User Manual
Noysim 2.2.0

Bert De Coensel
Griffith University & Ghent University
bert.decoensel@intec.ugent.be

September 29, 2011

Contents

1 Introduction **1**

2 Installation **2**
2.1 Installing Python 2
2.2 Installing Noysim 2

3 Configuring Aimsun **3**

4 Using Noysim **5**
4.1 Preparing Aimsun networks for using Noysim 5
4.2 Configuring and running Noysim 5
4.3 The output of Noysim 6
4.4 Using the viewer application 7

5 Known issues **7**

1 Introduction

Noysim is an Aimsun (`http://www.aimsun.com`) module for calculating instantaneous road traffic noise levels. Aimsun provides the possibility to create plug-ins, i.e. external applications that interact with Aimsun and have access to internal data during simulation runtime, through the Aimsun Application Programming Interface (AAPI). This way, it is possible to extend the functionality of Aimsun. As Aimsun does not have built-in capabilities to deal with traffic noise emission and propagation, a dedicated plug-in has to be written for this. Noysim calculates the sound power level generated by each vehicle in the simulation at each timestep (through the use of a vehicle emission model), and produces a time history of sound pressure levels at a number of specified receiver locations (through the use of a sound propagation model). Additionally, it provides the possibility to view the sound level at selected locations in real time during simulation.

2 Installation

2.1 Installing Python

The AAPI provides the opportunity to write plug-ins in C++ as well as in Python. The use of Python was chosen for Noysim, because plug-ins written in Python are easier to install, maintain and extend, and because there are a large number of external libraries available, that make it possible to include advanced functionality. The use of Noysim requires that a Python interpreter is installed on the host system (the computer that runs Aimsun), preferably version 2.6 (32-bit edition). This python distribution can be downloaded at `http://www.python.org/`. Additionally, a number of free and open source libraries have to be installed:

- **Numpy.** This is a widely used package for scientific computing with Python, providing functions for linear algebra, Fourier transformation and random numbers. More information can be found at `http://numpy.scipy.org/`.
- **Scipy.** This is another widely used package for scientific computing with Python, providing additional numerical routines on top of Numpy. More information can be found at `http://www.scipy.org/`.
- **Matplotlib.** This library provides plotting functionality for Python, producing publication quality figures, with a set of functions familiar to MATLAB users. More information can be found at `http://matplotlib.sourceforge.net/`.
- **wxPython.** This toolkit provides a wrapper around the popular wxWidgets cross-platform Graphical User Interface (GUI) library, allowing to create programs in Python with a robust, highly functional graphical user interface. More information can be found at `http://www.wxpython.org/`.
- **Xlwt.** This library makes it possible to use simple Python commands to generate spreadsheet files that are compatible with MS Excel. More information can be found at `http://pypi.python.org/pypi/xlwt/`.
- **OpenPyXL.** This library is similar to Xlwt, but makes it possible to generate xlsx spreadsheet files that are compatible with MS Excel 2007 and higher. More information can be found at `http://pypi.python.org/pypi/openpyxl/`.
- **RPyC.** This library implements remote procedure calls and facilities for distributed computing, useful to allow inter-process communication. More information can be found at `http://rpyc.wikidot.com/`.

2.2 Installing Noysim

Noysim consists of 3 parts: a python library, the plugin script (called `plugin.py`), and a script to run the viewer application (called `runviewer.pyw`). The noysim library has to be installed in the same way as the libraries described in the previous section; the easiest way is to use the installer (`noysim-2.2.0.win32.exe`).

3 Configuring Aimsun

It is assumed that a licensed copy of Aimsun 6.1 is installed on the host system, that the license includes using the AAPI, and that an Aimsun Dongle is present in the host system.

Because Aimsun loads plug-ins on a per network basis, it is not necessary to change general Aimsun settings to be able to run plug-ins. However, Noysim assumes some general settings in order to work well:

- Units are set to "metric".
- The North angle is set to 90 (degrees).
- The latitude/longitude is set to decimal.

These settings can be adjusted via `Edit` → `Preferences` → `Location` within Aimsun. Noysim works with right hand drive as well as left hand drive networks (this can also be changed via `Edit` → `Preferences` → `Location`, by setting the "Rule of the Road").

Aimsun can be started in various ways. For use with Noysim, it should preferably be started by double-clicking on the Aimsun network file (which has extension `.ang`). This way, all output of Noysim will be saved in the directory of the network (see also Section 5).

