ObsTools - Greenhill Observatory Observation Planning Dashboard

Thomas Plunkett

June 16, 2023

The Greenhill Observatory (GHO) 'ObsTools' dashboard is a tool to aid in planning and executing astronomical observations with the UTAS optical telescopes. It is written solely in Python, utilising the 'Dash' package to implement an interactive dashboard on your local machine. The main calculations use the Astropy, Astroplan and Skyfields packages. The dashboard contains a basic sky map, table with astronomical information and unit conversions, airmass plotter and 3 variations of exposure time calculation plots. In this document, I will outline how to install and use this tool.

Installation

It is recommended that the user downloads and installs Anaconda or Miniconda to manage their python distributions. First, download/clone the Github repository to your local machine: ObsTools Github. You will need to change into the downloaded directory, using:

cd [insert path to folder here]

Then, you will need to install the necessary requirements to run the dashboard. To do this, go to your terminal (or Anaconda Prompt on Windows) and create a new conda environment (for example 'obs'):

conda create -n obs python=3.9.16

You will then need to activate the environment:

conda activate obs

Then, assuming that 'pip' installed into this environment during the process, type:

pip install -r requirements.txt

The environment name should now display next to the current directory. You should now be able to use the dashboard! If you don't want to install locally, a copy of this code is installed on BT-RConsole3 in the Squid Lab.

How to Use

The dashboard has 3 main sections, which I will walk through here. To run the dashboard, activate your conda environment, change to the ObsTool directory and type:

python app.py

This will launch the dashboard on your local host. Go to a web browser and type the address given in the command line.

Part 1 - Sky Map and Table

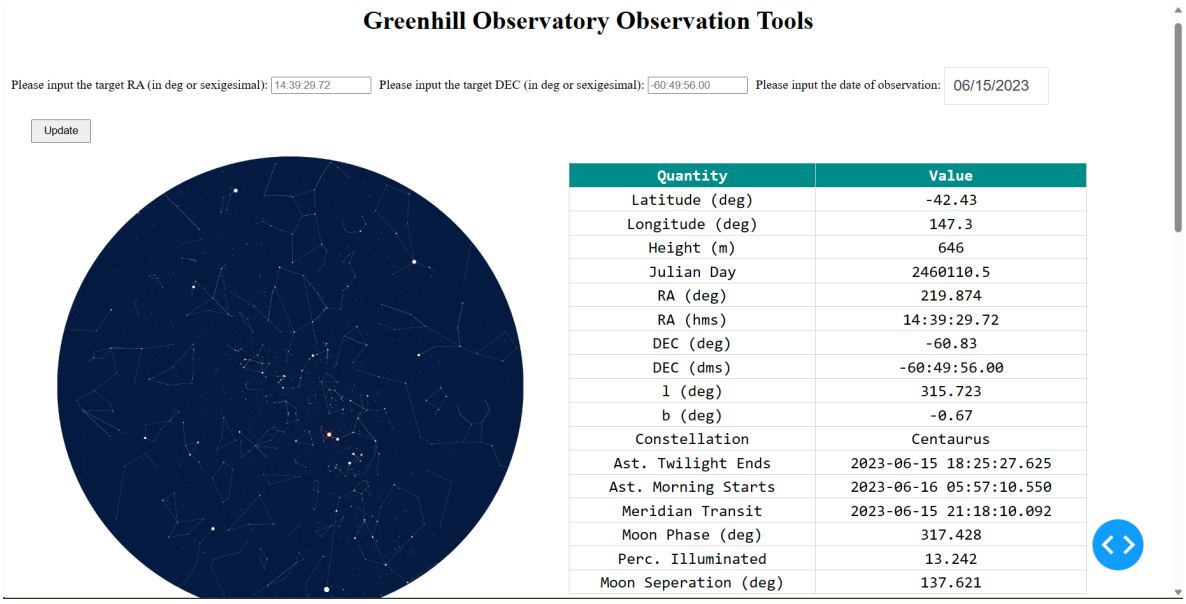

Figure 1: A screenshot of the first section of the dashboard, containing the sky map and table. The inputs for target RA/DEC and a calendar are above.

The first section of the dashboard contains a basic sky map, showing the visible constellations (based upon the time of year) and the location of the target (shown by red cross). Note, this view will be for midnight of the date picked. The star positions come from the 'hip_main.dat' file, which is a catalogue from the Hipparcos satellite. The constellations are defined in the 'constellations.fab' file, which comes from Stellarium. This sky map will update based upon the inputs in the RA, DEC boxes and the date selector/calendar, once the 'update button' is pressed. You can enter RA/DEC in either sexigesimal representation (i.e hour angle and degrees, separated with ':') or degrees (but please do not mix the two!).

