After the name of the sensor has been specified the sensor has to be defined as operating in the visible of the thermal infra-red light. A lot of parameters are listed and shortly explained in the following:

- Field of view:
  The field of view (FOV) is a circular part of the sky. By this parameter the size of the circular FOV is being defined. The orientation is defined in the sensor system settings (see 7.2.1.3). The FOV is one of three input parameters that are used to compute the focal length (see [1]). **NOTE:** For bigger FOVs it should be considered deactivating the Pre-Filter (cp. section 7.2.1.1), as such FOVs may bring its algorithms to their limits.

- Integration time:
  During the integration time the CCD chip is exposed to light. Often this parameter is also called “Exposure time”. In the following the data generated during the integration time will also be linked to the word “frame”.

- Gap time:
  The gap time is the time between two exposures of the CCD. Typically this period is required to readout the CCD.

- Min. number of consecutive detections:
  This value is used by the optical performance model for the detection decision. It represents parts of the sensor software. It is analyzed frame by frame

- Limiting Magnitude:
  The limiting magnitude is used by the optical performance model to fasten the simulation. A frame of an object will be skipped if not a single pixel contains an object magnitude lower than the defined limiting magnitude. The higher the value the more accurate is every frame computed by the software, based on the objects and the background signal.

- Diameter of Aperture:
  As given by the name this value is the diameter of the sensors aperture. This value is required to compute the objects and background signal, where the diameter of the aperture has an exponentially increasing impact.

- Pixel Size:
  First of all the pixel size is the physical size of the pixel. This parameter is one of three input values that are used to compute the focal length of the sensor system (see [1])
- Number of Pixels per Row:
  The software supports only square shaped CCDs. Thus the dimension of the CCD can be described by the number of pixels per row knowing that each row and column has the same number of pixels.

- Field of view per Pixel (*"scale"*):
  The "scale" value defines the part of the sky that is finally contained per pixel. The smaller the field of view per pixel the higher is the resolution of the sky on the CCD. This parameter is one of three input values that are used to compute the focal length of the sensor system (see [1])

- Full width at half maximum:
  To model the quality of the telescope for each point source (e.g. the object, discrete stars, etc.) a diffraction and aberration pattern is produced on the CCD matrix by a so called point spread function. Its distribution has been assumed in the form of a Gaussian. To adjust the quality of the telescope the parameter "full width at half maximum" can be used.

- Threshold parameter for detection:
  Because one cannot directly consider a user defined signal to noise ratio as detection criterion (SNR is not measurable in practice), the objects signal has to exceed the background signal (which is in practice measured in an area of the CCD without object signal in it) and the noise by a certain factor. The threshold is determined by the sum of the computed background signal and its standard variation. The latter one is scaled by the threshold parameters, which typically ranges from 1.2 to 2.5.

- Count rate constant for CCD readout noise:
  This value considers the errors in the received signal resulting from the CCD read out. It is considered when the S/N ratio is being computed by the sensor model to compute the detectability of an object.

- Count rate constant for dark noise:
  This value considers the errors that are caused by dark current, which is a relatively small electric current that flows through photosensitive devices even when no photons are entering the device. It is considered when the S/N ratio is being computed by the sensor model to compute the detectability of an object.

- ISR filename:
  This file specifies the instrumental spectral response to the incoming signal. It defines the percentage of the incoming signal that gets lost inside the sensor because of the optical devices (lenses, mirrors, etc.). Also the capability of the CCD to convert the incoming photons into an electronic charge can be defined in this file.

Below all these parameters errors for measurement generation can be specified. This part of the panel will be available only if the simulation is performed in the deterministic mode (see section 7.2.1.1 for the basic settings) and if the switch to generate measurements has been activated (see 7.2.1.1).

7.2.1.7 Radar Sensor Settings

The fourth sidebar button, which is used for the background databases, is for the “Radar Sensors” (Figure 55). The three sidebar buttons above have been discussed in section 7.2.1.4 (ground-based sensor sites) and 7.2.1.5 (space-based sensor orbits) and 7.2.1.6 (optical sensors).

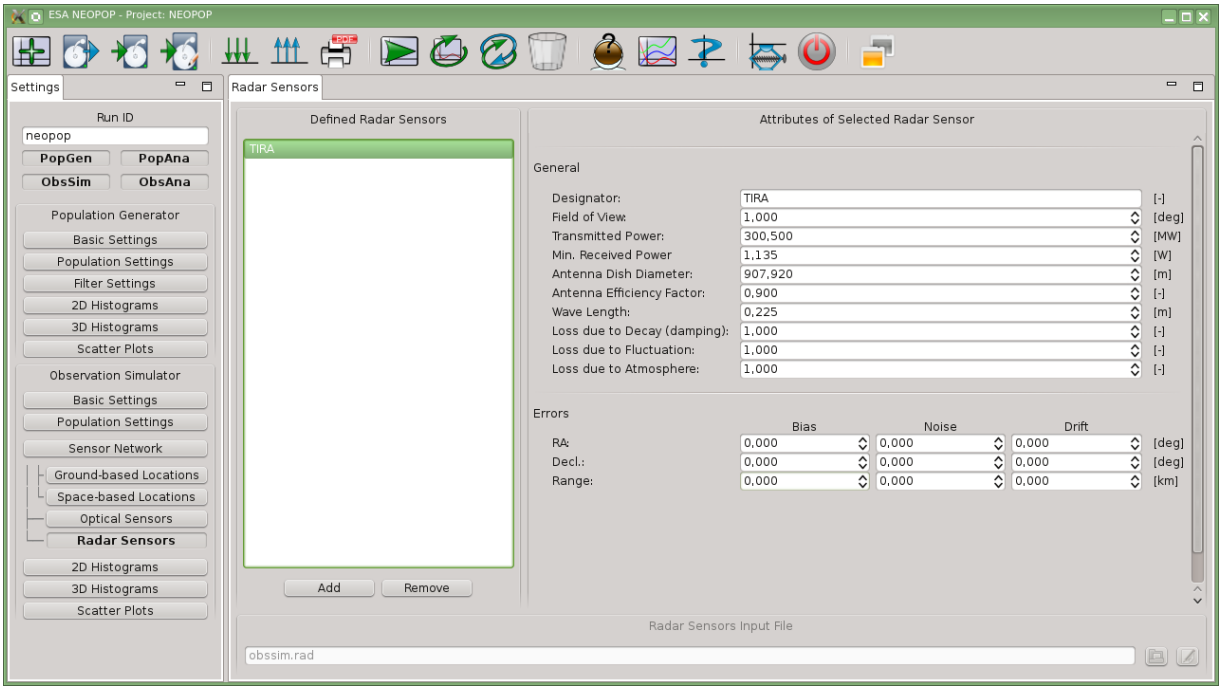

Figure 55: Radar Sensor Settings

As in all panels used for the database settings the panel for the “Radar Sensors” contains a list of all defined radar sensors next to the sidebar and it is possible to add or remove sensors by pressing the buttons below the list of sensors. The simulation assumes that continuous radars (as opposed to pulsed wave radars) are used. The following parameters are shown for the selected sensor on the right site of the panel:

- Designator:
  The designator defines the name of the sensor which is used to refer to this specific sensor. It is shown in the sensor list of the radar sensor settings and in the “Sensor” drop-down menus of the sensor system settings (section 7.2.1.3) if the sensor system’s sensor type is set to “Radar”.

- Field of View:
  The sensor is able to observe a volume shaped like a cone called the field of view. This parameter defines the “opening angle” of the cone. **NOTE:** For bigger FOVs it should be considered deactivating the Pre-Filter (cp. section 7.2.1.1), as such FOVs may bring its algorithms to their limits.

- Transmitted Power:

Defines the average power of the signal that is sent

- Minimal Received Power:
  The threshold for incoming signals to be detected

- Antenna Dish Diameter
  Diameter of a dish antenna

- Antenna Efficiency Factor:
  This factor describes the geometrical efficiency of an antenna to consider the ratio between the real and the effective antenna aperture.

- Wave Length:
  Defines the type of radar waves the sensor emits

- Loss due to Decay:
  Internal attenuation factors of the radar set on the transmitting and receiving paths. Losses are expressed as the denominator in a fraction, with "1" meaning that no loss is introduced, all values between 0 and 1 meaning a gain and all values larger than 1 meaning a loss.

