Heat sensor on wheels

Monica Calleja, Nuria Olivares and Hector Pascual.
Engineering Physics. Universitat Politecnica de Catalunya
North Campus, 08034 Barcelona

(Dated: May 31, 2019)

Data analysis is a process for obtaining raw data and converting it into useful information for decision-making by users. This process is present in all the advances of our surroundings regardless of the subject: technology, medicine, safety, etc. In this project two big areas of new technology are combined: data analysis and electronics. This project is based on the construction of a Rover (unmanned vehicle) which is later used to obtain and analyze heat, humidity and GPS data of a field.

I. INTRODUCTION

Drones have become a normality in our day-to-day basis. Until the past-decade, they were unknown and unused for most people. Only people related with the field of aerospace engineering or people who were interested in the subject knew from its existence. Nowadays, it is common to see them on the news and some models have become an affordable ”toy” that is mostly used in photography or just to have fun. But their applications go far beyond due to its key aspect, being an ”unmanned aerial vehicle”[1]. Every day people find new ways to benefit from its advantages and they have become an essential part of our world.
Rover is an advanced open source autopilot for guiding ground vehicles and boats. It can run fully autonomous missions that are defined using mission planning software or pre-recorded by the driver during a manual run.[2].
The rover of this project runs on the Pixhawk autopilot and its aim is to cover a field and collect a set of values of position, temperature and humidity in order to process them afterwards, and generate a heat map and a humidity map. This project can be useful in many different fields due to its target to obtain information from a location. Its different applications will be further commented later.

II. PURPOSE

The aim of this project is to generate a map with information of the temperature and the humidity of an area. To do so, a Rover will be used in order to go over a selected zone in an autonomous way, by being previously programmed with a route that covers the points of interest. External heat and humidity perturbations will be put so that the result shows some differences among the different points. For the data collection, the use of a microcomputer board will allow to program a heat sensor and a GPS to obtain the position of the different values that the sensor provides. The microcomputer will be controlled with an external device and a program called VNCviewer that allows to view the screen in a remote way. Once the data is obtained, it will need processing, therefore it will be exported in order to perform the final steps of the evaluation of the zone. The selected area used to perform the acquisition of data is a field near Castelldefels in Barcelona.

III. THE CONSTRUCTION

The construction can be subdivided in two parts: the building of the rover and the data acquisition elements that needed to be placed on the rover. In figure 1 are shown the following elements indicated:
First step of the construction, rover building:

1. **Pixhawk-Autopilot**: It is an independent open-hardware project that aims to provide the standard for autopilot hardware designs.[3].
2. **GPS Compass**: GPS connected to the Pixhawk so that the Autopilot has the information of the position needed to start the route/mission.
3. **Telemetry Radio**: A telemetry radio allows the Rover to communicate with the ground station remotely. This allows to interact with the missions in real time and receive streaming data from the vehicles components.
4. **Buzzer**: Used to play several sounds that indicate the state of the rover’s components.
5. **LiPo battery**: Battery for the rover.

Second step of the construction, data acquisition:

6. **LattePanda**: As previously mentioned, it is a complete microcomputer board integrated with Arduino.
7. **DHT11**: A temperature and humidity sensor compatible with Arduino.
8. **GPS Antaris**: An additional GPS to obtain the geographic position (Latitude and Longitude) of the Rover each second in order to synchronize the sensor’s data.
9. **Wifi Adaptor USB**: This item was necessary in order to have wireless internet access for the remote control of the LattePanda’s screen.
10. **Common portable battery**: Battery for the LattePanda.

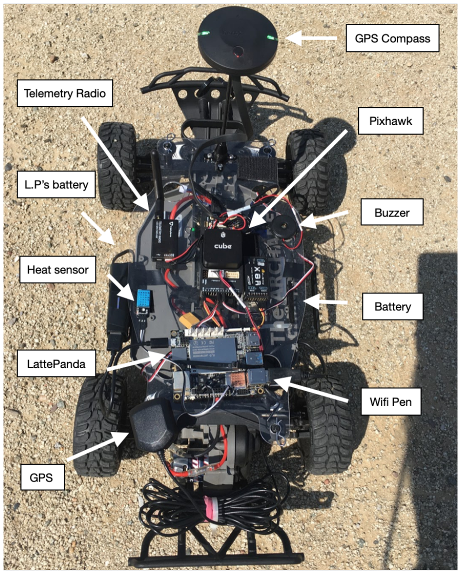

Figure 1: Rover with all its elements indicated

IV. THE MISSION PLAN

The rover can run fully autonomous missions that are defined using mission planning software. Mission Planner is a full-featured ground station application for the ArduPilot open source autopilot project.[4] This application was used not only to program via way-points the autonomous routes that the Rover followed, but also to calibrate the device and to later analyze the telemetry data. The steps were the following:

1. Mandatory Hardware Configuration. As part of first time setup, some hardware components must be calibrated, such as the compass and the accelerometer.