When a network is opened, the project window is shown, which contains the road network as well as all non-geometric information for a simulation. An example is shown in Figure 1. The icons on the left contain tools for editing the road network. Segment properties can be edited by double-clicking on the segment. For example, the posted speed limit for a road segment can be set by double-clicking the particular road segment and selecting "Road Type". The right hand side panel of the project window is used to set vehicle fleet properties and traffic demands for the simulation. Simulation settings are located under the "Scenarios" header.

Traffic demand data can be set using traffic states (demands per section) as well as using origin-demand matrices. For example, when traffic demands are specified as traffic states, the following good practice can be followed:

1. Define the vehicle types.
2. Define the traffic states. A traffic state contains, for a particular vehicle type, the desired traffic flow on each road segment. A default start time and duration can be set, but this can be overridden in the traffic demand setting.
3. Define the traffic demands. A particular traffic demand aggregates the various traffic states that are used for a single simulation. Traffic states for different vehicle types can be included or excluded. The traffic demand window also provides a rapid way of changing traffic volumes by setting the "Factor" (normally 100%). For example, changing to 50% reduces traffic flows simultaneously for all vehicle types by a factor 2; changing to 200% doubles them. Note that Noysim output and file names show the actual traffic flows, after taking into account this factor setting.

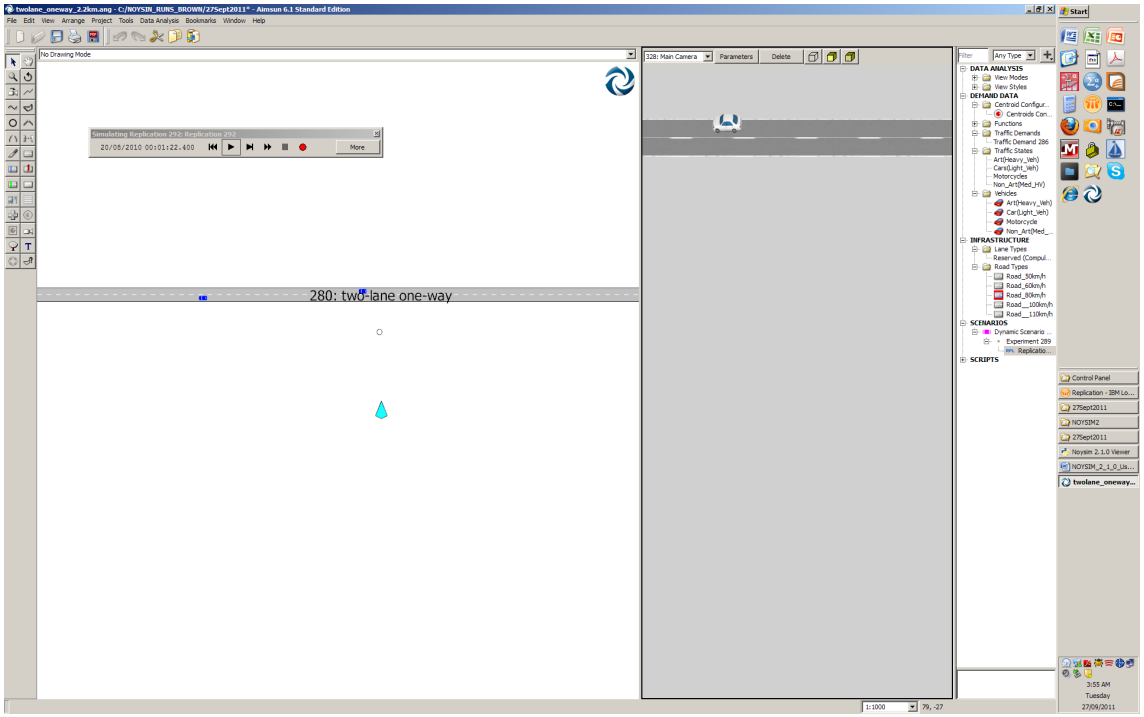

Figure 1: The project window of Aimsun, for a network consisting of a long, straight roadway. An optional 3D panel of the roadway near the receiver is shown; the camera is the blue segment in the left panel. The small circle in the left panel denotes a receiver at 15 m from the centerline of the nearest roadway lane.

