Mission Space Lab Phase 4 report outline

Team name: DgVeneto

Chosen theme: Life in Space

Organisation name: CoderDojo Fossò-Venezia

Country: Italy

1. Introduction

Our goal is to understand whether the Astro Pi computer is able to measure microgravity in the ISS.
To do this we decided to detect the movement of objects or people in front of the Astro Pi.
These are the key points of our idea:

- The main sensors to work with are the gyroscope, the accelerometer and the magnetometer. The camera was only used as a sensor, in accordance with the guidelines.
- Microgravity is measured in G, so it's an acceleration. After cleaning the data from saturations such as peaks in mG, mechanical vibrations such as peaks in micro-G, and so on, we can detect if there are fluctuations present only in the periods in which there are objects or bodies detected by the camera.
- We can be sure of the intensity match if we can detect the distance of the object or body from the Astro PI and can estimate its mass. With this data we can calculate the force of microgravity between Astro PI and the detected object or body.

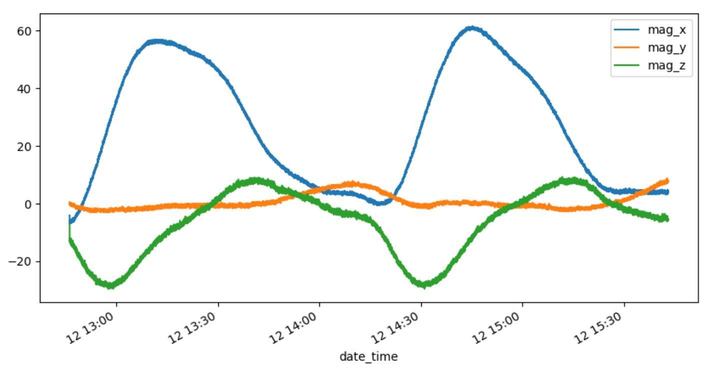
2. Method

Having decided to determine if the Astro Pi is capable of measuring microgravity, we used the IMU sensor, which included the gyroscope, the accelerometer, and the magnetometer.
We used the camera only as a sensor: we took a picture, saved it in memory and used a machine learning pretrained model for human detection, present in the cv2 (OpenCV) library to detect how many humans were present at any time.

We assumed the average weight of a human is 70kg and approximated the gravitational force that might pull the AstroPi, by using Newton's universal gravitation law.
It is important to note that the human detection function required the entire code to stop for about 7 seconds each time the Camera was accessed. Therefore, we used this function only once every 60 IMU sensor measurements, each of which took about 0.1 seconds. In total, about 50% of the measurements were lost, while the Astro Pi was busy taking pictures. Therefore, to account for this, during the analysis phase, we wrote a function to fill in these "gaps" in the received data. In future experiments, we might make use of the threading library to use the PiCamera without stopping the entire code.

3. Experiment results

The metrics of the magnetometer show a different behavior than that described in Vidhya's code.



It is strange that `mag_y` has an almost flat behavior and `mag_x` is so important: maybe it depends on the position of the Astro Pi relative to the Earth.

We have not detected any human by using the PiCamera as a sensor, as is shown by the following line of code:

```
data['humans_detected'].value_counts()
0    50520
```

Instead we have found movements by using MotionSensor, as is shown by the following line of code:

```
data['motion_detected'].value_counts()
0    45979
1     4541
```

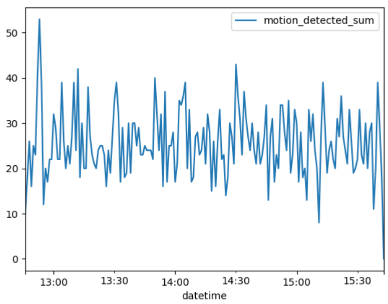
We calculate some motion_detected derivatives (interpolation, Simple Moving Average, ..) and we tested their values vs each metrics

- with the raw values directly

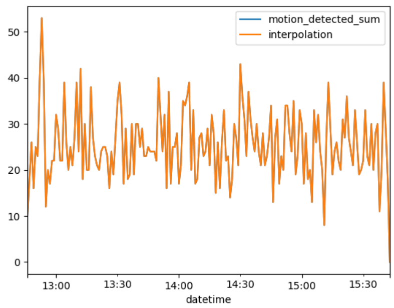
- with the delta between value and average
- with the difference between consecutive rows

The linear correlation (calculated using Pearson's coefficient) between the sensor metrics and the motion detection metric is very low.

