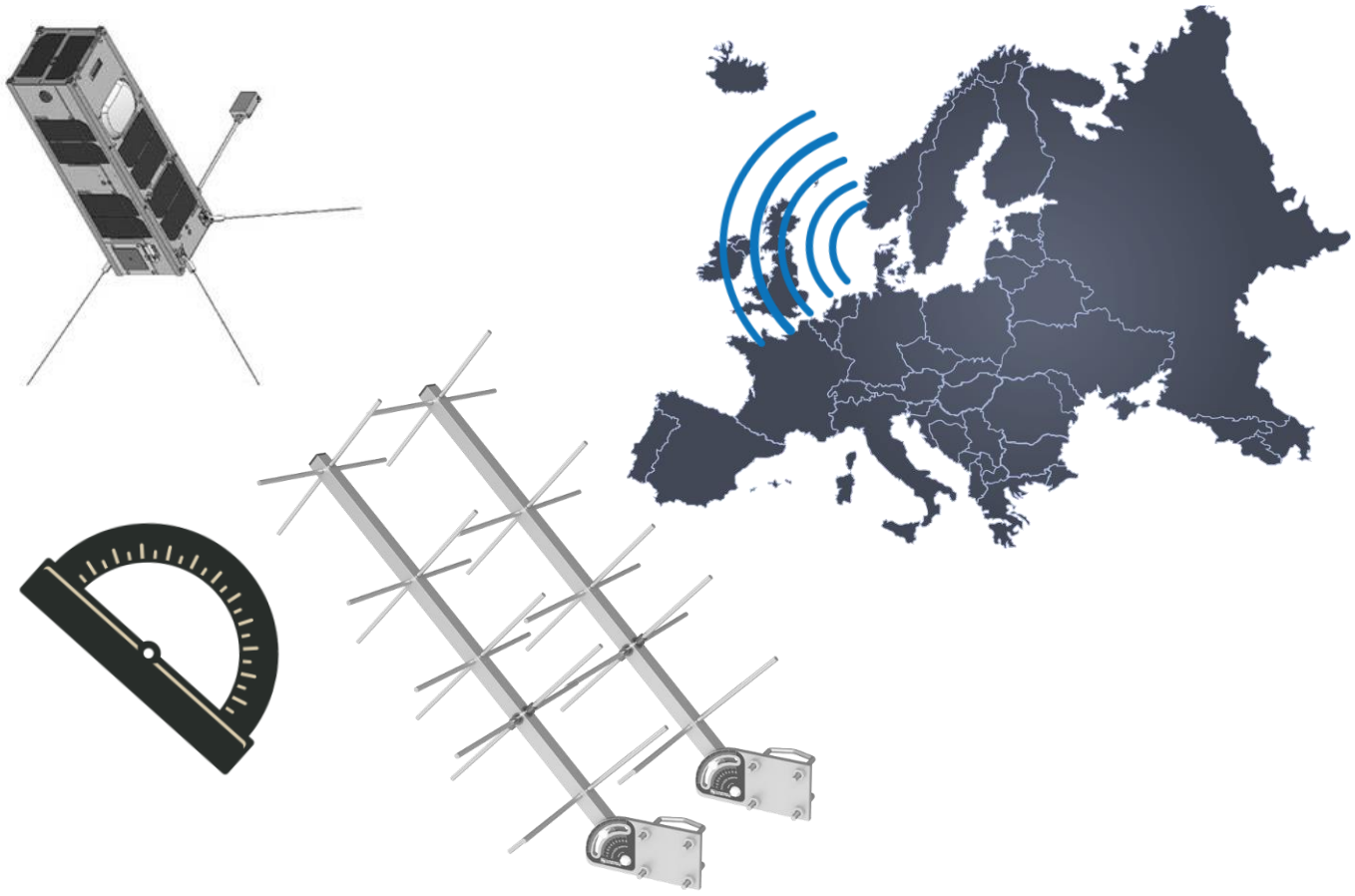
Aarhus University

Report

AARHUS UNIVERSITET

User Interface for Delphini Missions' Ground Station

By Esteban J. Struve T.
MSc Computer Engineering



Delivery for Delphini-1, Software and Satellite Operations Workshop [225171U019]
Winter 2017

CONTENTS

1. INTRODUCTION

The first satellite mission of Aarhus University, Delphini-1, calls for students, faculties and industry professionals to join forces and to take part in a multidisciplinary effort to build an educational, functional and scalable platform for building and operating this first *cubesat* and the next satellites to come. To contribute for the operations once the satellite is launched and also when calibrating, maintaining and using the ground station to establish communications with other entities, the control room must have interfaces to perform these tasks.

