

MASTER THESIS RO 2021
MICROENGINEERING SECTION

WIND ESTIMATION WITH MULTIROTOR UAVS

Meier Kilian

Supervisor EPFL : MER Skaloud Jan

External Expert (UNIS/NTNU) : Dr. Hann Richard

Assistant : Joseph Paul Kenneth

Geodetic Engineering Laboratory EPFL

Spring Semester 2021



ON STARTING

your ideas are fish
you are trying to catch
with your bare hands

only with a quiet mind
is the surface glassy enough
for you to plunge your arms below
 hold on to
 the squirming gift
 wide-eyed & flat
stunned at its own reflection
as it inhales out of the water

Date & Time, Phil Kaye

Acknowledgments

First, I would like to thank Jan Skaloud for accepting to supervise this thesis which was proposed by the University of Svalbard and to allow me to work from Lausanne due to the impossibility to go to Norway. Then, I would like to thank Richard Hann for supervising this thesis and for creating as many interesting opportunities for me as he could despite being on the other side of Europe. I would also like to thank Kenneth Joseph Paul for having proofread my thesis and for having explored with me the secrets of quadcopter drag. Many thanks to Aman Sharma for proofreading my thesis and improving my academic writing style. Thanks also to Jean-Baptiste Magnin and Aurélien Brun, fellow Master students with whom I share office, struggles and laughter. I would like to thank also all the other people working at TOPO for their support.

I would also like to thank the Solar Energy and Building Physics Laboratory (LESO), and in particular Laurent Deschamps, for providing me with very valuable ground-truth data from their weather mast. Thanks also to the Swiss Polar Institute which would have funded part of my journey to Norway, if it could have taken place.

Finally, a big thanks to my Roommates, Friends and Family for their support and encouragements.

Abstract

Unmanned Aerial Vehicles (UAV) benefited from a tremendous increase in popularity in the past decade, which generated a lot of novel and unique use cases. One of them being the use of UAVs in meteorological research, in particular for wind measurements. Research in this field using quadcopters showed promising results. However most of the published results rely on the assumption the drone is stationary and do not attempt to estimate vertical wind. This severely limits the use potential of quadcopters in meteorology. Hence this work will address both shortcomings by proposing a new dynamic model based approach, relying on the assumption the thrust force can be measured or estimated and the drag force can be related to air speed. This new method will be tested on empirical data gathered on a Phantom 4 RTK.

Keywords: UAV, Atmospheric Boundary Layer (ABL) Meteorology, Wind Estimation, Shear Wind Profile, UAV Motion Model, Drag Model, Blade Element Momentum (BEM) Theory

Contents

Acknowledgments	2
Abstract	3
1 Introduction	8
1.1 Motivation	8
1.2 State-of-the-art	9
1.3 Problem statement and approach	10
1.4 Thesis outline	11
2 Preliminaries	12
2.1 Notations	12
2.1.1 Scalars	12
2.1.2 Vectors	12
2.1.3 Quaternion	12
2.1.4 Matrices	12
2.1.5 Functions	12
2.1.6 Time derivatives	13
2.1.7 Estimated quantities	13
2.1.8 Measured quantities	13
2.2 Attitude representations	13
2.2.1 Rotation matrix	13
2.2.2 Quaternion	13
2.2.3 Euler angles	14
2.2.4 Rotating reference frames	15
2.3 Frames definition	15
2.3.1 Inertial frame (i-frame)	15
2.3.2 Earth frame (e-frame)	15
2.3.3 Local-level frame (l-frame)	16
2.3.4 Body frame (b-frame)	16
2.3.5 Tilt frame (t-frame)	16
2.3.6 Transformation from Body to Local-level frame	17
2.3.7 Transformation from Body to Tilt frame	17
2.4 Sensor model	19
2.4.1 Weather station sensors	19
2.4.2 Mobile sensor	19
2.4.3 Error modeling	20
2.5 Motion model	20
2.5.1 Equation of motion	21
2.5.2 Specific forces	21
2.6 Air density estimation	22
2.6.1 Compression factor	22
2.6.2 Molar fraction of water vapour	22
2.7 Wind triangle	23
3 Methodology	24
3.1 Data Collection	24
3.1.1 Flight data	24
3.1.2 Reference data	24
3.1.3 Flight campaign	25
3.2 Wind estimation from tilt	26
3.2.1 Introduction	26
3.2.2 Model calibration	28

3.2.3	Computing air velocity	29
3.2.4	Computing air direction	30
3.2.5	Computing wind vector	30
3.3	Wind estimation from dynamical model	30
3.3.1	Introduction	30
3.3.2	Computing thrust	31
3.3.3	Computing drag	31
3.3.4	Computing drag (without thrust)	32
3.3.5	Computing air speed (quadratic model)	32
3.3.6	Computing air speed (linear model)	34
3.3.7	Computing drag from force data	35
3.4	Software overview	35
3.4.1	Introduction	35
3.4.2	Data flow	35
3.4.3	Data files	37
3.4.4	External Parameters	40
3.4.5	Architecture	40
4	Results	42
4.1	Data Sets	42
4.1.1	Quality of ground truth	42
4.1.2	Lack of environment variation	42
4.1.3	Dataset size	42
4.2	Computing performance	43
4.2.1	Error, bias and standard deviation	43
4.2.2	Assessment of ground truth quality	44
4.2.3	Filtered Performance	44
4.3	Sample flights	44
4.3.1	Hover	45
4.3.2	Square	46
4.3.3	Vertical	47
4.4	Performance over wind speed	48
5	Discussion	52
5.1	Learning outcomes	52
5.1.1	Impact of vertical wind estimation	52
5.1.2	Linear or quadratic drag model	52
5.1.3	Ground truth quality	52
5.2	Method trade-off	53
5.3	Possible applicability to meteorological research	53
5.3.1	Estimating the aerodynamic roughness length	54
5.3.2	Open questions	55
6	Conclusion and future work	56
6.1	Future work	56
6.2	Final thoughts	57
A	DJI Phantom flight data extraction using DatCon	61
A.1	Required Hardware and Software	61
A.2	DatCon	61
A.3	Step-by-step procedure	62
B	Flight Campaign procedure	63

C Impact of drone flights on wild life	65
C.1 Context	65
C.2 Possible explanation of Seagull behaviour	65
C.3 Risk mitigation strategy	65
C.4 Conclusion	65
D Assumption List	67
E Dataset Overview Table	68
F Project description	69

List of acronyms

ABL Atmospheric Boundary Layer. 8, 9, 11

AC Air Craft. 61

BEM Blade Element Momentum. 52, 56

DCM Direct Cosine Matrix. 13

DM Dynamic Model. 30, 31, 53

ECEF Earth Centered Earth Fixed. 15

ECI Earth Centered Inertial. 15

EPFL Ecole Polytechnique Fédérale de Lausanne. 25, 43, 63

FRD Front-Right-Down. 16

GNSS Global Navigation Satellite System. 9, 10, 24

GPS Global Positioning System. 15, 61

IMU Inertial Measurement Unit. 15

LESO Solar Energy and Building Physics Laboratory. 64

LiDAR Light Detection And Ranging. 8

MoTUS Urban microclimate measurement mast. 64

NED North-East-Down. 16

RC Remote Control. 25, 63

RMS Root Mean Square. 10

RTK Real Time Kinematic. 24, 61

SODAR Sonic Detection And Ranging. 8

TOPO Geodetic Engineering Laboratory. 25, 56

TOPOAWS TOPO Automatic Weather Station. 63, 64

UAV Unmanned Aerial Vehicles. 3, 8–10, 42, 53, 54, 56, 66

UNIS The University Center in Svalbard. 24

XML Extensible Markup Language. 40

1 Introduction

1.1 Motivation

Boundary-layer meteorology In [1], *boundary-layer meteorology* is defined as "*a subdiscipline of meteorology [which] is concerned with the state of and processes in the air layer in immediate contact with the Earth's surface*". An essential parameter of the physical state of the Atmospheric Boundary Layer (ABL) is wind, i.e. air flow with respect to the ground. Thus, being able to measure wind in the ABL is important to understand its physics, but also for a wide variety of applications ranging from weather forecasting and climate modeling, to pollutant tracking [2] and understanding glacier ice melting [3]. Another proof of the importance of wind measurements is the wide number of available sensors and methods. Most of the ABL sensors are ground based, ranging from a simple cup-anemometer to more complex Light Detection And Ranging (LiDAR) or Sonic Detection And Ranging (SODAR) technologies. A comprehensive list of ground based ABL measurement methods can be found in [4]. To this, satellite based observations can be added as well as in-situ measurements using air-borne platforms such as weather balloons, tethered balloons, and UAV.



(a) DJI Phantom RTK quadcopter



(b) Sensfly eBee X fixed-wing.

Figure 2: Example of a commercial quadcopter and fixed-wing drone. Images are reproduced from their respective website.

UAV based wind measurements UAV is generic acronym which describes unmanned air-crafts ranging in weight from a couple grams to several thousand kilograms [5] and come in various shapes and architectures. Two of the most popular small UAVs (less than 25 [kg]) are quadcopters and delta fixed-wings. A commercial example of each group can be seen in Figure 2 (the Phantom 4 RTK will be used in this work). In this work "UAV", "drone" and "air-craft" will be used interchangeably. Using UAVs for wind measurements, allows making measurements in situation where "traditional" sensor cannot be used or are difficult or costly to deploy. For example, probing wind speeds at 100 meters above ground involves the installation of a tall mast which is costly and complex to deploy and maintain. Probing wind speeds in remote or dangerous areas, like above crevassed glaciers [6], is another example where traditional sensor fail. Indeed, this requires human probing on the glacier and involves high risks. Hence, wind measurements using UAVs have the following benefits:

- low cost of acquisition, deployment and maintenance
- easy access to various heights in the ABL¹
- easy access to remote and dangerous areas

¹The thickness of the ABL is typically around 1 [km] [1], quad-rotor drones are usually not able to reach this altitude and would thus be limited to the lower part of the ABL but fixed-wing drone may climb to higher altitudes.

- rapid deployment
- high spacial resolution along the flight path (continuous profiling)

However, these benefits come at a cost. Indeed, UAVs suffer from limitations which thus also limit their capability of measuring wind. Taking the Phantom 4 RTK as an example (all data is found in the drone's user manual [7]), UAVs tend to:

- have low endurance: at most 30 minutes for the phantom²
- only be able to fly in low wind conditions: less than 10 [m/s] for the Phantom
- be susceptible to other environmental conditions, such as snow, rain and fog or temperature: the drone should not be flown during snow, rain or fog conditions and operating temperature is between 0 [°C] and 40 [°C] for the Phantom
- be impacted by their flight environment : magnetic fields produced by power lines or conducted by large metallic objects may interfere with the drone's magnetic compass for example
- rely on external infrastructure: for example Global Navigation Satellite System (GNSS) is required for autonomous navigation

Most of these disadvantages do not apply to other wind measuring methods. Nonetheless, for all aforementioned reasons, enabling UAVs to measure wind, is an important tool for better understand ABL physics. This is the general objective of this work.

1.2 State-of-the-art

UAV based wind estimation can be classified in two categories: *on-board flow sensor based* and *inertial and power based* [5].

On-board flow sensor based A flow sensor is mounted on the UAV directly measures the quantity of interest³. This allows the usage of proven flow sensors and thus may lead to better accuracy and precision. But it comes at the cost of having to integrate an additional sensor on the UAV. This approach is commonly taken by fixed-wing drone due to the availability of pitot tubes for flight control. However, on rotor-crafts, integrating an air flow sensor is very challenging since the propellers heavily influence the airflow around the drone as can be see in Figure 3.

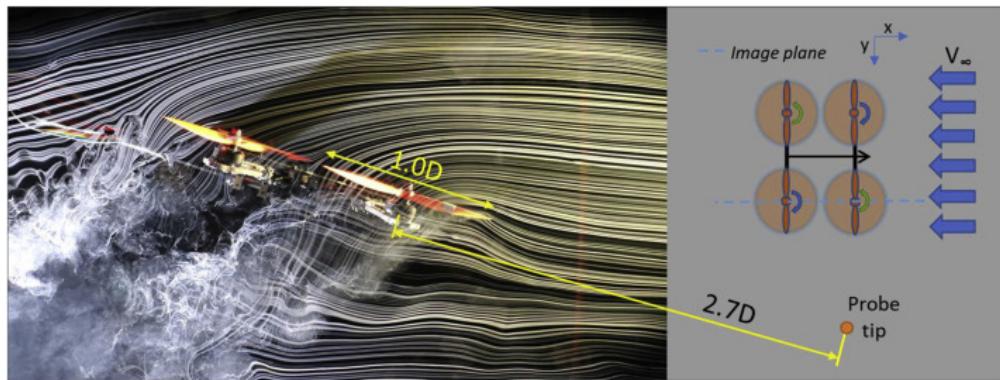


Figure 3: Air flow around a quadrotor. From [5].

²Fixed wing drone usually have larger endurance.

³The air flow with respect to the UAV is measured and the quantity of interest is the air speed with respect to the ground (aka wind).

Inertial and power based Only navigation data is used to infer wind speeds. In this approach, the UAV is considered as a dynamic system with an input: the autopilot commands; an output: the drone's position and attitude; and an external perturbation: the wind. Hence, provided the system's model in addition to the autopilot commands, the drone's position and attitude, the wind can be estimated. This approach has the downside that it highly depends on the quality of the acquired input/output data and the knowledge of the system. But on the other hand it does not rely on any additional hardware, thus greatly reducing hardware complexity.

This work will describe an inertial based approach and will use a quadrotor, hence the State-of-the-art description will also focus on this combination. The main publications on this topic from 2018 onward are listed in Table 1. The table classifies them regarding the used method, type of data the method was tested on, needed flight type and whether vertical wind speed is estimated (or only wind in the azimuthal plane).

Paper	Year	Method	Data Type	Flight type	Vertical Wind
[8]	2015	Tilt	Wind tunnel & Field	Hover & Moving	No
[9]	2017	Tilt	No wind & Field	Hover	No
[10]	2018	Tilt	Wind tunnel	Hover	No
[11]	2019	Kalman Filtering	Simulation	Hover	No
[12]	2019	Machine Learning	Wind tunnel	Hover	No
[13]	2019	System Identification	Simulation	Moving	Yes
[14]	2019	Kalman Filtering	Simulation	Moving	No
[15]	2020	Tilt	Simulation	Hover	No
[16]	2020	System Identification	Field	Hover & Vertical Profiling	No
[17]	2020	Machine Learning	Simulation	Hover	No
[18]	2020	System Identification	Simulation	Hover	No

Table 1: Main papers published from 2018 until 2021 using multirotor UAVs to estimate wind. This list is not exhaustive, plus wind estimation for navigation are not included. [8] and [9] are also included since their contribution is often cited.

The tilt method correlates the drone attitude to wind speed during hovering. All other methods are model-based and thus define a more or less generic [16] or complex [13] system and estimate the model parameters using various methods such as machine learning, system identification or filtering. Data used for parameter tuning may result from simulations, wind tunnel tests or field tests. Most of the listed methods assume a hovering drone. Finally, almost all listed papers do not estimate vertical wind or may even assume it to be zero. This can be understood since, usually, vertical wind has a small effect on the drone compared to horizontal wind, plus it removes an unknown in model-based approaches.

1.3 Problem statement and approach

This work follows up on the work of Arthur Garreau [19]. Using the same classification as in Table 1, [19] classifies as shown in Table 2. The used tilt approach is described in Section 3.2 and will serve as a baseline to this work. It is worth mentioning that the dataset used by Arthur Garreau comes from field flights performed in Svalbard (Norway). This is a valuable dataset since flights in high latitudes come with their own challenges: such as low temperatures or reduced GNSS coverage [20]. It is also worth mentioning that [19] attempted to apply this wind measuring technique to perform vertical wind profiling in order to calculate the aerodynamic roughness length⁴ of the overflowed terrain. The tilt estimation showed promising results with a Root Mean Square (RMS) error below 0.8 [m/s] for wind speed and thus it was decided to continue and improve on this work.

This thesis aims at addressing shortcomings of [19]. There are two major aspect that were identified as needing to be improved. First, the *need to hover*. The need to "stay still" to take measurements severely limits the potential use of a drone as a wind sensor. On one hand it considerably slows down the process of probing several points of interests and it prevents the possibility to do continuous profiling. On the other hand it makes the estimation depend on the autopilot and its ability to maintain position during hover. Second,

⁴For more information about aerodynamic roughness length see [1] or [3].

Paper	Year	Method	Data Type	Flight type	Vertical Wind
[19]	2020	Tilt	Field	Hover	No
This work	2021	Dynamic Model	Field (& Wind tunnel)	Moving	Yes

Table 2: Classification of Arthur Garreau's and this work using Table 1.

the assumption that *there is no wind in the vertical direction*. Even if this assumption may be acceptable in some situations, it is not true in general and several ABL effects involve vertical wind motions such as shear wind generated eddies or convective effects generated by surface heating [1]. Additionally, this thesis shall use empirical data collected during real condition flights. This avoids oversimplifications resulting from imperfect aerodynamic simulations or idealized conditions present in wind tunnels. Summing up, the new wind estimation proposed by this work shall:

1. **be able to perform during dynamic drone maneuvers**
2. **be able to estimate vertical wind speed**
3. **use real field data**

To answer the first and second objectives, a dynamic model based approach will be used. The forces acting on the air-craft are gravitational attraction, thrust (generated by the propeller) and drag (generated by the air flow around the drone). Thus, provided acceleration and all other forces are known, using Newton' second law of motion, drag can be estimated. Then drag can be related to air speed and wind speed using an appropriate drag model. To answer the third objective, a field campaign will be planned to acquire data in various wind conditions and during various flight trajectories. Thus this work can be classified as shown in Table 2.

There are two main challenges in this approach. First, thrust estimation, since most platform do not measure the thrust generated by their propellers. However in this work the rotor speeds are available and will be used to estimate thrust. Second, thrust and drag models feature several parameters that need to be tuned, computed or measured. In this work this will be achieved by relying on wind tunnel data provided by [21].

1.4 Thesis outline

The outline of this thesis is the following: Section 2 will define the notation used throughout this work, it will also define the used reference frames and derive important relations between them and it will describe the motion and sensor model used to describe the air-craft. Section 3 will present the baseline and new estimation method as well as how they were implemented in software. Section 4 will present the results and Section 5 will discuss them. Finally, Section 6 will conclude this work and discuss its possible future evolution.

2 Preliminaries

The correct use of data generated by mobile platform heavily relies on proper definition of attitude representations and reference frames and be supported by a clear and consistent notation scheme. This section aims at providing all of the above. Most of the wording and equations presented here are borrowed from [22] and [23].

2.1 Notations

The section will detail the notation conventions used throughout this work.

2.1.1 Scalars

Scalars are denoted using a single letter which may feature a subscript representing the axis along which the quantity is expressed. For example the following quantities are scalars:

$$g, P_x, \rho$$

2.1.2 Vectors

Vectors are noted using a single lowercase letter written in bold font. The letter may feature a superscript indicating the frame the vector is expressed in. For example a vector quantity \mathbf{r} expressed in the frame i featuring axes ix, iy, iz (see Section 2.3) can be written as:

$$\mathbf{r}^i = \begin{bmatrix} r_{ix} \\ r_{iy} \\ r_{iz} \end{bmatrix}$$

2.1.3 Quaternion

Quaternion are noted using a lowercase 'q' character written in bold font. It should feature a superscript and subscript as defined for rotation matrices, see Section 2.2. A quaternion $q = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k}$ describing a rotation from frame i to j is written as:

$$\mathbf{q}_i^j = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

Note that here \mathbf{i} and \mathbf{j} designates the basic quaternions, whereas i and j stand for two generic frames.

2.1.4 Matrices

Matrices are noted using a single uppercase letter written in bold font. The letter may feature a superscript and a subscript in the case of a rotation matrix (see Section 2.2). For example a rotation matrix describing a rotation from frame i to j is written as:

$$\mathbf{C}_i^j = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

2.1.5 Functions

Functions returning any of the above quantity can be formed by adding a set of parenthesis containing the input parameters of the function. For example, a function \mathbf{f} returning a vector in frame i and being dependent on a constant g and a matrix C will be written as:

$$\mathbf{f}^i(g, \mathbf{C})$$

2.1.6 Time derivatives

Time derivatives are denoted by adding a dot above the quantity of interest:

$$\dot{g}, \dot{\mathbf{r}}^i, \dot{\mathbf{q}}_i^j, \dot{\mathbf{C}}_i^j$$

2.1.7 Estimated quantities

Estimated quantities, if necessary, will be discerned from true quantities using a circumflex (hat):

$$\hat{g}, \hat{\mathbf{r}}^i, \hat{\mathbf{q}}_i^j, \hat{\mathbf{C}}_i^j$$

2.1.8 Measured quantities

Measured quantities, will be discerned from true quantities using a tilde:

$$\tilde{g}, \tilde{\mathbf{r}}^i, \tilde{\mathbf{q}}_i^j, \tilde{\mathbf{C}}_i^j$$

2.2 Attitude representations

Attitude describes the orientation of an object or a reference frame with respect to another. There are three common ways to describe this relative orientation, namely rotation matrices, quaternions and Euler angles. Each description has its benefits and drawbacks, which will be briefly summarised in this section.

2.2.1 Rotation matrix

A rotation matrix, \mathbf{C}_j^i , (also called Direct Cosine Matrix (DCM)) transforms a given vector quantity expressed in the j-frame \mathbf{r}^j to its corresponding quantity in the i-frame \mathbf{r}^i :

$$\mathbf{r}^i = \mathbf{C}_j^i \mathbf{r}^j \quad (1)$$

The columns of \mathbf{C}_j^i represent the unit vectors of the j-frame expressed in the i-frame axes. It can be proven that \mathbf{C}_j^i is orthonormal and thus $(\mathbf{C}_j^i)^T = (\mathbf{C}_j^i)^{-1} = \mathbf{C}_i^j$, which is useful to perform the inverse conversion:

$$\mathbf{r}^j = \mathbf{C}_i^j \mathbf{r}^i \quad (2)$$

Rotation matrices do not suffer from gimbal lock and are unambiguous, but are somewhat complex to build and have a lot of redundant parameters. In this work, they will be used to derive the equations of motions of the air craft.

2.2.2 Quaternion

Quaternions describe a rotation using four parameters and leverages Euler's rotation theorem stating that a transformation from one frame to another can be expressed as a single rotation of magnitude $\|\mathbf{u}\|$ about a vector \mathbf{u} , where $\|\cdot\|$ is the euclidean (L2) norm. The relation between \mathbf{q} and \mathbf{u} is the following:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\|\mathbf{u}\|}{2}\right) \\ \frac{u_x}{\|\mathbf{u}\|} \sin\left(\frac{\|\mathbf{u}\|}{2}\right) \\ \frac{u_y}{\|\mathbf{u}\|} \sin\left(\frac{\|\mathbf{u}\|}{2}\right) \\ \frac{u_z}{\|\mathbf{u}\|} \sin\left(\frac{\|\mathbf{u}\|}{2}\right) \end{bmatrix} \quad (3)$$

Note only unit quaternions, called *vectors*, are used to describe rotations. The equivalence between a quantity in the i-frame \mathbf{r}^i and j-frame \mathbf{r}^j is given by:

$$\mathbf{r}^i = rot(\mathbf{q}_j^i, \mathbf{r}^j) = \mathbf{q}_j^i \otimes \mathbf{r}^j \otimes \bar{\mathbf{q}}_j^i = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ r_{jx} \\ r_{jy} \\ r_{jz} \end{bmatrix} \otimes \begin{bmatrix} q_1 \\ -q_2 \\ -q_3 \\ -q_4 \end{bmatrix} \quad (4)$$

Where $\bar{\mathbf{q}}_j^i$ denotes the conjugate of quaternion \mathbf{q}_j^i and \otimes is the Hamilton product defined as follows:

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_1 p_1 - q_2 p_2 - q_3 p_3 - q_4 p_4 \\ q_1 p_2 + q_2 p_1 + q_3 p_4 - q_4 p_3 \\ q_1 p_3 + q_3 p_1 - q_2 p_4 + q_4 p_2 \\ q_1 p_4 + q_4 p_1 + q_2 p_3 - q_3 p_2 \end{bmatrix} \quad (5)$$

The opposite rotation from the i-frame to the j-frame is given by :

$$\mathbf{r}^j = \text{rot}(\mathbf{q}_i^j, \mathbf{r}^i) = \text{rot}(\bar{\mathbf{q}}_j^i, \mathbf{r}^i) \quad (6)$$

Note that Matlab implements a rotation by a quaternion with the `rotatepoint(...)` function. Finally, quaternions are no subject to gimbal lock and are unambiguous⁵, they are a more compact representation of a rotation than the rotation matrix but are less easy to understand on an intuitive level. In this work quaternions will be used to store and compute relative orientations.

