Manual DJI flight planner

Introduction

The DJI flight planner python program has been developed to plan flight missions for the DJI Matrice 300 RTK. It has specifically been developed in combination with the Yellowscan Mapper+ LiDAR module, however different LiDAR systems should be able to use the software as well.

This manual first explains all the python packages that are required, explains how the software should be started and how the software should be used. Known errors of the program are listed at the end.

If you have feedback or ideas how to improve the software, you can email: M.O.N.Brandwijk@student.tudelft.nl.

1. How to install the required python packages

Option 1: Using the .yml file to create an python virtual environment in conda

When Anaconda is installed on the computer the following method can be used to create an virtual environment to run the flight planner using the *environments.yml* file.

1. First open the Anaconda prompt. This can be done in windows by searching: Anaconda prompt and opening this.
2. Use the following code to install the virtual environment with the correct dependencies. The *environment.yml* file should be the path (or relative path) to the location you stored the file.

```
conda env create --name flight_planner --file=environments.yml
```

3. Activate the environment using:

```
conda activate flight_planner
```

4. When you are finished and no longer require the program you can remove the virtual environment using:

```
conda deactivate flight_planner

conda remove --name flight_planner --all
```

Option 2: Creating your own virtual environment with the following packages and python version. The packages can be installed with pip.

```
python==3.10.9
dash==2.8.1
dash-leaflet==0.1.23
dash-extensions==0.1.11
dash-bootstrap-components==1.3.1
pyproj==3.6.0
shapely==1.8.1
numpy==1.25.2
```

2. How to run the software

The following steps should be taken to run the software:

1. Open a terminal with the correct virtual environment activated.
2. Use the cd command to navigate to the folder *dji_flight_planner/flightplanner* in the terminal. This is important as some part of the code uses a relative path, which does not function when this step is omitted.

```
cd own_path_to_flightplanner_folder
```

3. Run the program using:

```
python main_application.py
```

4. Click the link that appears in the terminal to open the DJI flight planner program. Or go to an internet browser and use the following html: *http://127.0.0.1:7781/* .
5. The program should now appear as Figure 1 in an internet browser.
6. Shut down the program by using *ctrl+c* in the terminal.

3. How to use the software

Using the software can be done in two steps. First the area of interest needs to be chosen. Then the flight parameters can be set.

Selecting an area of interest

When opening the software, the screen should resemble Figure 1. In this window you can draw an polygon over the area of interest. The software will help determine a flight plan in the next step over this area with the preferred flight parameters. The interactive map on the right side of the screen can be used to move to the area of interest. By pressing the hexagon symbol on the left top side of the map, the polygon draw functionality will be started. By clicking with the mouse pointer on the preferred corners of a polygon and finally on the starting position, a polygon over the area of interest will be created. An example of a polygon in the software is shown in Figure 2. The layer symbol on the right top of the map can be used to view the map as google satellite layer. When the polygon is as desired, the green “Finished” button can be clicked on the left side of the screen. This will start the next step of the flight planning.

Current limitations to the polygon:

- Currently the flight planning tool only works in the Netherlands (this is caused by certain coordinate conversions).
- Polygons that deviate far from a convex hull might create bad unrealistic flight plans. This can however be seen in the following step.