Specific to ARC project at Griffith University

A predefined set of vehicle types is available for use with Noysim. These have been based on the types defined in the Queensland Department Main Roads template (see the file `template_QLD.ang`). The parameters of these vehicle types are taken from *Microsimulation Standards*, ARRB Group Technical Note 2 (May 2007). For Noysim, only four vehicle types (matching Imagine categories 1, 2, 3 and 4b) are used. The dimensions and dynamic characteristics of these vehicle types have been matched to `template_QLD.ang` as follows:

`template_QLD` vehicle type	Noysim vehicle type	Imagine category	Austroads category
Car_QLD	Car (Light_Veh)	1	1 & 2
LT (Light truck)	Non_Art (Med_Veh)	2	3
ST (Semi trailer)	Art (Heavy_Veh)	3	4–12
Motorcycle	Motorcycle	4b	N/A

4 Using Noysim

4.1 Preparing Aimsun networks for using Noysim

Noysim does not pose specific requirements on networks. In order to place the Noysim receivers correctly, one has to know the exact (x,y,z) coordinates of some relevant sections/junctions of the network; the coordinates of the cursor can be found in the Aimsun status bar (bottom right corner). As far as it is known to the author, there is no easy way for exactly specifying the coordinates of sections and junctions in Aimsun.

Given an Aimsun network with traffic demands (coded using one or more dynamic scenarios, experiments and replications), the Noysim plug-in can be loaded for a particular scenario, by double-clicking on the "Dynamic Scenario" icon, selecting the "Aimsun API" tab, and adding the file `plugin.py` to the bottom list (make sure the check box is marked). Subsequently, when replications in this scenario are run (e.g. by right-clicking on the "Replication" icon and selecting "Animated Simulation"), Noysim will be loaded and executed.

4.2 Configuring and running Noysim

When a replication of a dynamic scenario that uses Noysim is run, a small pop-up configuration window will appear at the start of the simulation (see Figure 2).

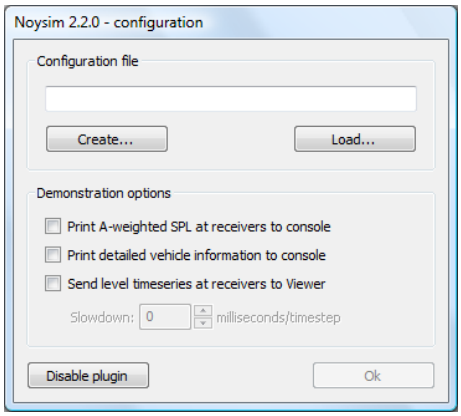

Figure 2: Noysim configuration window.

This window allows to select the Noysim configuration file that has to be used for the simulation run, and shows some additional options for visualization during the simulation. When no configuration file is available, a new file can be created, which will contain default settings that can be changed by the user if needed. When a file named `default.noysim` is available in the network folder (or in the same folder as `plugin.py`, depending on how Aimsun was started), this file is loaded automatically. Therefore, when the same configuration is used repeatedly, it can be useful to create a configuration file named `default.noysim`.

Noysim configuration files are regular text files with the extension `.noysim`. To have a feeling about the structure, and to create custom configuration files, one may open a

configuration file with a text editor. Noysim configuration files have the following sections:

- **emodel.** This section contains a number of settings for the emission model. The "name" field denotes the model type, and can be "Imagine", "Distribution" or "Lookup" (not yet implemented). The flags for the Imagine model turn a number of corrections on/off (more details about these corrections can be found in the Imagine Deliverable 11). In case the Distribution model is chosen (name field set to "Distribution"), the model additionally corrects the Imagine emissions with a random per-vehicle correction, sampled from a normal distribution with zero mean and with standard deviation supplied by the field "distribution-stdev" (5 comma-separated values, one for each Imagine vehicle class).

- **viewport.** This section defines the region in which vehicles have to be considered for noise calculations. The "rectangle" field contains the coordinates of the bottom left and top right corners of a rectangular area for consideration $(x_{BL}, y_{BL}, x_{TR}, y_{TR})$. The "dynamic" viewport type is not yet implemented; in future versions, when the "dynamic" field is set to True, a rectangular viewport will be automatically created, based on properties of the network (vehicle flows, distances of the receivers etc.), making a compromise between speed and accuracy.

- **pmodel.** This section contains a number of settings for the propagation model (currently, only the ISO 9613-2 model is implemented). Again, a number of flags turn a number of corrections on/off. The field "iso9613-ground-coeffs" contains the ground effect coefficients for the source, receiver and middle region resp., where 0.0 denotes a hard ground and 1.0 a soft ground (values in between are also possible).

- **receivers.** This section defines the locations of the receivers as a semicolon-separated list of (x,y,z) coordinates, and the (constant) background sound pressure level present at the location of the receivers.

- **output.** This section defines some additional output settings. The path and filename for the output file can be set here. When these fields are empty (the default), output files are stored in the network folder, and are given a descriptive filename containing the scenario, experiment and replication number, the maximum traffic flow and maximum speed for any section in the network, and a unique run number. When the field "spectra" is set to True, the octave-band spectra of the noise at the receivers is also included in the output (see below).