This report describes a simple but powerful tool to track the position of the antenna in a map according to the angles of the two rotors to move and point to the sky. Each section will describe how it was built to thoroughly describe its functionalities. Finally, some recommendations are made for other students to follow up in the development of the tool to meet future needs.

2. SOFTWARE

The software is two scripts: one small script (*anglerange.py*) for creating the ranges of angle values for Azimuth and Elevation and a second script (*uiantenna.py*) that runs the visual user interface that reads this values from the CSV file created by the first.

The CSV file is very simple: two columns of values separated by comas. The first column has the angles for the Azimuth angles, which describe how the antenna turns counterclockwise from the default position pointing north, moving towards the west (with no elevation angle), then south, east and finally back to north in a full range until the value 360 degrees (the same as 0 degrees). The second column has the values for the Elevation angles, which describe how the inclination of the antenna changes for the default totally horizontal position and opening the angle with respect to this line parallel to the floor, reaching 90 degrees value in full vertical position (where Azimuth values does not change the antennas pointing target) until reaching the full 180 degrees values facing the opposite way of 0 degrees in a totally horizontal position. The file named *angles.csv* is saved in the same path as the scripts.

The user interface is built using Python 2.7 with Pygame. It is considered by its creators as a Free and Open Source python programming language library for making multimedia applications. It is a cross-platform set of modules designed for writing video games and this user interface is based on it. The way it runs is simple: First the script reads and saves the angle values, performing some processing power consuming operations, then the Pygame modules are initialized, a window is created to display the information which is running in a while loop that performs the operations necessary to display the information

and interact with the user and closing the window with exit Pygame modules and stop the script. Everything necessary for the use of the interface is in the window.

In order to run the UI, Python 2.7 and Pygame must be install in the computer.

Having everything set up, the common way to use the tool is:

- Create the range values of Azimuth and Elevation you want to see the antenna perform. Modify *anglerange.py* and run it, the CSV *angles.csv* will be saved.
- Run the UI (*uiantenna.py*) and use the Mode and Zoom of preference to navigate these values and see the information displayed.

3. VISUAL LAYOUT

3.1 Screen Sections

The screen or window built as interface for the user to visualize the data and interact is split into in indication section and two interactive sections. This division describes the function of each section.

3.1.1 Indication Section

The section located in the upper right side of the screen will be the indication section. This part of the screen is reserved to display information about: Angle values (Azimuth and Elevation), position of the cursor (latitude and longitude) and time (elapsed in the simulation).

3.1.2 Interactive Section

The interactive sections are two: The map area and the button area.

The Map Area is the bigger left side of the screen. It consists of 800 by 800 pixels map the user can explore with the cursor and where the lines and blue circle target with indicate where the antenna is pointing at. In this section the user can interact with the direction keys (Manual Mode) and with the cursor (mouse).

The Button Area is the lower right section of the screen, reserved for buttons. In this section there is the Mode Switch to change Manual to Automatic Mode and vice versa and the buttons that will only appear in Automatic Mode.

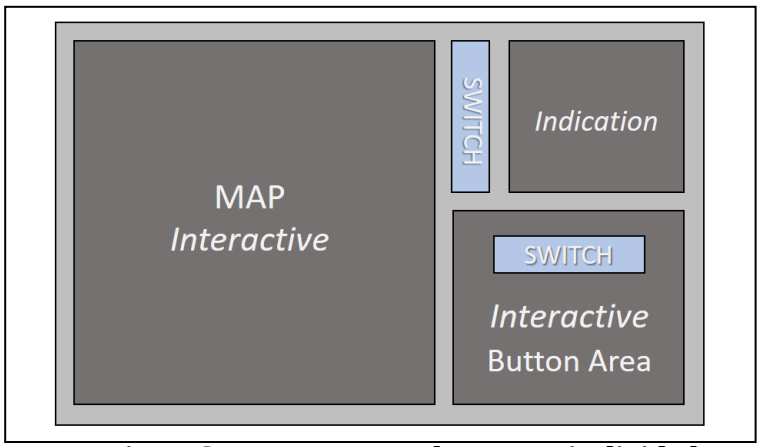


Fig. 1. Screen Layout. The screen is divided into the shown sections.

4. OPERATION MODES

To provide more ways to use the interface, this first version of the user interface has two modes of operation: Manual Mode and Automatic Mode.