2.2.3 Euler angles

Euler angles describe a sequence of three elementary rotations around their associated axis of the rotated frame (intrinsic rotations). This process is ambiguous since the chosen axis of rotation need to be specified as well as the order in which the rotation should be performed. In navigation, the most used rotation sequence is *ZYX*, which is named yaw-pitch-roll in the case of a rotation from the local-level to the body frame. The sequence of rotation can be described as follows in a matrix form:

$$\mathbf{C}_j^i = \mathbf{C}_X(r) \mathbf{C}_Y(p) \mathbf{C}_Z(y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(r) & \sin(r) \\ 0 & -\sin(r) & \cos(r) \end{bmatrix} \begin{bmatrix} \cos(p) & 0 & -\sin(p) \\ 0 & 1 & 0 \\ \sin(p) & 0 & \cos(p) \end{bmatrix} \begin{bmatrix} \cos(y) & \sin(y) & 0 \\ -\sin(y) & \cos(y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Where r , p and y are the roll, pitch and yaw angles respectively.⁶

Conversion from Euler angles to quaternion can be done as follows:

$$\mathbf{q} = \begin{bmatrix} \cos(y/2) \\ 0 \\ 0 \\ \sin(y/2) \end{bmatrix} \otimes \begin{bmatrix} \cos(p/2) \\ 0 \\ \sin(p/2) \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos(r/2) \\ \sin(r/2) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(r/2) \cos(p/2) \cos(y/2) + \sin(r/2) \sin(p/2) \sin(y/2) \\ \sin(r/2) \cos(p/2) \cos(y/2) - \cos(r/2) \sin(p/2) \sin(y/2) \\ \cos(r/2) \sin(p/2) \cos(y/2) + \sin(r/2) \cos(p/2) \sin(y/2) \\ \cos(r/2) \cos(p/2) \sin(y/2) - \sin(r/2) \sin(p/2) \cos(y/2) \end{bmatrix} \quad (8)$$

Note that Matlab implements this conversion with the `quaternion([y,p,r], 'euler', 'ZYX', 'frame')` function.

Conversion from quaternion to Euler angles can be done as follows:

$$\begin{bmatrix} r \\ p \\ y \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_1 q_2 + q_3 q_4), 1 - 2(q_2^2 + q_3^2)) \\ \text{asin}(2(q_1 q_3 - q_4 q_2)) \\ \text{atan2}(2(q_1 q_4 + q_2 q_3), 1 - 2(q_3^2 + q_4^2)) \end{bmatrix} \quad (9)$$

Note that Matlab implements this conversion with the `euler(q, 'ZYX', 'frame')` function.

As already mentioned Euler angles are ambiguous due to the various ways they can be defined. In addition to this, they suffer from the gimbal lock effect, which make them hard to handle mathematically. However, they are a compact and intuitive representation of the relative orientation of two objects, with which most people are familiar with. Hence, Euler angles will be used when the need for a visual representation of relative orientation arises.

⁵Provided versors are used.

⁶Note that in the literature these angle may also be referred to as bank, pitch and heading and may also be designated by the Greek letters ϕ , θ and ψ . Moreover, those angles may also go under the name of Tait-Bryan angles.

2.2.4 Rotating reference frames

Reference frames may rotate arbitrarily with respect to each other. In order to be able to compute motions in reference frames experiencing such relative change of orientation, it is necessary to be able to calculate the time derivative of a given relative orientation. Given a vector:

$$\omega_{ij}^j = \begin{bmatrix} \omega_{jx} \\ \omega_{jy} \\ \omega_{jz} \end{bmatrix} \quad (10)$$

Which describes the rotation rate of the j-frame with respect to the i-frame expressed in the j-frame, the time derivative of a rotation matrix can be written as:

$$\dot{C}_j^i = C_j^i \Omega_{ij}^j = C_j^i \begin{bmatrix} 0 & -\omega_{jz} & \omega_{jy} \\ \omega_{jz} & 0 & -\omega_{jx} \\ -\omega_{jy} & \omega_{jx} & 0 \end{bmatrix} \quad (11)$$

2.3 Frames definition

2.3.1 Inertial frame (i-frame)

An inertial frame is a non-accelerating and non-rotating reference frame, which is at rest or subjected to a uniform translational motion. In such a frame Newtonian mechanics are valid and thus will be the starting point of the derivation of the motion equations. There is no perfect inertial frame, but given the current sensitivity of Inertial Measurement Unit (IMU) a good approximation of an inertial frame is given by a frame described as follows:

- **Center:** Earth's center
- **x-axis:** Pointing toward the vernal equinox
- **y-axis:** Completing the orthogonal right-handed frame
- **z-axis:** Pointing toward the mean celestial north pole

This frame is known as the Earth Centered Inertial (ECI) frame.

2.3.2 Earth frame (e-frame)

The earth frame is fixed with respect to earth's rotation and thus represents a convenient way to describe a position on the globe. Due to earth's rotation this frame is not inertial. This frame is known as the Earth Centered Earth Fixed (ECEF) frame and is defined as:

- **Center:** Earth's center
- **x-axis:** Pointing toward the Greenwich meridian
- **y-axis:** Completing the orthogonal right-handed frame
- **z-axis:** Pointing towards the mean direction of rotation

One important realization of the earth frame is the WGS-84 being the reference frame used by Global Positioning System (GPS). Note that, in this frame, it is common to use an ellipsoidal coordinate system (instead of a Cartesian one). Hence, a position is described with two angles and a scalar: latitude, longitude and height:

$$\mathbf{r}^e = \begin{bmatrix} \phi \\ \lambda \\ h \end{bmatrix}$$

2.3.3 Local-level frame (l-frame)

The local-level frame is a local geodetic frame with an arbitrary origin. It is useful to describe motion in a space around a point of interest. It is defined as follows:

- **Center:** Arbitrarily chosen
- **x-axis/n-axis:** Pointing toward the geographical north
- **y-axis/e-axis:** Pointing toward the geographical east
- **z-axis/d-axis:** Pointing downwards (along the ellipsoid normal)

A local frame using this definition is known as a North-East-Down (NED) frame. Thus x-, y- and z-axis may be called n-, e- and d-axis respectively.

2.3.4 Body frame (b-frame)

The body frame is fixed to the body of interest (in this work, an air craft) and it is defined as follows:

- **Center:** Center of the body
- **x-axis:** Pointing forward
- **y-axis:** Pointing rightward
- **z-axis:** Pointing downward (along the height direction of the body)

A body frame using this definition is known as a Front-Right-Down (FRD) frame. All the above described frames are illustrated in Figure 4. More specifically the body frame related to the Phantom 4 RTK drone is shown on Figure 5.

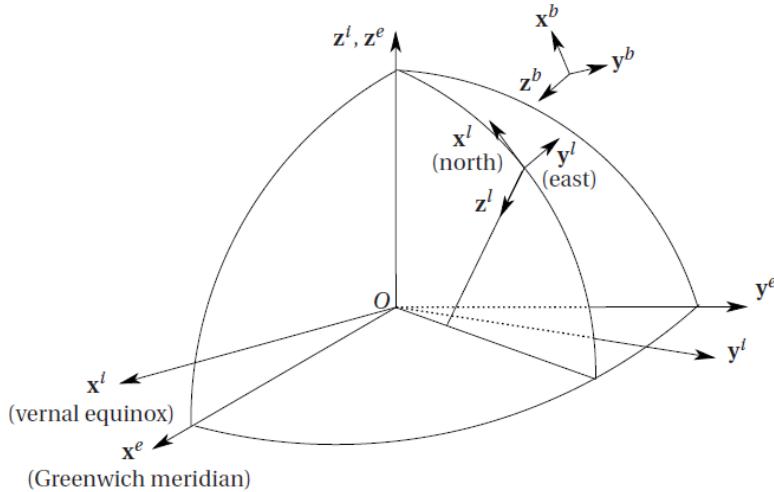


Figure 4: Inertial (i), Earth (e), Local-level (l) and Body (b) frames. O Represents earth's center and the sphere represents Earth' surface. This illustration was copied from [22].

2.3.5 Tilt frame (t-frame)

The tilt frame is not a frame commonly used in navigation, but it will prove useful in the context of this work. It is defined as follows:

- **Center:** Center of the body



Figure 5: Phantom 4 RTK body frame and propeller identification.

- **x-axis/tx-axis:** Pointing toward the tilt direction
- **y-axis/tx-axis:** Completing the orthogonal right-handed frame
- **z-axis/tz-axis:** Pointing in the same direction as the b-frame z-axis

Where the tilt direction is defined as the azimuthal direction toward which the minus-z-body-axis is pointing. In other words, assuming the drone is stationary, if a ball is placed at the origin of the body frame, and if this ball is capable of rolling on the xy-body-plane, then this ball will roll along the x-tilt-axis. For example, for an air-craft which is pitching forward (negative pitch angle), the tilt direction is equal to the yaw angle. Or for an air-craft rolling to its right (positive roll angle), the tilt direction is equal to the yaw angle plus 90 degrees. This definition means that this frame is not rigidly attached to the body. The x-, y- and z-axis may also be called tx-, ty- and tz-axis respectively. Figure 6 illustrates the tilt frame graphically. An animated version of this figure can be found in the software repository (introduced in Section 3.4) in the root folder as a file name [squareFlight_lowTimeRes.gif](#).

2.3.6 Transformation from Body to Local-level frame

The rotation from the l-frame to the b-frame is given by the quaternion \mathbf{q}_b^l or alternatively by the rotation matrix \mathbf{R}_b^l or the roll, pitch and yaw Euler angle $r \ p \ y$. The rotation quaternion will be measured by the drone's attitude sensor $\tilde{\mathbf{q}}_b^l$ (see Equation (21)). The corresponding rotation matrix and Euler angles can be derived using Equations (7) and (9). Finally, the opposite rotation, i.e. from Local-level to Body frame, is given by $\mathbf{q}_l^b = \bar{\mathbf{q}}_b^l$ (see Equation (6)).

2.3.7 Transformation from Body to Tilt frame

The rotation from the l-frame to the b-frame is given by the quaternion \mathbf{q}_l^t or alternatively by the rotation matrix \mathbf{R}_l^t . An expression for this quaternion will be given below and the rotation matrix can be computed from the quaternion using Equations (7) and (9). Based on the definition in Section 2.3.5, let's compute the minus-z-body vector in the local frame :

$$\mathbf{u}^l = \text{rot}(\mathbf{q}_b^l, \mathbf{u}^b) = \text{rot}(\mathbf{q}_b^l, \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}) \quad (12)$$

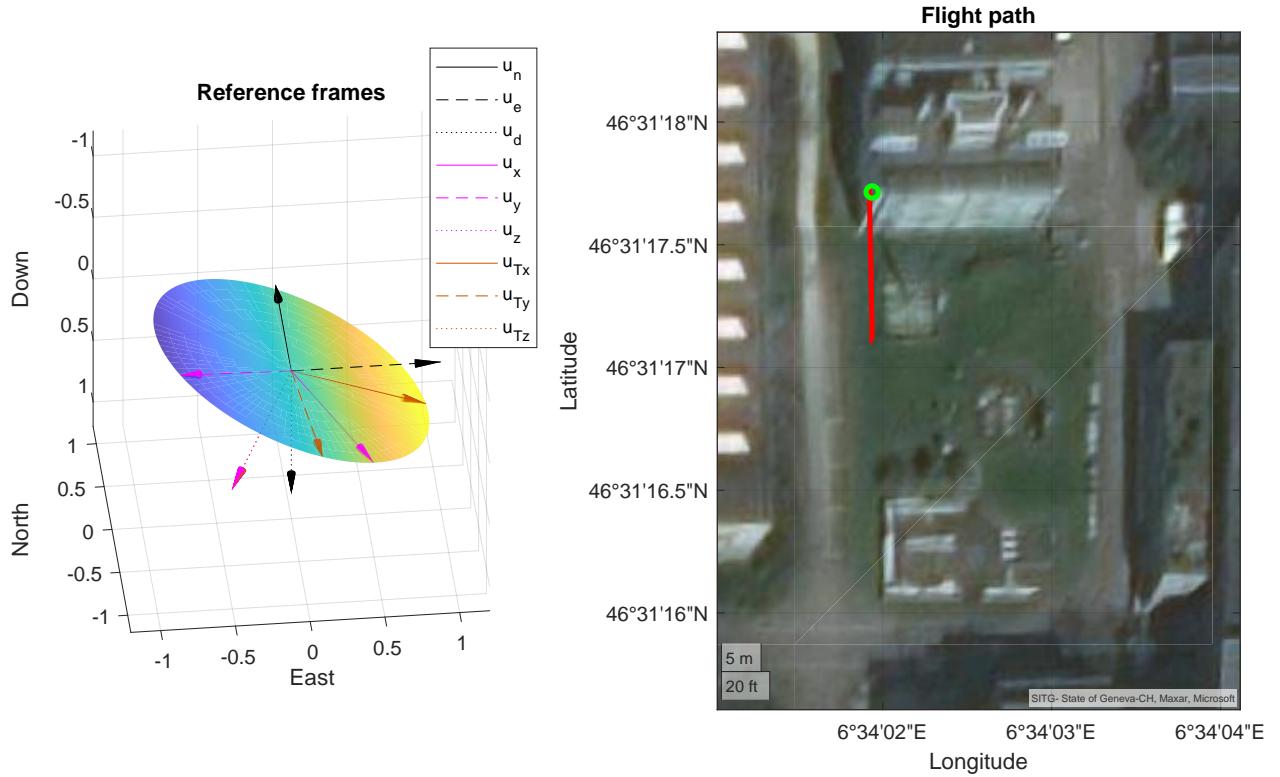


Figure 6: Reference frames during a typical flight. The plot on the right shows the already executed flight path (red line) and the current position of the drone (green circle). The left plot represents the drone (as a disk) in the local frame. Three reference frames are shown: l-frame (black) spanned by the u_n , u_e and u_d unit vectors, b-frame (magenta) u_x , u_y and u_z and t-frame (brown) u_{tx} , u_{ty} and u_{tz} . The heat map coloring the drone disk represents the proximity to ground: the hotter, the closer to the ground. Using the ball analogy again, a ball placed at the center of the disk will always roll toward the hottest point. An animated version is available [here](#).

Then, this vector, when projected on the azimuthal plane (ne-local-plane) points toward the tilt direction. Moreover, the angle between this vector and the local vertical (minus d-local-axis) equals the tilt angle. Hence, the tilt direction λ is given by:

$$\lambda = \text{atan2}(u_e, u_n) \quad (13)$$

Where $\text{atan2}()$ is the 4 quadrant arc-tangent function defined in the interval $[0, 2\pi]$. And the tilt angle α is given by:

$$\alpha = \arccos(-u_d) \quad (14)$$

The rotation from the body frame to the tilt frame is an elementary rotation around the z-body-axis (which is common to the tilt frame) of magnitude $\tilde{y} - \lambda$. Hence using (8) with $y = \tilde{y} - \lambda$ and $p = r = 0$ the rotation quaternion is given by:

$$\mathbf{q}_b^t = \begin{bmatrix} \cos((\tilde{y} - \lambda)/2) \\ 0 \\ 0 \\ \sin((\tilde{y} - \lambda)/2) \end{bmatrix} \quad (15)$$

Finally, the opposite rotation, i.e. from Tilt to Body frame, is given by $\mathbf{q}_t^b = \bar{\mathbf{q}}_b^t$ (see equation (6)).

2.4 Sensor model

In this work two type of sensors needs to be considered, the static weather station sensors and the sensors present on the mobile platform.

2.4.1 Weather station sensors

The weather stations used in this work are static with respect to the local-level frame and thus the measured vectorial quantities are measured in the l-frame. The only vectorial quantity measured in this work is wind speed. There are two ways of expressing the wind speed vector: either aligned with the air flux vector, i.e. the wind speed points in the direction where the air goes; or in the opposite direction, i.e. the wind speed points in the direction where the air is coming from. To solve this ambiguity the former will be called *physical wind speed* and the latter *meteorological wind speed*. The quantity of interest in this work in the physical wind speed, which will be written as:

$$\tilde{\mathbf{w}}^l = \begin{bmatrix} \tilde{w}_n \\ \tilde{w}_e \\ \tilde{w}_d \end{bmatrix} \quad (16)$$

However, most weather station measure the meteorological wind speed. Moreover, the meteorological wind speed is usually expressed using a cylindrical coordinate system: meteorological wind direction in the azimuthal plane (w_μ), wind speed magnitude in the azimuthal plane (w_r) and vertical wind speed (w_h). Using these coordinates physical wind speed can be written as:

$$\tilde{\mathbf{w}}^l = \begin{bmatrix} \tilde{w}_n \\ \tilde{w}_e \\ \tilde{w}_d \end{bmatrix} = \begin{bmatrix} -\tilde{w}_r \cos(\tilde{w}_\mu) \\ -\tilde{w}_r \sin(\tilde{w}_\mu) \\ -\tilde{w}_h \end{bmatrix} \quad (17)$$

The other quantities measured by the weather station are air temperature, air relative humidity and atmospheric pressure which are all scalar and thus do not need to be associated to a specific reference frame.

2.4.2 Mobile sensor

The mobile platform used in this work is a DJI Phantom 4 RTK. It is likely that this drone features a variety of sensors which are used to control its motion. However, since it is a proprietary platform the inner workings remain unknown and to access to autopilot data a community driven software which reverse engineered flight logs encoding was used (see Section 3.1.1 and Appendix A). Hence, one has no direct access to raw sensor data. Thus let's define a set of virtual sensors which will be assumed to be perfectly aligned and centered with the relevant reference frame (As.1). The following sensors are defined:

- **Position sensor:** Measuring the position in the earth frame:

$$\tilde{\mathbf{r}}^e = \begin{bmatrix} \tilde{\phi} \\ \tilde{\lambda} \\ \tilde{h} \end{bmatrix} \quad (18)$$

- **Velocity sensor:** Measuring the velocity in the local-level frame:

$$\tilde{\mathbf{r}}^l = \begin{bmatrix} \tilde{r}_n \\ \tilde{r}_e \\ \tilde{r}_d \end{bmatrix} \quad (19)$$

- **Accelerometer:** Measuring the specific force in the body frame:

$$\tilde{\mathbf{f}}^b = \begin{bmatrix} \tilde{f}_x \\ \tilde{f}_y \\ \tilde{f}_z \end{bmatrix} \quad (20)$$

- **Attitude sensor:** Measuring the relative orientation from the body frame to the local-level frame:

$$\tilde{\mathbf{q}}_b^l = \begin{bmatrix} \tilde{q}_1 \\ \tilde{q}_2 \\ \tilde{q}_3 \\ \tilde{q}_4 \end{bmatrix} \quad (21)$$

- **Gyroscope:** Measuring the angular rate of the body frame with respect to the local frame expressed in the body frame:

$$\tilde{\omega}_{lb}^b = \begin{bmatrix} \tilde{\omega}_x \\ \tilde{\omega}_y \\ \tilde{\omega}_z \end{bmatrix} \quad (22)$$

- **Rotor speed sensor:** Measuring the absolute angular rate of each rotor:

$$\tilde{\eta}_{RF}, \tilde{\eta}_{LF}, \tilde{\eta}_{LB}, \tilde{\eta}_{RB} \quad (23)$$

Where the subscripts R and L stand for Right and Left, respectively; and F, B for Front and Back, respectively. See Figure 5.

2.4.3 Error modeling

Every measurement is subjected to measurement error due to sensor imperfection, which can be expressed as:

$$\epsilon_x = x - \tilde{x} \quad (24)$$

Where x is a generic quantity, and \tilde{x} its measurement. In this thesis, measurement errors will not be assessed for each sensor. But it is implicitly understood that sensor readings are not perfect.

2.5 Motion model

This section will be dedicated to the derivation of the motion model of the air-craft in relation to the forces generating this motion. This will serve as a mathematical basis for the different methods of wind estimation, where drag is defined as an external perturbation force. This section will also highlight some important used for experimental confirmation.

2.5.1 Equation of motion

Using Newton's second law of motion for a rigid body in the i-frame, one can write:

$$\ddot{\mathbf{r}}^i = \mathbf{C}_b^i \mathbf{f}^b + \bar{\mathbf{g}}^i(\mathbf{r}^i) \quad (25)$$

Where $\bar{\mathbf{g}}(\mathbf{r})$ is the gravity model function, returning the gravitational acceleration for a given position in space. Now moving to the e-frame using the fact that $\mathbf{r}^e = \mathbf{C}_i^e \mathbf{r}^i$ and taking (11) into account when performing the time derivative:

$$\ddot{\mathbf{r}}^e = \mathbf{C}_b^e \mathbf{f}^b + \bar{\mathbf{g}}^e(\mathbf{r}^e) - 2\Omega_{ie}^e \dot{\mathbf{r}}^e - \Omega_{ie}^e \Omega_{ie}^e \mathbf{r}^e \quad (26)$$

In this derivation the angular velocity of the Earth with respect to the inertial frame Ω_{ie}^e is assumed constant (As.2). Since the gravitational acceleration and the centrifugal acceleration $\Omega_{ie}^e \Omega_{ie}^e \mathbf{r}^e$ both only depend on position and their main component is aligned with the local vertical they will be combined to form the local gravity vector:

$$\mathbf{g}^e(\mathbf{r}^e) = \bar{\mathbf{g}}^e(\mathbf{r}^e) - \Omega_{ie}^e \Omega_{ie}^e \mathbf{r}^e \quad (27)$$

Now moving to the local-level frame using (27) and using the fact that $\mathbf{r}^l = \mathbf{C}_e^l \mathbf{r}^e$ and taking (11) into account when performing the time derivative:

$$\ddot{\mathbf{r}}^l = \mathbf{C}_b^l \mathbf{f}^b + \mathbf{g}^l(\mathbf{r}^l) - (\Omega_{el}^l + 2\Omega_{ie}^l) \dot{\mathbf{r}}^l \quad (28)$$

The angular velocity vector of Earth expressed in the local frame is given by:

$$\omega_{ie}^l = \begin{bmatrix} \omega_{ie} \cos \phi \\ 0 \\ -\omega_{ie} \sin \phi \end{bmatrix} \quad (29)$$

Where ω_{ie} is the magnitude of the Earth's angular velocity. A typical value is $\omega_{ie} = 7.292115 \cdot 10^{-5}$ [rad/s] [24]. However, in this work the effect of Earth's rotation will be neglected (As.3). This assumption is realistic, since the resulting acceleration in the local frame is small compared to the gravity term. Indeed, the aircraft's speed in the local frame cannot exceed $\max(\dot{\mathbf{r}}^l) = 10$ [m/s] [7], which corresponds at most to an acceleration of $2\omega_{ie} \max(\dot{\mathbf{r}}^l) = 2.2 \cdot 10^{-4}$ [m/s²]. It is worth noting, that in the context of inertial navigation, this assumption may not be valid anymore. Indeed, due to cumulative processes (double integration over time to compute position) this small term may add up to a very large error [22], but no such cumulative processes will be used in this work. The angular velocity vector of the local frame with respect to the earth frame, named the local frame transport rate, is given by:

$$\omega_{el}^l = \begin{bmatrix} \dot{\lambda} \cos \phi \\ -\dot{\phi} \\ \dot{\lambda} \sin \phi \end{bmatrix} \quad (30)$$

This term is also very small and thus will be neglected (As.4). Indeed, again using a maximal speed of 16 [m/s], the maximal transport rate is $\max(\dot{\mathbf{r}}^l)/R_e = 2.5 \cdot 10^{-6}$ [m/s²] (along one axis), where $R_e = 6378$ [km] [24] is the Earth's radius. Hence rewriting (28) taking into account (As.3) and (As.4):

$$\ddot{\mathbf{r}}^l \approx \mathbf{C}_b^l \mathbf{f}^b + \mathbf{g}^l(\mathbf{r}^l) \quad (31)$$

Note that this equation corresponds to the motion equation in the i-frame (25), meaning that under the above-mentioned assumptions, the local frame is considered as inertial.

2.5.2 Specific forces

Two specific forces will be considered to act on the air-craft: *thrust*, which is generated by the propellers and *drag*, which is generated by the flow of air around the drone (As.5). Note that lift is not considered in this model since there are only very small lift-generating surfaces on the Phantom 4 RTK body⁷. Note that this also assumes that only flight scenarios are considered and no force generated by contact with the ground

⁷With the obvious exception of the propeller blades, but this is taken into account in the thrust force

needs to be considered. The relevant models for thrust and drag forces will be specific to each estimation method and thus will be described in their respective sections. However, the general expression of specific force holds:

$$\mathbf{f}^b = \mathbf{f}_T^b + \mathbf{f}_D^b = \frac{1}{m} \mathbf{F}_T^b + \frac{1}{m} \mathbf{F}_D^b \quad (32)$$

Where \mathbf{f}_T^b is the specific thrust force in the body frame, \mathbf{f}_D^b is the specific drag force in the body frame, \mathbf{F}_T^b is the mass thrust force in the body frame, \mathbf{F}_D^b is the mass drag force in the body frame and $m = 1.391$ [kg] [7] the mass of the drone.

2.6 Air density estimation

Air density needs to be estimated several time throughout this work. It will be estimated using the model given the *Bureau International des Poids et Mesures* for density of humid air as described in [25]. All equations and constants hereafter come from this source. Air density will be function of observed air temperature A_t , atmospheric pressure A_p and relative humidity A_h as follows:

$$\rho = \rho(A_t, A_p, A_h) = \frac{p M_a}{Z R T} \left(1 - x_v \left(1 - \frac{M_v}{M_a} \right) \right) \quad (33)$$

Where the parameters are described in Table 3.

Table 3: Air density base equation parameters (see Equation (33)).