- Loss due to Fluctuation:
  Changes of the signal due to extreme variations of the radar cross section (RCS) depending on signal frequency and observation angle. The value is expressed in the same way as “Loss due to Decay”.

- Loss due to Atmosphere:
  Absorption of the electromagnetic waves due to atmospheric conditions that result in damping of the signal. The value is expressed in the same way as “Loss due to Decay”.

Below all these parameters errors for measurement generation can be specified. This part of the panel will be available only if the simulation is performed in the deterministic mode (see section 7.2.1.1 for the basic settings) and if the switch to generate measurements has been activated (see 7.2.1.1).

7.2.1.8 Output Files

7.2.1.8.1 The OBSSIM results output file (*.res)

This file contains the observation simulation results and dumps data related to the crossing and detection analysis. The naming convention for this file is the following:

<*RUN-ID*>_**S**<*SYSTEM-ID*>_**MC**<*MC-Run*>**.res**

The first string is the run ID, then, separated by an underscore, the system ID and, again separated by an underscore, the number of the MC run. The file suffix is always 'res'.

The tabular data consists of 34 columns in total with the following information:

Type	Format	Examples (content within <>)	UNIT	Allowed Values
System number	I3	< 43>	[-]	$N = \{1{,}99\}$
Object ID	CHAR(10)	<'4332'>	[-]	(All)
Group ID	CHAR(3)	<'APO'>	[-]	(See section 7.1.1.4, PPF)
Factor	F9.3	< 1.000>	[-]	$Q = \{1,\}$
Status flag	INT	<1> Object is crossing <2> Gap time crossing (not detectable) <3> Gap time crossing (detectable) <4> Object has been detected	[-]	$N = \{1{,}4\}$
Diameter	F6.3	<34.567>	[km]	$Q = \{0{,}100\}$
Semi-major axis	F9.6	< 4.123456>	[AU]	$Q = \{0{,}100\}$
Eccentricity	F9.7	< 0.1234567>	[-]	$Q = \{0{,}1\}$
Inclination	F9.5	< 10.98768>	[deg]	$Q = \{0{,}180\}$
RAAN/LAN	F9.5	< 123.45678>	[deg]	$Q = \{0{,}360\}$
AoP	F9.5	< 123.45678>	[deg]	$Q = \{0{,}360\}$
Mean anomaly	F9.5	<123.45678>	[deg]	$Q = \{0{,}360\}$
H-value	F7.4	<12.3456>	[-]	$Q = \{0{,}30\}$
TCA (Time of Closest Approach)	F13.7	< 22.1234>	[h]	$Q = \{0,\}$
Range at TCA	F8.6	< 1.23456>	[AU]	$Q = \{0{,}100\}$
Range rate at TCA	F9.5	< 12.34567>	[km/s]	$Q = \{0,\}$
FOV dwell time	ES9.5	<1.56789E+01>	[min]	$Q = \{0,\}$
Min. path offset	ES9.4	<1.4242E+00>	[deg]	$Q = \{0,\}$
Mean angular velocity	ES9.3	<0.123E-04>	[deg/s]	$Q = \{0,\}$
Mean phase angle	ES9.4	<1.2345E-01>	[deg]	$Q = \{0,\}$
Mean object albedo	F6.4	<0.2341>	[-]	$Q = \{0{,}1\}$
Mean object irradiation	F8.3	<1.234E-02>	[W/m^2]	$Q = \{0,\}$
Mean object magnitude	F7.4	<1.200E+01>	[-]	Q
Mean backg. magnitude	F7.4	<1.200E+01>	[-]	Q
Mean SNR	ES9.3	<1.876E+01>	[-]	$Q = \{0,\}$
Power received by object	E8.2	<0.22E+01>	[kW]	$Q = \{0,\}$
Power received by sensor	E8.2	<0.48E-17>	[W]	$Q = \{0,\}$
Max. range for detection	F10.7	<0.0000438>	[AU]	$Q = \{0,\}$
Line of sight (e.g.AZI)	F7.3	<45.000>	[deg]	$Q = \{0{,}360\}$
Line of sight (e.g. ELE)	F7.3	<90.000>	[deg]	$Q = \{0{,}360\}$

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Obj. state vect. X	F13.6	<-1.239482>	[AU]	Q
Obj. state vect. Y	F13.6	<0.488972>	[AU]	Q
Obj. state vect. Z	F13.6	<0.177368>	[AU]	Q
Source ID	CHAR(3)	<'HUN'>	[-]	(See section 7.1.1.4, PPF)
Additional Information	CHAR(30)	<'Crossing Analysis only!'>	[-]	(See section 7.1.1.4, PPF)

Table 9: Format of the output file *.res

Among the output data can be found the:

- Time of Closest Approach (TCA):
  It refers to the time instant at which the minimum range value is found among all the identified object crossings through the sensor FOV during a certain observation window.

- Range at TCA
  This value is the lowest among all the range ones retrieved when analyzing the object crossings.

7.2.1.8.2 The OBSSIM MC summary output file (*_MC00.res)

The Monte Carlo summary output file is an internal interface and shall be generated, after the loops in the observation simulation are finished. It can be distinguished from the individual RES files (7.2.1.8.1) by looking at the two-digit number in front of the file suffix. Here, the index “00” is used, so while the individual MC runs are numbered with *_MC01.res, *_MC02.res, etc., the summary file shall have the name ending *_MC00.res.

It shall sample the individual RES files for each Monte Carlo run and combine the individual lines according to the following scheme, which is an example for four MC runs for simplicity:

Summary line 1: **Line 1 of MC 1**
Summary line 2: **Line 2 of MC 2**
Summary line 3: **Line 3 of MC 3**
Summary line 4: **Line 4 of MC 4**
Summary line 5: **Line 5 of MC 1**
Summary line 6: **Line 6 of MC 2**
⋮ ⋮ ⋮ ⋮
Summary line n: Line n of MC (n mod 4)

So this output file shall have the same format as for the RES file, however, it samples only $1/n_{MC}$ objects from each file, where n_{MC} is the number of MC runs.

7.2.1.8.3 The OBSSIM summary output file (*.sum)

The OBSSIM summary output file is an internal interface and shall provide a summary of an individual run, containing global information. It shall have the following naming convention:

<*RUN-ID*>_**OBSSIM.sum**

This file shall then display information like the observation simulation mode (deterministic vs. statistical), the number of MC runs, the number of systems and the associated output files.

7.2.2 “Observation Analysis” Module

This module can be used in combination with the module “Observation Simulation” or alone. Thus it is possible to analyze results of a fresh observation simulation or of an already existing one. This gives the advantage that the observation simulation, which can take a long computation time, does not have to be redone when changing the settings of to be generated plots. Thus, the only task of this functionality is to generate user defined plots of the generated observation results.

2D Histograms, 3D Histograms and Scatter Plots

The user may define 2D & 3D Histograms by using the side bars buttons having the same name. Several parameters can be defined, to specify up to 15 histograms. In general for both types of histograms there are the same parameters, but as the 3D histogram has two axes that

have to be defined by the user there are two sets of parameters required. Note that 3D plots show (only) detections.

