Case: Data scientist

Background

A major challenge we have when doing home deliveries is to keep our service open as much as possible, even in high winds. For a while our limitation has been 10 m/s, but we have made some improvements that will allow us to increase it. In order to test such solutions, it's not enough to just look at the forecast: even though the forecast shoes 7 m/s at a particular 2.5km grid point in Yr's simulation, the real wind at the drone can be anything from 3 m/s to 14 m/s. For this reason, we need some way of measuring what the wind *actually* was for a given flight, based on sensor data on the drone.

Definitions

1. Wind and ground speed are 2D vectors, while airspeed is a scalar.
2. The wind magnitude at any given time is defined as the difference between ground speed magnitude and airspeed when flying straight into the wind. Wind and ground speed are 2D vectors, while airspeed is a scalar.
3. The **mean wind** is defined as the static wind field that best explains the airspeed and ground speed throughout the full flight.
4. The **maximum 3-second gust** is defined as the maximum wind throughout the flight, averaged over 3-second time windows.

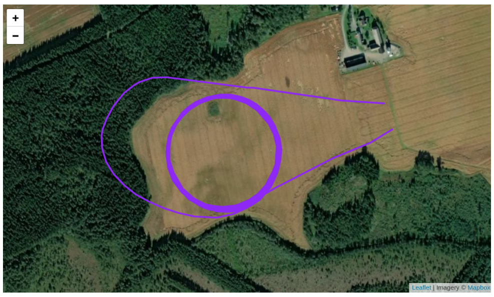
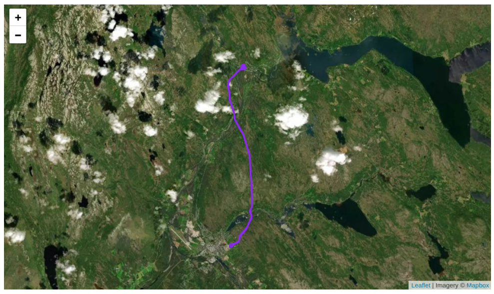
Tasks

1. Formulate the mathematical equation that relates ground speed to wind and airspeed.
2. Write a python script that takes a CSV input as described below, and outputs the **mean wind** and **maximum 3-second gust** for that flight.
3. Run the script on the two flights provided below. Visualize the results in a format that is understandable and intuitive for a non-technical pilot:
    a. Relevant raw data
    b. Results
    c. Uncertainty in the results
4. Make a presentation of maximum 10 minutes. We will ask follow-up questions to probe your understanding.

Flights

You are provided logs from two flights: One circular flight which makes data extraction simple, and one realistic flight which has more nuances. We recommend starting with the circular flight, and then moving on to the realistic flight.

To better undestand what is happening in the flights, see this video: https://www.youtube.com/watch?v=iH8gsDCDJU4

	Circular flight	**Realistic flight**
Flight path		
Forecasted wind	Average 7 m/s from 270 degrees, gusting up to 10 m/s	Average 8 m/s from 225 degrees, gusting up to 10 m/s
Mission description	1. Takes off in multirotor and transitions to fixed-wing. 2. Flies in a circle for ~20 minutes. 3. Transitions back to multirotor and lands straight down.	1. Takes off in multirotor and transitions to fixed-wing. 2. Flies in fixed-wing for ~10 minutes to the destination. 3. Transitions back to multirotor and winches down the package for approximately 3 minutes. 4. Transitions to fixed-wing. 5. Flies in fixed-wing for ~10 minutes back to the origin. 6. Transitions back to multirotor and lands straight down.

Datasets	`case_circular_flight_vtol_vehicle_status_0.csv` `case_circular_flight_vehicle_gps_position_0.csv` `case_circular_flight_distance_sensor_0.csv` `case_circular_flight_actuator_armed_0.csv`	`case_realistic_flight_vtol_vehicle_status_0.csv` `case_realistic_flight_vehicle_gps_position_0.csv` `case_realistic_flight_distance_sensor_0.csv` `case_realistic_flight_actuator_armed_0.csv`

Datasets

Each flight has four datasets attached:

- `vtol_vehicle_status_0` contains information needed to filter the correct intervals
  - `.vtol_in_rw_mode` indicates that the drone is in multirotor mode (rw = rotary wing).
  - `.vtol_in_trans_mode` indicates that the drone is in doing a transition.
  - `.timestamp` is the number of microseconds since bootup
- `actuator_armed_0` contains information needed to filter the correct intervals
  - `.armed` indicates if the motors are spinning or not.
  - `.timestamp` is the number of microseconds since bootup
- `vehicle_gps_position_0` contains the information you need for the model
  - `.vel_n_m_s` is the North velocity in m/s
  - `.vel_e_m_s` is the East velocity in m/s
  - `.timestamp` is the number of microseconds since bootup
- `distance_sensor_0` is included to give you some context on how the drone is flying
  - `.current_distance` is the distance from the drone to the ground

> Though you don't need it, you can use other fields in the datasets if you want. You can see their definitions at https://github.com/aviant-tech/PX4-Autopilot/tree/aviant/1.13/msg

Important info about the datasets

1. When `vtol_vehicle_status_0.vtol_in_rw_mode == 1`, the drone is in multirotor mode, i.e. flying as a "normal drone". In that mode, the flight controller tries to track a series of ground speed setpoints. You do not have access to the ground speed setpoints, but during delivery the drone is trying to stay still in the air.
2. When `vtol_vehicle_status_0.vtol_in_trans_mode == 1`, the drone is doing a transition to/from fixed-wing/multirotor, and the logged data is not very clean.
3. When `actuator_armed_0.armed == 0`, the drone is on the ground without spinning motors.
4. Otherwise (when `vtol_in_rw_mode==0`, `vtol_in_trans_mode==0` and `armed==1`) the drone is flying and in fixed-wing mode. Then you can (and should) assume that the drone keeps the airspeed at 23 m/s at all times. Hence you don't need the airspeed measurements for this task, and they are not provided for the circular flight. They are available in the "realistic flight" under `vtol_vehicle_status_0.airspeed_filtered` if you are interested anyway.

Evaluation

You are evaluated as follows:

1. How well did you solve the tasks?
2. How well did you answer the follow-up questions?

⚠ This is the first time we have made this exact case, so there might be information missing or unclear. If you have questions, feel free to ask. This will not impact your evaluation.

Good luck, we are excited to see your solution!