Symbol	Description	Value
p	Atmospheric pressure	A_p [Pa]
M_a	Molar mass of dry air	$28.9635 \cdot 10^{-3}$ [kg/mol]
Z	Compression factor	See Section 2.6.1
R	Molar gaz constant	8.31441 [J/mol/K]
T	Thermodynamic air temperature	A_t [K]
x_v	Molar fraction of water vapour	See Section 2.6.2
M_v	Molar mass of water	$18.015 \cdot 10^{-3}$ [kg/mol]

2.6.1 Compression factor

The compression factor Z is given by:

$$Z = 1 - \frac{p}{T} (a_0 + a_1 t + a_2 t^2 + (b_0 + b_1 t)x_v + (c_0 + c_1 t)x_v^2) + \frac{p^2}{T^2} (d + e x_v^2) \quad (34)$$

Where the parameters are described in Table 4.

2.6.2 Molar fraction of water vapour

The molar fraction of water vapour x_v is given by:

$$x_v = h_r f \frac{p_{SV}}{p} \quad (35)$$

Where the augmentation factor f is given by:

$$f = \alpha + \beta p + \gamma t^2 \quad (36)$$

And the vapour saturation pressure is given by:

$$p_{SV} = \exp \left(AT^2 + BT + C + \frac{D}{T} \right) \quad (37)$$

And the remaining parameters a given in Table 5.

Table 4: Compression factor equation parameters (see Equation (34)).

Symbol	Description	Value
p	Atmospheric pressure	$A_p [Pa]$
T	Thermodynamic air temperature	$A_t [K]$
t	Air temperature	$A_t [{}^\circ C]$
x_v	Molar fraction of water vapour	See Section 2.6.2
a_0	Constant	$1.62419 \cdot 10^{-6} [K/Pa]$
a_1	Constant	$-2.8969 \cdot 10^{-8} [1/Pa]$
a_2	Constant	$1.0880 \cdot 10^{-1} [1/K/Pa]$
b_0	Constant	$5.757 \cdot 10^{-6} [K/Pa]$
b_1	Constant	$-2.589 \cdot 10^{-8} [1/Pa]$
c_0	Constant	$1.9297 \cdot 10^{-4} [K/Pa]$
c_1	Constant	$-2.285 \cdot 10^{-6} [1/Pa]$
d	Constant	$1.73 \cdot 10^{-11} [K^2/Pa^2]$
e	Constant	$-1.034 \cdot 10^{-8} [K^2/Pa^2]$

Table 5: Molar fraction of water vapour equation parameters (see Equations (35), (36) and (37)).

Symbol	Description	Value
h_r	Relative humidity	$A_h []$
f	Augmentation factor	See Equation (36)
p_{SV}	Vapour saturation pressure	See Equation (37)
p	Atmospheric pressure	$A_p [Pa]$
α	Constant	$1.00062 []$
β	Constant	$3.14 \cdot 10^{-8} [1/Pa]$
γ	Constant	$5.6 \cdot 10^{-7} [1/K]$
T	Thermodynamic air temperature	$A_t [K]$
A	Constant	$1.2811805 \cdot 10^{-5} [1/K^2]$
B	Constant	$-1.9509874 \cdot 10^{-2} [1/K]$
C	Constant	$34.04926034 []$
D	Constant	$-6.3536311 \cdot 10^3 [K]$

2.7 Wind triangle

In this work the air speed with respect to a moving platform will be estimated \mathbf{V}^b . However the air speed with respect to the local-level frame is needed, i.e. the physical wind \mathbf{w}^l (see Section 2.4.1). The relation between air speed and wind speed (known as the aviation triangle) depends on the platform speed \mathbf{r}^l and is expressed in the local-level frame as follows⁸:

$$\mathbf{w}^l = \mathbf{r}^l + \text{rot}(\mathbf{q}_b^l, \mathbf{V}^b) \quad (38)$$

⁸The "traditional" aviation triangle subtracts the air speed from the ground speed, but in this version of the triangle, it is the platform speed with respect to air that is used and not the air speed with respect to the platform as defined in this work. Thus changing the sign.

3 Methodology

3.1 Data Collection

Data used in this project come from various sources and can be classified in two categories: flight data and reference data. Flight data contains sensor output generated by an aircraft during a flight. This is the data used to make the wind estimation. Reference data contains readings of direct wind sensors (weather station). This data is used to validate and/or calibrate the wind estimation. All data acquired is time-stamped using *UTC time* to ensure synchronisation and acquired at 10 [Hz] .

3.1.1 Flight data

Flight data was produced by two different sources:

- DJI Phantom 4: this drone was used in [19] and produced a dataset of hovering flights performed in Norway.
- DJI Phantom 4 RTK: this drone was used in this thesis and produced a dataset of various flights performed in Switzerland (see Section 4.1 for flight details). This drone is Real Time Kinematic (RTK) enabled which results in centimeter level accuracy in the drone's GNSS measurements ([@ DJI website](#)).

The signals extracted from each source are the same and are described in Table 6.

The data is extracted from the drones using DatCon, the exact procedure is described in Appendix A.

Symbol	Field Name	Unit	Description
ϕ	lati	[deg]	AC Latitude (WGS84)
λ	long	[deg]	AC Longitude (WGS84)
h	alti	[m]	AC Altitude (WGS84)
\dot{r}_n	vn	[m/s]	AC Speed in local frame (NED)
\dot{r}_e	ve	[m/s]	AC Speed in local frame (NED)
\dot{r}_d	vd	[m/s]	AC Speed in local frame (NED)
f_x	ax	[m/s ²]	AC Acceleration in body frame (XYZ)
f_y	ay	[m/s ²]	AC Acceleration in body frame (XYZ)
f_z	az	[m/s ²]	AC Acceleration in body frame (XYZ)
q_1	q1		
q_1	q2		Quaternion describing the rotation from
q_2	q3		body frame to local-level frame.
q_3	q4		
ω_x	gyroX	[rad/s]	AC Angular rate in the roll direction
ω_y	gyroY	[rad/s]	AC Angular rate in the pitch direction
ω_z	gyroZ	[rad/s]	AC Angular rate in the yaw direction
η_{RF}	motRpm_RF	[rpm]	Right-Front rotor speed
η_{LF}	motRpm_LF	[rpm]	Left-Front rotor speed
η_{LB}	motRpm_LB	[rpm]	Left-Back rotor speed
η_{RB}	motRpm_RB	[rpm]	Right-Back rotor speed

Table 6: Flight data signals.

3.1.2 Reference data

Reference data was produced by three different sources:

- UNISAWS: The University Center in Svalbard (UNIS) Automatic weather station, situated in Advent-dalen (Norway). The station measures wind at 2m and 10m above ground and several other atmospheric parameters. See [19] for details.



(a) Picture of MoTUS mast (on the right) and flying drone (small on the left).



(b) Picture of the TOPO weather station (on the left) and of the MoTUS mast (on the right).

Figure 7: Weather stations.

- MoTUS: Urban microclimate measurement mast ([@ MoTUS website](#)), situated in Lausanne (Switzerland). The mast features seven sonic anemometers, spread vertically up to a height of approx. 22m. Figure 7a and 7b shows a picture of the MoTUS mast.
- TOPOAWS: Geodetic Engineering Laboratory (TOPO) automatic weather station. This is small portable weather station developed by TOPO ([TOPOAWS Wiki](#), access restricted). Figure 7b shows a picture of the weather station.

Every reference data source does not provide the same signals, the collected signals available sources are described in Table 7.

3.1.3 Flight campaign

During this work 56 flights spread over 6 different days were executed. A flight is defined as the time interval between take-off and landing. Flights were performed using a DJI Phantom 4 RTK (see [DJI website](#)). Flight were performed on EPFL's campus around in the flight zone show in Figure 8. This location was chosen because of the presence of the MoTUS weather mast providing wind reference measurements. This location proved to be convenient since it is very close to the TOPO office and since it is a flight zone already recognized by Ecole Polytechnique Fédérale de Lausanne (EPFL).

The flights were performed using the Phantom's autopilot in *waypoint mode*, which allows the definition of a set of waypoints the drone will cross and a maximal cruising speed. A waypoint is defined by its latitude, longitude, altitude and heading. A set of waypoints and their cruising speed is stored as a flight plan in the drone Remote Control (RC) and will be called a *flight type* in this work. Ten different flight types were defined that can be grouped in four categories:

Full Name	Field Name	Unit	UNISAWS	MoTUS	TOPOAWS
Horizontal wind direction @ 1.5 m	windHDir_0150cm	[deg]	YES	YES	YES
Horizontal wind magnitude @ 1.5 m	windHMag_0150cm	[m/s]	YES	YES	YES
Vertical wind speed @ 1.5 m	windVert_0150cm	[m/s]	YES	YES	
Horizontal wind direction @ 11.4 m	windHDir_1140cm	[deg]	YES	YES	
Horizontal wind magnitude @ 11.4 m	windHMag_1140cm	[m/s]	YES	YES	
Vertical wind speed @ 11.4 m	windVert_1140cm	[m/s]	YES	YES	
Horizontal wind direction @ 14.7 m	windHDir_1470cm	[deg]		YES	
Horizontal wind magnitude @ 14.7 m	windHMag_1470cm	[m/s]		YES	
Vertical wind speed @ 14.7 m	windVert_1470cm	[m/s]		YES	
Horizontal wind direction @ 18.0 m	windHDir_1800cm	[deg]		YES	
Horizontal wind magnitude @ 18.0 m	windHMag_1800cm	[m/s]		YES	
Vertical wind speed @ 18.0 m	windVert_1800cm	[m/s]		YES	
Horizontal wind direction @ 21.3 m	windHDir_2130cm	[deg]		YES	
Horizontal wind magnitude @ 21.3 m	windHMag_2130cm	[m/s]		YES	
Vertical wind speed @ 21.3 m	windVert_2130cm	[m/s]		YES	
Air temperature	tempRef	[°C]	YES		(YES)
Atmospheric pressure	pressRef	[hPa]	YES		YES
Relative humidity	humidRef	[%]	YES		YES

Table 7: Reference data signals.

- **Hover:** Using two waypoints⁹, the drone moves to an altitude of approximately 20m above ground and 10m to the south of the weather mast. Once the final waypoint reached the drone hovers, hold its position, camera looking toward north. The pilot can decide when the hover flight should end, in this work position was held for typically 5 to 10 minutes.
- **Square:** The drone moves in an approximate square around the weather mast with an approximate minimum distance of 10m to it. The heading angle is such that the camera is always looking at the weather mast.
- **Constant speed (cstSpeed):** The drone moves approximately from the north-east corner of the flight zone to its south-west corner then back and then to the south-west corner again (three segments in total). Heading is always in the travel direction. This flight plan is repeated four times for a cruising speed of 2, 6, 10 and 13 [m/s]. Thus resulting in four flight types.
- **Vertical:** The drone moves to the space location has for hovering but at around 15m above ground. Then it moves up and down three times to approximately 30m above ground and back to 15m. During the maneuver the camera is always looking north. This flight is repeated four times at a cruising speed of 2, 3, 4 and 5 [m/s]. Thus resulting in four flight types.

An example of each category can be seen in Figure 9. A typical flight campaign is described in Appendix B. The acquired data will be discussed in Section 4.1 and list of all flights can be found in Table 15.

3.2 Wind estimation from tilt

3.2.1 Introduction

Motivation This section will describe the approach referred as the *simple method* in [19]. This estimation will serve as a baseline for the new estimation scheme proposed in this work and will be referred to as *tilt method* or *tilt estimation*.

General description The idea behind this approach is very elegant by its simplicity. Assuming the drone is perfectly stationary, i.e. the drone has an autopilot capable of keeping the drone at the same position

⁹A flight plan with only one waypoint is not valid.



Figure 8: Flight zone overview. Center of the zone is at $46^{\circ}31'17.0''\text{N}$ $6^{\circ}34'02.5''\text{E}$

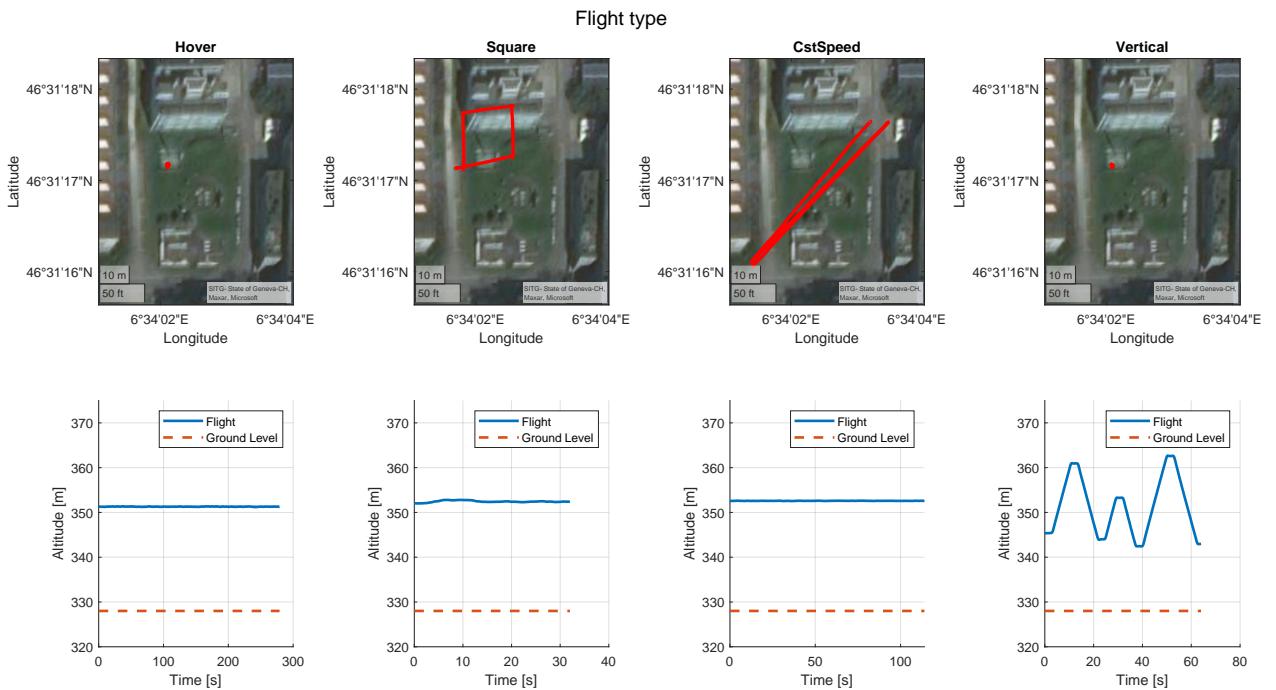


Figure 9: Flight types. Plots on the upper row show a typical flight as seen from above (to be compared with Figure 8. Plots on the lower row show the drone's altitude over time. Plotted flights are hover, first square, cstSpeed2ms and Vertical2ms taken on 07.06.2021 (see Table 15).

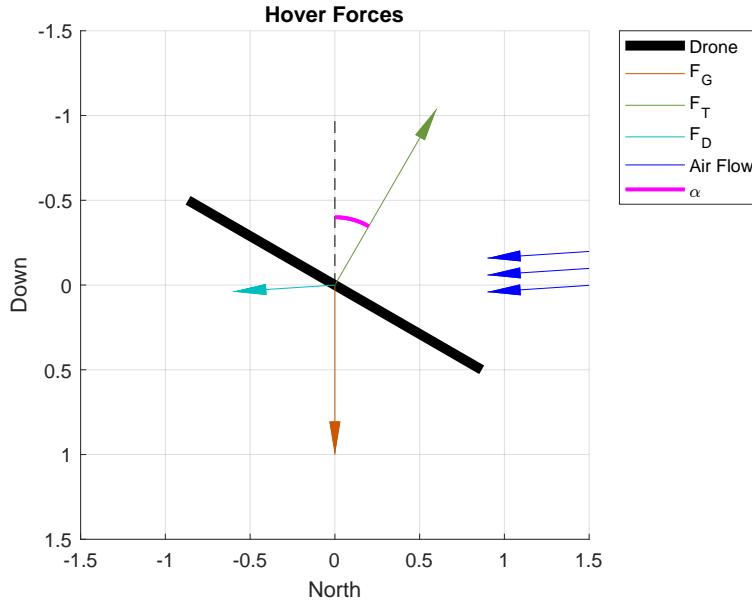


Figure 10: Forces acting on a hovering drone in the wind. Where F_G (in orange) is the gravitational force, F_T (in green) the thrust force, F_D (in blue) the drag force. The air flow direction is shown in blue and the tilt angle α is shown in pink.

regardless of the wind conditions, then the tilt angle of the drone is correlated to the wind velocity. Indeed, the autopilot has to tilt the drone such that the thrust force compensates for the drag forces generated by the wind. Figure 10 shows a diagram of the force situation. From this figure it becomes clear that the bigger the drag force, the bigger the tilt angle α .

Step-by-step, while highlighting the needed inputs, the estimation procedure can be summarised as follows in Table 8.

Step	Description	Needed Input	Section
1	Establish model relating tilt and wind	Flight dataset	3.2.2 and [19]
2	Compute tilt	Altitude measurements	2.3.7
3	Compute air velocity	Tilt angle	3.2.3
4	Compute air direction	Tilt direction	3.2.4
5	Compute wind vector	Air velocity and direction	2.7

Table 8: Wind from tilt estimation algorithm summary.

3.2.2 Model calibration

In order to choose and calibrate the model relating tilt angle and wind velocity, A. Garreau used a dataset containing several flights next to a weather station serving as a wind reference. Part of this dataset was used for calibration the other for validation. Figure 11 shows the calibration dataset for a DJI Phantom 4 Pro and a DJI Mavic 2 Enterprise and two different attempt to fit the datasets. Note that the Figure shows the correlation between the tangent of the tilt angle $\tan(\alpha)$ and the air velocity squared $\|\mathbf{V}^b\|^2$, which are noted $\tan(\gamma)$ and v^2 respectively in [19]. Moreover, this model only holds under this assumption that the drone is stationary, i.e. the position of the drone in the local-level frame is constant (As.7) and that the wind is contained in the azimuthal plane (As.6). The chosen model for the DJI Phantom 4 Pro is the following:

$$\|\mathbf{V}^b\|^2 = \begin{cases} a_0 \tan^2(\gamma) & \text{for } \gamma < \gamma_{crit} \\ a_1 \tan(\gamma) + a_2 & \text{for } \gamma \geq \gamma_{crit} \end{cases} \quad (39)$$

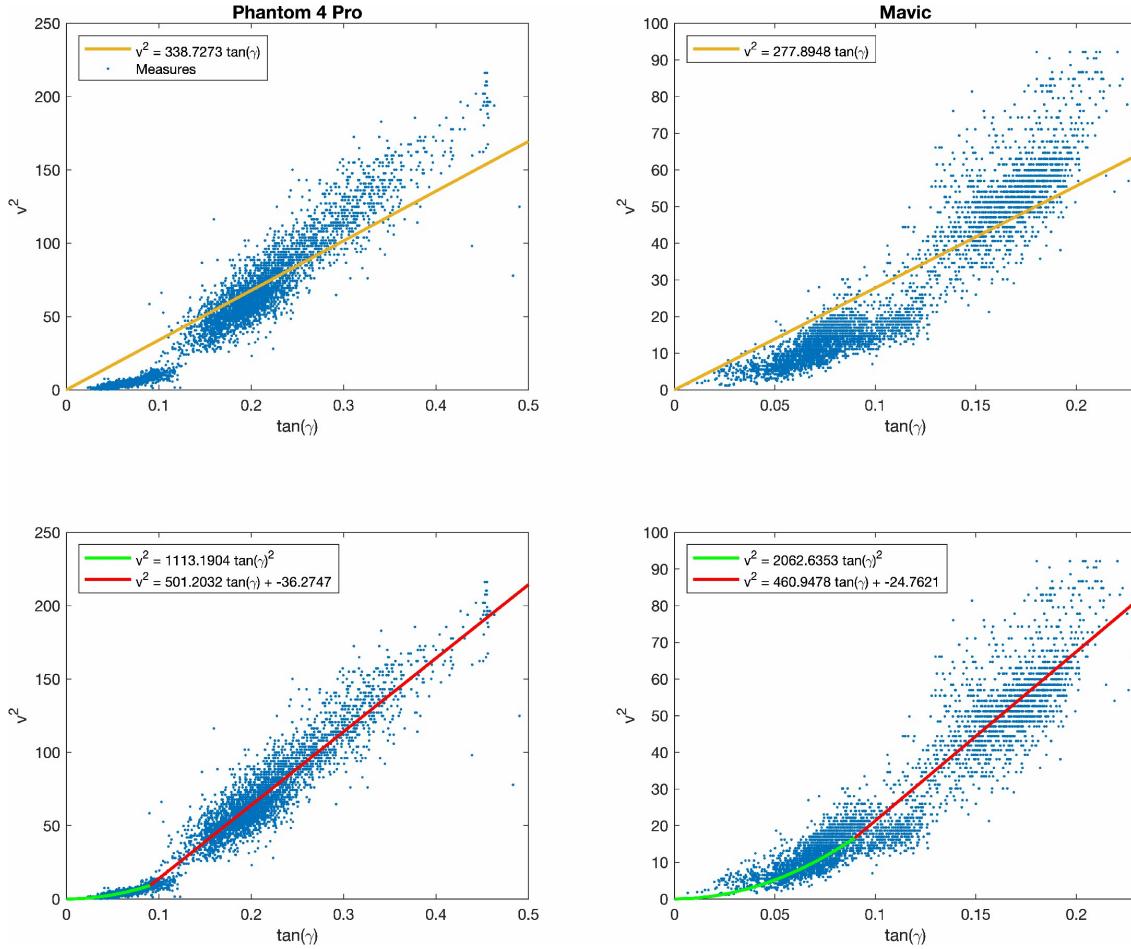


Figure 11: Tilt to wind correlation for a DJI Phantom 4 Pro and DJI Mavic 2 Enterprise drone. This Figure is taken from [19].

Where the constants are defined in Table 9. The detail behind the choice of this model will not be exposed here and the reader should refer to [19].

Symbol	Value
a_0	$1113.2 \text{ [m}^2/\text{s}^2]$
a_1	$501.2032 \text{ [m}^2/\text{s}^2]$
a_2	$-36.2747 \text{ [m}^2/\text{s}^2]$
γ_{crit}	0.091 [rad]

Table 9: Tilt to air velocity model parameter (see Equation (39)).

3.2.3 Computing air velocity

Once the tilt angle α is computed using Equation 14 the air velocity can directly be computed using Equation (39). But for this to be true it must be assumed that the Phantom 4 RTK used in this work has the same aerodynamic properties than the Phantom 4 Pro used to establish the parameters in Table 9 (As.9).

3.2.4 Computing air direction

The air direction in the azimuthal plane is directly given by the tilt direction λ which can be computed using Equation (13).

3.2.5 Computing wind vector

Let's first build the air speed vector in the local-level frame given the air velocity and direction:

$$\mathbf{V}^l = \begin{bmatrix} \|\mathbf{V}^b\| \cos(\lambda) \\ \|\mathbf{V}^b\| \sin(\lambda) \\ 0 \end{bmatrix} \quad (40)$$

The air speed vector can be used in Equation (38) to compute the physical wind speed vector:

$$\mathbf{w}^l = \dot{\mathbf{r}}^l + \text{rot}(\mathbf{q}_b^l, \mathbf{V}^b) = \mathbf{V}^l \quad (41)$$

Where the second equality follows from the assumption that the drone is stationary (As.7).

3.3 Wind estimation from dynamical model

3.3.1 Introduction

Motivation The assumption that the drone is stationary (As.7) used in the *wind from tilt* estimation (see Section 3.2) is a hard assumption. Indeed, on one side, if the autopilot is not good enough and results in oscillations around the hover point then this assumption fails. On the other side, this assumption also prevents the aircraft to perform continuous wind profiling over position. In other words, under this assumption, the measurement of a wind profile would need the drone to move to a sample point, hover for a while and repeat this for each sample point. If this assumption could be removed then the drone could simply fly through all sample points without the need to stop. Hence, this new estimation scheme will aim at removing the stationarity assumption.

General description Instinctively the idea behind this estimation is the following. The multi-rotor is under the influence of two specific forces: drag and thrust. Hence if the total specific force and the thrust is known (observed) then the drag is known as well. Fortunately, the total specific force is known thanks to the accelerometer and thrust can be estimated from the rotor speed (which are measured). Drag is the product of wind (air flow) and the resistance response of the aircraft structure. Hence if the former is known the latter can be estimated, provided an appropriate drag model. Finally, the wind speed can be deduced by knowing the drone's velocity and the air speed around it. Step-by-step, while highlighting the needed inputs, the procedure can be summarised as follows in Table 10.

Step	Description	Needed Input	Section
1	Compute thrust	Rotor speed	3.3.2
2	Compute drag (without thrust)	Thrust and Acc. measurements	3.3.4
2bis	Compute drag	Thrust and Acc. measurements	3.3.3
3	Compute air speed (linear drag)	Drag	3.3.6, 2.3.7 and 3.3.7
3bis	Compute air speed (quadratic drag)	Drag	3.3.5, 2.3.7 and 3.3.7
4	Compute wind speed	Air speed and drone speed	2.7

Table 10: Dynamic Model estimation algorithm summary.