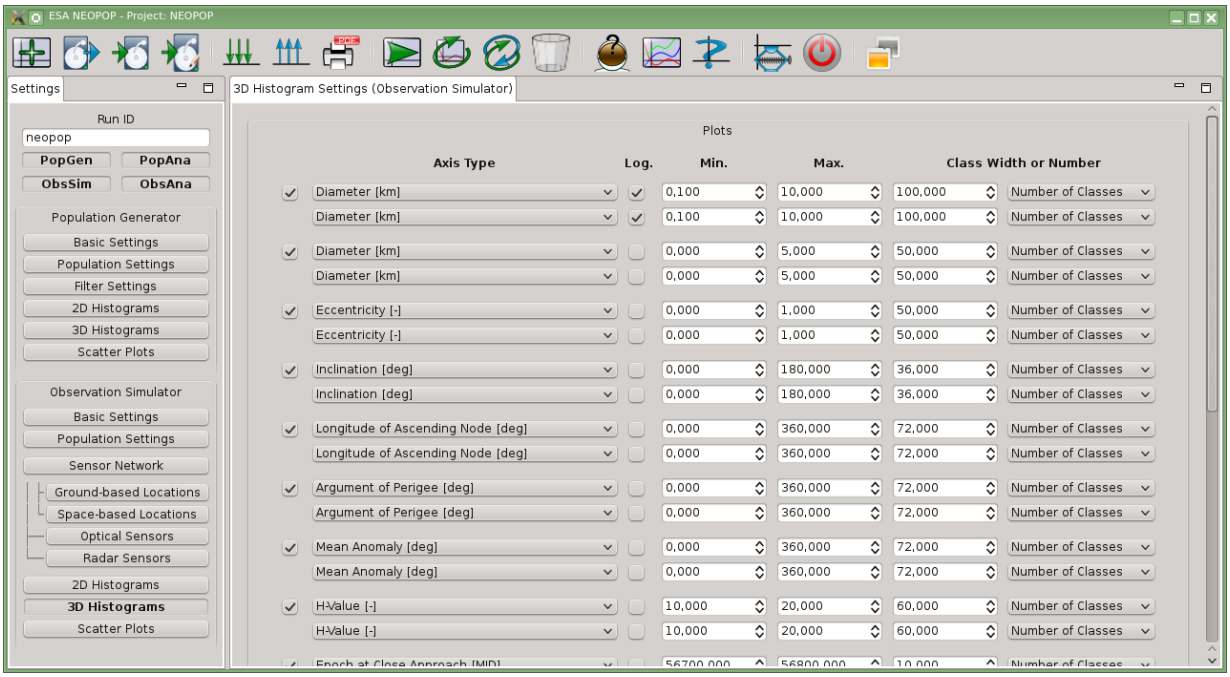

Figure 56: 3D Observation Simulator histogram settings . For defining 2D Histograms only a single line per plot is required, while for defining scatter plots the class definition is not required.

The following content has to be specified for the plot generation:

- Axis Type:
  Via a dropdown menu the kind of data that shall be considered for the plot has to be specified. While the user has a list of parameters within the graphical user interface the input file in ASCII format contains only a number code, which is documented in the header of the input file (default name: *obssim.plt*). Note that the plot axis types containing “Close Approach” refer to the closest approach between sensor and object while the object is within the field-of-view.

- Logarithmic / Linear scale:
  The axis can be defined in a logarithmic or linear scale.

- Minimum:
  The given value is used as lower limit for the axis shown in the plot. This means also that data below this value is not represented in the resulting plot.

- Maximum:
  The given value is used as upper limit for the axis shown in the plot. This means also that data above this value is not represented in the resulting plot.

- Class Width / Class Number
  The resolution of the histogram can be defined in two ways. The user may specify either the class width to be used for generating the histogram classes (or bins) or a predefined number of classes. In the latter case the class width is computed internally by dividing the range between the given minimum and maximum by the given class number. If the class width is predefined the software computes the number of classes internally with a possible increase of the upper limit (Maximum.

  Because there is only a single value to be defined as class width or as class number the user has to define by a switch the meaning of this value.

For scatter plots the same parameters have to be defined except for the class settings which are not needed.

Skyplots

The Observation Analysis automatically creates skyplots, if the Observation Simulation output data used contains corresponding information (for more information on the content of the plots and associated software behavior , cp. section 7.2.1.1). If a crossing analysis has been performed by Observation Simulator, one skyplot showing these crossings is created per selected sensor system by the Observation Analysis. If detection analysis has been performed by the Observation Simulation as well, another skyplot is created per selected sensor system showing the detections. The skyplots are handled as 3D-plots, because they give a number of objects as function sensor orientation (e.g. azimuth and elevation).

Solar System Plots

This module also creates Solar System Plots – just like the Population Analysis (for more details on the plots, cp. section 7.1.2). That's why their axes can be switched on and off as well: in Basic Observation Simulator Settings. The differences, however, are:

- Both Solar System Plot types are generated for each activated sensor system
- The meaning of the dots is different: green dots stand for crossings, red dots indicate detections.

8 Using the Command-Line Tool

The Command-Line Tool (CLT) can be found in folder `default` which is in NEOPOP's installation folder. It is named `neopop` (Linux) or `neopop.exe` (Windows) and can be executed "as is". However, you shouldn't execute the tool in the `default` folder as it houses the default project as well – with all the default settings. Instead you should copy the folder to a place of your choice. You don't have to copy the executable itself though. And it is not recommended to copy the two `data` folders, either, as their content takes up lots of disk space. There are two alternative ways to use external data folders (e. g. those of the default project):

- Create symbolic folder links that replace the `data` folders in `01-POPGEN` and `02-OBSSIM` and that point to the central data folders you chose.
- Change the data folder settings in `01-POPGEN/popgen.cfg` and `02-OBSSIM/obssim.cfg` appropriately.

After setting up your project you can then start a test run of the tool by executing it in your new project folder. It will start by reading `neopop.cfg` and from there on read in all other referenced files that are needed.

8.1 Project Folder Content

The following table gives an overview of the content of a project folder:

File / Folder	Meaning / Content
`neopop.cfg`	NEOPOP configuration file ● Module selection (Population Generation, Population Analysis, Observation Simulation, Observation Analysis) ● Definition of Population Generator and Observation Simulator configuration files
`01-POPGEN`	Folder of Population Generator component
`01-POPGEN/popgen.cfg`	Population Generator configuration file ● Definition of data, input and output folder ● Definition of data and input files
`01-POPGEN/data`	Population Generator data folder ● Contains input data that usually does not change very often – see section 4.1.1 for details
`01-POPGEN/input`	Population Generator input folder

File / Folder	Meaning / Content
	● Contains input for Population Generator component
01-POGGEN/input/popgen.inp	Main Population Generator Input File ● Mode selection for Population Generation and Analysis modules ● Basic settings
01-POPGEN/input/popgen.fil	Filter Settings Input File ● Contains settings for filtering populations analyzed by Population Analysis module.
01-POPGEN/input/popgen.plt	Population Generator Plot Input File ● Defines what Population Analysis plots and how
01-POPGEN/output	Population Generator output folder ● All output data of this component is put here – see later in this chapter for details.
02-OBSSIM	Folder of Observation Simulator component
02-OBSSIM/obssim.cfg	Observation Simulator configuration file ● Definition of data, input and output folder ● Definition of data and input files
02-OBSSIM/data	Observation Simulator data folder ● Contains input data that usually does not change very often – see section 4.1.1 for details
02-OBSSIM/input	Observation Simulator input folder ● Contains input for Observation Simulator component
02-OBSSIM/input/obssim.inp	Main Observation Simulator Input File ● Selection of Observation Simulation and Analysis modes ● Basic settings for this component
02-OBSSIM/input/obssim.src	Source Input File ● Defines input population of Observation Simulation
02-OBSSIM/input/obssim.net	Network Input File ● Contains definition of the Sensor Systems that make up your Sensor Network

File / Folder	Meaning / Content
	● A Sensor System is a Sensor at a specific Location, both of which are defined in the following four input files
`02-OBSSIM/input/obssim.g_b`	Ground-based Location Input File ● Defines Ground-based Locations (on Earth) that a Sensor System may be located at
`02-OBSSIM/input/obssim.s_b`	Space-based Location Input File ● Defines Space-based Locations (i. e. orbits) that a Sensor System may be located at (as part of a satellite, for example)
`02-OBSSIM/input/obssim.opt`	Optical Sensor Input File ● Defines Optical Sensors that can be part of a Sensor System
`02-OBSSIM/input/obssim.rad`	Radar Sensor Input File ● Defines Radar Sensors that can be part of a Sensor System
`02-OBSSIM/input/obssim.plt`	Observation Simulator Plot Input File ● Defines what Observation Analysis plots and how
`02-OBSSIM/output`	Observation Simulator output folder ● All output data of this component is put here – see later in this chapter for details.

The format of configuration and input files is always the same:

- ASCII text files
- Comment lines explain what values can be changed, which values are allowed and what unit is used. They are indicated by a "#" as the first character (for a complete comment line) or a "!" (for a comment taking up the rest of the line)
- Actual values are:
    - Switches, with "1" and "0" as possible values meaning "Yes"/"True" and "No"/"False"
    - Choices, with "1", …, "<n>" as possible values meaning the selection of one fixed value among others.
    - Integers
    - Doubles, floating-point decimal numbers.
    - Strings, character chains. They should be put into "".

You can find more detail about the configuration and input files in chapter 9.