When the configuration file contains errors (e.g. typos, missing or non-defined values), the Aimsun log window will show an error (red icon, with a hopefully useful explanation), and the plugin is disabled. Options for visualization include writing the A-weighted SPL at each receiver location to the Aimsun log window at each timestep, writing detailed information of each vehicle at each timestep to the Aimsun log window, and supplying A-weighted sound pressure levels to the viewer application (see below).

4.3 The output of Noysim

At the end of a succesful simulation run, a Noysim output file is saved (location and filename as defined above). This is an MS Excel file containing several worksheets:

- **simulation.** This sheet contains general information about the simulation, such as the duration of the warm-up period or the time step, together with the speed limits and total demands for all sections in the network.
- **plugin.** This sheet contains the plugin configuration used for this simulation.
- **counts.** This sheet contains actual vehicle counts for each section, for the complete duration of the simulation (so no need to add detectors to the network). Additionally, this sheet contains detailed information of vehicle pass-by's at the origin along the x-axis (only useful for test networks with a single road along the x-axis).
- **levels.** This sheet contains a time history of the A-weighted sound pressure level at the location of each receiver (the corresponding coordinates can be found in the sheet "plugin"), taking into account the background level. Additionally, a column with the pass-by's at the origin along the x-axis is given.
- **spectraxx [optional].** These sheets contain timeseries of the (non-weighted) octave-band spectrum at the location of each receiver. Note that here, the background is not taken into account, because nothing is known about the spectral shape of the background noise.

4.4 Using the viewer application

In order to view noise levels in real time during simulation, the option "Send level timeseries at receivers to Viewer" has to be selected in the configuration window. A slowdown option is also available, which can be useful when a small network is run on a fast computer. Additionally, it is necessary to start the viewer application before the simulation is started, by double-clicking `runviewer.pyw`. The Noysim viewer (see Figure 3) shows the time history of the A-weighted sound pressure level at each receiver location, and contains basic functionality to adjust the axes.

5 Known issues

The Python API of Aimsun does not provide an easy way to retrieve the type of a vehicle (car, motorcycle, truck, etc.) in a non-ambiguous way (a type number can be retrieved, but the meaning of this number may change between networks). The type of a vehicle is of course an important factor when calculating noise emissions. Therefore, Noysim uses the dimensions of a vehicle to "estimate" its type. The length and width ranges that are currently coded into Noysim are shown in Table 1. When Noysim encounters a vehicle in the network that it can not classify according to this table, it will raise an error and the plugin will be disabled. Therefore it is necessary that all vehicle types, for which traffic demand is defined in the network, have their dimensions inside one of the intervals in Table 1.

Unlike the C++ API, the Python API of Aimsun does not provide the possibility to retrieve the location and name of the current network. Therefore, the output MS Excel file does not include the name of the network. Also, the output location of this file will be the location from where Aimsun was started. If Aimsun was started by clicking on the icon

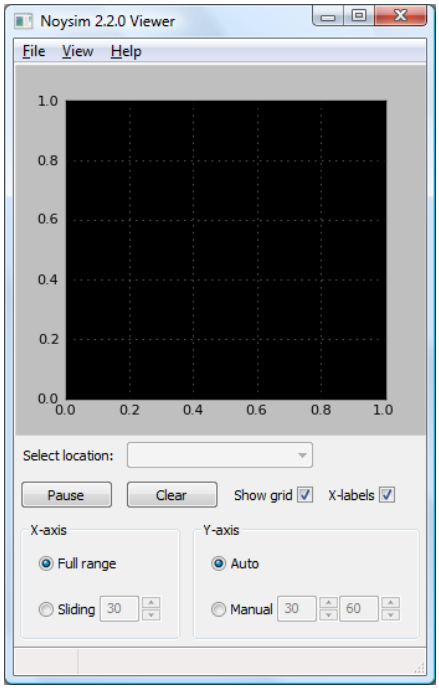

Figure 3: Noysim viewer application.

Table 1: Length and width ranges for determining the vehicle type.

Vehicle type	Length	Width	Imagine category
Car	3.0–5.0	2.0–2.0	1
Van	5.0–9.0	2.0–2.0	1
Light truck	10.0–14.0	2.5–2.5	2
Semi trailer	19.0–19.0	2.5–2.5	3
B-double	25.0–25.0	2.5–2.5	3
Motorcycle	1.5–1.5	0.6–0.6	5

of a particular network, the output will be saved at the path of the network. However, if Aimsun was started by clicking on the icon on the desktop, all output files will be saved to the desktop. As far as the author is aware of, there is no easy way around this.