

MASTER THESIS

---

# WIND ESTIMATION WITH MULTIROTOR UAVS

---

Meier Kilian  
Supervisor EPFL : MER Skaloud Jan  
Supervisor UNIS : Dr. Hann Richard  
External Expert: ???  
Assistant: Joseph Paul Kenneth  
Geodetic Engineering Laboratory EPFL

Spring Semester 2021





## Acknowledgments

SPI

## Abstract

# Contents

<b>Acknowledgments</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>1 Introduction</b>	<b>6</b>
1.1 Motivation . . . . .	6
1.2 State of the art . . . . .	6
1.3 Approach . . . . .	6
<b>2 Preliminaries</b>	<b>7</b>
2.1 Notations . . . . .	7
2.1.1 Scalars . . . . .	7
2.1.2 Vectors . . . . .	7
2.1.3 Quaternion . . . . .	7
2.1.4 Matrices . . . . .	7
2.1.5 Functions . . . . .	7
2.1.6 Time derivatives . . . . .	8
2.1.7 Estimated quantities . . . . .	8
2.1.8 Measured quantities . . . . .	8
2.2 Attitude representations . . . . .	8
2.2.1 Rotation matrix . . . . .	8
2.2.2 Quaternion . . . . .	8
2.2.3 Euler angles . . . . .	9
2.2.4 Rotating reference frames . . . . .	10
2.3 Frames definition . . . . .	10
2.3.1 Inertial frame (i-frame) . . . . .	10
2.3.2 Earth frame (e-frame) . . . . .	10
2.3.3 Local-level frame (l-frame) . . . . .	11
2.3.4 Body frame (b-frame) . . . . .	11
2.3.5 Tilt frame (t-frame) . . . . .	11
2.3.6 Transformation from Body to Local-level frame . . . . .	11
2.3.7 Transformation from Body to Tilt frame . . . . .	12
2.4 Sensor model . . . . .	12
2.4.1 Weather station sensors . . . . .	12
2.4.2 Mobile sensor . . . . .	13
2.5 Motion model . . . . .	13
2.5.1 Equation of motion . . . . .	13
2.5.2 Specific forces . . . . .	14
2.6 Air density estimation . . . . .	15
2.6.1 Compression factor . . . . .	15
2.6.2 Molar fraction of water vapour . . . . .	15
2.7 Wind triangle . . . . .	16
<b>3 Methodology</b>	<b>17</b>
3.1 Data Collection . . . . .	17
3.1.1 Flight data . . . . .	17
3.1.2 Reference data . . . . .	17
3.1.3 Flight campaign . . . . .	18
3.2 Wind estimation from tilt . . . . .	18
3.2.1 Introduction . . . . .	18
3.2.2 Model calibration . . . . .	19
3.2.3 Computing air velocity . . . . .	21
3.3 Computing air direction . . . . .	21

3.4	Computing wind vector . . . . .	21
3.5	Wind estimation from dynamical model . . . . .	21
3.5.1	Introduction . . . . .	21
3.5.2	Computing thrust . . . . .	22
3.5.3	Computing drag . . . . .	22
3.5.4	Computing air speed . . . . .	23
3.5.5	Computing drag from force data . . . . .	24
3.6	Software overview . . . . .	24
3.6.1	Introduction . . . . .	24
3.6.2	Data flow . . . . .	24
3.6.3	Architecture . . . . .	24
<b>4</b>	<b>Results</b>	<b>25</b>
4.1	Data Sets . . . . .	25
4.2	Direct wind estimation from tilt . . . . .	25
4.3	Direct wind estimation from dynamical equation . . . . .	25
4.4	Filtered wind estimation using EKF . . . . .	25
4.5	Optimized wind estimation using DN . . . . .	25
<b>5</b>	<b>Discussion</b>	<b>26</b>
5.1	Method performance comparison . . . . .	26
5.2	Method tradeoff . . . . .	26
5.3	Applicability to meteorological research . . . . .	26
<b>6</b>	<b>Conclusion and future work</b>	<b>27</b>
<b>A</b>	<b>DJI Phantom flight data extraction using DatCon</b>	<b>29</b>
A.1	Required Hardware and Software . . . . .	29
A.2	DatCon . . . . .	29
A.3	Step-by-step procedure . . . . .	31
<b>B</b>	<b>Impact of drone flights on wild life</b>	<b>32</b>
<b>C</b>	<b>Assumption List</b>	<b>33</b>