8.2 Output Folder Content

The **Population Generator** output folder can contain the following data:

File / Folder	Meaning / Content
`<run id>.<"dys" | "mpc" | "des">`	Population File ● Depending on population file format chosen, this file contains basic information about generated NEOs like their absolute magnitude and their orbital parameters ● See section 7.1.1.4 for details
`<run id>.ppf`	Physical Properties File ● Supplements a population file with physical properties like albedo ● See section 7.1.1.4 for details ● Path can be changed when generating a PPF only
`<run id>.sum`	Population Generation Summary File ● Gives a general overview of what has been generated
`<run id>_ANA.res`	Population Analysis Result File ● Lists all objects that were analyzed by Population Analysis ● Objects that are filtered out don't show up in this file
`<run id>_ANA.gnu`	Main Population Analysis gnuplot File ● References all other generated gnuplot files of this run ● When executed with gnuplot, all plot pictures of run <run id> are created
`<run id>_ANA_<"2D" | "3D" | "SC">_<plot category number>_<"d" | "c" | "r">.spc`	Spectra ● Files containing data used in individual gnuplot files to define plots ● See directly beneath for details about name
`<run id>_ANA_<"2D" | "3D" | "SC">_<plot category number>_<"d" | "c" | "r">.gnu`	Individual gnuplot Files ● Also called "plot drivers" ● Define 2D and 3D histogram as well as scatter plots

File / Folder	Meaning / Content
	● "d" stands for "differential" plot which is the default ● Cumulative ("c") and reverse-cumulative ("r") plot variants exist for 2D histograms only ● Can be executed by gnuplot creating a PNG file with the same filename
<run id>_ANA_<"SLP" | "SLS">.gnu	Solar System gnuplot Files ● Define Polar Solar System Plots ("SLP") and Solar System Side Plots ("SLS")

The **Observation Simulator** output folder can contain the following data:

File / Folder	Meaning / Content
<run id>.ana	Observation Simulation Results to be used by Observation Analysis ● Produced by Observation Simulation, read by Observation Analysis
<run id>_OBSSIM.sum	Observation Simulation Summary File ● Gives a general overview of what has been simulated ● See section 7.2.1.8 for details.
<run id>_S<sensor system>_MC<MC run>.res	Observation Simulation Results File ● Lists FOV crossings and detections by a sensor system during a Monte-Carlo run ● "<sensor system>" is a two-digit number which relates to one of the defined and simulated sensor systems. Sensor System "00" is the Sensor Network as a whole. ● "<MC run>" is a two-digit number which stands for the Monte-Carlo run this results file applies to. MC run "00" is the only run in deterministic Observation Simulation mode and stands for the statistical mean in statistical mode. ● See section 7.2.1.8 for details.
<run id>_OBSSIM.gnu	Main Observation Analysis gnuplot file ● References all Sensor gnuplot Files of this run ● When executed with gnuplot, all plot pictures of run <run id> are created

File / Folder	Meaning / Content
<run id>_S<sensor system>.gnu	Sensor gnuplot File ● References all generated individual gnuplot files of this run ● When executed with gnuplot, all plot pictures of sensor system <sensor system> in run <run id> are created ● "<sensor system>": See Observation Simulation Results File
<run id>_S<sensor system>_<"2D" | "3D" | "SC">_<plot category>_<"d" | "c" | "r">.spc	Spectra ● Files containing data used in individual gnuplot files to define plots ● See directly beneath for details about name
<run id>_ S<sensor system>_<"2D" | "3D" | "SC">_<plot category>_<"d" | "c" | "r">.gnu	Individual gnuplot Files ● Also called "plot drivers" ● Same as in Population Analysis, but with additional sensor system number. See Observation Simulation Results File for details
<run id>_S<sensor system>_SL_MC<MC run>.spc	Spectra for Solar System Plots ● Contain data shown in Solar System Plots ● "<sensor system>", "<MC run>": See Observation Simulation Results File for details
<run id>_S<sensor system>_<"SLP" | "SLS">.gnu	Solar System gnuplot Files ● Same as in Population Analysis, but with additional sensor system number. See Population Analysis Output Folder Content and Observation Simulation Results File for details
<run id>_SKY_S<sensor system>_d.spc	Spectra for Skyplots ● Contain data shown in Skyplots ● "<sensor system>": See Observation Simulation Results File
<run id>_SKY_S<sensor system>_<"crs" | "det">.gnu	Skyplot gnuplot Files ● gnuplot Files defining skyplots showing crossings ("crs") and detections ("det") ● "<sensor system>": See Observation Simulation Results File
<run id>_S<sensor system>_MC00.mes	Measurement File ● Contains measurements for detected objects

File / Folder	Meaning / Content
	● "<sensor system>": See Observation Simulation Results File

8.3 Executing gnuplot

Unlike the GUI the CLT does not execute gnuplot automatically to generate pictures from gnuplot files. In order to do that on your own on the command-line:

Population Analysis	Observation Analysis

Execute
`gnuplot <run id>_ANA.gnu`
in
`01-POPGEN/output`
	Execute
	`gnuplot <run id>_OBSSIM.gnu`
	in
	`02-OBSSIM/output`

This will create all plots of a specific run of Population or Observation Analysis. You can also just recreate an individual plot by replacing the ".gnu" file being executed with the respective plot's one. In the case of Observation Analysis you also have the possibility to use <run id>_S<sensor system>.gnu to (re-)create all plot pictures of a specific sensor system.

Make sure that you execute gnuplot in the respective output folder. Otherwise gnuplot will not be able to locate the required spectra files.

9 Configuration and Input Files Specification

This chapter describes the parameters that can be changed in the configuration and input files of NEOPOP CLT along with their respective allowed values.

9.1 neopop.cfg

Type	Format	Examples (content within <>)	UNIT	Allowed Values
# Component switches				
Population Generation	INT	<0>,<1>	[-]	$N = \{0,1\}$
Population Analysis	INT	<0>,<1>	[-]	$N = \{0,1\}$
Observation Simulation	INT	<0>,<1>	[-]	$N = \{0,1\}$
Observation Analysis	INT	<0>,<1>	[-]	$N = \{0,1\}$
# Configuration files				
Population Generator	CHAR(230)	<01-POPGEN/popgen.cfg>	[-]	all
Observation Simulator	CHAR(230)	<02-OBSSIM/obssim.cfg>	[-]	all

Table 10: Parameters of the NEOPOP main configuration file (neopop.cfg)

9.2 obssim.cfg

Type	Format	Examples (content within <>)	UNIT	Allowed Values
# Run-ID used for the simulation				
Runid	CHAR(12)	<obssim>	[-]	all, except folder delimiters
# Comment lines				
Line 1	CHAR(30)	<Cmtln 1>	[-]	-
Line 2	CHAR(30)	<Cmtln 2>	[-]	-
# Project folder				
Path to project folder	CHAR(170)	<`.´>	[-]	-
# Path definition				
Path to data folder	CHAR(200)	<'data'>	[-]	-
Path to input folder	CHAR(200)	<'input'>	[-]	-
Path to output folder	CHAR(200)	<'output'>	[-]	-
# Data files				
SPICE kernel SPK	CHAR(30)	<`de430.bsp´>	[-]	-
SPICE kernel PCK	CHAR(30)	<`pck00010.tpc´>	[-]	-
SPICE kernel TLS	CHAR(30)	<`naif0010.tls´>	[-]	-
Observatory code list	CHAR(30)	<`obscodes.dat´>	[-]	-
Airglow spec. distr.	CHAR(30)	<`optical.air´>	[-]	-
Atmospheric extinction	CHAR(30)	<`optical.atm´>	[-]	-

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Starlight spec. distr.	CHAR(30)	<`optical.ssz´>	[-]	-
Sun spec. irradiance	CHAR(30)	<`optical.sun´>	[-]	-
Patched stellar catalog	CHAR(30)	<`optical.pat´>	[-]	-
Zodiacal light data	CHAR(30)	<`optical.zod´>	[-]	-
Scattered sunlight	CHAR(30)	<`optical.ssl´>	[-]	-
Scattered moonlight	CHAR(30)	<`optical.sml´>	[-]	-
Lunar phase factor	CHAR(30)	<`optical.lpf´>	[-]	-
Extragal. Background	CHAR(30)	<`optical.ebl ´>	[-]	-
V-band extinction	CHAR(30)	<`optical.vbe´>	[-]	-
Star cat. – VIS	CHAR(30)	<`optical.str´>	[-]	-
Star cat. – TIR	CHAR(30)	<`optical.stc´>	[-]	-
# Input files				
Main Input file	CHAR(30)	<'obssim.inp'>	[-]	-
Source definition file	CHAR(30)	<'obssim.src'>	[-]	-
Network definition file	CHAR(30)	<'obssim.net'>	[-]	-
Database for ground-based locations	CHAR(30)	<'obssim.g_b'>	[-]	-
Database for space-based orbits	CHAR(30)	<'obssim.s_b'>	[-]	-
Database for optical sensors	CHAR(30)	<'obssim.opt'>	[-]	-
Database for radar sensors	CHAR(30)	<'obssim.rad'>	[-]	-
Plot definition file	CHAR(30)	<'obssim.plt'>	[-]	-