4.1 Manual Mode

The Manual Mode consists of simply controlling the antenna with the direction keys in the computer's keyboard. The user can user the up and down keys to navigate the Elevation values of the angles in the CSV file that is read at the beginning when running the interface. Likewise, the left and right keys can navigate the Azimuth values in the same file. In this operation mode, only the switches, cursor (inside the map) and the keys can be used, no other buttons will appear in the Button Area.

4.2 Automatic or Simulation Mode

To provide a swift and interactive way to show the movement of the antenna, the Automatic Mode will provide the way to use the timer to navigate the angle values simultaneously to see how these angle values translate into a maneuver. When the antenna is tracking a satellite both angles are change accordingly to follow the path of the satellite in the sky. These values can be inputted in a compatible CSV file and a simulation of the maneuver can be seen in the map. How to build a file for this use case is out of the scope of this report.

The buttons will appear in the Button Area when selecting this operation mode. The *Play* button will start the timer and change to *Playing* when timer is running. The *Stop* button will pause the timer and allow to resume pressing *Play* again. The *Reset* button will turn the timer to 0 again to start again the simulation. The design follows the KISS principle.

5. MAP FEATURES

The map is a square map with two Zoom Modes: The 500 m per 100 pixel scale and the 50 km per 100 pixel scale.

5.1 Zoom Modes

The two Zoom Modes can be change using the Zoom Switch locate vertically aligned in the right side of the map. Clicking each side of the switch will change the Zoom Mode.

The small-scale map allows the user two see the position of the antenna with respect to Aarhus city. The majority of the streets in a 2 km radius can be resolved and buildings can be used as reference. In this map with a range of 400 km of radius, only the Elevation angles close to 90 degrees can be showed, due to the maps size. In the code the scale is 1:9 for the angles in this map.

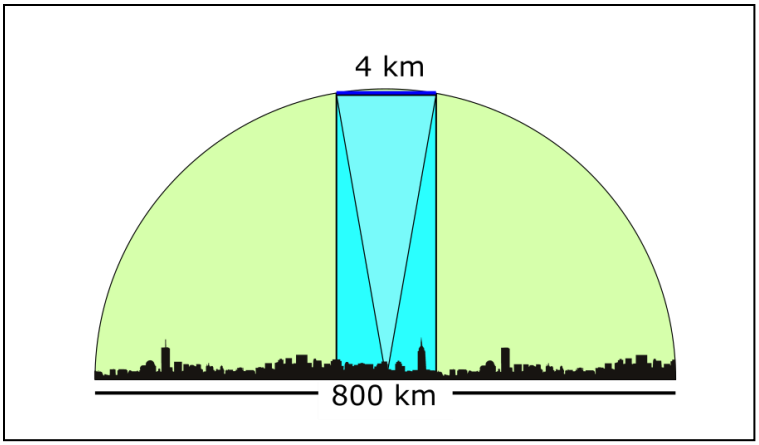


Fig. 2. Scale of map 1 is 4km represented in the navy blue horizontal line.

The big-scale map allows the user to see the position of the antenna with respect to other cities and countries. In this scale with the same range, more angle values can be resolved in a full 200 km radius of the map. In the code the scale is 1:3 for the angles shown.

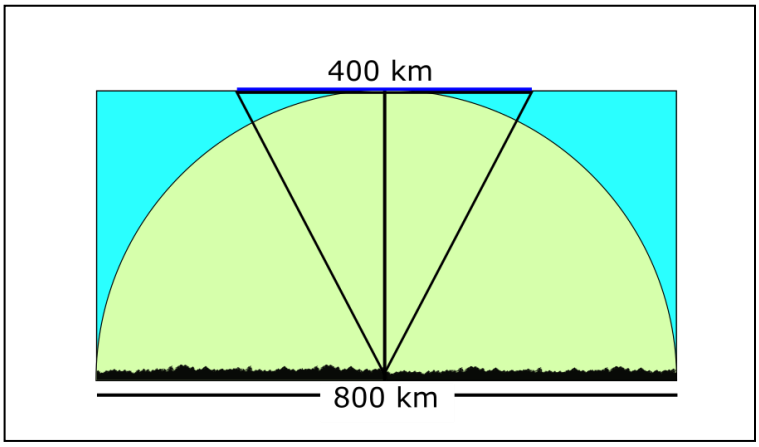


Fig. 3. **Scale of map 1 is 400km represented in the navy blue horizontal line.**

5.2 Lines and Equations

To describe the principles used to calculate coordinates properly for the interface, trigonometry and the equation of the line was constantly used in several parts of the design and code.