## List of acronyms

**AC** Air Craft. 29

**DCM** Direct Cosine Matrix. 8

**ECEF** Earth Centered Earth Fixed. 10

**ECI** Earth Centered Inertial. 10

**FRD** Front-Right-Down. 11

**GNSS** Global Navigation Satellite System. 17

**GPS** Global Positioning System. 10, 29

**IMU** Inertial Measurement Unit. 10

**NED** North-East-Down. 11

**RTK** Real Time Kinematic. 17, 29

**TOPO** Geodetic Engineering Laboratory. 18

**UNIS** The University Center in Svalbard. 17

## 1 Introduction

Dummy citation [1]

### 1.1 Motivation

### 1.2 State of the art

### 1.3 Approach

## 2 Preliminaries

The proper use of data generated by mobile platform heavily relies on proper definition of attitude representations and reference frames, it should also 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 [2] and [3].

### 2.1 Notations

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

#### 2.1.1 Scalars

Scalars are noted 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  $r$  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

Quaternions 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  $q = q_1 + q_2i + q_3j + q_4k$  describing a rotation from frame  $i$  to  $j$  is written as:

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

#### 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  $f$  returning a vector in frame  $i$  and being dependent on a constant  $g$  and a matrix  $C$  will be written as:

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

### 2.1.6 Time derivatives

Time derivatives are note by adding a dot above the quantity to derive:

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

### 2.1.7 Estimated quantities

Estimated quantities, if necessary, will be differentiated 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 differentiated 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 an other. A quantity  $\mathbf{r}$  expressed in the a frame  $i$  is noted  $\mathbf{r}^i$ . 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 briefly be summarised in this section as well as the relations between all representations.

### 2.2.1 Rotation matrix

A rotation matrix  $\mathbf{C}$  (also called Direct Cosine Matrix (DCM)) is a matrix such that given a quantity in the  $i$ -frame  $\mathbf{r}^i$  and the same quantity in the  $j$ -frame  $\mathbf{r}^j$  the following relation holds:

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

The columns of  $\mathbf{C}_j^i$  represent the unit vectors of the  $j$ -frame projected onto 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 are mathematically stable and 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, since their use is more widely spread than quaternions and thus renders the comprehension of the motion equation easier.

### 2.2.2 Quaternion

Quaternions describe a rotation using only four parameters and leverages Euler's 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 that quaternion used for rotation have a length of 1 and thus only represents a subset of all quaternions. The equivalence between a quantity in the i-frame  $\mathbf{r}^i$  and j-frame  $\mathbf{r}^j$  is given by:

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

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 :

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

Note that Matlab implements a rotation by a quaternion with the *rotatepoint(...)* function.

Finally, quaternion are mathematically stable and unambiguous<sup>1</sup>, 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 quaternion 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 sequence yaw-pitch-roll and can be described as follows in a matrix form:

$$\mathbf{C}_j^i = \mathbf{C}_1(r) \mathbf{C}_2(p) \mathbf{C}_3(y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(r) & \sin(r) \\ 0 & -\sin(r) & \cos(r) \end{bmatrix} \begin{bmatrix} \cos(p) & 0 & -\sin(p) \\ 0 & 1 & 0 \\ \sin(p) & 0 & \cos(p) \end{bmatrix} \begin{bmatrix} \cos(y) & \sin(y) & 0 \\ -\sin(y) & \cos(y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Where  $r$ ,  $p$  and  $y$  are the roll, pitch and yaw angles respectively.<sup>2</sup>

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

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

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.

<sup>1</sup>Provided the subset of unit length quaternion is used.

<sup>2</sup>Note 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  $\phi$ ,  $\theta$  and  $\psi$ . 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 call 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.

### 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
- **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 call tx-, ty- and tz-axis respectively.

### 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  $\bar{\mathbf{q}}_b^l$  (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 :

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

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  $\lambda$  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  $\alpha$  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 is 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 ( $w_\mu$ ), 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:

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

## 2.5 Motion model

This section will be dedicated to the derivation of the translational motion model of the air-craft and to the study of the forces generating this motion. This will serve as a mathematical basis for the different method of estimation experimented in this work and will highlight some important assumptions that need to be made.