Table 11: Configuration file of the component "Observation Simulator" (obssim.cfg)

9.3 obssim.g_b

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Location definition flag	INT	<0> Obsrv. Code <1> Geocentric <2> Geodetic <3> Earth fixed	[-]	$N = \{0,3\}$
Longitude/x-value	F12.6	< 130.000000>	[deg/km]	(depends)
Latitude/y-value	F12.6	< 90.000000>	[deg/km]	(depends)
Geoc. Distance/altitude/z-value	F12.6	< 90.000000>	[km]	(depends)
Ambient temperature	F7.2	< 20.00>	[K]	$Q = \{200,340\}$
Atm. Pressure	F8.2	< 1013.20>	[hPa]	$Q = \{300,1100\}$
Relative humidity	F6.2	< 50.00>	[%]	$Q = \{0,100\}$
Tropospheric lapse rate	F5.2	< 6.50>	[K/100km]	$Q = \{0,10\}$
Designator	CHAR(30)	<Location-01>	[-]	-

Table 12: Ground-based sensor file parameters (obssim.g_b)

9.4 obssim.inp

Type	Format	Examples (content within <>)	UNIT	Allowed Values
# Mode of Observation Simulation (1=statistic, 2=deterministic)				
Switch	INT	<1>,<2>	[-]	$N = \{1,2\}$
# General Switches				
Pre-Filtering	INT	<0>,<1>	[-]	$N = \{0,1\}$
Crossing analysis	INT	<0>,<1>	[-]	$N = \{0,1\}$
--Light travel time	INT	<0>,<1>	[-]	$N = \{0,1\}$
--Tropospheric refraction	INT	<0>,<1>	[-]	$N = \{0,1\}$
Detection analysis	INT	<0>,<1>	[-]	$N = \{0,1\}$
--global physical data	INT	<0>,<1>	[-]	$N = \{0,1\}$
Measurement Generation	INT	<0>,<1>	[-]	$N = \{0,1\}$
# Time interval for observation simulation				
Start of observation	I4,5(1X,I2)	<2012 09 30 00 00 00>	[-]	–
Duration of observation	6(1x,I2)	<00 00 01 02 59 59>	[-]	–
# Sky Plot Generation				
Switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Grid Size for mesh	F6.3	<5.000>	[deg]	$Q = \{0,45\}$, $Q = 90/a$, $a \in N$
# General settings				
Seed for Random number generator	INT	<5>	[-]	$N = \{0,10000\}$
No. of MC runs	INT	<20>	[-]	$N = \{1,99\}$
Number of steps in FOV	INT	<400>	[-]	$N = \{1,\}$
Min. elevation (g-b sensor)	F6.3	<10.100>	[deg]	$Q = \{0,90\}$
Max. mag. for discrete stars	INT	<6>	[-]	$N = \{0,25\}$

Table 13: Main Observation Simulator input file (obssim.inp)

9.5 obssim.net

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Consider system switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Sensor designator	CHAR(30)	<Sensor-1_GB>	[-]	–
Location	INT	<1> ground-based <2> space-based	[-]	$N = \{1,2\}$
Sensor type	INT	<1> Radar <2> Telescope	[-]	$N = \{1,2\}$

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Automated processing sw.	INT	<0>,<1>	[-]	$N = \{0,1\}$
OSD file name	CHAR(30)	<default.osd>	[-]	–
Sensor location design.*	CHAR(30)	<Location-1>	[-]	–
Sensor type designator*	CHAR(30)	<Sensor-1>	[-]	-
Coordinate system*	INT	<1> loc. hor. <2> topoc. eq. <3> topoc. ecl.	[-]	$N = \{1,3\}$
Azimuth/RA/Ecl. Lon.*	F7.3	<120.000>	[deg]	$Q = \{0,360\}$
Elevation/Decl./Ecl. Lat.*	F7.3	< 60.000>	[deg]	$Q = \{0,90\}$
Min. range between sensor and object*	F7.3	< 0.000>	[AU]	$Q = \{0,50\}$
Max. range between sensor and object*	F7.3	< 5.000>	[AU]	$Q = \{5,100\}$
Bistatic switch	INT	<0>,<1>	[-]	$N = \{0,1\}$

Table 14: The network definition file parameters (obssim.net)

9.6 obssim.opt

* Valid POV is calculated based on FOV and pixels per row

Type	Format	Examples (content within <>)	UNIT	Allowed Values
VIS or IR switch	INT	<1> Visual band <2> Thermal IR	[-]	$N = \{1,2\}$
Field of view	FLOAT	<0.709>	[deg]	$Q = \{0,360\}$
Integration time	FLOAT	<5.0>	[sec]	$Q = \{0,3600\}$
Gap time	FLOAT	<58.0>	[sec]	$Q = \{0,3600\}$
Min. number of consecutive detections	INT	<2>	[-]	$N = \{1,100\}$
Limiting magnitude	FLOAT	<17.0>	[-]	$Q = \{0,30\}$
Aperture	FLOAT	<1.016>	[m]	$Q = \{0,100\}$
Pixel size	FLOAT	<10.0>	[µm]	$Q = \{0,1000\}$
Pixels per row	INT	<2048>	[-]	$N = \{3220480,\}$
FOV per pixel	FLOAT	<1.24>	['']/pix]	[?]*
FWHM	FLOAT	<1.210>	[-]	$Q = \{0,100\}$
Detection threshold	FLOAT	<2.100>	[-]	$Q = \{0,1E6\}$
Count rate const. for CCD readout noise	FLOAT	<8.000>	[-]	$Q = \{0,1E6\}$
Count rate const. for dark noise	FLOAT	<0.000>	[-]	$Q = \{0,1E6\}$
Sensor Az. Bias	FLOAT	<0.040>	[deg]	$Q = \{0,10\}$
Sensor Az. Noise	FLOAT	<0.050>	[deg]	$Q = \{0,9.999\}$
Sensor Az. Drift	FLOAT	<0.060>	[deg]	$Q = \{0,9.999\}$
Sensor Elev. Bias	FLOAT	<0.040>	[deg]	$Q = \{0,10\}$

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Sensor Elev. Noise	FLOAT	<0.050>	[deg]	$Q = \{0,9.999\}$
Sensor Elev. Drift	FLOAT	<0.060>	[deg]	$Q = \{0,9.999\}$
Designator	CHAR(30)	<Sensor-1>	[-]	-
ISR File name	CHAR(30)	<`optical.isr´>	[-]	-

Table 15: Optical sensor properties file parameters (obssim.opt)

9.7 obssim.plt

Type	Format	Examples (content within <>)	UNIT	Allowed values
# 2D histogram settings (one line, blank separated values) **# 3D histogram settings** (two lines, where "Spectra switch" is only in first line!)				
Spectra switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Data Identifier	I2	<01>,<18>	[-]	$N = \{1,18\}$
Logscale switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Lower limit*	F7.3	< 0.000>	[*]	$Q = \{0,?\}$
Upper limit*	F7.3	< 1.000>	[*]	$Q = \{0,?\}$
Switch class width (1) / no. of classes (0)	INT	<1>	[-]	$N = \{0,1\} N = \{0,1E5\}$
Class width	I3	< 5>	[-]	
# Scatter plot settings				
Spectra switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Data Identifier	I2	<01>,<18>	[-]	$N = \{1,18\}$
Logscale switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Lower limit*	F7.3	< 0.000>	[*]	$Q = \{0,?\}$
Upper limit*	F7.3	< 1.000>	[*]	$Q = \{0,?\}$
# Plot settings (lines displayed in plots)				
Switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Linetype	INT	<-1>,<2>	[-]	$Z = \{-1,\}$
Linewidth	INT	<1>,<2>	[-]	$N = \{1,\}$
Label	CHAR(30)	<'Crossing'>	[-]	-

# Solar System Plot Settings				
Axes switch	INT	<0>, <1>	[-]	$N = \{0,1\}$

Table 16: Observation Simulator plot settings input file (obssim.plt)