As can be seen in the table, there are two alternatives for step 2 and for step 3 and they can be combined independently. Hence this section is describing four estimation methods which will be named:

- Dynamic Model (DM), Linear and No Vertical Drag (using step 2 and 3)
- DM, Linear and Vertical Drag (using step 2bis and 3)

- DM, Quadratic and No Vertical Drag (using step 2 and 3bis)
- DM, Quadratic and Vertical Drag (using step 2bis and 3bis)

3.3.2 Computing thrust

A common simple thrust model in the body frame is given by (As.8):

$$\mathbf{F}_T^b = \begin{bmatrix} 0 \\ 0 \\ -\rho b(\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2) \end{bmatrix} \quad (42)$$

Where ρ is the air density and b is the motor thrust constant. The air density can be estimated from the method described in Section 2.6. The motor thrust constant also needs to be estimated. Here this constant will be inferred from data produced in [21]. The data contains force measurements at a given rotor speed (same for all four rotors) for a DJI Phantom 3 drone. In this case the model can be expressed as follows:

$$\mathbf{F}_T^b = \begin{bmatrix} 0 \\ 0 \\ -\bar{b}\bar{\eta}^2 \end{bmatrix} \quad (43)$$

Where \bar{b} is the drone thrust constant and $\bar{\eta}$ is the overall rotor speed, which will be defined in (44). Using a linear least-square regression, one can estimate $\bar{b} = 4.9 \cdot 10^{-7} [N/RPM^2]$, this model is plotted on Figure 12. Assuming that the thrust produced by the DJI Phantom 4 RTK is the same, then the same model can be used (As.9). However, in [21], the rotor speed is the same for all four rotors, which is not the case during a typical flight. Hence, $\bar{\eta}$ needs to be computed. Due to the quadratic relation of rotation rates and thrust, $\bar{\eta}$ should be computed as follows:

$$\bar{\eta} = \frac{1}{2} \sqrt{\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2} \quad (44)$$

Indeed, if (44) is used in (43):

$$\mathbf{F}_T^b = \begin{bmatrix} 0 \\ 0 \\ -\rho \bar{b} \frac{1}{4} (\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\rho b(\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2) \end{bmatrix} \quad (45)$$

Where the second equality assumes the drone is symmetric ($\bar{b} = 4b$) and thus shows that the $\bar{\eta}$ computation is coherent with the single motor thrust model. Finally, thrust in the body frame can be expressed as:

$$\mathbf{F}_T^b = \begin{bmatrix} 0 \\ 0 \\ -\rho \bar{b} \left(\frac{1}{2} \sqrt{\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2} \right)^2 \end{bmatrix} \quad (46)$$

3.3.3 Computing drag

Using (32) the drag force can be written as:

$$\mathbf{F}_D^b = m\mathbf{f}^b - \mathbf{F}_T^b \quad (47)$$

Moving to the local-level frame:

$$\mathbf{F}_D^l = m\mathbf{C}_b^l \mathbf{f}^b - \mathbf{C}_b^l \mathbf{F}_T^b = rot(\mathbf{q}_b^l, m\mathbf{f}^b) - rot(\mathbf{q}_b^l, \mathbf{F}_T^b) \quad (48)$$

In this expression \mathbf{q}_b^l and \mathbf{f}^b are known through their direct measurement $\tilde{\mathbf{q}}_b^l$ and $\tilde{\mathbf{f}}^b$. Thus only the thrust force remains to be estimated, which can be done as described in Section 3.3.2.

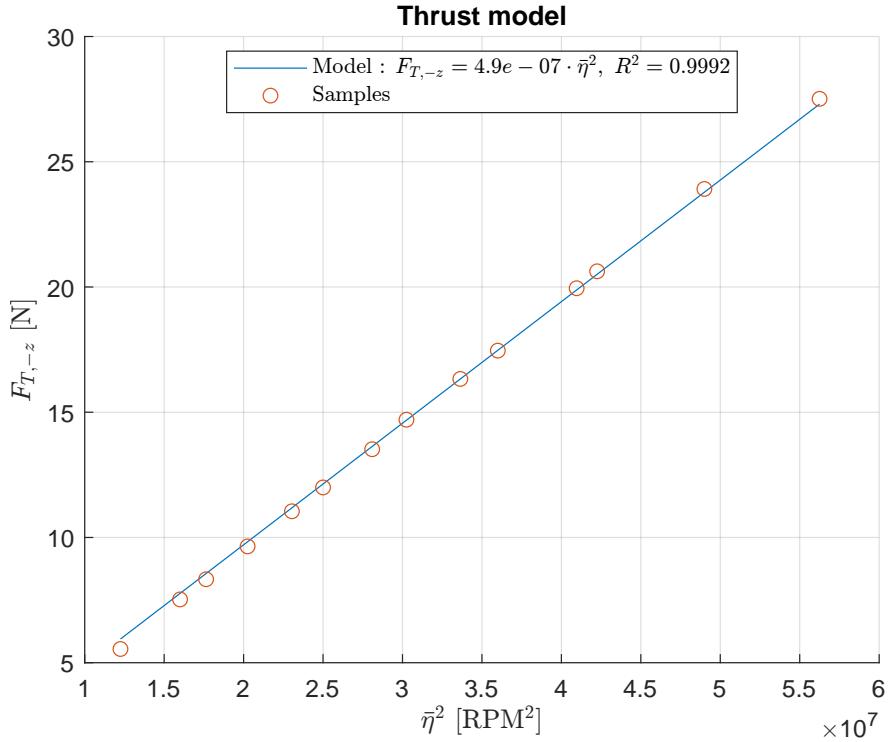


Figure 12: Thrust model fitted on data from [21].

3.3.4 Computing drag (without thrust)

Alternatively, drag can be estimated without using thrust by assuming there is no drag in the vertical direction. In this case, (48) can be rewritten as:

$$\begin{bmatrix} F_{D,n} \\ F_{D,e} \\ 0 \end{bmatrix} = m \mathbf{C}_b^l \mathbf{f}^b - \mathbf{C}_b^l \begin{bmatrix} 0 \\ 0 \\ F_{T,z} \end{bmatrix} \quad (49)$$

Grouping all three unknowns ($F_{D,n}$, $F_{D,e}$ and $F_{T,z}$) in a vector \mathbf{x} and rearranging (49), \mathbf{x} can be written as:

$$\begin{bmatrix} F_{D,n} \\ F_{D,e} \\ F_{T,z} \end{bmatrix} = \mathbf{x} = \mathbf{A}^{-1} \mathbf{b} \quad (50)$$

Where \mathbf{A} is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & C_b^l(1,3) \\ 0 & 1 & C_b^l(2,3) \\ 0 & 0 & C_b^l(3,3) \end{bmatrix} \quad (51)$$

\mathbf{b} is given by

$$\mathbf{b} = m \mathbf{C}_b^l \mathbf{f}^b \quad (52)$$

And \mathbf{A}^{-1} indicates the inverse matrix of \mathbf{A} and $C_b^l(i,j)$ designates the component on the i^{th} line and j^{th} column of the C_b^l matrix.

3.3.5 Computing air speed (quadratic model)

The goal of this section is to find a relation between the drag force \mathbf{F}_D^b and air speed with respect to the body frame \mathbf{V}^b . Defining an axis j which points in the same direction as the air speed vector. The drag

force along the j -axis can be related to air speed in the following way:

$$F_{D,j} = \frac{1}{2} \rho_j A_j C_j V^2 = \rho_j K_j V_j^2 \quad (53)$$

Where ρ_j is the air density, A_j is the cross-section area, along the j -axis, of the object exposed to air flow, C_j the drag coefficient of that given cross-section and V_j the air speed magnitude. In this model, K_j is unknown and needs to be measured or estimated. Here again the data from [21] will be leveraged. During a given experimental run in [21], air flows along an axis p with a speed V_p . In the dataset the drone can only pitch, hence the p axis is always contained in the xz -body-plane and forms an angle γ_p with the x -body-axis. γ is called the *incidence angle*. Hence, using equation (53), the following relation can be written:

$$F_{D,p} = \rho_p K_p V_p^2 \quad (54)$$

In each experimental run of [21] air density and wind speed are the same and are given as: $V_p = V_R = 6$ [m/s] and $\rho_p = \rho_R = 1.22$ [kg/m³]. The experimental data also contains measurements of forces acting on the drone's body for various p axes, i.e γ_p angles. This data can be used to compute $F_{D,p}$ as is detailed in Section 3.3.7. $F_{D,p}$ also depends on average rotor speed $\bar{\eta}$ as defined in (44). Making this dependency explicit, the drag force can be expressed as :

$$F_{D,p} = D_R(\gamma_p, \bar{\eta}) \quad (55)$$

Where D_R is detailed in Section 3.3.7. Hence, isolating K_p since it is the only unknown, (54) can be rewritten as:

$$K_p = \frac{D_R(\gamma_p, \bar{\eta})}{\rho_R V_R^2} \quad (56)$$

In the situation where the j axis is the same as the p axis, then $K_j = K_p$ and (53) can be written as:

$$F_{D,j} = \rho_j \frac{D_R(\gamma_j, \bar{\eta})}{\rho_R V_R^2} V_j^2 \quad (57)$$

Isolating V_j :

$$V_j = \sqrt{V_R^2 \frac{\rho_R}{\rho_j} \frac{F_{D,j}}{D_R(\gamma_j, \bar{\eta})}} \quad (58)$$

However, in the general case, the j axis is not the same as the p axis since the p axis is always contained in the xz -body-axis of a drone that can only pitch and the j axis can be in any orientation. To solve this problem, let's use the tilt frame: see its definition in Section 2.3.5 and the calculation of the rotation quaternion in 2.3.7. This frame is oriented such that the drone is purely "pitching" around the ty-tilt-axis. Hence, if the air speed vector is contained in the txtz-plane with an incidence angle γ_{txtz} , then the situation is equivalent to the experiment found in [21]. But only under the assumption that the drone has a cylindrical symmetry around its z-body-axis (As.10). Note that, under the same assumption, if the air speed vector is colinear with the ty-tilt-axis, then the situation is also equivalent the experiment found in [21] but with an incidence angle $\gamma_{ty} = 0$. Hence for an arbitrary j -axis, the problem can be decomposed by projecting air speed and drag force onto the txtz-tilt-plane and onto the ty-tilt-axis. Figure 13 illustrates this projection.

This can be written down mathematically as follows. First, compute drag in the tilt frame:

$$\mathbf{F}_D^t = rot(\mathbf{q}_b^t, \mathbf{F}_D^b) \quad (59)$$

Second, compute air speed in the txtz-tilt-plane, where the incidence angle is $\gamma_{txtz} = \pi - \arctan(F_{D,tz}/F_{D,tx})$:

$$V_{txtz} = \sqrt{V_R^2 \frac{\rho_R}{\rho} \frac{F_{D,txtz}}{D_R(\gamma_{txtz}, \bar{\eta})}} \quad (60)$$

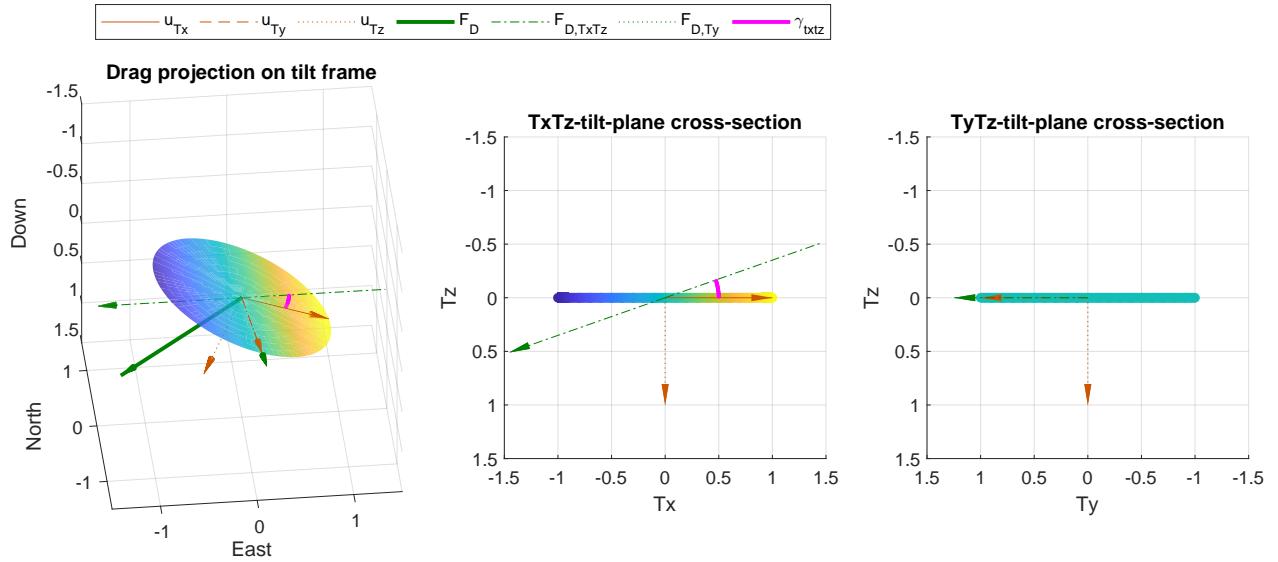


Figure 13: Drag projection in the tilt frame. The leftmost plot shows a 3D view of the situation as introduced in Figure 6. With the drag force F_D shown in green. The center plot shows a cross-section along the $txtz$ -plane with the drag vector projected in the same plane, note also the incidence angle γ_{txtz} (in pink). The rightmost plot shows a cross-section in the $tytz$ -plane with the drag vector projected along the ty -axis.

Third, compute air speed along the y -tilt-axis:

$$V_{ty} = \sqrt{V_R^2 \frac{\rho_R}{\rho} \frac{F_{D,ty}}{D_R(0, \bar{\eta})}} \quad (61)$$

Last, recompose the air speed vector and rotate it to the body frame:

$$\mathbf{V}^b = \text{rot} \left(\mathbf{q}_t^b, \begin{bmatrix} V_{txtz} \cos(\gamma_{txtz}) \\ V_{ty} \text{Sgn}(F_{D,ty}) \\ V_{txtz} \sin(\gamma_{txtz}) \end{bmatrix} \right) \quad (62)$$

3.3.6 Computing air speed (linear model)

Alternatively, another drag model will be tested where the drag force relates linearly to the air speed. In this case (53) becomes:

$$F_{D,j} = \rho K_j V_j \quad (63)$$

And thus following the same reasoning as in Section 3.3.5, air speed can be expressed as:

$$V_j = V_R \frac{\rho_R}{\rho} \frac{F_{D,j}}{D_R(\gamma_j, \bar{\eta})} \quad (64)$$

Finally, the passage through the tilt frame, again follows the same reasoning as in Section 3.3.5 and thus the air speed vector is given by:

$$\mathbf{V}^b = \text{rot} \left(\mathbf{q}_t^b, \begin{bmatrix} V_{txtz} \cos(\gamma) \\ V_{ty} \text{Sgn}(F_{D,ty}) \\ V_{txtz} \sin(\gamma) \end{bmatrix} \right) \quad (65)$$

Where the air speed in the tilt frame is given by:

$$V_{txtz} = V_R \frac{\rho_R}{\rho} \frac{F_{D,txtz}}{D_R(\gamma, \bar{\eta})} \quad (66)$$

And:

$$V_{ty} = V_R \frac{\rho_R}{\rho} \frac{F_{D,ty}}{D_R(0, \bar{\eta})} \quad (67)$$

3.3.7 Computing drag from force data

In [21], used here to estimate drag coefficients, the drag force is not directly estimated, but the forces acting on the body are measured. The detailed derivation of the relation between drag magnitude and measured force can be found in Appendix A of [19] and in [26]. The final expression, once updated to correspond to this work's notation reads:

$$D_R(\gamma, \bar{\eta}) = \tilde{F}_x \cos(\gamma) - (\tilde{F}_z + F_{T,z}) \sin(\gamma) \quad (68)$$

Where γ is the incidence angle, $\tilde{\mathbf{F}}^b$ is the measured force acting on the body and $\mathbf{F}_T(\bar{\eta})$ the thrust force which can be estimated as described in Section 3.3.2. To make the model continuous, values between sample points are linearly interpolated, the resulting model can be seen on Figure 14.

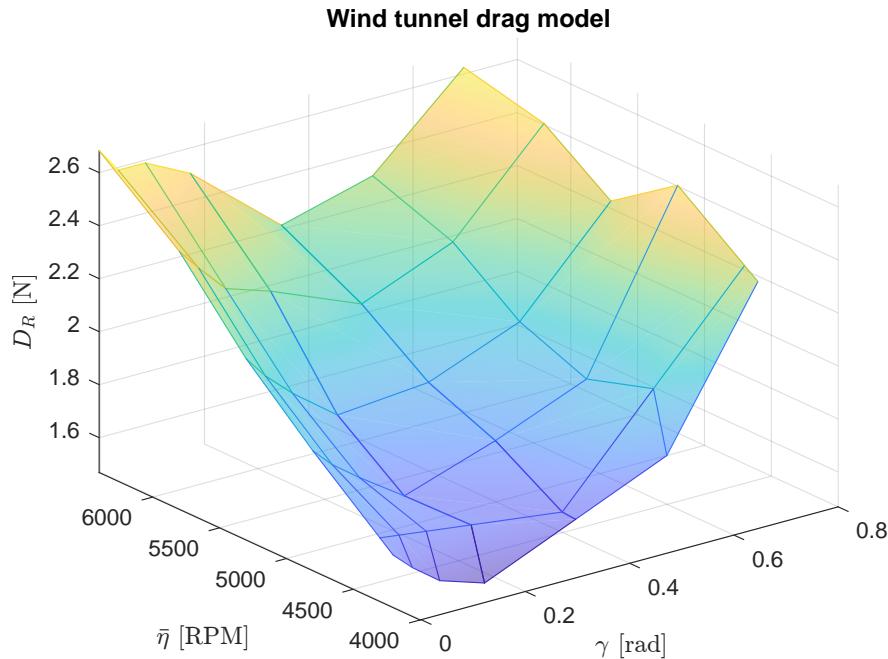


Figure 14: Wind tunnel drag force model.

3.4 Software overview

3.4.1 Introduction

To process the data used in this work, to perform the estimations and to evaluate and display the results a software package was developed. The section aims at describing its architecture and its internal data flow. The software was built using Matlab 2021a. This choice was motivated by the popularity of Matlab among academic circles as opposed to other scripting languages such as Python, despite Matlab not being free to use. However the software was constructed such that it could be executed using [Matlab Runtime](#) with very little modifications, more details will be given in the following sections. The full software can be found on GitHub at the following link : <https://github.com/meierkilian/WEMUAV>.

3.4.2 Data flow

The software is decomposed into three distinct parts: *pre-processing*, *estimation* and *evaluation*. These parts are highlighted in yellow, blue and red respectively in Figure 15. Each part depends on the data produced

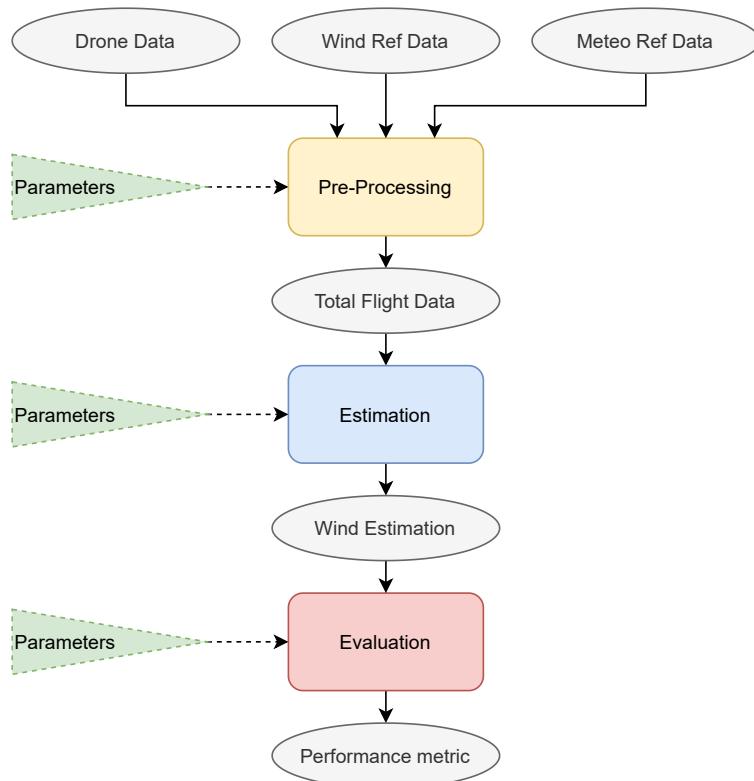


Figure 15: Software process flow. Scripts are shown as boxes, data as ellipses and external parameters as triangles.

by the previous part, but they can be executed separately. Each part is also controlled by a set of parameter defining its behaviour (see Section 3.4.4). The responsibilities of each part are:

- **Pre-processing:**

- Parse *Drone Data*, *Wind Ref Data* and *Meteo Ref Data*
- Synchronize all the data sources
- Extract data for the time interval(s) of interest
- Store the resulting data in a common *Total Flight Data* file

- **Estimation:**

- Estimate wind from *Total Flight Data* using the desired estimation method(s)
- Store each resulting estimation in a *Wind Estimation* file

- **Evaluation:**

- Compute the performance metrics (error, bias, standard deviation...) from the *Wind Estimation* files
- Display desired graphs and plots

The format of the files mentioned in this process are described in Section 3.4.3. In the software package these three parts can be found in three folders named *PrePro*, *Estimator* and *Eval* inside the *src* folder. The easiest way to execute any of these sub-modules is to use the appropriate main script present in the *src* folder, namely *main_PrePro.m*, *main_Estimate.m* and *main_Eval.m* respectively. All sub-modules can be executed in sequence using the *main.m* script.

3.4.3 Data files

This section aims at describing the data storage format and field of interest in each data file used in this work. The data files can be classified in different categories, which are shown in the four horizontal layers with ellipses in Figure 15. The first layer, corresponding to raw sensor data input, is complex, since there are several different data sources that need to be synchronized and will be the main focus of this section.

Raw sensor inputs One way to describe the goal of the pre-processing sub-module is to take raw sensor files and to populate the *field of interest* that will be present in the *Total Flight Data*. These fields are described in Tables 6 and 7. There are six different data sources and for each a parser was implemented:

- DJI Phantom data extracted with DatCon3
- DJI Phantom data extracted with DatCon4
- UNISAWS data
- MoTUS data
- TOPOAWS FW1
- TOPOAWS FW2

Sample data file for each source can be found in the GitHub repository in the [sampleData/rawSensorData](#) folder. Table 11 shows which sensor is providing which field of interest and under what header name this field is present in the file.

Field of Interest	DatConV3	DatConV4	MoTUS	UNISAWS	TOPOAWS
lati	IMU_ATTI_0_Latitude	IMU_ATTI_0_Latitude	-	-	-
long	IMU_ATTI_0_Longitude	IMU_ATTI_0_Longitude	-	-	-
alti	-	IMU_ATTI_0_alth.D	-	-	-
vn	-	IMU_ATTI_0_velN	-	-	-
ve	-	IMU_ATTI_0_velE	-	-	-
vd	-	IMU_ATTI_0_velD	-	-	-
ax	IMU_ATTI_0_accel_X	IMU_ATTI_0_accelX	-	-	-
ay	IMU_ATTI_0_accel_Y	IMU_ATTI_0_accelY	-	-	-
az	IMU_ATTI_0_accel_Z	IMU_ATTI_0_accelZ	-	-	-
roll	IMU_ATTI_0_roll	-	-	-	-
pitch	IMU_ATTI_0_pitch	-	-	-	-
yaw	IMU_ATTI_0_yaw	-	-	-	-
q1	-	IMU_ATTI_0_quatW.D	-	-	-
q2	-	IMU_ATTI_0_quatX.D	-	-	-
q3	-	IMU_ATTI_0_quatY.D	-	-	-
q4	-	IMU_ATTI_0_quatZ.D	-	-	-
gyroX	IMU_ATTI_0_gyro_X	IMU_ATTI_0_gyroX	-	-	-
gyroY	IMU_ATTI_0_gyro_Y	IMU_ATTI_0_gyroY	-	-	-
gyroZ	IMU_ATTI_0_gyro_Z	IMU_ATTI_0_gyroZ	-	-	-
motRpm_RF	Motor_Speed_RFront	Motor_Speed_RFront	-	-	-
motRpm_LF	Motor_Speed_LFront	Motor_Speed_LFront	-	-	-
motRpm_LB	Motor_Speed_LBack	Motor_Speed_LBack	-	-	-
motRpm_RB	Motor_Speed_RBack	Motor_Speed_RBack	-	-	-
windHDir_0150cm	-	-	windHDir_0150cm	WindDir2m	WindDir
windHMag_0150cm	-	-	windHMag_0150cm	WindSpeed2m	WindSpeed
windVert_0150cm	-	-	windVert_0150cm	-	-
windHDir_1140cm	-	-	windHDir_1140cm	WindDir10m	-
windHMag_1140cm	-	-	windHMag_1140cm	WindSpeed10m	-
windVert_1140cm	-	-	windVert_1140cm	-	-
windHDir_1470cm	-	-	windHDir_1470cm	-	-
windHMag_1470cm	-	-	windHMag_1470cm	-	-
windVert_1470cm	-	-	windVert_1470cm	-	-
windHDir_1800cm	-	-	windHDir_1800cm	-	-
windHMag_1800cm	-	-	windHMag_1800cm	-	-
windVert_1800cm	-	-	windVert_1800cm	-	-
windHDir_2130cm	-	-	windHDir_2130cm	-	-
windHMag_2130cm	-	-	windHMag_2130cm	-	-
windVert_2130cm	-	-	windVert_2130cm	-	-
windHDir_other	-	AirSpeed_windFromDir	-	-	-
windHMag_other	-	AirSpeed_windSpeed	-	-	-
windVert_other	-	AirSpeed.comp_alth.D	-	-	-
tempMotus	-	-	temp_2130cm	-	-
tempRef	-	-	-	AirTemp1	AirTemp1
pressAC	-	-	-	-	-
pressRef	-	-	-	AtmPressure	AtmPressure
humidAC	-	-	-	-	-
humidRef	-	-	-	AirHumidity1	AirHumidity

Table 11: Summary table showing which field of interest is present in which sensor source and under what name.