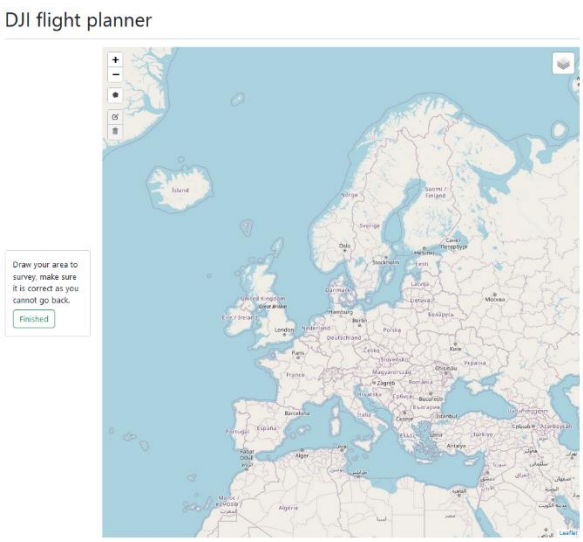

Figure 1: Opening window

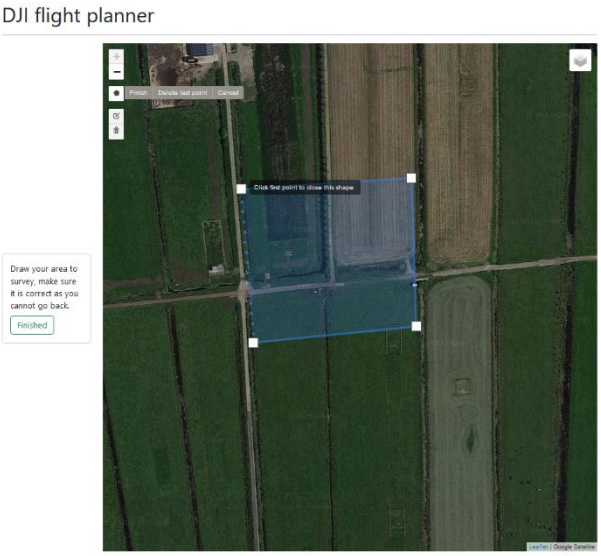

Figure 2: Selecting area of interest.

Setting flight parameters

The setting of flight parameters window will look similar to Figure 3. The buttons on the left side of the screen can be used to set the preferred flight parameters. When the preferred flight parameters are selected, the flight plan can be downloaded using the download button on the bottom left of the screen. The base filename of the downloaded mission can be changed with the text box in the left bottom corner.

Furthermore, it is possible to download multiple different flight plans and perform them close after each other. For example, two flights with similar parameters but with orthogonal flight lines can be used to perform a “grid” flight.

Set mode

The set mode option on the top left side screen can be used to use different flight parameters as input for the flight plan. In Standard mode the distance between flight lines needs to be set in meters. In Yellowscan Mapper+ mode, the UAV LiDAR parameters, height and overlap parameters are used to calculate the corresponding distance between flight lines.

Flight info

The flight info on the left side of the screen, gives estimates of flight characteristics of the flight that is shown on screen. It should be noted that the estimated point density has not been validated yet. Furthermore, the estimated flight time is of significant importance. The UAV LiDAR system has a flight time of around 23 minutes per battery set. Therefore it is convenient to limit the flight time of a flight plan to this time. However it is possible to partially fly a mission, change to a new battery set and continue the mission.

Flight parameters

- Angle [^{o}]: Range (0^{o}, 360^{o}). The angle the flight lines make to North in degrees.
- Offset [-]: Range (0, 0.5). This value moves the flight lines perpendicular to itself over the polygon of interest. The amount of shift is the fraction of the distance between flight lines.

- Buffer [m]: Range (0m, 20m). Polygon of interest is increased with this size in all directions. This parameter can make sure the turns are performed outside the region of interest.
- Damping [m]: Range (0.2m, 50m). Maximum radius of the turns. When the this value is to large for the radius of the turn, the maximum radius possible for the turn is selected.
- Height [m]: Range (0m, 300m). Height of the UAV LiDAR system above the take off location. The minimum height depends on the take-off location and objects in the area of interest. With the current UAV license a maximum height of 120 meters above the ground is allowed.
- Speed [m/s]: Range (0.1 m/s, 15 m/s). The speed the mission is flown. For most purposes a value between 5 m/s and 10 m/s is chosen. A too low value will cause the IMU to drift significantly. A too high value will lower the point density.
- Overlap [-]: Range (0, 0.5). Fraction of overlap between two adjacent flight lines.
- Distance flight lines [m]: Range (0m, 120m). Distance between flight lines.