9.8 obssim.rad

Type	Format	Examples (content within <>)	UNIT	Allowed Values

 # Radar Parameters

Field of view	F6.3	<0.709>	[deg]	$Q = \{0{,}360\}$
Transmitted power	F10.5	<12345>	[MW]	$Q = \{0{,}9999\}$
Min. received power	F6.3	<21.123>	[W]	$Q = \{0{,}99\}$
Antenna Diameter	F7.3	<321.123>	[m]	$Q = \{0{,}999\}$
Antenna efficiency	F4.2	<1.12>	[-]	$Q = \{0{,}9.99\}$
Wavelength	F8.5	<21.12345>	[m]	$Q = \{0{,}99\}$
Loss (system)	F6.3	<21.123>	[-]	$Q = \{0{,}99\}$
Loss (fluctuation)	F6.3	<21.123>	[-]	$Q = \{0{,}99\}$
Loss (atmosphere)	F6.3	<21.123>	[-]	$Q = \{0{,}99\}$
Designator	CHAR(30)	<Sensor-1>	[-]	-

# Sensor errors for measurement generation				
RA Bias	F6.3	<0.000>	[deg]	$Q = \{0{,}10\}$
RA Noise	F6.3	<0.000>	[deg]	$Q = \{0{,}9.999\}$
RA Drift	F6.3	<0.000>	[deg]	$Q = \{0{,}9.999\}$
Decl. Bias	F6.3	<0.000>	[deg]	$Q = \{0{,}10\}$
Decl. Noise	F6.3	<0.000>	[deg]	$Q = \{0{,}9.999\}$
Decl. Drift	F6.3	<0.000>	[deg]	$Q = \{0{,}9.999\}$
Range Bias	F6.3	<0.000>	[km]	$Q = \{0{,}1000\}$
Range Noise	F6.3	<0.000>	[km]	$Q = \{0{,}9.999\}$
Range Drift	F6.3	<0.000>	[km]	$Q = \{0{,}9.999\}$

Table 17: Radar properties file parameters (obssim.rad)

9.9 obssim.s_b

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Orbit type	INT	<1> Geocentric <2> Heliocentr.	[-]	$N = \{1{,}2\}$
Type of state vector	INT	<1> Kepler el. <2> Cartesian	[-]	$N = \{1{,}2\}$
Semi-major axis OR x-value	FLOAT	< 1.00000> <-2.34500>	[km|AU] [km|AU]	$Q = \{0,\}$ Q
Eccentricity OR y- value	FLOAT	< 0.20000> <-3.44500>	[-] [km|AU]	$Q = \{0{,}1\}$ Q

Type	Format	Examples (content within <>)	UNIT	Allowed Values
Inclination OR z- value	FLOAT	< 30.0000> < 2.15000>	[deg] [km|AU]	$Q = \{0,180\}$ Q
RAAN/LNODE OR v_x- value	FLOAT	< 30.0000> < 2.15000>	[deg] [km/s|AU/d]	$Q = \{0,360\}$ Q
AoP OR v_y- value	FLOAT	< 30.0000> < 2.15000>	[deg] [km/s|AU/d]	$Q = \{0,360\}$ Q
Mean anomaly OR v_z- value	FLOAT	< 30.0000> < 2.15000>	[deg] [km/s|AU/d]	$Q = \{0,360\}$ Q
Orbit epoch	FLOAT	<2013112.345>	[YYYYDDD.ddd]	-
Designator	CHAR(30)	<Orbit-01>	[-]	-
BSP File	CHAR(250)	<soho_orbit.bsp> relative path to SPK kernel file	[-]	-
TLS File	CHAR(250)	<naif0010.tls> relative path to auxiliary file	[-]	-

Table 18: Space-based sensors input file (obssim.s_b)

9.10 obssim.src

Type	Format	Examples (content within <>)	UNIT	Allowed Values
# NEO population groups to be used				
Amor switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Apollo switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Atens switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Atiras switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Main Belt asteroids switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Trojans switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Others switch	INT	<0>,<1>	[-]	$N = \{0,1\}$

# NEO population sources to be used				
Hungarias switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Phocae switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
nu6 switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Jupiter 3:1 switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Jupiter 5:2 switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Jupiter 2:1 switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
JFC switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Unknowns switch	INT	<0>,<1>	[-]	$N = \{0,1\}$

# Population Files 1 (e.g. the NEO population)				
Population switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
File name	CHAR(30)	<`allnum.cat´>	[-]	–
PPF name	CHAR(30)	<`1to999.ppf´>	[-]	–
Format of population switch	INT	<1> NEODyS/AstDys <2> MPC <3> DES	[-]	$N = \{1,3\}$
# Population Files 2 (e.g. the asteroid population)				
Population switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
File name	CHAR(30)	<`allnum.cat´>	[-]	–
PPF name	CHAR(30)	<`1to999.ppf´>	[-]	–
Format of population switch	INT	<1> NEODyS/AstDys <2> MPC <3> DES	[-]	$N = \{1,3\}$
# Physical object data				
Visual albedo Error in visual albedo	F6.3 F6.3	< 3.000> < 0.023>	[-] [-]	$Q = \{0,1\} Q = \{0,1\}$

Table 19: The Observation Simulator sources input file parameters (obssim.src)

9.11 popgen.cfg

Name	Format	Examples (content within <>)	UNIT	Allowed Values
# Run-ID used for the simulation				
Runid	CHAR(12)	<'popgen'>	[-]	all, except folder delimiters
# Comment lines				
Line 1	CHAR(30)	<'Cmtln 1'>	[-]	all
Line 2	CHAR(30)	<'Cmtln 2'>	[-]	all
# Project Folder				
Project Folder	CHAR(200)	<'.'>	[-]	all
# Path definition				
Path to data folder	CHAR(200)	<'data'>	[-]	all
Path to input folder	CHAR(200)	<'input'>	[-]	all
Path to output folder	CHAR(200)	<'output'>	[-]	all
# Data files				
Known NEOs H<15	CHAR(30)	<'known_NEOs_up_to_H15.dat'>	[-]	all
Old Model: Src Region File 1	CHAR(30)	<'31all.res'>	[-]	all

Name	Format	Examples (content within <>)	UNIT	Allowed Values
Old Model: Src Region File 2	CHAR(30)	<'imc_update_exttarget.res'>	[-]	all
Old Model: Src Region File 3	CHAR(30)	<'comet_jfc_only.res'>	[-]	all
Old Model: Src Region File 4	CHAR(30)	<'nu6all.res'>	[-]	all
Old Model: Src Region File 5	CHAR(30)	<'hung.res'>	[-]	all
Old Model: Src Region File 6	CHAR(30)	<'pho.res'>	[-]	all
Old Model: Src Region File 7	CHAR(30)	<'ob_superext.res'>	[-]	all
New Model file	CHAR(30)	<'gmb_model.dat'>	[-]	all
SPICE Leap seconds file	CHAR(30)	<'naif0010.tls'>	[-]	all
SPICE JPL ephemerides kernel	CHAR(30)	<'de421.bsp'>	[-]	all
SPICE PCK file	CHAR(30)	<'pck00010.tpc'>	[-]	all
# Input files				
Main Input file	CHAR(30)	<'popgen.inp'>	[-]	all
Filter Settings	CHAR(30)	<'popgen.fil'>	[-]	all
Plot Settings	CHAR(30)	<'popgen.plt'>	[-]	all

Table 20: Population Generator Configuration File (popgen.cfg)

9.12 popgen.fil

Type	Format	Examples (content within <>)	UNIT	Allowed values
# Asteroid groups selection switches				
Amors	INT	<0>, <1>	[-]	$N = \{0,1\}$
Apollos	INT	<0>, <1>	[-]	$N = \{0,1\}$
Atens	INT	<0>, <1>	[-]	$N = \{0,1\}$
Atiras	INT	<0>, <1>	[-]	$N = \{0,1\}$
Main Belt	INT	<0>, <1>	[-]	$N = \{0,1\}$
Trojans	INT	<0>, <1>	[-]	$N = \{0,1\}$