Due to an early design choice, three input files are given called *Flight Data*, *Wind Ref Data* and *Meteo Ref Data* (see Figure 15), each of these file has an input type defined in the external parameters which will choose the appropriate parser. It is good practice for clarity to pass Phantom data as the flight data file, but there would be no difference in the software behaviour if the drone data was passed as wind ref data.

Note that there are more fields present in the data files of each sensor than the one described in Table 11. So for completeness let's copy here the header line of each type of sensor data, except for the DatCon generated data since it contains more than a thousand fields.

- **UNISAWS data:** Timestamp, RecordNbr, ID, AirTemp1, AirTemp2, AirTemp3, AirHumidity1, AirTemp4, AirHumidity2, AtmPressure, WindSpeed2m, WindDir2m, WindSpeed10m, WindDir10m
- **MoTUS data:** ID, windHDir, windHMag, windVert, unit, soundSpeed, temp, Date, port

- **TOPOAWS FW1:** TOW, AirTemp1, AirHumidity, AtmPressure, WindSpeed, WindDir, GPSTime_Legacy, Lati, Long, Alti, GPSNbrSat, GPSLockFlag, TempCPU, TempSens, BattCharge, FanOn
- **TOPOAWS FW2:** date, GPStime, temp, humidty, pression, windSpeed, windDir, lat, lng, alt, nbtSat, isValid, tCpu, tSens, battery, fan

More details can be found about the MoTUS logging format in [27] (section 7) and about the TOPOAWS on its [Wiki](#) (restricted access).

Finally, let's focus on the synchronisation between each data source. Synchronisation is achieved using the time field present for each sensor. In detail and for each sensor, those are the used fields and what time reference they represent:

- DJI Phantom data extracted with DatCon3
 - Time reference : UTC
 - Used fields : GPS_dateTimeStamp, offsetTime¹⁰
- DJI Phantom data extracted with DatCon4
 - Time reference : GPS
 - Used fields : gps_utc_data:gps_year:D, gps_utc_data:gps_month:D, gps_utc_data:gps_day:D, gps_utc_data:gps_hour:D, gps_utc_data:gps_minute:D, gps_utc_data:gps_sec:D, Clock:Tick#
- UNISAWS data
 - Time reference : UTC
 - Used fields : Timestamp
- MoTUS data
 - Time reference : UTC
 - Used fields : Date
- TOPOAWS FW1
 - Time reference : GPS Time of week
 - Used fields : TOW
- TOPOAWS FW2
 - Time reference : GPS
 - Used fields : date, GPStime

Total Flight data This file is generated by the pre-Processing module. The pre-processing module generates one file for each flight. It is stored as a Matlab parameter file (.mat file) which contains a single variable called *totalTT*. This variable is a Matlab Timetable which is a Matlab data structure designed to contain time stamped data (see [Timetable documentation](#)). *totalTT* contains all the fields described in 11 sampled at 10 [Hz] and has three custom properties namely: ID, FlightName and FlightType; which are described in Section 3.4.4. A sample file can be found in the GitHub repo at [sampleData/totalFlightData](#).

Wind Estimation This file is generated by the estimation module. The estimation module generates one file for each estimation method for each flight. Data is stored in the same way as the total flight data, except the timetable variable is named *tt*. It contains all data present in the total flight data file, plus the computed estimations: *windHDir_est*, *windHDMag_est* and *windVert_est*. It also features the same custom properties as the total flight data timetable with the additional *method* property, which is the name of the method used to perform the wind estimation. A sample file can be found in the GitHub repo at [sampleData/windEstimation](#).

¹⁰Experiment using DatCon4 suggests that this might not be accurate, but was not tested on DatCon3. So this must be used with caution.

3.4.4 External Parameters

A non-Matlab-licensed user could run the complete Matlab based software once compiled using [Matlab Runtime](#). However, any user should be able to influence the behaviour of the software by passing it a set of parameters; which, thus, should not be contained in a Matlab script or Matlab data file. Hence, it was decided to define parameters in an external Extensible Markup Language (XML) which could be passed to the software as an input and the XML file itself can be modified with any XML-viewer of the user's choice. A sample parameter file can be found at in the GitHub repository in the [para](#) folder. For convenience, for licensed Matlab users, parameter generator scripts were written, which can produce a well formatted XML file by calling *main_ParaGen.m* present in the [para](#) folder. This section will not go through each parameter present in the parameter file. However, the parameter generator scripts contain description and purpose of each parameter and thus they are the best resource to list and understand all parameters. One exception will be made for the *datasetOverviewPath* present in the pre-processing parameters. This parameter contains a path to a table describing the available dataset. An example of this table can be seen on Table 15 in Appendix E. The table contains the following columns:

- **DataStartTimeString:** Datetime string defining the start of a flight in UTC time.
- **DataEndTimeString:** Datetime string defining the end of a flight in UTC time.
- **ID:** Numerical identifier of a given flight.
- **FOLDER:** Path to a folder containing all data of a given flight. This folder is expected to contain at least a *FLIGHT* and *WEATHER* sub-folder.
- **FLIGHT:** Contains the name of the *drone data* file, which should be in the *FLIGHT* folder. Hence, the concatenation of *FOLDER*, "*FLIGHT*" and *FLIGHT* gives the total path to the data.
- **REF:** Contains the name of the *wind ref data* file, which should be in the *WEATHER* folder. Hence, the concatenation of *FOLDER*, "*WEATHER*" and *REF* gives the total path to the data.
- **REFMETEO:** Contains the name of the *meteo ref data* file, which should be in the *WEATHER* folder. Hence, the concatenation of *FOLDER*, "*WEATHER*" and *REFMETEO* gives the total path to the data.
- **FlightType:** Contains the flight type of each flight. See Section 3.1.3 for a description of the different flight types.
- **Mean windHMag [m/s]:** Average horizontal wind during a given flight.

Hence, each line describes a flight and the custom properties named *ID* and *flightType* of the *total flight data* and *wind estimation* tables refer directly to the corresponding fields in this table. And the *flightName* property is formed from the concatenation of *FLIGHT*, *DataStartTimeString* and *FlightType*.

3.4.5 Architecture

To conclude this software overview this section will briefly describe the software architecture through its dependency graph. The main dependency graph can be seen on Figure 16. Each node represents a file and each arrow represents a dependency (for example *main_PrePro.m* depends on *PrePro.m*). The coloring of the nodes follows the color code introduced in Figure 15: yellow-preprocessing, blue-estimation, red-evaluation. Additionally, purple represents main nodes. Note that the parameter generator scripts are not shown in the graph. Note also, that the dependency to the Sensor Fusion and Tracking Matlab Toolbox is not shown in this graph either. Indeed, *PrePro.m*, *Est.DirectDynamicModel.m* and *Util.Frame.m* use the toolbox' implementation of quaternions and conversion to Euler angles. In Figure 16, from left to right, four depth layers can be seen. The first contain the main scripts, which act as entry points. The second layer, for the pre-processing and estimator modules acts as a switch, calling the appropriate parser or estimation method respectively. For the evaluation module this contains all the error computations and plotting scripts. In the third layer, there are the parser and estimation methods mentioned before. Finally, in the fourth layer, there are utility classes which can be used by multiple estimation methods. Thanks to this switch architecture it is easy to add another parser (for a new sensor) or to add a new estimation method.

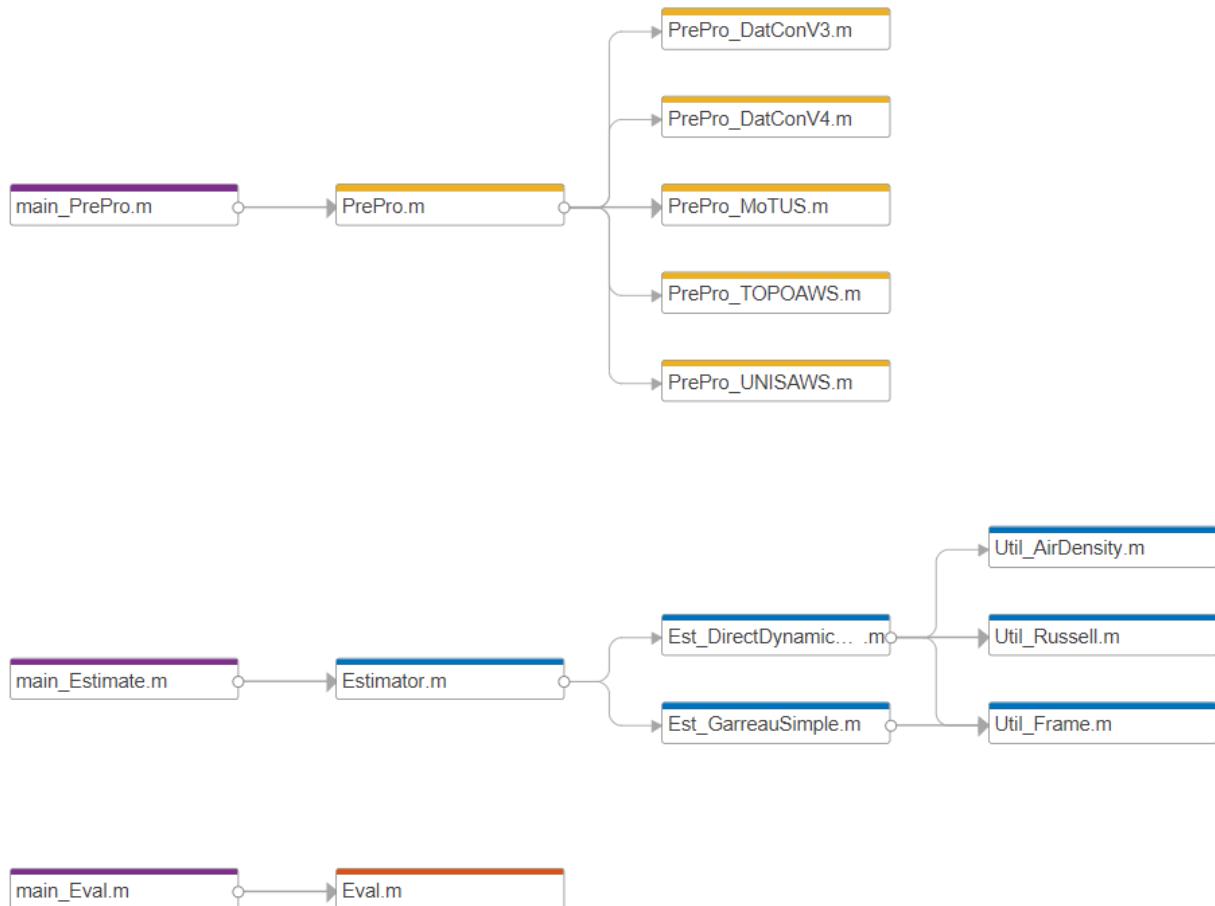


Figure 16: Main dependency graph. Parameter generators are not show, nor is the dependency to the Sensor Fusion and Tracking toolbox.

4 Results

4.1 Data Sets

The dataset used in this work is composed of 56 flights spread over 6 different days. The methodology to acquire this dataset is described in Section 3.1.3 and the file formatting and storage location are described in Section 3.4.3 and in Table 15. The flights can be divided into four categories: *hover*, *square*, *constant speed* and *vertical*, which are also described in Section 3.1.3. This section will focus on assessing the quality and the relevance of this dataset.

4.1.1 Quality of ground truth

The reference sensors used here are three-axis ultrasonic anemometer from GILL (see user manual [27]), which provide a high accuracy 3D wind vector. This sensor provides a high accuracy wind measurement. However, ground truth measurements in UAV based wind measurements are always a challenge since it is impossible to measure wind precisely at the UAV's position with a reference sensor since wind is changing over time and the UAV should not crash into the reference sensor or interfere with its measurements through the UAV's downwash. Even worse, further the reference sensor is placed from the drone, the bigger the chance that there are local wind gusts that either only effect the drone or only the reference sensor. In the edge case where the flow is purely turbulent, it is impossible to compare wind speed from two sensors if they are not at the same position¹¹. In this dataset the the drone was flying at approximately 10 to 15 [m] away from the reference sensor (except for constant speed flights which take the drone further away). This distance ensured a proper safety distance with respect to the reference sensor, while being as close as possible to it. Additionally, the flight zone (see Figure 8) is situated between buildings, hence to minimize the impact of turbulences, all flights were performed above roof height, i.e. approximately higher than 15 [m] above ground. Finally, in an attempt to filter out local effects the ground truth wind is defined as the instantaneous mean of the three upper sensor present on the mast:

$$\mathbf{w}^l(t) = \frac{1}{3} (\tilde{\mathbf{w}}_{14.7m}^l(t) + \tilde{\mathbf{w}}_{18m}^l(t) + \tilde{\mathbf{w}}_{21.3m}^l(t)) \quad (69)$$

Where $\tilde{\mathbf{w}}_h^l(t)$ is the wind vector as measured by the sensor placed at height h at time t .

4.1.2 Lack of environment variation

This dataset only contains data acquired during flights on one flight zone (Figure 8), which has its typical wind and other environmental conditions. In particular, the wind is usually coming from the south and temperatures range from 15 to 25 degrees for example. It would be interesting to introduce more variability to environmental conditions to explore their effects. Interesting environments to explore could be for example: open field (without buildings), high altitude above sea level and above ground, low temperature (close to freezing), high latitude, terrain with high convection current... Obviously the main challenge in flying in another environment is how to get a reliable ground truth measurement to evaluate the estimation performance. It was attempted to include data from [19] to add this diversity. This data features flights with up to 10 [m/s] of mean wind speed, in low temperatures and high latitude (dataset was acquired in Svalbard). However, for the moment, this attempt remained unsuccessful for various technical reasons (unknown rotation sequence of Euler angles used to describe attitude, possible miss-synchronisation with reference already mentioned by A. Garreau in his work) and due to lack of time.

4.1.3 Dataset size

This dataset is interesting since it features a variety of different flight types repeated at different wind conditions which allow comparisons among flights of a same type but also between different types. In particular, the dataset contains dynamic flights (as opposed to hovering flights) which are not usually studied in wind estimations (see Section 1.2). Nonetheless, the dataset could benefit from additional flights, especially under higher wind conditions. There were two main reasons that prevented the realisation of more flight

¹¹If the experiment takes place in a wind tunnel these problem can be avoided at the cost of a less realistic wind behaviour.

campaigns during the time frame of this thesis. First, despite EPFL having a dedicated flight zone, health and safety department needed to be notified of a flight one week in advance, which made it difficult to anticipate interesting flight days since wind is hard to forecast so long in advance. Second, on two occurrences flight campaigns had to be aborted due to seagulls threatening to attack the drone, which could have caused damage on infrastructure (drone, reference sensors ...) and harmed the birds themselves. These events are discussed in more details in Appendix C. As already discussed in Section 4.1.2, it was attempted to add data from [19], but this was not successful.

4.2 Computing performance

4.2.1 Error, bias and standard deviation

Computing performance could be done in a lot of different ways. A common way to do so is to compute the absolute error between the true (\mathbf{x}) and estimated ($\hat{\mathbf{x}}$) quantity, which can be defined using the euclidean distance as follows:

$$e = \|\mathbf{x} - \hat{\mathbf{x}}\| = |x - \hat{x}| \quad (70)$$

Where $\|\cdot\|$ is the euclidean norm, $|\cdot|$ the absolute value and where the second equality holds for 1-dimensional quantities. In the case of wind estimation, it is common (see literature presented in Section 1.2), to compute error on horizontal wind speed (w_r) and direction (w_μ) separately. Even if this makes sense intuitively, since most wind measurements are expressed using this polar coordinate system, it actually deforms the perception of the error, especially at low wind speed. Plus it is not consistent with the definition in Equation (70). This is best illustrated with an example. Let's consider two wind situations. In the first scenario, the true wind vector is given by:

$$\mathbf{w}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

And the estimated wind is:

$$\hat{\mathbf{w}}^1 = \begin{bmatrix} 0.995 \\ 0.1 \\ 0 \end{bmatrix}$$

The magnitude difference in this situation is zero ($w_r - \hat{w}_r = 0$), the angular difference is approximately 6 degrees ($w_\mu - \hat{w}_\mu = 0.1 = 6^\circ$) and the Euclidean distance ($\|\mathbf{w}^1 - \hat{\mathbf{w}}^1\|$) is 0.1. In the second scenario the true and estimated vectors are given by:

$$\mathbf{w}^1 = \begin{bmatrix} 0.05 \\ 0 \\ 0 \end{bmatrix}$$

And:

$$\hat{\mathbf{w}}^1 = \begin{bmatrix} -0.05 \\ 0 \\ 0 \end{bmatrix}$$

Here the error is again zero in magnitude but the angular error is 180 degrees and the euclidean distance still is 0.1. Intuitively, having a 180 degree error is perceived as being a lot, whereas the definition in (70) concludes that the absolute error is the same in both scenario. Hence the distance error should be preferred over the component-wise polar error. Additionally, expressing errors on an angle comes with other challenges, such as angle wrapping: the angular distance between 355° and 5° is 10°, however using the absolute difference would result in 350°; or such as the limit case where the wind magnitude is zero then the angular error is undefined. For all these reasons, error will be computed as described in (70). However, even if in the above description all three spatial direction of the local frame were considered together, the results hereafter will separate the vertical direction (local down axis) from the horizontal directions (local north and east axes). This is motivated by the fact that wind processes are different vertically than horizontally as was already discussed in Section 1.2 and by the fact that two estimation methods consider vertical wind to be zero. Hence horizontal wind error is defined as:

$$e_{w,h} = \left\| \begin{bmatrix} w_n \\ w_e \end{bmatrix} - \begin{bmatrix} \hat{w}_n \\ \hat{w}_e \end{bmatrix} \right\| \quad (71)$$

And the vertical wind error as:

$$e_{w,v} = |w_d - \hat{w}_d| \quad (72)$$

Additionally, since wind is a time varying process, performance over time should also be defined. To do so two parameters will be defined: bias and error standard deviation. *Bias* is the systematic error performed during a measurement period, in other words it gives a sense of accuracy and is defined horizontally as:

$$\mathbf{b}_{\mathbf{w},\mathbf{h}} = \frac{1}{N} \sum_{i=0}^N \left(\begin{bmatrix} w_n(i) \\ w_e(i) \end{bmatrix} - \begin{bmatrix} \hat{w}_n(i) \\ \hat{w}_e(i) \end{bmatrix} \right) \quad (73)$$

Where $x(i)$ represents the i^{th} sample of quantity x and N is the total number of samples. Similarly for the vertical direction:

$$b_{w,v} = \frac{1}{N} \sum_{i=0}^N (w_d(i) - \hat{w}_d(i)) \quad (74)$$

Standard deviation gives a sense of the spread of the error, in other words it gives a sense of precision and is defined horizontally as follows:

$$\sigma_{w,h} = \left\| \sqrt{\frac{1}{N} \sum_{i=0}^N \left(\begin{bmatrix} w_n(i) \\ w_e(i) \end{bmatrix} - \begin{bmatrix} \hat{w}_n(i) \\ \hat{w}_e(i) \end{bmatrix} - \mathbf{b}_{\mathbf{w},\mathbf{h}} \right)^2} \right\| \quad (75)$$

Where the square root, the sum and the square are applied element-wise. And vertically:

$$\sigma_{w,v} = \sqrt{\frac{1}{N} \sum_{i=0}^N (w_d(i) - \hat{w}_d(i) - b_{w,v})^2} \quad (76)$$

Note that using standard deviation to express spread assumes that the data follows a normal distribution. This assumption proves to be valid for all estimation methods and flights except for tilt estimations which are not computed for hover flights. Hence the standard deviation in these cases should be taken with caution.

4.2.2 Assessment of ground truth quality

As discussed in Section 4.1.1, due to wind turbulences, the reference sensors cannot measure the wind at the drone's position. To estimate the magnitude of the ground truth error the aforementioned performance figures are also computed between two anemometers. \mathbf{w} can be replaced by $\tilde{\mathbf{w}}_{14.7m}^1(t)$ and $\hat{\mathbf{w}}$ by $\tilde{\mathbf{w}}_{21.3m}^1(t)$ in the equations from (71) to (76) to compute the ground truth bias and standard deviation.

4.2.3 Filtered Performance

The performance of filtered data will also be evaluated. The chosen filter is a low-pass finite impulse response filter with a cutoff frequency of 0.1 [Hz] and of order 50. This filter was applied twice on the data of interest, once in the forward direction and once backward. This results in a zero-phase filter with a squared magnitude response and a doubled filter order. The magnitude response of the filter can be seen in Figure 17.

4.3 Sample flights

This section will show the performance of three arbitrarily chosen flights from the acquired data set. One *hover* flight depicted in Figure 18 and of which the performance data can be seen in Table 12. One *square* flight, Figure 19 and Table 13. And one *vertical* flight, Figure 20 and Table 14. Note that no flight of type "cstSpeed" was selected here, since they show very similar performance to square flights at high flight speeds and to hover flights at low speed. Hence, they were omitted in favour of conciseness. In spite of the discussion in Section 4.2, the figures show the wind vector decomposed as horizontal wind speed (top plot), horizontal wind direction (center plot), vertical wind (bottom plot). This allows for a more instinctive understanding of the wind and estimation behaviour. On each plot the thick blue curve represents the ground-truth data acquired from the external anemometer and the other curves represent the data resulting from each estimation method. In the performance tables, it is worth noting that for the horizontal wind, the norm of the bias is shown and not the bias itself. Let's consider each of these three chosen flights individually.

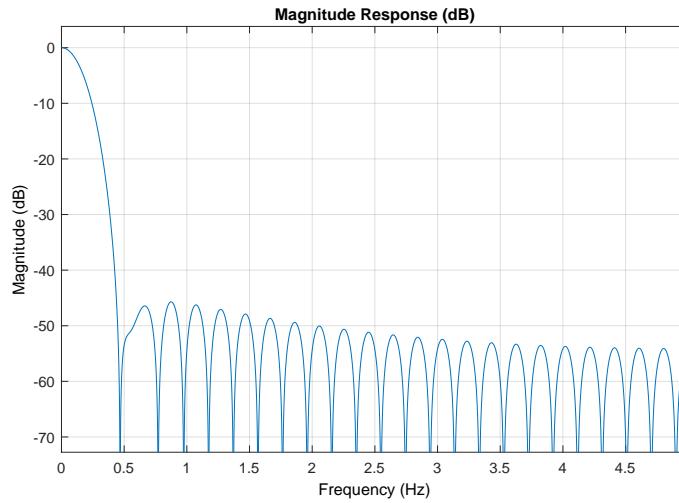


Figure 17: Filter magnitude response. Cutoff frequency : 0.1 [Hz]. Filter order 50.

4.3.1 Hover

	Horizontal Wind				Vertical Wind			
	Not Filtered Bias	Not Filtered Std	Lowpass Filtered Bias	Lowpass Filtered Std	Not Filtered Bias	Not Filtered Std	Lowpass Filtered Bias	Lowpass Filtered Std
DN, Linear and No Vertical Drag	0.46	1.08	0.46	0.84	0.04	0.50	0.04	0.44
DN, Linear and Vertical Drag	0.29	1.07	0.29	0.82	1.53	0.82	1.53	0.68
DN, Quadratic and No Vertical Drag	1.90	1.78	1.89	1.25	0.04	0.50	0.04	0.44
DN, Quadratic and Vertical Drag	1.42	1.75	1.42	1.17	2.25	0.98	2.25	0.79
Tilt	0.15	0.93	0.15	0.70	0.04	0.50	0.04	0.44

Table 12: Performance figures for sample *hover* flight shown in Figure 18.

The hover flight type is important to consider since it is the only flight type where the assumptions underlying the *Tilt* estimation method are respected. Thus in some sense it is the only flight type where the tilt estimation method has a "fair" comparison. On the other hand, any new estimation method should ideally perform at least as well as the tilt method in this simpler case. Note that the data plotted in Figure 18 shows only three minutes of the total flight. This was done to make this plot somewhat less dense.

Horizontal wind speed Starting with the horizontal wind magnitude and looking at the tilt estimation (light blue curve), it can be seen that it produces a good approximation of the actual horizontal wind. Variations in the wind speed seem to be reflected in the estimation. An exception to this could be the small wind gust present around 09:33. In the reference data this gust crosses the 2 [m/s] mark at 09:32:59 and in the tilt estimation at 09:32:54, this is a difference of 5 seconds. Knowing that the drone flies approximately 10 [m] to the south of the wind reference and that the wind is coming from the south, it seems reasonable to think that the cause of this delay is simply the propagation time of this wind gust and not a data synchronisation issue. Now looking the new estimation methods, they can be grouped in pairs: the two methods using the linear drag model perform very similarly and are also a good estimation of the true wind speed; and the two methods using the quadratic drag model perform also very similarly, but seem to feature a important bias. This highlights two important observations: first, assuming that the vertical wind speed is zero seems to have only little impact on the estimation, this will be discussed in a general way in Section 5.1.1; second, the choice of drag model seems to have a significant impact on the estimation, this will be discussed in a general way in Section 5.1.2.