2. In the same program, select the box named "Flight Plan" in order to design a route. Select an area on the satellite map in which the route will be designed through different way-points.

Two plans were carried out to cover all the field in 'Castelldefels School of Telecommunications and Aerospace Engineering'.

Figure 2: First route on the left and second route on the right

3. Chose an appropriate speed: In this case, 1m/s in order to obtain enough data to create the heat map, since the DHT11 sensor has a frequency lecture of 5 seconds.

4. Save the flight plan and write it into the Rover's memory through telemetry. After the mission, an analysis of the results obtained through telemetry can be carried out.

V. LABVIEW PROGRAM TO OBTAIN DATA

LabVIEW is a very useful tool when it comes to work with sensors and the obtaining of data. The main VI consisted on a block diagram which opened a VISA resource and configured a COM port. It then performed a serial port read displayed on the front panel. When the read was performed, this VI waited until the specified number of bytes were received at the port and then displayed them. The number of bytes specified in our case were 100, enough to read a full string. Two different block diagrams were needed, one for the GPS and another for the sensor.

However, the purpose of using a LabVIEW program was not only to read the serial port bytes, but also to store them in the Lattepanda's memory as a text file which could be later processed.

After being converted to doubles, the arrays corresponding to the GPS and the sensor data were plugged each one into a datalog function, which stored them in the Lattepanda's memory. In order to synchronize the data of the GPS and the sensor, a function to get the time and data strings was added and also plugged into the arrays of the sensor and the GPS.

VI. MATLAB TO DRAW THE HEAT MAP

The first step for drawing the heat map is to filter the data obtained from the sensor and the GPS. Because of the mismatched periodicity of the sensor and the GPS, the first filter applied was to synchronize the data and obtain a matrix with the location, humidity and temperature values. Since some data was missing in the matrix due to perturbations on the GPS, the matrix was altered with the data corresponding to the posterior position since they were less than a meter apart.
The data was obtained in a field of about $100m^2$ so not much change on the temperature and humidity of the area was expected. This is why during the development of the mission, external perturbations were introduced in order to obtain differences between the data points in the map, so that it could show variations. The perturbations were created by several things: creating a shadow over the sensor in order to decrease the temperature or approaching a heat source in order to increase it. Furthermore, as it can be observed in the results, a wet cloth was put onto the sensor so that both the humidity and temperature would be altered.

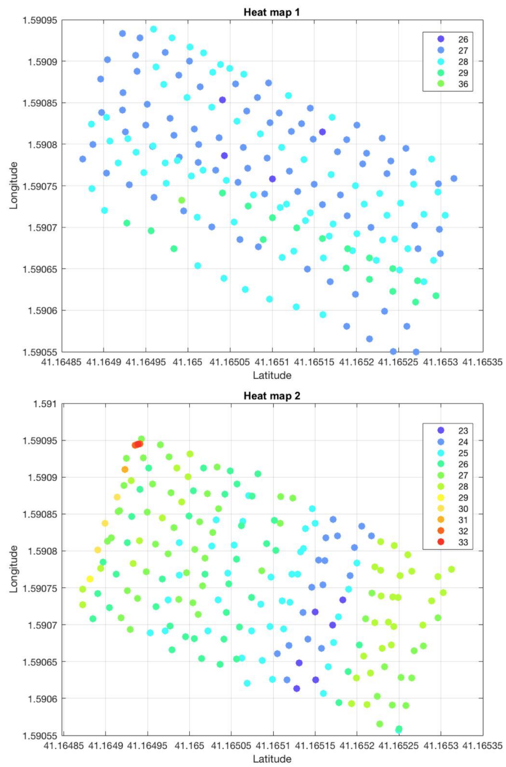

Figure 3: Heat maps or the route 1 above and the route 2 below.