Type	Format	Examples (content within <>)	UNIT	Allowed values
Others	INT	<0>, <1>	[-]	$N = \{0,1\}$
# Asteroid source selection switches				
Hungaria	INT	<0>, <1>	[-]	$N = \{0,1\}$
Phocea	INT	<0>, <1>	[-]	$N = \{0,1\}$
(nu6) secular resonance	INT	<0>, <1>	[-]	$N = \{0,1\}$
Jupiter 3/1 resoance	INT	<0>, <1>	[-]	$N = \{0,1\}$
Jupiter 5/2 resoance	INT	<0>, <1>	[-]	$N = \{0,1\}$
Jupiter 2/1 resoance	INT	<0>, <1>	[-]	$N = \{0,1\}$
Jupiter Family Comets	INT	<0>, <1>	[-]	$N = \{0,1\}$
Unknowns	INT	<0>, <1>	[-]	$N = \{0,1\}$
# Filter settings				
Switch a Min. value Max. value	INT F6.3 F6.3	<0>, <1> <0.000> <5.000>	[-] [AU] [AU]	$N = \{0,1\}$ $Q = \{0,100\}$ $Q = \{0,100\}$
Switch q Min. value Max. value	INT F6.3 F6.3	<0>, <1> <0.000> <5.000>	[-] [AU] [AU]	$N = \{0,1\}$ $Q = \{0,100\}$ $Q = \{0,100\}$
Switch Q Min. value Max. value	INT F6.3 F6.3	<0>, <1> <0.000> <5.000>	[-] [AU] [AU]	$N = \{0,1\}$ $Q = \{0,100\}$ $Q = \{0,100\}$
Switch e Min. value Max. value	INT F5.3 F5.3	<0>, <1> <0.000> <1.000>	[-] [-] [-]	$N = \{0,1\}$ $Q = \{0,1\}$ $Q = \{0,1\}$
Switch i Min. value Max. value	INT F5.1 F5.1	<0>, <1> < 0.0> < 5.0>	[-] [deg] [deg]	$N = \{0,1\}$ $Q = \{0,180\}$ $Q = \{0,180\}$
Switch Ω Min. value Max. value	INT F5.1 F5.1	<0>, <1> <123.0> <345.0>	[-] [deg] [deg]	$N = \{0,1\}$ $Q = \{0,360\}$ $Q = \{0,360\}$
Switch ω Min. value Max. value	INT F5.1 F5.1	<0>, <1> <123.0> <345.0>	[-] [deg] [deg]	$N = \{0,1\}$ $Q = \{0,360\}$ $Q = \{0,360\}$
Switch M Min. value Max. value	INT F5.1 F5.1	<0>, <1> <123.0> <345.0>	[-] [deg] [deg]	$N = \{0,1\}$ $Q = \{0,360\}$ $Q = \{0,360\}$
Switch H Min. value Max. value	INT F5.3 F5.3	<0>, <1> <0.000> <5.000>	[-] [deg] [deg]	$N = \{0,1\}$ $Q = \{0,30\}$ $Q = \{0,30\}$
Switch MOID Min. value Max. value	INT F6.3 F6.3	<0>, <1> < 0.000> < 2.000>	[-] [AU] [AU]	$N = \{0,1\}$ $Q = \{0,100\}$ $Q = \{0,100\}$
Switch coll. prob. Min. value	INT F5.3	<0>, <1> <0.123>	[-] [-]	$N = \{0,1\}$ $Q = \{0,1\}$

Type	Format	Examples (content within <>)	UNIT	Allowed values
Max. value	F5.3	<1.000>	[-]	$Q = \{0,1\}$
Switch diameter	INT	<0>, <1>	[-]	$N = \{0,1\}$
Min. value	F8.3	< 0.000>	[km]	$Q = \{0,1000\}$
Max. value	F8.3	<1000.000>	[km]	$Q = \{0,1000\}$
Switch dist. Sun	INT	<0>, <1>	[-]	$N = \{0,1\}$
Min. value	F6.3	<0.000>	[AU]	$Q = \{0,100\}$
Max. value	F6.3	<5.000>	[AU]	$Q = \{0,100\}$
Switch dist. Earth	INT	<0>, <1>	[-]	$N = \{0,1\}$
Min. value	F6.3	<0.000>	[AU]	$Q = \{0,100\}$
Max. value	F6.3	<5.000>	[AU]	$Q = \{0,100\}$

Table 21: Filter Settings input file (popgen.fil)

9.13 popgen.inp

Name	Type	Example (content within <>)	UNIT	Allowed values
# Population Generation				
Mode	INT	<2>	[-]	$N = \{0,4\}$
# Population Analysis				
Mode	INT	<1>	[-]	$N = \{0,2\}$
# In case of generating a synthetic population				
Epoch	I4,5(X,I2)	<2012 05 01 00 00 00>	[-]	-
Min. H-value	FLOAT	<18.4>, <15.32>	[-]	$Q = \{5,30\}$
Max. H-value	FLOAT	<18.4>, <15.32>	[-]	$Q = \{5,10\}$
Scaling Factor	FLOAT	<1.46>	[-]	$Q = \{0.001,1000\}$
Extr. switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Slope 1	F5.3	<3.000>	[-]	$Q = \{0,10\}$
Breaking pt.	F5.3	<28.000>	[-]	$Q = \{25,30\}$
Slope 2	F5.3	<5.000>	[-]	$Q = \{0,10\}$
# In case of generating a fictitious population				
Min. a	F6.3	<0.000>	[AU]	$Q = \{0,100\}$
Max. a	F6.3	<5.000>	[AU]	$Q = \{0,100\}$
Min. e	F4.3	<0.000>	[-]	$Q = \{0,1\}$
Max. e	F4.3	<1.000>	[-]	$Q = \{0,1\}$
Min. i	F6.3	< 0.0>	[deg]	$Q = \{0,180\}$
Max. i	F6.3	< 5.0>	[deg]	$Q = \{0,180\}$
Min. Ω	F6.3	<123.0>	[deg]	$Q = \{0,360\}$
Max. Ω	F6.3	<345.0>	[deg]	$Q = \{0,360\}$
Min. ω	F6.3	<123.0>	[deg]	$Q = \{0,360\}$
Max. ω	F6.3	<345.0>	[deg]	$Q = \{0,360\}$
Min. M	F6.3	<123.0>	[deg]	$Q = \{0,360\}$
Max. M	F6.3	<345.0>	[deg]	$Q = \{0,360\}$
Apollos-switch	INT	<1 5000>	[-]	$N = \{0,1\}$

Name	Type	Example (content within <>)	UNIT	Allowed values
Num. of obj.	INT		[-]	N
Atens-switch Num. of obj.	INT INT	<1 5000>	[-] [-]	$N = \{0,1\}$ N
Amors-switch Num. of obj.	INT INT	<1 5000>	[-] [-]	$N = \{0,1\}$ N
Atiras-switch Num. of obj.	INT INT	<1 5000>	[-] [-]	$N = \{0,1\}$ N
PHO-switch Num. of. obj. Max. Earth Dist.	INT INT FLOAT	<1 5000 0.0153>	[-] [-] [AU]	$N = \{0,1\}$ N $Q = \{0,0.1\}$
# Additional Settings				
Seed for RDN	INT	<1>	[-]	N
MOID	INT	<0>, <1>	[-]	$N = \{0,1\}$
Stat.Coll.Prob	INT	<0>, <1>	[-]	$N = \{0,1\}$
# Format of the user population file to be generated or provided (1=NEODyS/AstDyS, 2=MPC, 3=DES)				
Switch	INT	<1>, <2>, <3>	[-]	$N = \{1,3\}$
# Population file paths				
Path/name of population file to be used	CHAR(230)	<input/allnum.cat>	[-]	-
Path/name of PPF to be used	CHAR(230)	<input/allnum.ppf>	[-]	-
# Format of the population to be analyzed (1=NEODyS/AstDyS, 2=MPC, 3=DES)				
Switch	INT	<1>, <2>, <3>	[-]	$N = \{1,3\}$
# Population file paths				
Path/name of population file to be used	CHAR(230)	<input/allnum.cat>	[-]	-
Path/name of PPF to be used	CHAR(230)	<input/allnum.ppf>	[-]	-
# Analysis epoch (objects will be propagated to given epoch)				
Switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Reference epoch	I4,5(X,I2)	<2012 05 01 00 00 00>	[-]	-
# Close approach analysis				
Switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Start epoch	I4,5(X,I2)	<2013 05 01 00 00 00>	[-]	-
End epoch	I4,5(X,I2)	<2014 05 01 00 00 00>	[-]	-
Max. distance object - Earth	F7.5	<0.10000>	[AU]	$Q = \{0,1\}$
# Filtered population dump settings				
Switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Dump format	INT	<1>, <2>, <3> (See user-provided population above)	[-]	$N = \{1,3\}$
# Filtered population dump path				