Horizontal wind direction Now observing the wind direction on the middle plot, it can be seen that all five estimation methods perform very similarly and give a good estimation of the wind direction. The

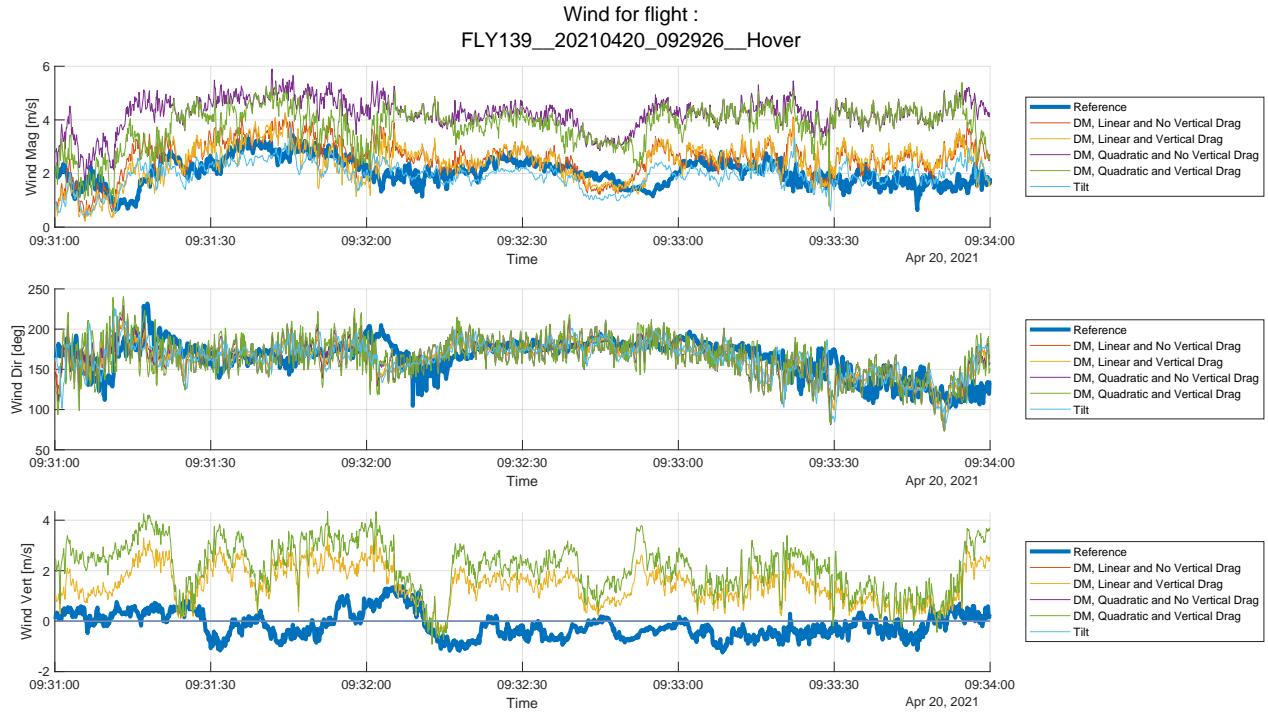


Figure 18: Wind estimation for sample *hovering* flight. Thick blue line represents the ground truth and other lines represent tested estimation methods.

only slight difference concerns the two methods using the quadratic drag model, which have a slightly bigger variance. There seems to be a "directional wind gust" around 09:20, which results in a timing miss-match between estimation and measurement. But, as for wind speed, it seems fair to assume that this is due to gust propagation time.

Vertical wind Finally, looking at the vertical wind speed, it can first be seen that even if the vertical wind speed is not zero, it evolves close to the 0 [m/s] per second mark and thus assuming that there is no vertical wind speed seems to be a fair assumption. And as a matter of fact, the methods trying to estimate vertical wind have a worse performance than simply assuming the vertical wind is null.

Performance data The performance data shown in 12 reflect the visual observations made above in a numerical way. Additionally, the filtered data shows an improved performance of approximately 20 [%] on the standard deviation, as could be expected from filtering.

4.3.2 Square

	Horizontal Wind				Vertical Wind			
	Not Filtered		Lowpass Filtered		Not Filtered		Lowpass Filtered	
	Bias	Std	Bias	Std	Bias	Std	Bias	Std
DN, Linear and No Vertical Drag	0.93	1.84	0.90	1.32	0.46	0.54	0.46	0.46
DN, Linear and Vertical Drag	0.69	1.68	0.68	1.09	1.42	1.55	1.40	0.92
DN, Quadratic and No Vertical Drag	1.51	2.47	1.50	1.64	0.46	0.54	0.46	0.46
DN, Quadratic and Vertical Drag	1.14	2.31	1.14	1.44	1.90	1.42	1.87	0.80
Tilt	0.40	7.94	0.39	4.34	0.46	0.54	0.46	0.46

Table 13: Performance figures for sample *square* flight shown in Figure 19.

The square flight is interesting due to its very high dynamics: acceleration are high when reaching or leaving

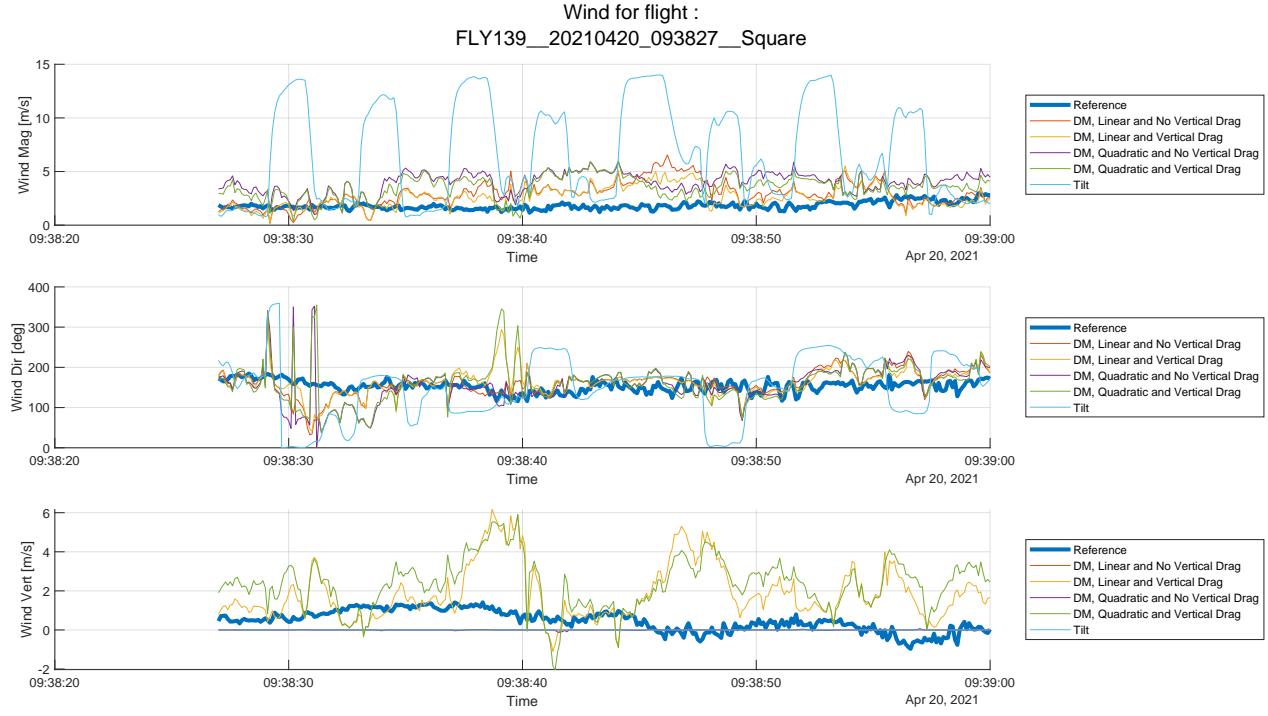


Figure 19: Wind estimation for sample *square* flight. Thick blue line represents the ground truth and other lines represent tested estimation methods.

the corner of the square and thus induces tilting not related to drag force. The first obvious observation to be made about the estimations seen on Figure 19, is that the tilt estimation is failing at estimating wind speed and direction in such a flight scenario, which was expected. The error peaks can be traced back to the moments of drone acceleration or deceleration. On the other hand the dynamic-model-based methods, seem to be able to filter out the impact of the vehicle acceleration. Even if it seems to come at the cost of some accuracy, as can be seen by comparing the performance figures in Table 12 and 13. Wind direction seems to be estimated correctly most of the time by the dynamic-mode-based methods, except for some occurrences where the wind direction estimation is completely wrong, such as around 09:38:30 or 09:38:40. It is unclear why this is happening. Concerning vertical wind, the same observations can be made as for the hover flight.

4.3.3 Vertical

	Horizontal Wind				Vertical Wind			
	Not Filtered Bias	Lowpass Filtered Std						
DN, Linear and No Vertical Drag	0.41	1.78	0.40	0.92	0.05	0.36	0.06	0.29
DN, Linear and Vertical Drag	0.58	1.64	0.55	0.80	0.00	1.54	0.06	0.90
DN, Quadratic and No Vertical Drag	0.96	3.12	1.01	1.65	0.05	0.36	0.06	0.29
DN, Quadratic and Vertical Drag	0.50	2.78	0.54	1.37	0.16	2.03	0.22	0.98
Tilt	0.48	1.46	0.34	1.32	0.05	0.36	0.06	0.29

Table 14: Performance figures for sample *vertical* flight shown in Figure 20.

The vertical flight is interesting for its application in shear wind estimation (see Section 5.3). Here the most important observation to make is that there are three phases where the estimation features a lot of error, namely around 14:39:20, around 14:39:40 and around 14:39:55. These correspond to the moments where the drone is descending. Moving downward is notoriously difficult for a drone since it flies inside the downwash produced by its propellers. Hence this downward flight is very unstable and probably results in bad attitude

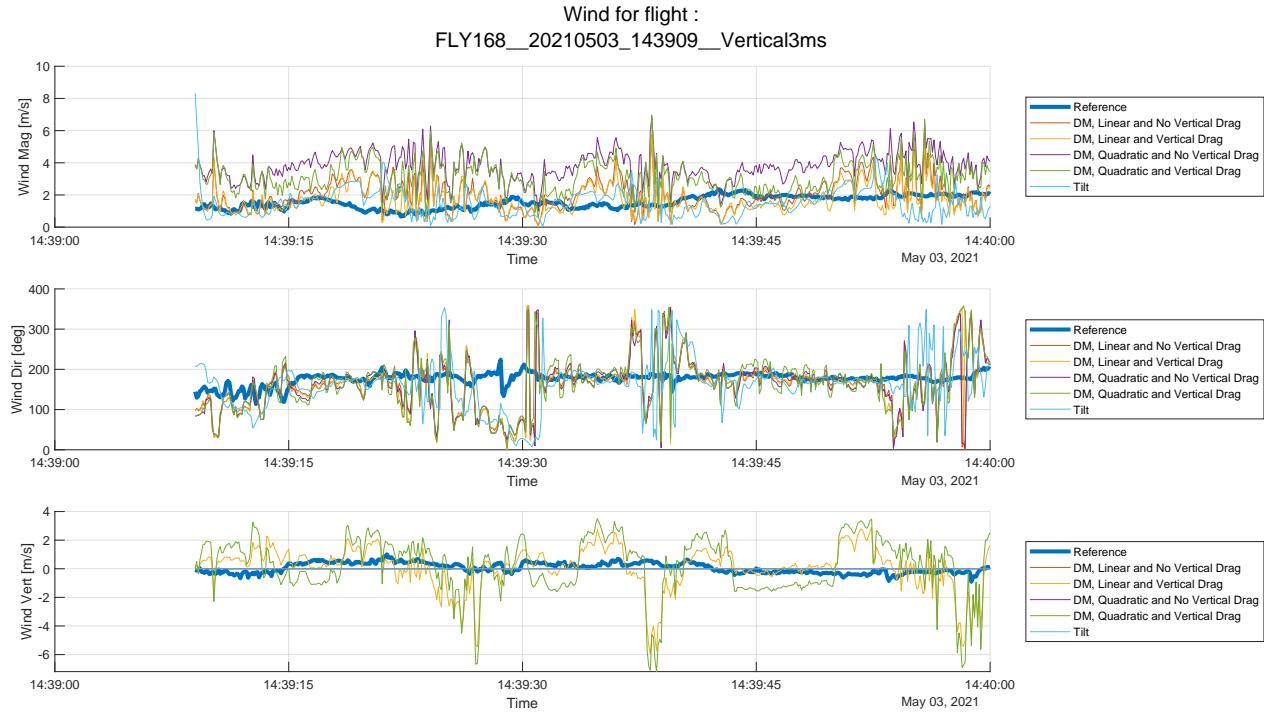


Figure 20: Wind estimation for sample *vertical* flight. Thick blue line represents the ground truth and other lines represent tested estimation methods.

measurements, leading to poor wind estimations. However estimation during the ascending phase seem to be good, in particular for tilt and linear drag dynamic model estimations.

4.4 Performance over wind speed

This section will focus on the estimation performance at various wind speeds. To do so, the figures mentioned hereafter will plot, for a given flight subset, bias and standard deviation with respect to mean wind speed over a flight (blue). This will be done for each estimation method. Additionally, bias and standard deviation of ground truth as defined in Section 4.1.1 is also plotted (orange). Figure 21 shows the performance of *hover* flights over wind speed, Figure 22 the performance of *square* flights, Figure 23 the performance of *vertical flights* and Figure 24 the performance of all flights. Additionally, Figure 25 show the performance over all flight of vertical wind estimation. In the plots, each circle on the continuous blue line represents the bias of a given flight and the dashed line 1σ deviation. Even if some patterns can be seen studying the dependency on wind speed, it should be kept in mind that the used dataset is small, especially when considering only a given flight type. For example, there are only 6 hover flights, spread between 0 and 6 [m/s] mean wind, and 5 of the flights are below 3 [m/s] wind. Hence, generalizing from these observations should be done with caution and, ideally, more data should be acquired. But let's nonetheless analyse the performance of the selected flight subsets.

Hover flights By observing Figure 21, the same trend can be observed as in Section 4.3, namely tilt performs better than the two linear-drag model-based methods, which perform better than the quadratic-drag model-based methods. However this difference seems to fade out at higher wind speeds. But in these hover flights tilt seems to result in the least error across all wind speeds.

Square flights Again, Figure 22 shows similar results as in Section 4.3. The tilt method is not able to estimate wind during a square flight regardless of the wind speed. And as for hover flights, linear- performs better than quadratic-drag model methods at low wind speeds and equally at higher wind speeds.

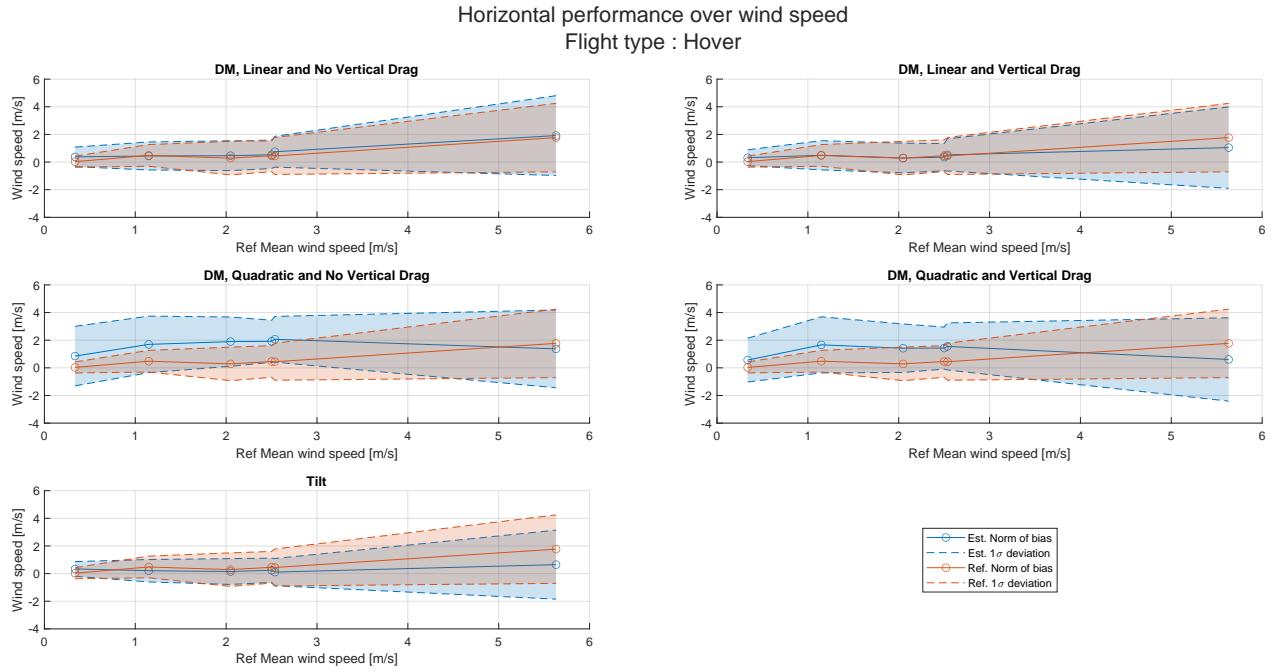


Figure 21: Performance of hover flights over mean wind speed.

Vertical flights Here also the vertical flights shown in Figure 23 correspond to the observations made in Section 4.3. However it is worth noting that looking in Figure 23, standard deviation seems to vary a lot from flight to flight also independently from mean wind speed. But this figure includes all different cruising speeds (see Section 3.1.3) and the faster the cruising speed, the more difficult the descending flight through the downwash becomes (see discussion about downwash in Section 4.3) and thus the bigger the error gets.

All flights Figure 24 shows the performance of all flights contained in the dataset and thus can give a feeling of estimation performance during a flight following an arbitrary path. Looking at Figure 25, it is easy to see that assuming no vertical wind is a good estimation, whereas the attempts to estimate vertical drag lead to poor results with big biases and deviations.

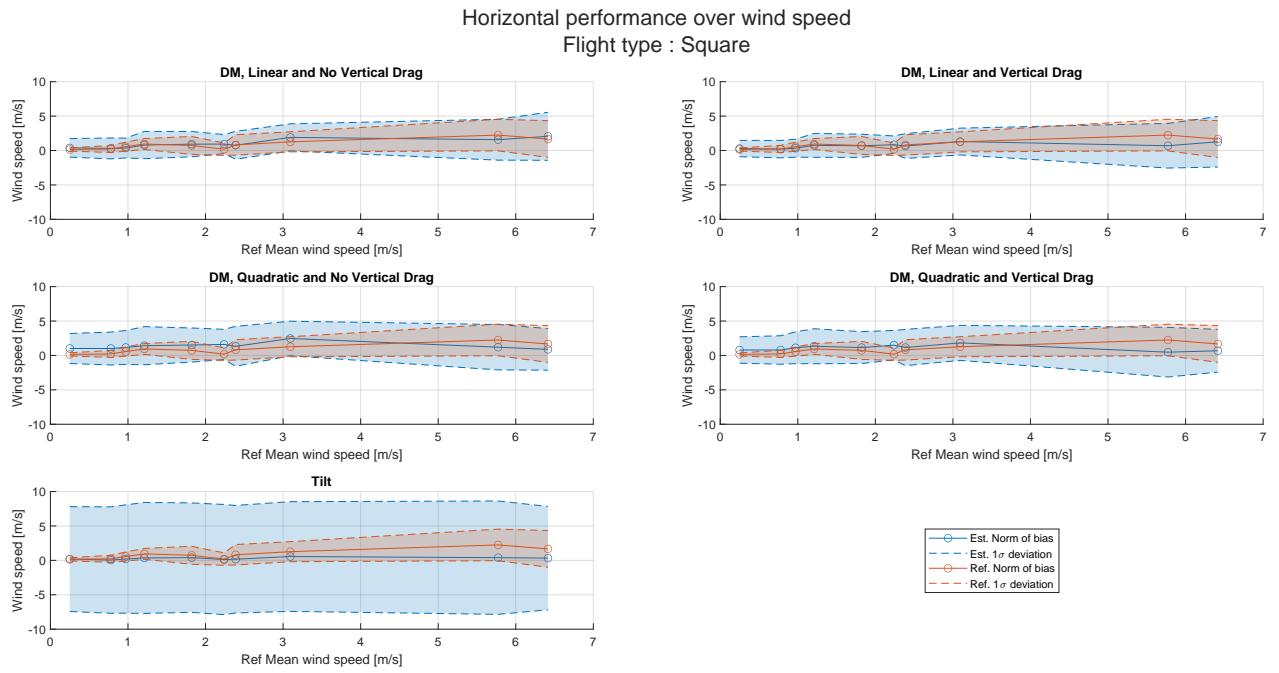


Figure 22: Performance of square flights over mean wind speed.

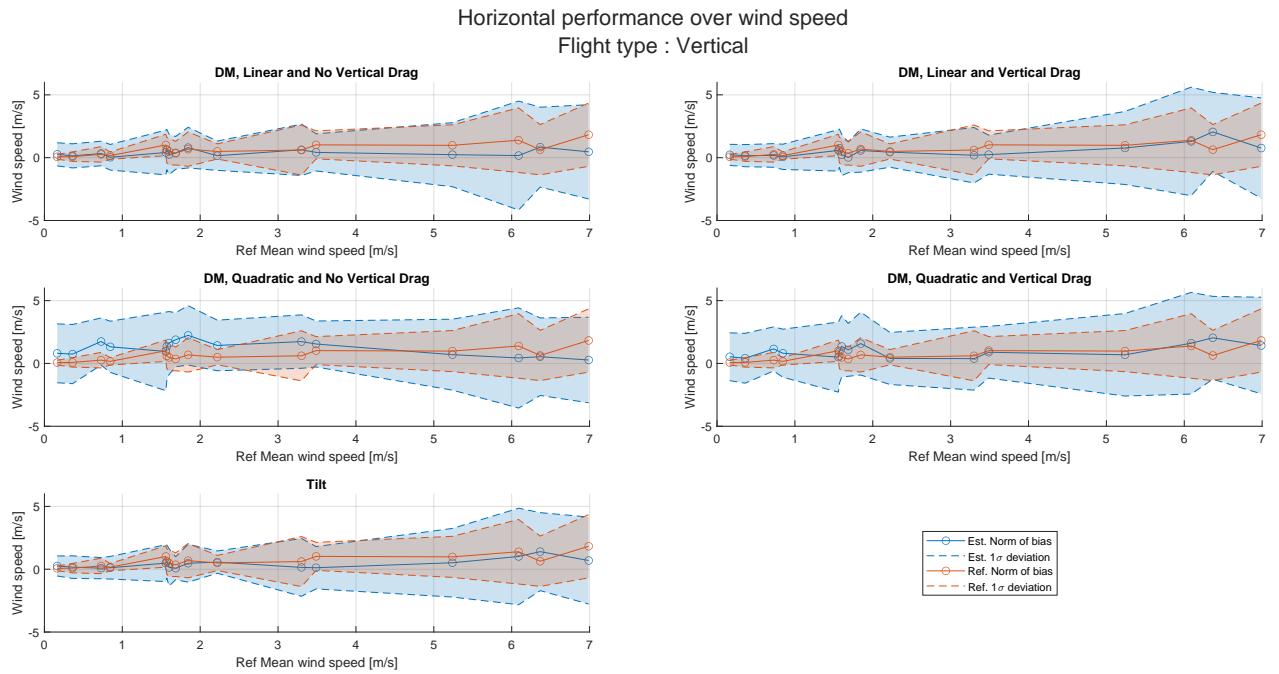


Figure 23: Performance of vertical flights over mean wind speed.