### 2.5.1 Equation of motion

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

$$\ddot{\mathbf{r}}^i = \mathbf{C}_b^i \mathbf{f}^b + \bar{\mathbf{g}}^i(\mathbf{r}^i) \quad (24)$$

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:

$$\ddot{\mathbf{r}}^e = \mathbf{C}_b^e \mathbf{f}^b + \bar{\mathbf{g}}^e(\mathbf{r}^e) - 2\Omega_{ie}^e \dot{\mathbf{r}}^e - \Omega_{ie}^e \Omega_{ie}^e \mathbf{r}^e \quad (25)$$

In this derivation the angular velocity of the earth with respect to the inertial frame  $\Omega_{ie}^e$  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:

$$\mathbf{g}^e(\mathbf{r}^e) = \bar{\mathbf{g}}^e(\mathbf{r}^e) - \Omega_{ie}^e \Omega_{ie}^e \mathbf{r}^e \quad (26)$$

Now moving to the local-level frame using (26) 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:

$$\ddot{\mathbf{r}}^l = \mathbf{C}_b^l \mathbf{f}^b + \mathbf{g}^l(\mathbf{r}^l) - (\Omega_{el}^l + 2\Omega_{ie}^l) \dot{\mathbf{r}}^l \quad (27)$$

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

$$\omega_{ie}^l = \begin{bmatrix} \omega_{ie} \cos \phi \\ 0 \\ -\omega_{ie} \sin \phi \end{bmatrix} \quad (28)$$

Where  $\omega_{ie}$  is the magnitude of the earth's angular velocity. A typical value is  $\omega_{ie} = 7.292115 \cdot 10^{-5} [\text{rad/s}]$  [4]. 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  $\max(\dot{\mathbf{r}}^l) = 16 [\text{m/s}]$  [5], which corresponds at most to an acceleration of  $2\omega_{ie} \max(\dot{\mathbf{r}}^l) = 2.2 \cdot 10^{-4} [\text{m/s}^2]$ . 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 [2], 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:

$$\omega_{el}^l = \begin{bmatrix} \dot{\lambda} \cos \phi \\ -\dot{\phi} \\ \dot{\lambda} \sin \phi \end{bmatrix} \quad (29)$$

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  $\max(\dot{\mathbf{r}}^l)/R_e = 2.5 \cdot 10^{-6} [\text{m/s}^2]$  (along one axis), where  $R_e = 6378 [\text{km}]$  [4] is the earth's radius. Hence rewriting (27) taking into account (As.3) and (As.4):

$$\ddot{\mathbf{r}}^l = \mathbf{C}_b^l \mathbf{f}^b + \mathbf{g}^l(\mathbf{r}^l) \quad (30)$$

Note that this equation corresponds to the motion equation in the i-frame (24), 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 is no surface on the Phantom 4 RTK body which seem to have been designed to generate lift<sup>3</sup>. Note this also assumes that only flight scenarios are considered and no force generated by contact with the ground 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:

$$\mathbf{f}^b = \mathbf{f}_T^b + \mathbf{f}_D^b = \frac{1}{m} \mathbf{F}_T^b + \frac{1}{m} \mathbf{F}_D^b \quad (31)$$

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  $m = 1.391 [\text{kg}]$  [5] the mass of the drone.

<sup>3</sup>With the obvious exception of the propeller blades, but this is taken into account in the thrust force

## 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 [6]. Air density will be function of air temperature  $A_t$ , atmospheric pressure  $A_p$  and relative humidity  $A_h$  as follows:

$$\rho = \rho(A_t, A_p, A_h) = \frac{pM_a}{ZRT} \left( 1 - x_v \left( 1 - \frac{M_v}{M_a} \right) \right) \quad (32)$$

Where the parameters are described in Table 1.

Table 1: Air density base equation parameters (see Equation (32)).

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:

$$Z = 1 - \frac{p}{T} (a_0 + a_1 t + a_2 t^2 + (b_0 + b_1 t)x_v + (c_0 + c_1 t)x_v^2) + \frac{p^2}{T^2} (d + e x_v^2) \quad (33)$$

Where the parameters are described in Table 2.

Table 2: Compression factor equation parameters (see Equation (33)).

Symbol	Description	Value
$p$	Atmospheric pressure	$A_p$ [Pa]
$T$	Thermodynamic air temperature	$A_t$ [K]
$t$	Air temperature	$A_t$ [°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 <sup>2</sup> /Pa <sup>2</sup> ]
$e$	Constant	$-1.034 \cdot 10^{-8}$ [K <sup>2</sup> /Pa <sup>2</sup> ]

### 2.6.2 Molar fraction of water vapour

The molar fraction of water vapour  $x_v$  is given by:

$$x_v = h_r f \frac{p_{SV}}{p} \quad (34)$$

Where the augmentation factor  $f$  is given by:

$$f = \alpha + \beta p + \gamma t^2 \quad (35)$$

And the vapour saturation pressure is given by:

$$p_{SV} = \exp \left( AT^2 + BT + C + \frac{D}{T} \right) \quad (36)$$

And the remaining parameters are given in Table 3.