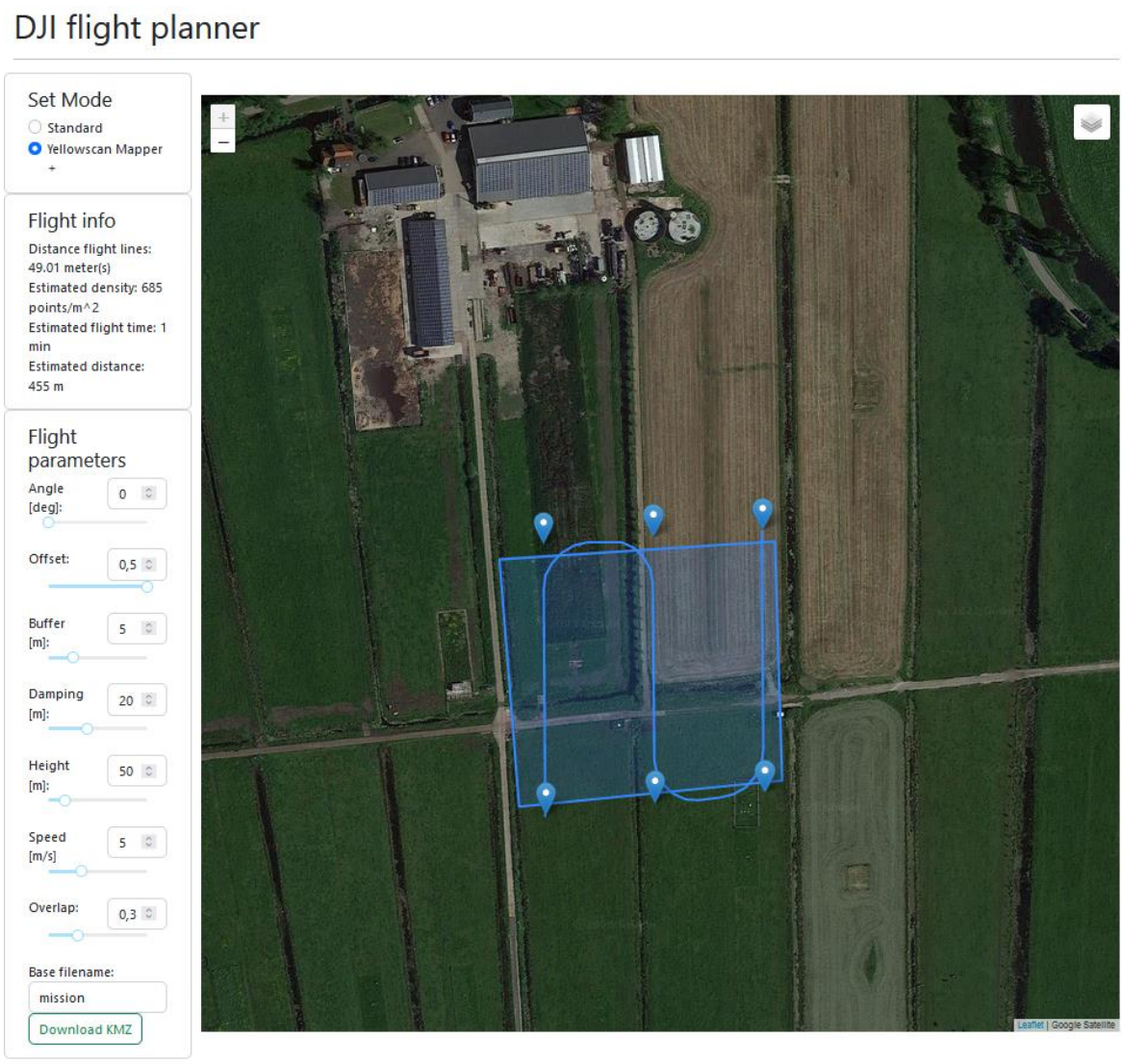

Figure 3: Choosing flight parameters

Current limitations of the flight planning:

- When flight parameters are given as input that should not be a possible combination according to the software the original flight parameters are not updated.
- When flight lines are set to be very close, the corner radius between flight lines becomes very small. This has not yet been tested, but will likely cause the speed of the UAV to lower significantly in the corner and result in a not smooth corner. This might lower the IMU accuracy slightly

4. Known errors

- Red border around flight parameters: This red border is caused by values that should not be allowed by the step size of the number box of the parameter. However, the value that is drawn in the map corresponds to the final flight plan that is downloaded. Therefore, if the flight plan on the map looks as desired, this should not cause a problem in the downloaded flight plan.
- Map of the flight parameter window will not load: This error occurs when during the first step the polygon has been drawn and “Finished” is pressed. The second window that opens has in this case a not functioning map. Usually this can be fixed by reloading the webpage and drawing the polygon again. This error is caused by the fact that the second interactive leaflet map is hidden and only later revealed. When this leaflet map is not loaded before it is hidden it will not function. This error cannot be easily fixed with another method.