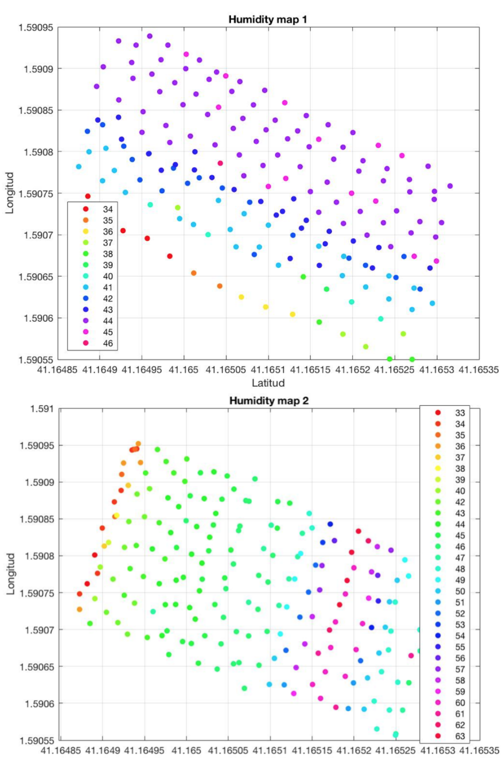

Figure 4: Humidity maps or the route 1 above and the route 2 below.

In the first route, only perturbations of heat increase were externally put, therefore the range of temperatures is [26 - 29]°C. We consider the value of 36°C to be non valid since it is far from the maximum of all the other points. The range of humidity varies between [34 - 46]%. Since the sensor wouldn't detect a temperature change unless it was very noticeably, the first heat map is very monotonous.
For the second route, the goal was to diminish the temperature and try to change the humidity values. The range of values of the temperature is [23 - 33]°C and the range for the humidity is [33- 63]%. The first external perturbation was to create a shadow over the sensor so the temperature would decrease. The results can be observed on the center-left side of the second heat map which has an average temperature of 25°C. Taking into consideration that the sensor was previously heated by the sunlight, such perturbation implied a decrease in the average temperature of the zone. On the center-right side of both maps a decrease in the temperature and a increase of the humidity can be detected, in view of the fact that this path was altered by putting a wet cloth over the sensor.

VII. ROVER TELEMETRY ANALYSIS AND SIMULATION COMPARISON USING SITL-MISSION PLANNER

Aside from creating the heat map of the zone by performing two routes, further analysis on the behaviour of the rover during the mission was performed.

Software In The Loop (SITL) is a simulator that allows you to test your missions before carrying them out with your actual vehicle. However, SITLs simulations are not perfect. In fact, SITL uses different ground models that are not perfectly according the reality and has a generic model for the Rover that does not take into account neither the configuration nor the weight. Keeping this in mind, a simulation can be made with the same parameters and the same flight plan of the real flight. Analyzing the data stored through telemetry in the Mission Planner during the real flight, it is possible to determine how reliable the simulation is, and take it into consideration when planning future missions where the rover has to go near obstacles or complete a manoeuvre at a particular velocity.

The telemetry data was obtained from the '.log' files of the missions and it was uploaded to the Mavlink Log Graph in order to plot different relevant parameters of the flights.

As it is expected, there are some differences between the simulation, which tends to an ideal path with theoretical errors, and the real path. These differences can be observed in the following graphs. They also show the z velocity component, which gives information about the height difference between the points of the route and shows the time stamps where there were slopes in the field.

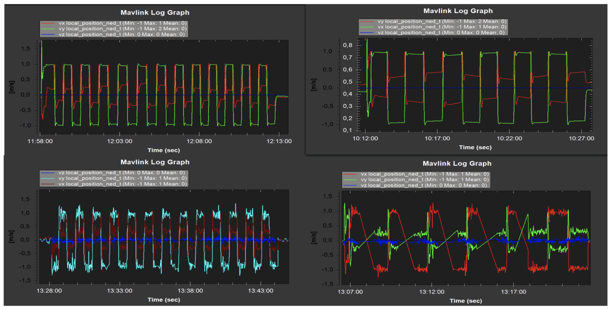

Figure 5: Comparison between the simulation (above) and the performance of the rover (below) for the route 1 on the right and the route 2 on the left.