Table 3: Molar fraction of water vapour equation parameters (see Equations (34), (35) and (36)).

Symbol	Description	Value
$h_r$	Relative humidity	$A_h []$
$f$	Augmentation factor	See Equation (35)
$p_{SV}$	Vapour saturation pressure	See Equation (36)
$p$	Atmospheric pressure	$A_p [Pa]$
$\alpha$	Constant	1.00062 []
$\beta$	Constant	$3.14 \cdot 10^{-8} [1/Pa]$
$\gamma$	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 follows:

$$\mathbf{w}^l = \mathbf{r}^l + \text{rot}(\mathbf{q}_b^l, \mathbf{V}^b) \quad (37)$$

### 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 GPS time to ensure synchronisation.

##### 3.1.1 Flight data

Flight data was produced by three different sources:

- DJI Phantom 4: this drone was used in [7] 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 Global Navigation Satellite System (GNSS) measurements ([@ DJI website](#)).

The signals extracted from each source are the same and are described in Table 4.

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

Table 4: Flight data signals.

Symbol	Field Name	Unit	Description
$\phi$	lati	[deg]	AC Latitude (WGS84)
$\lambda$	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 <sup>2</sup> ]	AC Acceleration in body frame (XYZ)
$f_y$	ay	[m/s <sup>2</sup> ]	AC Acceleration in body frame (XYZ)
$f_z$	az	[m/s <sup>2</sup> ]	AC Acceleration in body frame (XYZ)
$q_1$	q1		
$q_1$	q2		Quaternion describing the rotation from body frame to local-level frame.
$q_2$	q3		
$q_3$	q4		
$\omega_x$	gyroX	[rad/s]	AC Angular rate in the roll direction
$\omega_y$	gyroY	[rad/s]	AC Angular rate in the pitch direction
$\omega_z$	gyroZ	[rad/s]	AC Angular rate in the yaw direction
$\eta_{RF}$	motRpm_RF	[rpm]	Right-Front rotor speed
$\eta_{LF}$	motRpm_LF	[rpm]	Left-Front rotor speed
$\eta_{LB}$	motRpm_LB	[rpm]	Left-Back rotor speed
$\eta_{RB}$	motRpm_RB	[rpm]	Right-Back rotor speed

##### 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 [7] 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 1: 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 1a and 1b 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 1b 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 5.

### 3.1.3 Flight campaign

Location Flight type appendix on procedure for setup and flights

## 3.2 Wind estimation from tilt

### 3.2.1 Introduction

**Motivation** This section will describe the approach referred as the *simple method* in [7]. This estimation will serve as a baseline for the new estimation scheme proposed in this work and will be named *wind from 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 regardless of the wind conditions, then the tilt angle of the drone is correlated to the wind velocity. Indeed,

Table 5: Reference data signals.

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

the autopilot has to tilt the drone such that the thrust force compensates for the drag forces generated by the wind. Figure 2 shows a diagram of the force situation. From this figure it becomes clear that the bigger the drag force, the bigger the tilt angle  $\alpha$ .

Step-by-step, while highlighting the needed inputs, the estimation procedure can be summarised as follows in Table 6.

Table 6: Wind from tilt estimation algorithm summary.

Step	Description	Needed Input	Section
1	Establish model relating tilt and wind	Flight dataset	3.2.2 and [7]
2	Compute tilt	Altitude measurements	2.3.7
3	Compute air velocity	Tilt angle	3.2.3
4	Compute air direction	Tilt direction	3.3
5	Compute wind vector	Air velocity and direction	2.7

### 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 3 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 [7]. 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:

$$\|\mathbf{V}^b\|^2 = \begin{cases} a_0 \tan^2(\gamma) & \text{for } \gamma < \gamma_{crit} \\ a_1 \tan(\gamma) + a_2 & \text{for } \gamma \geq \gamma_{crit} \end{cases} \quad (38)$$

Where the constants are defined in Table 7. The detail behind the choice of this model will not be exposed here and the reader should refer to [7].



Figure 2: Forces acting on a hovering drone in the wind.