Name	Type	Example (content within <>)	UNIT	Allowed values
Path/name for population subset	CHAR(230)	<path/name-of-subset.cat>	[-]	-
Path/name PPF of subset	CHAR(230)	<path/name-of-subset.ppf>	[-]	-

Table 22: Main input file of the component "Population Generation" (popgen.inp)

9.14 popgen.plt

Type	Format	Examples (content within <>)	UNIT	Allowed values
# 2D histogram settings (one line, blank separated values) **# 3D histogram settings** (two lines, where "Spectra switch" is only in first line!)				
Spectra switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Data Identifier	I2	<01>,<18>	[?]	$N = \{1,18\}$
Logscale switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Spec. lower limit	F7.3	< 0.000>	[?]	$Q = \{0,?\}$
Spec. upper limit	F7.3	< 1.000>	[?]	$Q = \{0,?\}$
Switch class width (1) / no. of classes (0)	INT	<1>	[-]	$N = \{0,1\}N = \{0,\}$
Class width	I3	< 5>	[-]	
# Scatter plot settings				
Spectra switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Data Identifier	I2	<01>,<18>	[?]	$N = \{1,18\}$
Logscale switch	INT	<0>, <1>	[-]	$N = \{0,1\}$
Spec. lower limit	F7.3	< 0.000>	[?]	$Q = \{0,\}$
Spec. upper limit	F7.3	< 1.000>	[?]	$Q = \{0,\}$
# Plot settings (lines displayed in plots)				
Switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Amors switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Apollos switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Atens switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Atiras switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Main Belt switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Trojans switch	INT	<0>,<1>	[-]	$N = \{0,1\}$
Others switch	INT	<0>,<1>	[-]	$N = \{0,1\}N = \{0,1\}$
Unknown switch	INT	<0>,<1>	[-]	
Linetype	INT	<-1>,<2>	[-]	$Z = \{-1,\}$
Linewidth	INT	<1>,<2>	[-]	$N = \{1,\}$
Label	CHAR(30)	<'Amors'>	[-]	_

# Solar System Plot Settings				
Axes switch	INT	<0>, <1>	[-]	$N = \{0,1\}$

Table 23: Population Generator plot settings input file (popgen.plt)

10 Troubleshooting and Feedback

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 Software User Manual and the related Final Report [1]
- the gnuplot Manual [2] (<installdir>/docu/gnuplot.pdf)
- the warning and error messages generated by the software tool and the GUI

In Figure 57 the pop-up window as displayed to the user during the execution of NEOPOP is shown. Warning messages will appear in the logfile while errors will be written to the dedicated error file. This will only be generated if an error occurred during execution.

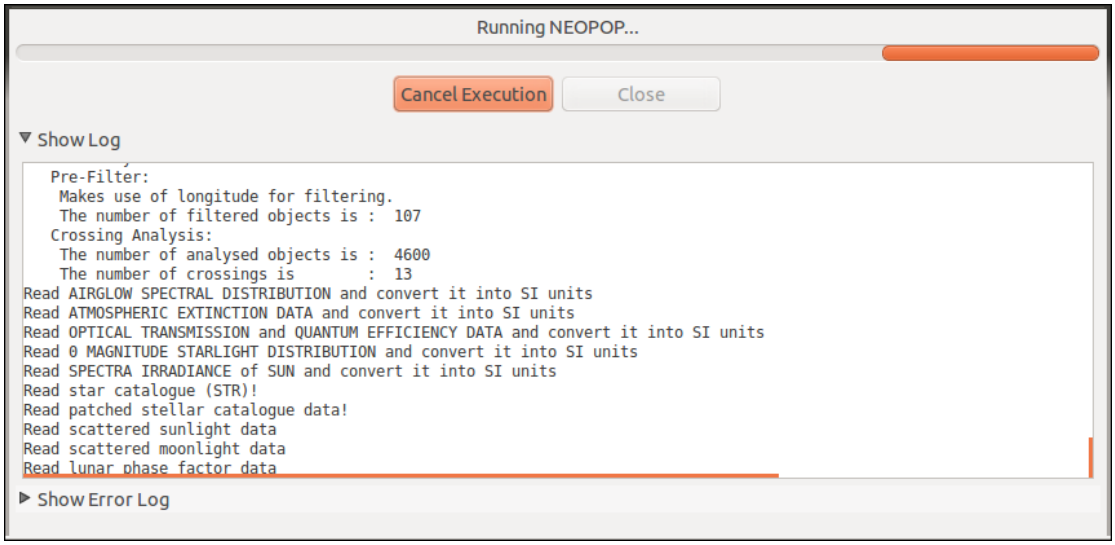

Figure 57: Run dialog with expanded log area

If problems with the GUI occur there is the possibility to look into the log files created in `<personal folder>/neopop-gui/conf`. If that is not sufficient to solve a problem, you may try shutting down the GUI, deleting all its configuration files as listed in section 4.2 and starting it up again.

Upon all that, the next sections list specific problems that may arise and offer suggestions to solve them.

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

http://neo.ssa.esa.int/neo-population

10.1 Crash on Project Load

A known problem of NEOPOP GUI is that it crashes when users open a project with invalid input values in them. If the project which is opened on GUI start-up is the problematic project, the GUI won't start. In these cases a "neopop.err" should be created in the GUI's configuration folder (<Your personal home folder>/neopop-gui/conf). Diagnosing the error messages in this file may give you a hint about the problematic configuration or input file of the project and the input value which is wrong. If nothing helps, try creating a new project and, one after another, copy the configuration and input files into the new project, followed by a use of the "Reset Input to Previously-saved Input Files" function via the Reset Toolbutton in the Toolbar. If the problematic project is the one NEOPOP GUI tries to open on start-up, rename the project folder. NEOPOP GUI will try opening the "NEOPOP" project in your workspace then. It will create that project if it doesn't exist.

10.2 Invalid Value Error on Software Run

Issue: When trying to run a simulation or population generation in NEOPOP, an error message pops up indicating that an input value is invalid even though all necessary input values are correct.
Solution: NEOPOP also checks the validity of values in deactivated/deselected input fields (e.g. deactivated plots). It is possible that one of these values is invalid. If so, activate the respective input field, change the input to a valid value, and run the simulation/population generation again.

10.3 Problems with External Population Files

When reading in external population files, keep in mind that comment lines – including header and column caption lines! – must always begin either with a "!" or "#" character. The character must be the first character in the line; no spaces or something like that are allowed before that. Blank lines also count as comment lines. NEOPOP ignores all comment lines and tries to read all non-comment lines.

10.4 GUI does not Start

One possible explanation for a GUI that does not start is that the GUI bit-architecture and the bit-architecture of the installed (default) Java runtime do not match. You can find out the Java bit-architecture by executing "`java -version`" in a console.

On Windows you can open a console by typing "cmd" into the search field of the start menu. Windows should show you a result named "cmd.exe". Click on this result to open the console.

If the "`java -version`" output contains something like "64-bit", the (default) Java on your system is a 64-bit one. This means that you should install NEOPOP 64-bit. Otherwise, NEOPOP 32-bit is the correct one.

10.5 Forgetting about Global Options

Both NEOPOP variants (32-bit and 64-bit) can function on the same system (when switching between 32-bit and 64-bit Java). However, both use the same GUI configuration folder resulting in "forgetting" global options as, for example, the selected workspace. This information needs to be reentered after switching between the two NEOPOP variants.

11 References

[1] J. Gelhaus, et. al., "*The Near Earth Object Population Observation Program*", Final Report, ESA/ESTEC Contract No: 4000106274, June 2014

[2] T. Williams, C. Kelley, et.al., "*GNUPLOT - An Interactive Plotting Program*" , Version 4.4.4, November 12th, 2011 (Source: http://www.gnuplot.info/docs_4.4/gnuplot.pdf, 09.05.2013)

[3] Bottke, W. F., Jedicke, R., et al., *"Understanding the Distribution of Near-Earth Asteroids"*, Science 288, 2190-2194, 2000.

[4] Bottke, W. F., Morbidelli, et al., *"Debiased Orbital and Absolute Magnitude Distribution of the Near-Earth Objects"*, Icarus 156, 399-433, 2002.