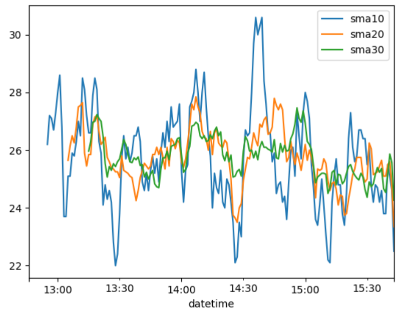
That is the distribution per minutes of `motion_detected` values:



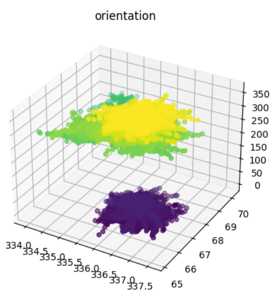
Interpolation is not sufficient to see anything significant:



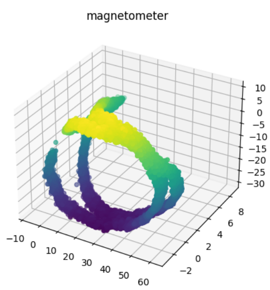
We also analyzed the Simple Moving Average (applied every 10, 20 and 30 samples) of the motion detection data:



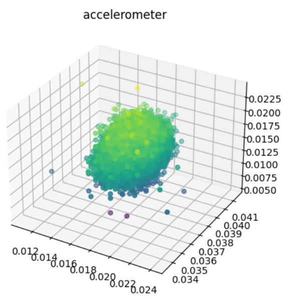
orientation



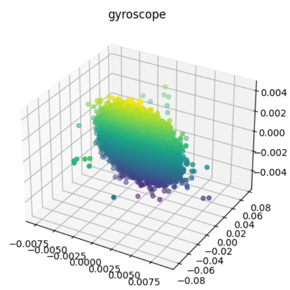
magnetometer



accelerometer



gyroscope



We elaborated the motion data by doing two things:

- Each measurement is associated to a timestamp (in either microseconds or seconds) relative to the start of the experiment.
- All the "gaps" that are present inside the measurement data are filled by adding estimated measurements that can be of any value (-1 or 0 for example). The estimated measurements are evenly spaced inside the "gaps". The number of estimated measurements inside each "gap" is determined by dividing the duration of the hole by the average delta of each measurement.

More info: [README.md]() and [GeneralAnalysis.ipynb]()

4. Learnings

Not living in the same city, we met mostly online, and only twice in person.

To coordinate our tasks we mainly used a WhatsApp group.
The documents were written using Google Workspace.
We used GitHub for code developing, which not all the team members knew about.
The greatest difficulties we faced were the definition of the initial idea, the division of tasks and supporting the single students.
WhatsApp is not a very suitable tool for this purpose. If the young people do not know each other, collaboration is difficult and a greater number of face-to-face meetings should be organized, especially in the start-up phase.
We have learned that a tool or a method is needed to develop the idea of the experiment in a shared way to encourage everyone to be creative.
Moreover, next time we will be sure to better study the limits and potential of Astro Pi.

More info: README.md

5. Conclusion

The metrics of the magnetometer show a different behavior than that described in [Vidhya's code]().

The Pearson's correlation is very low between metrics and the feature named `motion_detection`.

The microgravity needs at least 10^{-8} m/s^2 and the resolution detected may be useful to detect it. Now it is difficult to use only `motion_detection` and delta of acceleration values because we don't have the distance detection.
It would be interesting to add an ultrasonic distance sensor to use also the class DistanceSensor and then be able to detect the distance of the motion: if there are positive `motion_detection`, a `distance_detection` interesting and a swing of acceleration, we can define if that swing is only noise or microgravity.
Another strategy could be to detect the distance of the object with a machine learning algorithm by using the PiCamera as a sensor.

We came to the conclusion that according to the data we collected and processed, it could be possible to detect microgravity with Astro Pi but we need to take advantage of Python threads to remove gaps and adding distance detection to confirm if the swing is microgravity.

More info: OurAim.ipynb and GeneralAnalysis.ipynb