The table next to the map contains basic information about the observatory, observing conditions and relevant times, along with coordinate conversions for the target. Again, this updates based upon the inputs from the user and the date picked. It is worth noting two choices in this table. First, the 'Ast. Twilight Ends' and 'Ast. Morning Starts' are a proxy for when it will be dark enough to observe. These are defined by the Sun being -18 degrees below the horizon. Next, the 'Moon Phase' given is the angle between the moon's ecliptic longitude and the Sun, meaning that a 'new moon' will display 0 degrees and a 'full moon' will display 180 degrees (with a range from 0 to 360). These quantities are calculated using the Astroplan package.

Part 2 - Airmass Plots

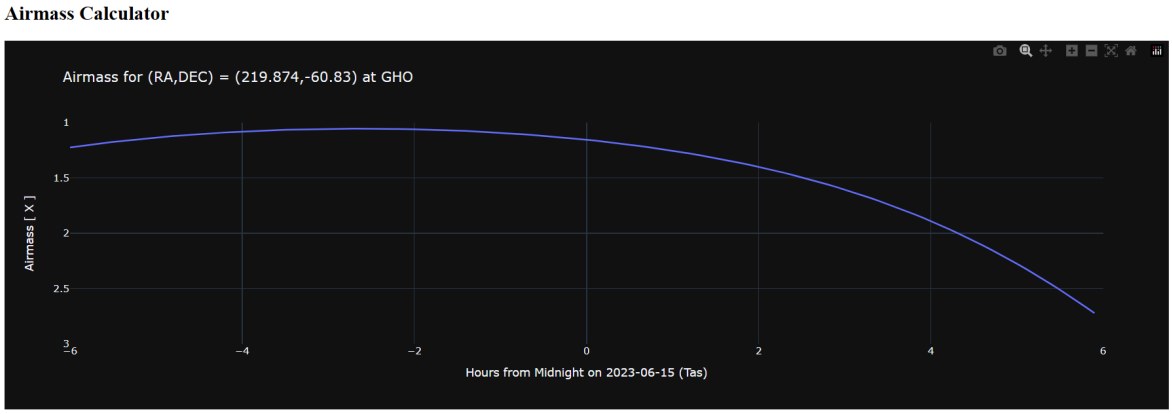

Figure 2: A screenshot of the airmass plot, with time given as hours from midnight on the specified date.

This section plots the airmass as a function of time for a specified target. This will update based upon the RA, DEC values given in the boxes and the date chosen (from the first section). The 'airmass' is defined as:

$$\chi = sec(z) \tag{1}$$

Where, z is the object's altitude above the horizon. The airmass values are calculated using the Astroplan package, by defining the observatory location using the Observer class and then creating a 'FixedTarget' from the RA,DEC given. The 'altaz' function is then called and finally converted by taking sec() of the target altitude. Note, this plot defaults to a range of $\chi = 1$ to 3, as it is generally not a good idea to observe above this (below in terms of altitude). However, you can rescale the plot to show the full range using the 'autoscale' on the top right of the plot. You can also save this figure locally (as a .png file) using the camera button in the same location. As with all Plotly plots, you can hover over the line to get values at each sampled point and can zoom into regions of interest.