The green line represents the antenna's direction with 0 degrees of Elevation and the yellow line represents the antenna's direction with 180 degrees of Elevation. These two lines split the map in to halves and change in angle according to the angle of Azimuth. The blue circle is a target representing the place antenna is pointing to, projected onto the map shown.

5.3 Uses for Cursor

The cursor inside the map is represented as yellow arrow pointing down to the example pixel in the map, which corresponds to the latitude and longitude values display in the Indication Section in the right side of the screen. The four margin markers will follow the cursor moving inside the map to provide a natural way for the user to relate to the pixel coordinates of the screen. All thing elements are restricted to the Map Area.

An additional action is allowed when using the cursor: saving a location. If the user wishes to mark a particular location in the map to see the latitude and longitude and then use some other tool to find some extra information, it can be done using the right click of the mouse. When the cursor is pointing the desired location, right-clicking will save the location by stop the update in the latitude and longitude display in the screen. This way the user can see this information for as long as need. To drop the save location and continue using the cursor in the map, a right-click again will do.

6. RECOMMENDATIONS

The scope of the first version of this user interface focused on creating a friendly, simple and interactive way of displaying the information of the antenna position. There are many improvements that can be made following this finished version.

The input values can be extracted directly from the instructions of the server based on the TLE of the object to be tracked. This way maneuver can be viewed in the interface before, during or after it happens. The operation modes can be much more and the features in each mode can be also enriched with new ones. For example, the way the simulation is controlled can be improved by having a timeline with a cursor to navigate the simulation faster to the point of interest. Additional Zoom Modes can be added to provide a wider range of scale and information. Also, providing ways to retrieve the maps directly from the services like the Google Maps API can proof to be very useful.

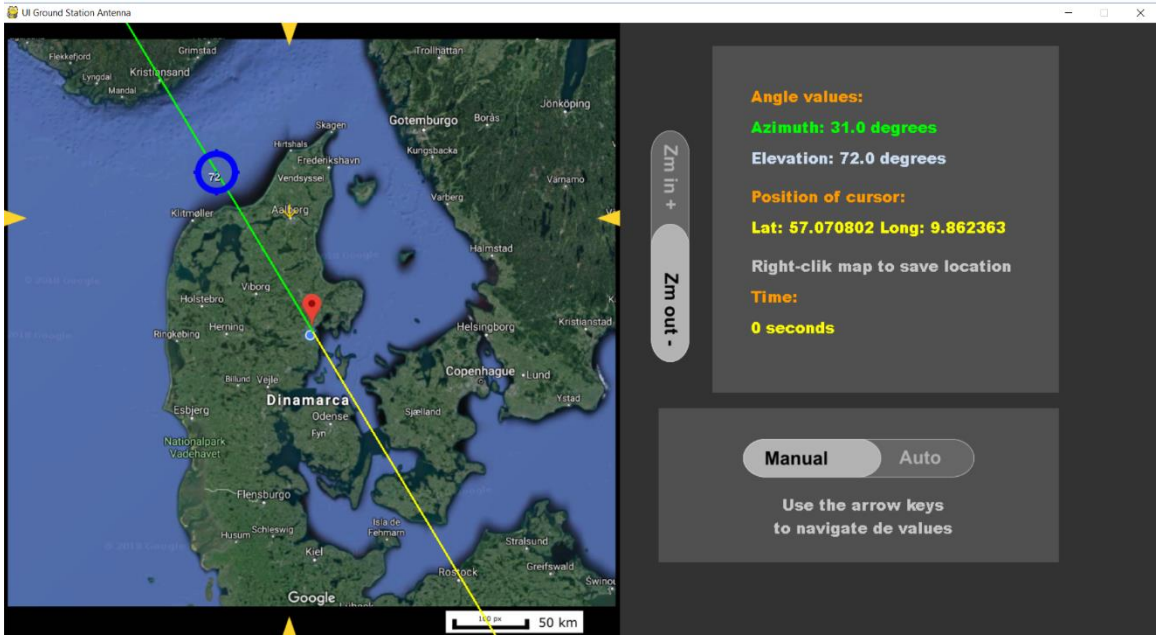
For all these, changes in the uiantenna.py must be made accordingly and the code was built to be comprehensive enough with variable convenient variable naming and abundant comments throughout the script.

APPENDIX A

A.1 Photo of Antenna

A.2 Rotors, controller and antenna base

A.3 Photo of User Interface (Zoom Out)



A.4 Photo of User Interface (Zoom In)