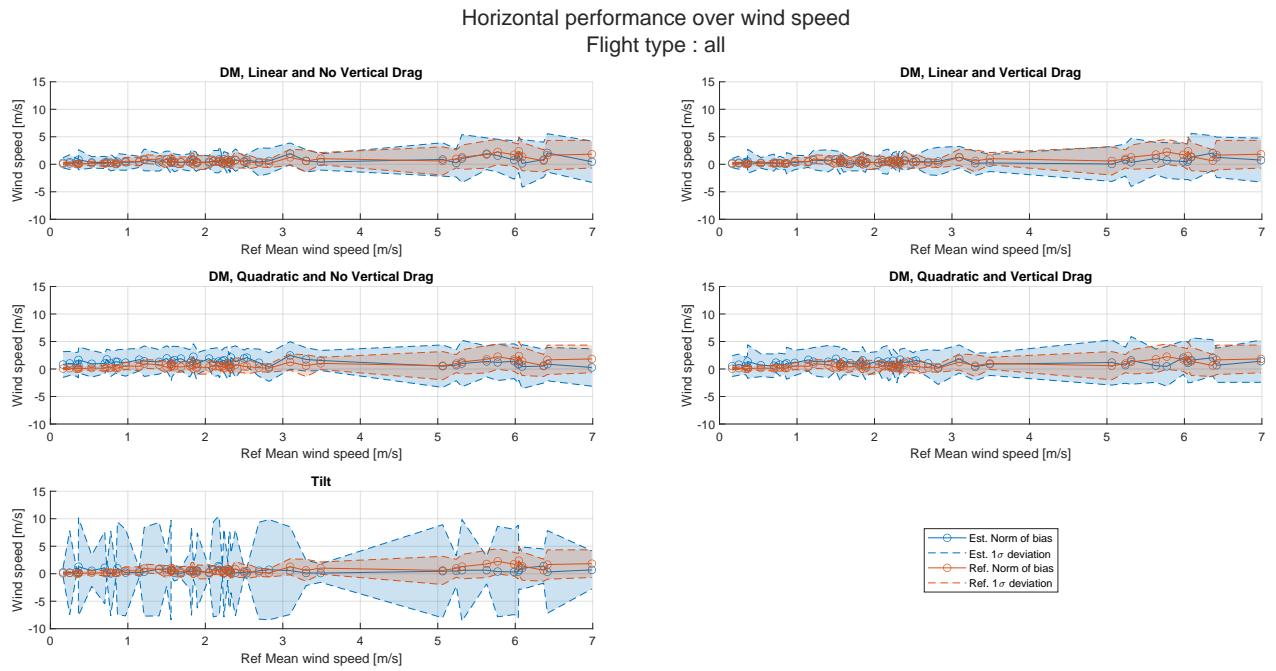


Figure 24: Performance for all flights over mean wind speed.

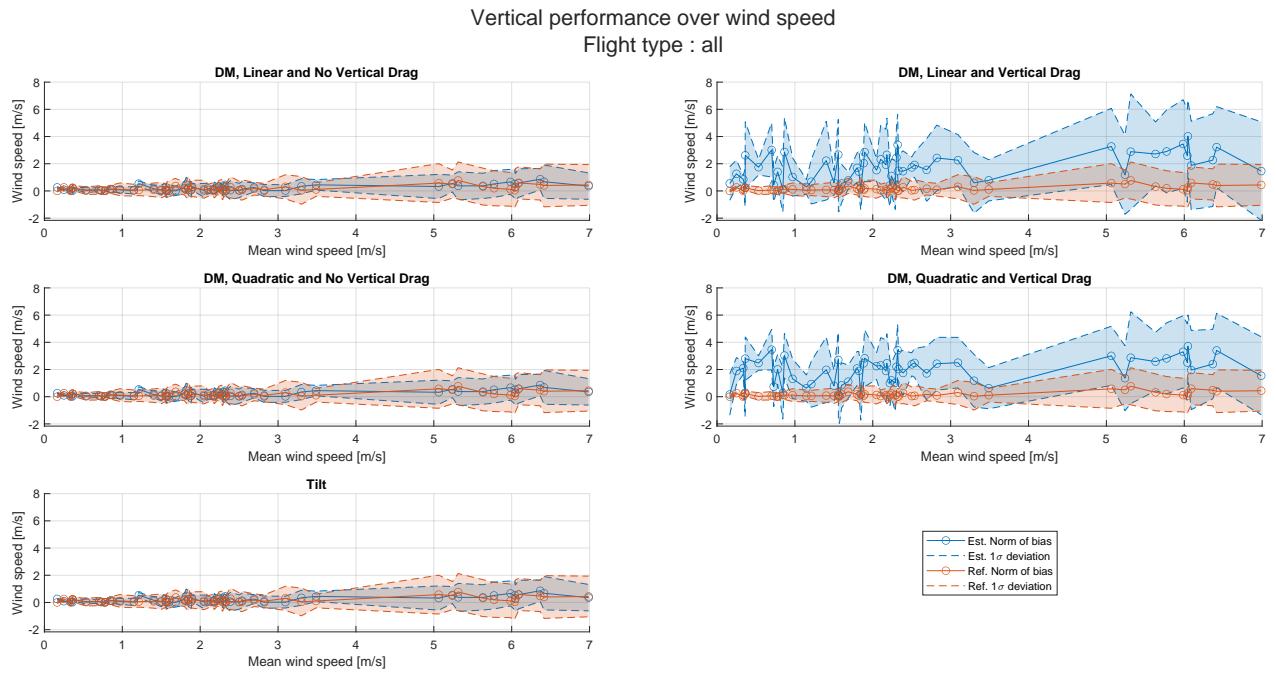


Figure 25: Performance for all flights over mean wind speed, for vertical wind measurements.

5 Discussion

5.1 Learning outcomes

5.1.1 Impact of vertical wind estimation

As presented in Section 1.2, vertical wind is rarely considered in the literature, either for simplicity or to remove the need of thrust estimations. Two important observations can be made based on the results presented in Section 4. First, the performance at estimating horizontal wind of the methods assuming no vertical wind are not different from the one estimating vertical wind. This is likely to be the result of the vertical wind being actually close to zero in all the flights in the dataset. Thus estimating thrust using the wind tunnel data leads to similar results as computing it from Equation (50). However, before choosing one approach or the other, it would be relevant, on one hand, to compare the performance of both approaches on data featuring significant vertical winds; and on the other hand, to investigate, how often no vertical winds are to be expected, with respect to environmental conditions, position and also altitude. Second, for methods estimating vertical wind, the estimation is poor. The error is smaller by assuming the value of vertical wind is zero. This is surprising since air speed is computed in the tilt frame and thus the estimation should not be different vertically than horizontally, so this may point toward an implementation error.

5.1.2 Linear or quadratic drag model

Two drag models were explored during this work: a quadratic model, presented in 3.3.5 and a linear model, presented in 3.3.6. The quadratic model is well tested for spherical objects and widely used also to model drag forces on quadcopters [28]. However, in this work, for low wind speeds, using a linear drag model leads to better wind speed estimations. This is a surprise since the publications presented in Section 1.2 that use a physical model for drag, all use a quadratic drag. But, this is not a new idea in quadcopter modeling: for example [28] analytically derives using Blade Element Momentum (BEM) theory and experimentally validates a linear drag relation. Plus, the well-known Gazebo simulator also implements a linear drag model for quadcopters [29]. According to this model the total drag force is dominated by drag generated by air passing through the rotors in the xy-body-plane, i.e. perpendicularly to the thrust direction. More explicitly the drag model for a single rotor is given by:

$$\mathbf{F}_D^b = \eta K_{BEM} \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix} \quad (77)$$

This is not the linear model implemented in this work, but both model are equivalent under the assumptions that the rotor angular rates are constant and the drone is not tilting. Unfortunately this BEM-based model could not be implemented due to lack of time. Note also that BEM-based approaches start to fail at high air speeds and other approaches are needed to correct drag modeling, see for example [30], where BEM is combined with a neural network trained to estimate drag residuals BEM theory is not able to predict.

5.1.3 Ground truth quality

The ground truth quality was assessed by computing the error present on two sensors on the weather mast separated by 7 [m] vertically (the drone being at least 10 [m] or more away from the sensors). As could be seen in Section 4.4, the bias and deviation between the two sensors is important, leading to the conclusion that turbulences of the air in the flight zone was underestimated. This is a hard limit to the development of new estimation methods, because it prevents the evaluation of their performance beyond this limit. It is essential to find some alternative reference measurements, providing readings closer to wind experienced by the air-craft. This is an open question, but here are two possible approaches to explore. First, only consider time averaged data over long period of times (in the order of several minutes): this approach is easy to implement and should remove most of the high frequency variance (but this needs to be investigated); however this comes at the cost of reducing the estimation's bandwidth. Second, perform indoor flights where there is no wind: in this scenario the air speed is equal to the vehicle speed, thus the method can be validated by verifying how well it estimates vehicle speed for which a ground truth can be provided by a motion capture

system. This approach is similar to what is done in [30]. However, this solution must assume that the effect of travelling at a given speed in still air is the same as standing still in a wind of the same speed, which might not be true.

5.2 Method trade-off

This section will list pros and cons in the usage of each method. The estimation methods using the quadratic drag model are not included in this list since they are always out-performed by the other methods and do not have any benefit over the linear drag methods. However they should not be discarded yet since behaviour at higher wind speeds, were they some to perform at least as well as the linear methods, could not be observed in this dataset.

- **DM, Linear and No Vertical Drag**

- Pros:
 - * Most precise and accurate during dynamic flights, thus enabling continuous profiling
 - * Does not need to estimate thrust
 - * Does only rely on the autopilot's pose estimation not on its control
- Cons:
 - * Will not estimate vertical wind and vertical wind may impact estimation accuracy
 - * Less precise than tilt in hovering conditions
 - * Needs wind tunnel data to compute drag coefficients

- **DM, Linear and Vertical Drag**

- Pros:
 - * Most precise and accurate during dynamic flights, thus enabling continuous profiling
 - * Does only rely on the autopilot's pose estimation not on its control
 - * (attempts to estimate vertical wind, but results are poor)
- Cons:
 - * Less precise than tilt in hovering conditions
 - * Needs wind tunnel data to compute drag coefficients

- **Tilt**

- Pros:
 - * Most accurate and precise during hover flights.
 - * Simple to describe and implement.
 - * simple to extrapolate to other platforms, provided calibration flights are possible.
- Cons:
 - * Will only work on hover and slow ascending flights
 - * Will not estimate vertical wind and vertical wind may impact estimation accuracy
 - * Depends on the autopilot control performance

5.3 Possible applicability to meteorological research

This section will focus on one specific application, namely the measurement of shear wind profiles in order to determine aerodynamic roughness lengths, since this was one of the original motivations behind this project (see [19]). But, obviously wind estimation using UAVs could have many other use cases. This application will only be considered from a theoretical point of view and thus was not validated experimentally. But this discussion could serve as a starting point for developing this application.

5.3.1 Estimating the aerodynamic roughness length

A shear wind profile displays wind speeds with respect to height, under neutral stability the wind profile follows a logarithmic law [31]:

$$w_h(z) = \left\| \begin{bmatrix} w_n(z) \\ w_e(z) \\ 0 \end{bmatrix} \right\| = \frac{w_*}{\kappa} \ln \frac{z}{z_0} \quad (78)$$

Where $w_h(z)$ is the norm of the horizontal wind speed at height z , z is the height above the ground and thus equals $-x_d$ provided the origin of the local-level frame in on the ground, w_* is the friction velocity, κ is the von Karman constant and equals 0.41 (unit-less) and z_0 is the aerodynamic roughness length. Figure 26 shows a sample shear wind profile. The quantity of interests in this equation are the friction velocity and the aerodynamic roughness length. Hence using a UAV and an estimation method presented above wind estimations at different heights can be collected. Let's call the set of wind observations \mathbf{w}_h and the set of corresponding heights \mathbf{z} , both stored as a column vector. To compute the aerodynamic roughness length a linear regression can be used. So let's rewrite Equation (78):

$$\mathbf{w}_h = \frac{w_*}{\kappa} \ln(\mathbf{z}) - \frac{w_*}{\kappa} \ln(z_0) \quad (79)$$

This can be written as a over-determined linear system of the form:

$$\mathbf{y} = [\mathbf{x} \quad \mathbf{1}] \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix} = [\mathbf{x} \quad \mathbf{1}] \boldsymbol{\beta} \quad (80)$$

Where $\mathbf{y} = \mathbf{w}_h$, $\mathbf{x} = \ln(\mathbf{z})$, $\mathbf{1}$ is column vector composed of ones of the same size as \mathbf{x} , $\beta_1 = \frac{w_*}{\kappa}$ and $\beta_0 = -\frac{w_*}{\kappa} \ln(z_0)$. Performing a linear regression $\boldsymbol{\beta}$ can be solved for and friction velocity can thus be expressed as:

$$w_* = \beta_0 \kappa \quad (81)$$

And aerodynamic roughness length as:

$$z_0 = \exp(-\frac{\beta_0}{\beta_1}) \quad (82)$$

Thanks to the linear regression, noise present on the wind and height measurements are likely to be filtered out and thus could result in an accurate estimation of the aerodynamic roughness length.

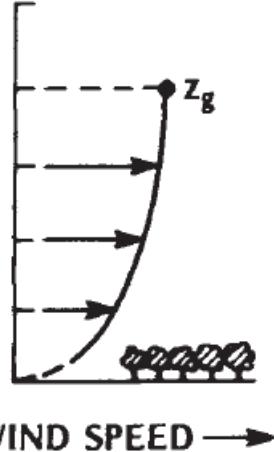


Figure 26: Logarithmic shear wind profile. Reproduced from [31].

5.3.2 Open questions

The derivation present in the previous section describe how to estimate the aerodynamic roughness length, but two important questions need to be answered before being able to test this method. First, what is the optimal flight plan to do this estimation. Obviously an ascending (not descending due to downwash turbulences) path should be taken, but what is the ideal cruising speed? Since there is a compromise between being fast enough to measure the whole wind profile before it changes and being slow enough to get the most accuracy out of the estimation method. The second question to answer is how to validate this estimation, since shear wind profiles are not easy to estimate.

6 Conclusion and future work

To sum up, to main contribution of this work are:

- To propose a novel estimation scheme able to compete with the tilt approach described in [19] in hovering conditions and at the same time capable to handle dynamic flights.
- To discuss the assumption that there are no vertical wind and to assess the impact of this assumption on horizontal wind estimation.
- To highlight the need to model drag based on BEM theory, which leads to better wind estimation compared to drag simply proportional to the air speed squared.

6.1 Future work

There are still a lot to be explored and tested. This section will list some paths that could lead to better wind estimations or address some limitations of UAV based wind estimations. Some items where already discussed but are repeated here for completeness.

- The sample rate of the data retrieved from the DatCon software is 10 [Hz], however higher sample rates can be asked for. Thus, assuming a 10 [Hz] wind estimation should be kept, it could be possible to improve wind estimation by using a higher sample rate and average or filter the data, as it has been shown to improve estimation precision.
- The dataset could be improved by performing more flights, in particular at higher wind speed and under environment conditions, see Section 4.1. Data from [19] could also be included.
- The ground truth quality could be improved following up on the discussion in Section 5.1.3.
- The dynamic model used in this work focused only on forces and not moments. However according to [28] air flow also generates a drag moment. This could be used in the same way as the drag force in this work to estimate wind and thus could contribute to improve the wind estimation.
- Several interaction with the doctoral work of K. Joseph Paul, here at TOPO, are possible. His doctoral work follows up on [32]. This work aims at improving quadcopter pose estimation using a dynamic model of the vehicle (VDM). In particular, the wind estimation of this work could serve as an input to the VDM sensor or the VDM wind estimate could be compared to the estimation achieved in this work.
- TOPO also has a custom build UAV which feature thrust sensors (force sensors placed at the propeller base). This could be used to validate the thrust model used in this work or to directly perform wind estimation without having to rely on a thrust model or having to make the no vertical wind assumption.
- Since rotor based UAVs cannot probe wind for longer than half an hour, the possibility to perform autonomous periodic flights could be assessed. This would allow for the generation of time series that could span an arbitrary long time period, provided charging is possible between flights. (Note this would require beyond line of sight operation, which makes the operation category of the drone from *open to specific* according to [EASA eRules](#)).

6.2 Final thoughts

This thesis was a very rich, intense and inspiring experience. It allowed me to get more familiar with the academic and research world and thus adds to possible career paths available to me at the end of this work. Despite the fact I was not able to travel to the arctic yet, this project also opened various opportunities related to sensing in the arctic, which, I hope, I will be able to use wisely.

Thank you for reading this thesis.

02.07.2021

Kilian Meier

A handwritten signature in blue ink, appearing to read "Kilian Meier".

References

- [1] X. Lee, *Fundamentals of Boundary-Layer Meteorology*, en, ser. Springer Atmospheric Sciences. Cham: Springer International Publishing, 2018, ISBN: 978-3-319-60853-2. DOI: [10.1007/978-3-319-60853-2](https://doi.org/10.1007/978-3-319-60853-2). [Online]. Available: <http://link.springer.com/10.1007/978-3-319-60853-2> (visited on 03/17/2021).
- [2] Haiyang Chao and YangQuan Chen, “Surface wind profile measurement using multiple small unmanned aerial vehicles,” in *Proceedings of the 2010 American Control Conference*, Baltimore, MD: IEEE, Jun. 2010, pp. 4133–4138, ISBN: 978-1-4244-7425-7. DOI: [10.1109/ACC.2010.5530609](https://doi.org/10.1109/ACC.2010.5530609). [Online]. Available: <http://ieeexplore.ieee.org/document/5530609/> (visited on 06/21/2021).
- [3] A. Dachauer, R. Hann, and A. J. Hodson, “Aerodynamic roughness length of crevassed tidewater glaciers from UAV mapping,” *Glaciers/Remote Sensing*, preprint, May 2021. DOI: [10.5194/tc-2021-113](https://doi.org/10.5194/tc-2021-113). [Online]. Available: <https://tc.copernicus.org/preprints/tc-2021-113/> (visited on 06/21/2021).
- [4] S. Emeis, *Surface-Based Remote Sensing of the Atmospheric Boundary Layer*, ser. Atmospheric and Oceanographic Sciences Library. Dordrecht: Springer Netherlands, 2011, vol. 40, ISBN: 978-90-481-9340-0. DOI: [10.1007/978-90-481-9340-0](https://doi.org/10.1007/978-90-481-9340-0). [Online]. Available: <http://link.springer.com/10.1007/978-90-481-9340-0> (visited on 06/21/2021).
- [5] S. Prudden, A. Fisher, M. Marino, A. Mohamed, S. Watkins, and G. Wild, “Measuring wind with Small Unmanned Aircraft Systems,” en, *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 176, pp. 197–210, May 2018, ISSN: 01676105. DOI: [10.1016/j.jweia.2018.03.029](https://doi.org/10.1016/j.jweia.2018.03.029). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S016761051730942X> (visited on 06/21/2021).
- [6] R. Hann, A. Hodson, and M. Jonassen, “Glacier mapping and wind estimation with uavs on svalbard,” Dec. 2019.
- [7] “Phantom 4 RTK, User Manual,” DJI, Tech. Rep., Jul. 2020. [Online]. Available: https://dl.djicdn.com/downloads/phantom_4_rtk/20200721/Phantom_4_RTK_User_Manual_v2.4_EN.pdf.
- [8] P. P. Neumann, “Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit,” en, p. 11, 2015.
- [9] R. T. Palomaki, N. T. Rose, M. van den Bossche, T. J. Sherman, and S. F. J. De Wekker, “Wind Estimation in the Lower Atmosphere Using Multirotor Aircraft,” en, *Journal of Atmospheric and Oceanic Technology*, vol. 34, no. 5, pp. 1183–1191, May 2017, ISSN: 0739-0572, 1520-0426. DOI: [10.1175/JTECH-D-16-0177.1](https://doi.org/10.1175/JTECH-D-16-0177.1). [Online]. Available: <https://journals.ametsoc.org/view/journals/atot/34/5/jtech-d-16-0177.1.xml> (visited on 03/17/2021).
- [10] Y. Song, B. Luo, and Q.-H. Meng, “A rotor-aerodynamics-based wind estimation method using a quadrotor,” *Measurement Science and Technology*, vol. 29, no. 2, p. 025801, Feb. 2018, ISSN: 0957-0233, 1361-6501. DOI: [10.1088/1361-6501/aa8a9d](https://doi.org/10.1088/1361-6501/aa8a9d). [Online]. Available: <https://iopscience.iop.org/article/10.1088/1361-6501/aa8a9d> (visited on 06/22/2021).
- [11] Z. Xing, Y. Zhang, C.-Y. Su, Y. Qu, and Z. Yu, “Kalman Filter-based Wind Estimation for Forest Fire Monitoring with a Quadrotor UAV,” en, in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, Hong Kong, China: IEEE, Aug. 2019, pp. 783–788, ISBN: 978-1-72812-767-5. DOI: [10.1109/CCTA.2019.8920637](https://doi.org/10.1109/CCTA.2019.8920637). [Online]. Available: <http://ieeexplore.ieee.org/document/8920637/> (visited on 03/17/2021).
- [12] L. Wang, G. Misra, and X. Bai, “A K Nearest Neighborhood-Based Wind Estimation for Rotary-Wing VTOL UAVs,” en, *Drones*, vol. 3, no. 2, p. 31, Mar. 2019, ISSN: 2504-446X. DOI: [10.3390/drones3020031](https://doi.org/10.3390/drones3020031). [Online]. Available: <https://www.mdpi.com/2504-446X/3/2/31> (visited on 06/22/2021).
- [13] G. Perozzi, D. Efimov, J.-M. Biannic, and L. Planckaert, “Using a quadrotor as wind sensor: Time-varying parameter estimation algorithms,” en, *International Journal of Control*, pp. 1–12, Jun. 2020, ISSN: 0020-7179, 1366-5820. DOI: [10.1080/00207179.2020.1780324](https://doi.org/10.1080/00207179.2020.1780324). [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/00207179.2020.1780324> (visited on 06/22/2021).

- [14] Y. Qu, K. Wang, and X. Wu, "Wind estimation with uavs using improved adaptive kalman filter," in *2019 Chinese Control And Decision Conference (CCDC)*, IEEE, 2019, pp. 3660–3665.
- [15] P. Abichandani, D. Lobo, G. Ford, D. Bucci, and M. Kam, "Wind Measurement and Simulation Techniques in Multi-Rotor Small Unmanned Aerial Vehicles," en, *IEEE Access*, vol. 8, pp. 54 910–54 927, 2020, ISSN: 2169-3536. doi: [10.1109/ACCESS.2020.2977693](https://doi.org/10.1109/ACCESS.2020.2977693). [Online]. Available: <https://ieeexplore.ieee.org/document/9020170/> (visited on 03/17/2021).
- [16] J. González-Rocha, S. F. J. De Wekker, S. D. Ross, and C. A. Woolsey, "Wind Profiling in the Lower Atmosphere from Wind-Induced Perturbations to Multirotor UAS," en, *Sensors*, vol. 20, no. 5, p. 1341, Feb. 2020, ISSN: 1424-8220. doi: [10.3390/s20051341](https://doi.org/10.3390/s20051341). [Online]. Available: <https://www.mdpi.com/1424-8220/20/5/1341> (visited on 03/17/2021).
- [17] S. Allison, H. Bai, and B. Jayaraman, "Wind estimation using quadcopter motion: A machine learning approach," en, *Aerospace Science and Technology*, vol. 98, p. 105 699, Mar. 2020, ISSN: 12709638. doi: [10.1016/j.ast.2020.105699](https://doi.org/10.1016/j.ast.2020.105699). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1270963819324034> (visited on 06/22/2021).
- [18] G. Loubimov, M. P. Kinzel, and S. Bhattacharya, "Measuring Atmospheric Boundary Layer Profiles Using UAV Control Data," en, in *AIAA Scitech 2020 Forum*, Orlando, FL: American Institute of Aeronautics and Astronautics, Jan. 2020, ISBN: 978-1-62410-595-1. doi: [10.2514/6.2020-1978](https://doi.org/10.2514/6.2020-1978). [Online]. Available: <https://arc.aiaa.org/doi/10.2514/6.2020-1978> (visited on 06/22/2021).
- [19] A. Garreau, "Validation and Application of Novel Wind Estimation Methods with Quadcopter UAVs in the Arctic," en, M.S. thesis, The University Center in Svalbard, École Nationale de la Météorologie, Svalbard, 2020. [Online]. Available: https://folk.ntnu.no/richahan/Publications/2020_Garreau_Thesis.pdf.
- [20] K. de Jong, M. Goode, X. Liu, and M. Stone, "Precise GNSS Positioning in Arctic Regions," in *All Days*, Houston, Texas: OTC, Feb. 2014, OTC-24651-MS. doi: [10.4043/24651-MS](https://doi.org/10.4043/24651-MS). [Online]. Available: <https://onepetro.org/OTCARCTIC/proceedings/14OARC/All-14OARC/Houston,%20Texas/172546> (visited on 06/23/2021).
- [21] C. Russell, J. Jung, G. Willink, and B. Glasner, "Wind Tunnel and Hover Performance Test Results for Multicopter UAS Vehicles," en, p. 20,
- [22] M. Khaghani, "Vehicle Dynamic Model Based Navigation for Small UAVs," en, Ph.D. dissertation, EPFL, 2018.
- [23] Y. Stebler, "Modeling and Processing Approaches for Integrated Inertial Navigation," en, Ph.D. dissertation, EPFL, 2013.
- [24] "WORLD GEODETIC SYSTEM 1984, Its Definition and Relationships with Local Geodetic Systems," en, Department of Defense, NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY (NGA) STANDARDIZATION DOCUMENT, Jul. 2014, p. 207. [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj08Y0R3YDxAhVK_KQKHZ5hC6MQFjABegQIBRAD&url=https%3A%2F%2Fearth-info.nga.mil%2Fphp%2Fdownload.php%3Ffile%3Dcoord-wgs84&usg=A0vVaw2vHA9yXMOT_5xZpyifj0yW.
- [25] "Formule pour la détermination de la masse volumique de l'air humide," fr, Bureau International des Poids et Mesures, Sèvres, France, Tech. Rep. BIPM-81/8, 1981.
- [26] F. Schiano, J. Alonso-Mora, K. Rudin, P. Beardsley, R. Siegwart, and B. Sicilianok, "Towards Estimation and Correction of Wind Effects on a Quadrotor UAV," en, 2014. doi: [10.3929/ETHZ-A-010286793](https://doi.org/10.3929/ETHZ-A-010286793). [Online]. Available: <http://hdl.handle.net/20.500.11850/92099> (visited on 06/08/2021).
- [27] "Windmaster User Manual," en, Gill Instruments Limited, Tech. Rep., Nov. 2019. [Online]. Available: <http://gillstruments.com/data/manuals/1561-PS-0001%20WindMaster%20Windmaster%20Pro%20Manual%20Issue%2013.pdf> (visited on 06/18/2021).
- [28] P. Martin and E. Salaun, "The true role of accelerometer feedback in quadrotor control," in *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK: IEEE, May 2010, pp. 1623–1629, ISBN: 978-1-4244-5038-1. doi: [10.1109/ROBOT.2010.5509980](https://doi.org/10.1109/ROBOT.2010.5509980). [Online]. Available: <https://ieeexplore.ieee.org/document/5509980/> (visited on 06/29/2021).