Part 3 - Exposure Time Calculator

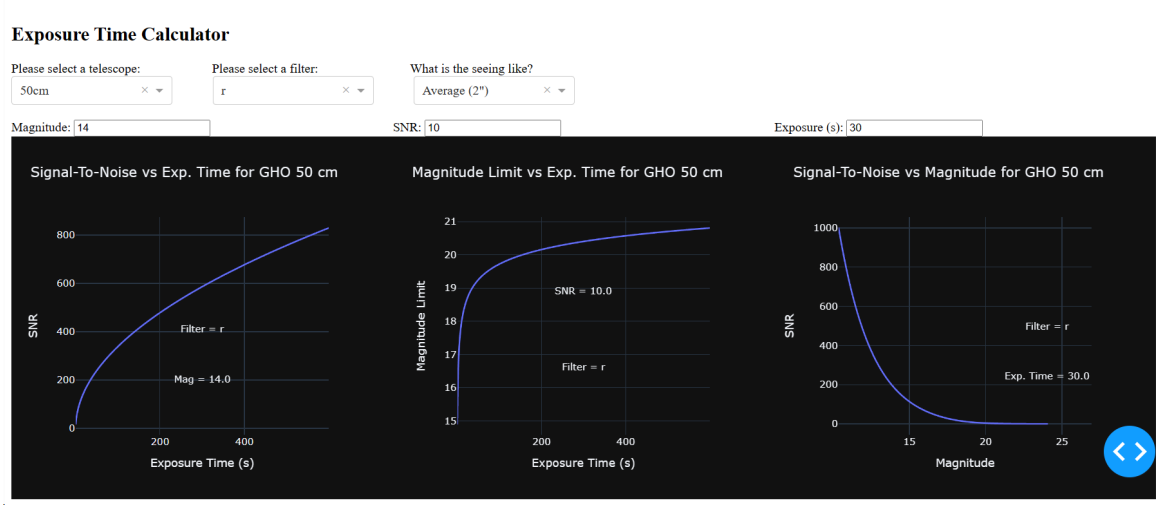

Figure 3: A screenshot of the Exposure Time Calculator section. The left plot shows SNR as a function of exposure time, given the magnitude of the target. The middle plot shows the magnitude limits as a function of exposure time, given a desired SNR. The right plot shows SNR as a function of magnitude, for a fixed exposure time.

This final section provides 3 plots related to signal-to-noise, exposure time and magnitude limits. These plots will update based upon the middle (filter selector) and right (seeing) dropdowns, along with the values inputted into the boxes above each plot. Furthermore, the date chosen by the user will also influence the calculations, based upon the moon phase on the specified date. Note, currently only the 50cm is implemented, so the telescope dropdown does nothing.

The calculations for these plots rely on a mixture of theory and experimental fitting. The central equation is the so-called 'CCD Equation' (Howell 2011):

$$SNR = \frac{E\Delta t}{\sqrt{E\Delta t + n_{pix}(R\Delta t + D\Delta t + rn^2)}} \tag{2}$$

Where, E is the average target signal (in e/sec), n_{pix} is the number of pixels to consider, R is the sky background rate (in e/pix/sec), D is the dark current (in e/pix/sec), rn is the read-noise (in e/pix) and Δt is the exposure time (in seconds). In my calculations, I have used:

$$n_{pix} = \frac{2 \times FWHM}{pixel\ scale} \approx \frac{2 \times FWHM}{0.8} \tag{3}$$

Here, the FWHM refers to the seeing (i.e the stellar FWHM). The signal rate is given by:

$$E = F_{obj}\,(\pi r_{tel}^2)\,T\;q\,\Delta\lambda \tag{4}$$

Where, F_{obj} is the object flux density (photons/sec/cm^2/nm), q is the quantum efficiency of the detector, $\Delta\lambda$ is the bandpass and T is the transmission efficiency (i.e of atmosphere, telescope and camera).

When using this tool, the following guidelines can help to interpret the SNR result. An SNR of less than 3 is a poor quality observation, the target is barely detectable. You should generally aim for $10 < \text{SNR} < 1000$, which will allow for a solid detection without saturation. It is worth remembering that, for a good detection:

$$\delta m \approx \frac{1}{SNR} \tag{5}$$

Final Remarks

I hope this tool will make your life easier. I aim to keep updating this tool, hopefully including the 1.3 metre and eventually improving the sky background model. If you have any suggestions for improvements or find any bugs, feel free to email me. Clear skies!

References/Packages

The Astropy Collaboration: Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., et al. (2022). "The Astropy Project: Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package". *The Astrophysical Journal*, 935(2), 167.
Package avilable at: https://github.com/astropy/astropy

Dash Package (by Plotly): https://github.com/plotly/dash

Howell, S. B. (2011). "Handbook of CCD astronomy". *Cambridge University Press*.

Morris, B. M., Tollerud, E., Sipocz, B., Deil, C., Douglas, S. T., Medina, J. B., Vyhmeister, K., Smith, T. R., Littlefair, S., Price-Whelan, A. M., Gee, W. T., & Jeschke, E. (2018). "Astroplan: An open source observation planning package in python". *The Astronomical Journal*, 155(3), 128.
Package available at: https://github.com/astropy/astroplan

Rhodes, B. (2019). "Skyfield: High precision research-grade positions for planets and Earth satellites generator". *ASCL.net - Astrophysics Source Code Library*. http://ascl.net/1907.024.
Package available at: https://github.com/skyfielders/python-skyfield