Because of these slopes on the ground, and some other factors, the velocity of the rover is not constant at the value of 1m/s. With every turn, it diminished its velocity and compensated it afterwards. Noticeably, the main difference in velocity that is produced corresponds to the turns on the routes.The following figure shows the evolution of the velocity of the rover with the time and the simulation of the parameter, not taking into consideration the slopes:

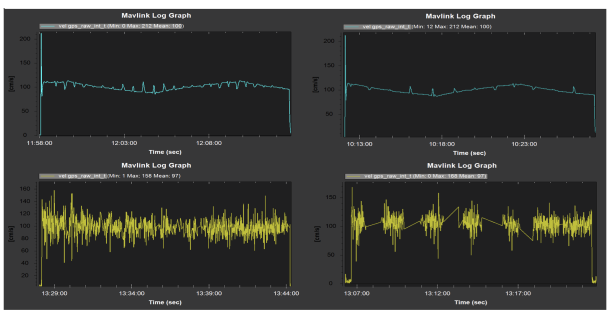

Figure 6: Comparison between the velocity of the simulation (above) and the one of the rover (below) for the route 1 on the right and the route 2 on the left.

Another parameter that can be studied when analyzing the performance of the rover is the number of satellites that the GPS was receiving information from at each time. In both routes the GPS captured signal from 15 and 16 satellites on average. This number is a good result since it makes the value of the position of the rover more reliable.

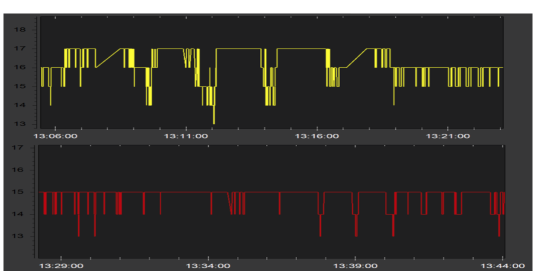

Figure 7: Number of satellites that the GPS was receiving information from at each time for route 1 on the top and the route 2 on the bottom.

VIII. APPLICATIONS

This project can be useful in a wide range of fields. The Heat sensor on wheels has fully autonomy to go through large fields without any worries.

One of the main problems of rice field is Rice blast caused by Pyricularia oryzae, teleomorph Magnaporthe grisea. It is the most destructive of all rice diseases. Many attempts have been made to develop models to forecast rice blast. The most frequent input variable has been air temperature, followed by relative humidity and rainfall. Critical factors for the pathogenesis, such as leaf wetness, nitrogen fertilization and variety resistance have had limited integration in the development of these models.[5]
An improved version of this Rover could be used to explore a rice fields and take the data of the temperature and humidity of this rice fields to try to avoid this problem or at least try to learn the source of this disease.

This is just one example, but this rover could have much more applications.

IX. CONCLUSIONS

The main intention of the project was to build and fly a drone, but due to limitations when it came to the legalities of flying a remote control object near the airport, obtaining data to create an application was not possible. However, thanks to the investigation towards it, our knowledge of the drone's applications has increased.
Not being able to fly a drone in the aereal espace near the laboratory led us to use the rover, which was a good option since the mission plan could be performed without much problem and there was a big variety of applications that could be implanted on it.
Different programs were used during the project, we chose the ones that were more practical to use in order to have a more dynamic progress. When collecting the data, Labview turned out to be the best option since it compatibilizes well with Arduino and could be easily installed in the Lattepanda. Also Matlab is always a good ally when it comes to processing data, therefore it was our choice for drawing the maps.
As commented on a previous section, not much change on the temperature and humidity of the area was expected because the data was obtained in a small field. But due to the external perturbations that were added, it is possible to observe some changes in the maps.
There were several inconveniences when it came to collect the data that showed the limitations of our project, i.e. the rover and the Lattepanda were not protected, therefore collection was not possible during a rainy day. Finally, the remote control of the screen of the microcomputer needed internet access, which makes the project dependent of a source of internet close to it.
Despite the inconveniences explained above, we are extremely satisfied with the project as it has proved to show really good results.

X. ACKNOWLEDGEMENTS

We would like to thank professor Pablo Royo for his help in the introduction and construction of the Rover and also for teaching us its usage. We would also like to thank professor Oscar Casas, who has helped us configure the data acquisition part of this project. Both have provided us with the elements that we needed in order to perform the project.

---

[1] Wikipedia, "Unmanned aerial vehicle," https://en.wikipedia.org/wiki/Unmanned$_aerial_vehicle$.
[2] Rover Home, "Ardupilot dev team," http://ardupilot.org/rover.
[3] Ramin Sangesari, "Getting started with lattepanda," https://www.hackster.io/idreams/getting-started-with-lattepanda-c093e0.
[4] Media Assets, "Pixahawk," http://pixhawk.org.
[5] Christos Dramalisand Pau Puigdollers Dimitrios Katsantonis, Kalliopi Kadoglidou, "Rice blast forecasting models and their practical value: a review," .