- [29] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, “RotorS—A Modular Gazebo MAV Simulator Framework,” in *Robot Operating System (ROS)*, A. Koubaa, Ed., vol. 625, Cham: Springer International Publishing, 2016, pp. 595–625, ISBN: 978-3-319-26054-9. DOI: [10.1007/978-3-319-26054-9_23](https://doi.org/10.1007/978-3-319-26054-9_23) [Online]. Available: http://link.springer.com/10.1007/978-3-319-26054-9_23 (visited on 06/29/2021).
- [30] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza, “NeuroBEM: Hybrid Aerodynamic Quadrotor Model,” *arXiv:2106.08015 [cs]*, Jun. 2021. [Online]. Available: [http://arxiv.org/abs/2106.08015](https://arxiv.org/abs/2106.08015) (visited on 06/29/2021).
- [31] T. R. Oke, *Boundary layer climates*, en. 2009, ISBN: 978-0-203-40721-9. [Online]. Available: <https://login.lacollegerlibrary.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&AN=74015> (visited on 03/17/2021).
- [32] K. Joseph Paul, “Introducing vehicle dynamic models in dynamic networks for navigation in uavs,” p. 51, 2019. [Online]. Available: <http://infoscience.epfl.ch/record/276234>.

A DJI Phantom flight data extraction using DatCon

This appendix describes how to extract flight data from a DJI Phantom 4 RTK drone using DatCon. This procedure may apply to other drone with minor adjustments.

A.1 Required Hardware and Software

The following hardware is needed:

- DJI Phantom 4 RTK drone
- USB-A to USB-micro-B cable
- Computer

The following software is needed:

- DJI Assistant 2 For Phantom, used version 2.0.10 ([@ Download DJI Assistant](#))
- DatCon, used version 4.0.5 ([@Download DatCon](#))

A.2 DatCon

DatCon is an open-source software¹² which converts DJI flight logs (.DAT files) to a human readable tabular file (.CSV). This was achieved by reverse engineering the encoding of the DJI flight logs, hence it is not officially supported by DJI. This implies there is no guarantee of the quality of the conversion nor that the DatCon software will be maintained in the future. That being said, there is a fair community of DatCon users and there are regular updates. Figure 27 shows a snapshot of DatCon.

The configuration used to convert logs are the following:

- Preferences ;
 - Check if new version available on startup: NO
 - Load last .DAT file on startup: NO
 - Show units in column heading: NO
 - Smart Time Axis processing: YES
 - Validate Coords: YES
- Signal Groups: ALL SIGNAL GROUPS
- Time Axis:
 - Offset: FLIGHT START
 - Lower: GPS LOCK
 - Upper: MOTOR STOP
- CSV
 - Sample Rate: 10 Hz
 - .CSV: YES
 - Event log: YES

Note that, as can be seen in the logs at bottom of Figure 27, DatCon is not able to recognise the Air Craft (AC) and assumes a bad clock frequency which results in a wrong time axis. This has to be corrected for before using the extracted data. By comparison with the GPS and RTK time it was measured that the DJI internal clock has a frequency of 4687453.408 [Hz].

¹²At least partly. Source code for version 4 could not be found, but version 3 can be found on [GitHub](#)

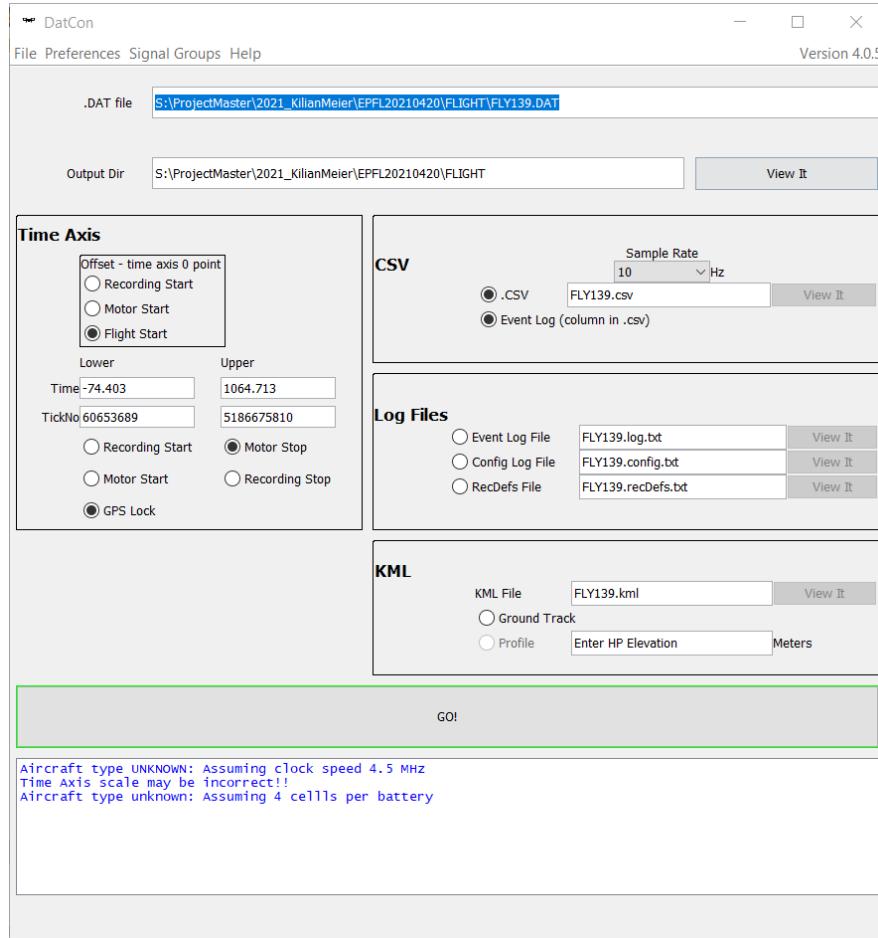


Figure 27: Snapshot of running DatCon software.

A.3 Step-by-step procedure

1. Remove protective covers from camera.
2. Plug the USB cable to drone and computer.
3. Insert battery in the drone and power it up.
4. Run DJI Assistant 2
5. Click on the "Phantom 4 RTK" icon that should appear after a while.
6. Click on the "Flight Data" menu and click "Confirm". This will mount the drone as an external memory storage named "DJI FLY LOG".
7. Run DatCon and configure it as described in Section A.2.
8. Select the desired flight log (.DAT file) in the data storage mounted by the DJI Assistant.
9. Wait until the file pre-analysis is done.
10. Set the output directory.
11. Click "GO!" to perform the data extraction.

B Flight Campaign procedure

A typical flight campaign is described hereafter. The typical total outdoor time is about 1 hour.

1. At least one week before the planned flight, notify EPFL about the intent of flying in accordance with the document called *Drone flight authorization on EPFL campus, Geodetic Engineering Laboratory – TOPO*.
2. At least half a day before the planned flight, check charge level of the TOPO Automatic Weather Station (TOPOAWS), DJI Phantom drone batteries (at least two fully charged batteries are needed) and DJI Phantom RC battery.
3. Needed material :
 - DJI Phantom 4 RTK case, which contains:
 - Drone
 - RC
 - Propellers
 - At least, 2 fully charged drone batteries
 - RC battery
 - TOPOAWS case, which contains at least:
 - Weather station (charged)
 - Anemometer
 - GPS Antenna
 - Compass
 - TOPOAWS Tripod and mast
4. Setup the TOPOAWS at the place marked on Figure 8:
 - (a) Place and level the tripod.
 - (b) Plug the GPS antenna to the weather station.
 - (c) Plug the anemometer to the weather station.
 - (d) Turn-on the weather station.
 - (e) Calibrate the wind direction sensor (see [TOPOAWS Wiki](#), access restricted).
 - (f) Mount the anemometer on the tripod and orient it using the compass.
 - (g) Place the weather station in the shadow to avoid false temperature readings.
 - (h) Check GPS lock occurred.
 - (i) Check data is being recorded.
5. Prepare the drone for flight:
 - (a) Place the drone on its home point, see Figure 8.
 - (b) Plug the battery.
 - (c) Mount the propellers.
 - (d) Remove the camera protection gear.
 - (e) Turn-on the drone (short press and then long press on power button).
6. Prepare RC:
 - (a) Plug the battery.
 - (b) Turn-on the RC (short press and then long press on power button).

- (c) Go to the "fly" menu and wait for the drone to be read to fly.
- 7. Perform flights by going the "flight plan" menu and then "planned" (icon in the left center of the screen). See Section 3.1.3 for a list and description of the flight plans (flight types). **CAUTION** Be aware of safety at all time during flight, in particular flight should not be performed with winds above 30km/h and seagull may want to attack the drone, see Appendix C.
- 8. When asked to change the drone's battery, by turning it off, replacing the battery and turning it on again.
- 9. Once all flights are completed, turn everything off and stow everything in their respective cases.
- 10. Extract drone data according the procedure described in Appendix A.
- 11. Extract TOPOAWS data (the [wsDataPreviewer.m](#) script can be used to preview the data stored in the weather station).
- 12. Kindly ask for Urban microclimate measurement mast (MoTUS) data. Solar Energy and Building Physics Laboratory (LESO) is responsible for the mast and Laurent Deschamps can be contacted.

C Impact of drone flights on wild life

C.1 Context

Twice during the total duration of the flight campaign performed for this work, once at the end of May and once mid June, flights needed to be aborted due to a bird threatening to collide with the drone. It was decided to cancel the flight since the bird was endangering the integrity of the drone, surrounding infrastructure (in case of uncontrolled landing due to propeller failure for example). Plus, the bird would probably suffer sever injuries if caught in the spinning propellers. The bird was identified as a Seagull. Both time the behaviour of the bird was the same. It approached the drone shrieking loudly. Then, flew in a circle a couple meters above the drone. To finally plunge toward the drone and avoid it at the last instant and repeat the whole cycle again until the drone landed. A composite image of one of the plunge toward the drone can be see in Figure 28.



Figure 28: Composite image of Seagull attack. The red circle indicates the drone hovering next to the MoTUS mast. The birds comes in from the top right, flies toward the drone and leaves on the top left of the image.

C.2 Possible explanation of Seagull behaviour

The observed behaviour of the seagull seemed to be aggressive toward the drone. Most birds (and animals in general) behave aggressively for three main reason: for sexual selection, the strongest bird that fought-off all other concurrent gets to mate; for hunting; or for territorial defence. The first hypothesis is unlikely since the mating season of Seagull is around February-March. The second hypothesis is unlikely as well since the bird always avoided the drone and did not try to catch it. The third hypothesis is likely since May-June corresponds to nesting season for Seagulls. Plus, a lot of birds are known to nest on the roofs of EPFL's buildings. However, the Seagull did not attack each time a flight was performed in June. Hence, it is unclear why the bird would defend its nest some times only (the flight paths were always the same).

C.3 Risk mitigation strategy

If the nesting theory is true, the most obvious mitigation strategy, would be to avoid flights during nesting season or in the vicinity of nests. But, better understanding the bird's behaviour might lead to other risk mitigation means. In general it is important to be focused on the flight the drone is performing and to be ready to abort the flight if necessary.

C.4 Conclusion

Birds attacking drones are a known phenomena, but usually observed for fixed-wing drones attacked by birds of pray. Probably because a fixed-wing drone behaves a lot more like a bird. But it is important to be aware

of the risk wildlife is causing to drones, but also, and maybe more importantly, the threat a drone is for wild-life. This is especially true if flights are performed in wild or remote areas, which is one of possibilities offered by UAV based estimation. Hence, I would recommend assessing the drone - wildlife mutual impact prior to executing any flights.

D Assumption List

This appendix groups all the assumptions in this work. In the text they are referenced as (As.x), where x stand for the assumption number.

1. All sensors in the DJI Phantom 4 RTK are perfectly aligned with the body frame.
2. The angular velocity of the earth with respect to the inertial frame Ω_{ie}^e is constant.
3. The angular velocity of the earth with respect to the inertial frame Ω_{ie}^l is zero.
4. The local frame transport rate Ω_{el}^l is zero.
5. The only specific forces acting on the air-craft are thrust f_T and drag f_D .
6. Wind from Tilt : The wind vector is contained in the azimuthal plane, i.e. there is no vertical wind component.
7. Wind from Tilt : The air craft is stationary, i.e. the position over time in the local-level frame is constant.
8. Wind from Dynamic Model: The thrust force of a quadcopter along its z-body-axis is given by:

$$-\rho b(\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2)$$
9. The thrust and drag properties of the DJI Phantom 3 are the same as the thrust and drag properties of the DJI Phantom 4 Pro and DJI Phantom 4 RTK.
10. Wind from Dynamic Model: The DJI Phantom 3, 4 Pro and 4 RTK are cylindrically symmetric around their z-body-axis.

E Dataset Overview Table

DataStream	DataEndTimeString	ID	FOlder	REFLIGHT	REF	REFMETEO	FlightType	Mean windHMag [m/s]
19-Apr-2021 09:31:07	19-Apr-2021 09:41:52	001	S:\ProjectMaster\2021_KilianMeier\EPFL29210419	FLY134	2021-04-19.Mesures.MoTUS	18-Apr-2021	Hover	2.53
19-Apr-2021 09:46:35	19-Apr-2021 09:48:33	002	S:\ProjectMaster\2021_KilianMeier\EPFL29210419	FLY135	2021-04-19.Mesures.MoTUS	18-Apr-2021	Square	2.28
19-Apr-2021 09:56:25	19-Apr-2021 09:57:58	003	S:\ProjectMaster\2021_KilianMeier\EPFL29210419	FLY135	2021-04-19.Mesures.MoTUS	18-Apr-2021	CstSpeed2m/s	2.2
20-Apr-2021 09:26:42	20-Apr-2021 09:27:32	004	S:\ProjectMaster\2021_KilianMeier\EPFL29210420	FLY139	2021-04-20.Mesures.MoTUS	18-Apr-2021	Vertical	2
20-Apr-2021 09:29:26	20-Apr-2021 09:36:53	005	S:\ProjectMaster\2021_KilianMeier\EPFL29210420	FLY139	2021-04-20.Mesures.MoTUS	18-Apr-2021	Hover	2.07
20-Apr-2021 09:38:27	20-Apr-2021 09:39:06	006	S:\ProjectMaster\2021_KilianMeier\EPFL29210420	FLY139	2021-04-20.Mesures.MoTUS	18-Apr-2021	Square	2.32
20-Apr-2021 09:41:59	20-Apr-2021 09:42:44	007	S:\ProjectMaster\2021_KilianMeier\EPFL29210420	FLY140	2021-04-20.Mesures.MoTUS	18-Apr-2021	CstSpeed2m/s	2.78
20-Apr-2021 09:44:04	20-Apr-2021 09:44:50	008	S:\ProjectMaster\2021_KilianMeier\EPFL29210420	FLY140	2021-04-20.Mesures.MoTUS	18-Apr-2021	CstSpeed0m/s	1.93
20-Apr-2021 09:46:22	20-Apr-2021 09:47:16	009	S:\ProjectMaster\2021_KilianMeier\EPFL29210420	FLY140	2021-04-20.Mesures.MoTUS	18-Apr-2021	CstSpeed0m/s	2.38
20-Apr-2021 09:48:53	20-Apr-2021 09:50:15	010	S:\ProjectMaster\2021_KilianMeier\EPFL29210420	FLY140	2021-04-20.Mesures.MoTUS	18-Apr-2021	CstSpeed2m/s	1.98
03-May-2021 10:14:33	03-May-2021 10:21:54	011	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY167	2021-05-03.Mesures.MoTUS	02-May-2021	Hover	2.02
03-May-2021 14:23:44	03-May-2021 14:24:15	012	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY167	2021-05-03.Mesures.MoTUS	02-May-2021	Square	2.02
03-May-2021 14:25:42	03-May-2021 14:27:43	013	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY167	2021-05-03.Mesures.MoTUS	02-May-2021	CstSpeed2m/s	1.67
03-May-2021 14:29:27	03-May-2021 14:30:21	014	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY167	2021-05-03.Mesures.MoTUS	02-May-2021	CstSpeed0m/s	2.03
03-May-2021 14:31:39	03-May-2021 14:32:29	015	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY167	2021-05-03.Mesures.MoTUS	02-May-2021	CstSpeed0m/s	2.06
03-May-2021 14:33:54	03-May-2021 14:34:38	016	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY167	2021-05-03.Mesures.MoTUS	02-May-2021	CstSpeed2m/s	1.41
03-May-2021 14:36:44	03-May-2021 14:37:47	017	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY168	2021-05-03.Mesures.MoTUS	02-May-2021	Vertical2m/s	1.6
03-May-2021 14:39:09	03-May-2021 14:40:00	018	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY168	2021-05-03.Mesures.MoTUS	02-May-2021	Vertical	1.9
03-May-2021 14:41:27	03-May-2021 14:42:15	019	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY168	2021-05-03.Mesures.MoTUS	02-May-2021	Vertical	1.83
03-May-2021 14:43:48	03-May-2021 14:44:35	020	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY168	2021-05-03.Mesures.MoTUS	02-May-2021	Vertical5m/s	2.26
03-May-2021 14:45:59	03-May-2021 14:46:30	021	S:\ProjectMaster\2021_KilianMeier\EPFL29210503	FLY168	2021-05-03.Mesures.MoTUS	02-May-2021	Square	1.2
25-May-2021 11:18:35	25-May-2021 11:26:46	022	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY172	2021-05-25.Mesures.MoTUS	23-May-2021	Hover	5.65
25-May-2021 11:28:48	25-May-2021 11:33:13	023	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY172	2021-05-25.Mesures.MoTUS	23-May-2021	Square	3.18
25-May-2021 11:31:19	25-May-2021 11:33:13	024	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY172	2021-05-25.Mesures.MoTUS	23-May-2021	CstSpeed2m/s	6.04
25-May-2021 11:35:00	25-May-2021 11:35:56	025	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY172	2021-05-25.Mesures.MoTUS	23-May-2021	CstSpeed0m/s	5.51
25-May-2021 11:37:15	25-May-2021 11:38:04	026	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY172	2021-05-25.Mesures.MoTUS	23-May-2021	CstSpeed0m/s	5.37
25-May-2021 11:41:07	25-May-2021 11:41:56	027	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY173	2021-05-25.Mesures.MoTUS	23-May-2021	CstSpeed0m/s	4.12
25-May-2021 11:43:23	25-May-2021 11:44:10	028	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY173	2021-05-25.Mesures.MoTUS	23-May-2021	CstSpeed2m/s	5.05
25-May-2021 11:45:38	25-May-2021 11:46:42	029	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY173	2021-05-25.Mesures.MoTUS	23-May-2021	Vertical2m/s	6.52
25-May-2021 11:49:01	25-May-2021 11:49:54	030	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY173	2021-05-25.Mesures.MoTUS	23-May-2021	Vertical	7.08
25-May-2021 11:51:18	25-May-2021 11:52:08	031	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY173	2021-05-25.Mesures.MoTUS	23-May-2021	Vertical	4.83
25-May-2021 11:53:28	25-May-2021 11:54:16	032	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY173	2021-05-25.Mesures.MoTUS	23-May-2021	Vertical	5.64
25-May-2021 11:55:44	25-May-2021 11:56:19	033	S:\ProjectMaster\2021_KilianMeier\EPFL29210525	FLY173	2021-05-25.Mesures.MoTUS	23-May-2021	Square	7.35
03-Jun-2021 06:11:55	03-Jun-2021 06:21:50	034	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	Hover	6.76
03-Jun-2021 06:12:23	03-Jun-2021 06:21:56	035	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	Square	0.34
03-Jun-2021 06:26:20	03-Jun-2021 06:28:14	036	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	CstSpeed2m/s	0.78
03-Jun-2021 06:30:18	03-Jun-2021 06:30:48	037	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	Vertical	0.36
03-Jun-2021 06:33:04	03-Jun-2021 06:33:50	039	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	CstSpeed0m/s	0.87
03-Jun-2021 06:35:15	03-Jun-2021 06:35:24	040	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	CstSpeed2m/s	0.36
03-Jun-2021 06:37:19	03-Jun-2021 06:38:14	041	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	Vertical2m/s	0.73
03-Jun-2021 06:41:55	03-Jun-2021 06:40:35	042	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	CstSpeed0m/s	0.85
03-Jun-2021 06:42:45	03-Jun-2021 06:42:54	043	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	Vertical	0.36
03-Jun-2021 06:46:05	03-Jun-2021 06:46:39	044	S:\ProjectMaster\2021_KilianMeier\EPFL29210603	FLY173	2021-06-03.Mesures.MoTUS	30-May-2021	Vertical	0.16
03-Jun-2021 12:53:56	03-Jun-2021 12:55:17	046	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY181	2021-06-07.Mesures.MoTUS	06-Jun-2021	Hover	1.15
07-Jun-2021 13:01:24	07-Jun-2021 13:01:54	047	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY181	2021-06-07.Mesures.MoTUS	06-Jun-2021	Square	0.97
07-Jun-2021 13:04:56	07-Jun-2021 13:04:57	048	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY181	2021-06-07.Mesures.MoTUS	06-Jun-2021	CstSpeed2m/s	1.5
07-Jun-2021 13:06:23	07-Jun-2021 13:07:17	049	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY181	2021-06-07.Mesures.MoTUS	06-Jun-2021	CstSpeed0m/s	1.9
07-Jun-2021 13:08:48	07-Jun-2021 13:09:34	050	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY181	2021-06-07.Mesures.MoTUS	06-Jun-2021	CstSpeed0m/s	1.4
07-Jun-2021 13:12:15	07-Jun-2021 13:15:17	052	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY182	2021-06-07.Mesures.MoTUS	06-Jun-2021	CstSpeed2m/s	2.17
07-Jun-2021 13:14:13	07-Jun-2021 13:17:23	053	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY182	2021-06-07.Mesures.MoTUS	06-Jun-2021	Vertical2m/s	1.85
07-Jun-2021 13:18:54	07-Jun-2021 13:19:40	054	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY182	2021-06-07.Mesures.MoTUS	06-Jun-2021	Vertical	3.49
07-Jun-2021 13:21:39	07-Jun-2021 13:22:28	055	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY182	2021-06-07.Mesures.MoTUS	06-Jun-2021	Vertical	3.3
07-Jun-2021 13:23:45	07-Jun-2021 13:24:18	056	S:\ProjectMaster\2021_KilianMeier\EPFL29210607	FLY182	2021-06-07.Mesures.MoTUS	06-Jun-2021	Square	2.39

Table 15: Overview of the data acquired during the flight campaign. The mounting point S : corresponds to the following server \\\enac1files.enfh.ch\topo (restricted access). Details about this Table can be found in Sections 3.1.3, 3.4.4 and 4.1. This file can be found at \\\enac1files.enfh.ch\topo\ProjectMaster\2021_KilianMeier

F Project description



Description	<p>Unmanned aerial vehicles (UAVs) can be used to estimate wind speed and wind direction. This is relevant in the field of meteorology for studying the lower atmospheric boundary layer. Multirotor drones can be used to measure vertical profiles of wind speed and direction instead of conventional weather balloons. This allows for example to take wind measurements in areas that are difficult to access such as crevassed glacier fronts.</p> <p>This study investigates an indirect method to estimate wind. Attitude (roll, pitch, yaw) and heading data are correlated to experimental anemometer measurements. This information can be used to build a model for wind speed and direction estimation. The goal of this study is to mature an existing method for wind estimation on DJI Phantom and Mavic drones, previously developed at UNIS. In addition, the method will be expanded with an existing approach of estimating sensible heat fluxes from drone profiles data. The work will include fieldwork in Adventdalen. Additional fieldwork options are possible.</p>
	<p>Fig. 1: Met mast measurements on Svalbard</p>
	<p>Fig. 2: Effect of wind on aircraft.</p>
Tasks	<ul style="list-style-type: none"> ▪ Literature review on the topic of wind estimation with UAVs. ▪ Adapt & improve the existing method for wind estimation. ▪ Generate additional validation data for the existing method. ▪ Implement estimation of sensible heat fluxes. ▪ Conduct meteorological observations in Adventdalen. ▪ Critically discuss the results
Requirements	<ul style="list-style-type: none"> ▪ Basic interest in meteorology and atmospheric research. ▪ Hands-on experience with drone flying is an advantage. ▪ Independent and reliable work style with a detail-oriented mindset.
Contact	<ul style="list-style-type: none"> ▪ Richard Hann (richard.hann@ntnu.no) ▪ Marius Jonassen (marius.jonassen@unis.no)