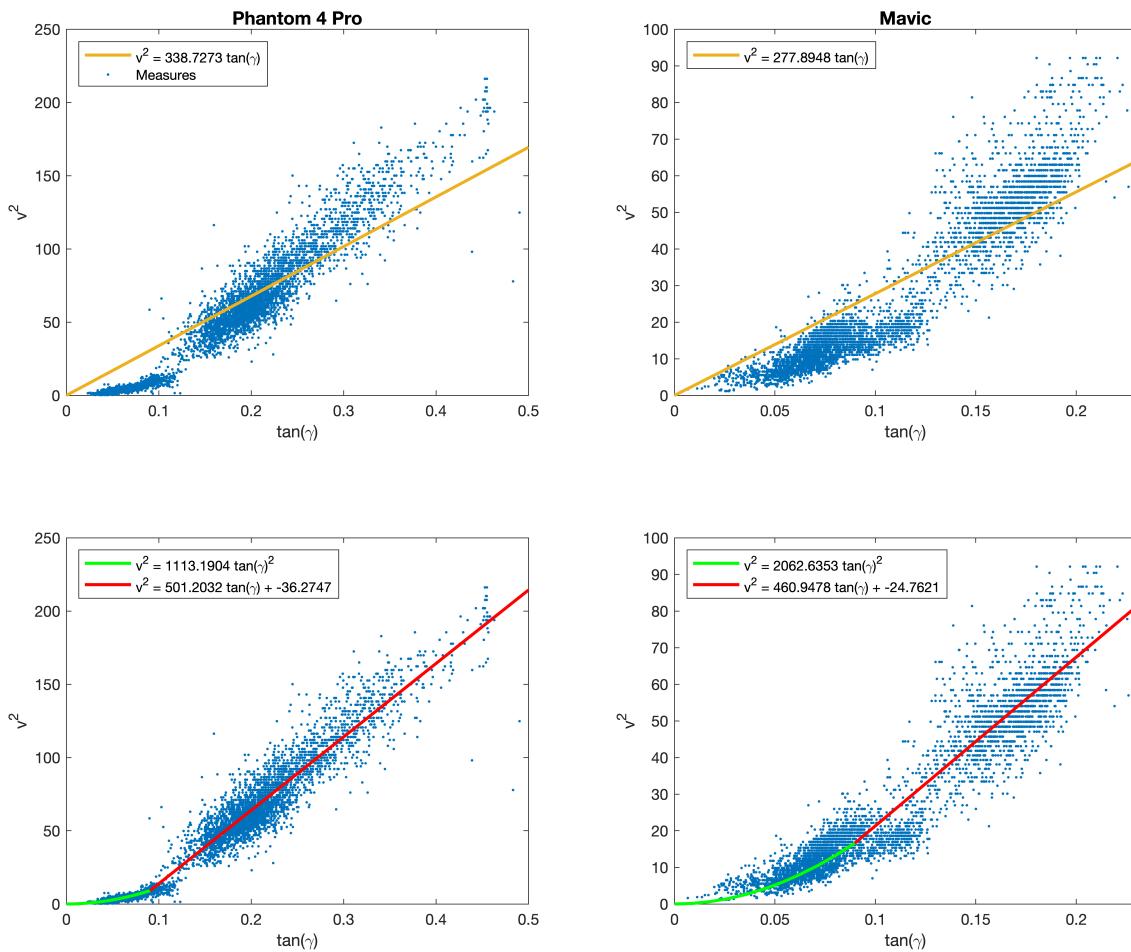


Figure 3: Tilt to wind correlation for a DJI Phantom 4 Pro an DJI Mavic 2 Enterprise drone. This Figure is taken from [7].

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

Table 7: Tilt to air velocity model parameter (see Equation (38)).

### 3.2.3 Computing air velocity

Once the tilt angle  $\alpha$  is computed using Equation 14 the air velocity can directly be computed using Equation (38). 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 7 (As.9).

## 3.3 Computing air direction

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

## 3.4 Computing wind vector

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

$$\mathbf{V}^l = \begin{bmatrix} \|\mathbf{V}^b\| \cos(\lambda) \\ \|\mathbf{V}^b\| \sin(\lambda) \\ 0 \end{bmatrix} \quad (39)$$

The air speed vector can be used in Equation (37) to compute the physical wind speed vector:

$$\mathbf{w}^l = \dot{\mathbf{r}}^l + \text{rot}(\mathbf{q}_b^l, \mathbf{V}^b) = \mathbf{V}^l \quad (40)$$

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

## 3.5 Wind estimation from dynamical model

### 3.5.1 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 air craft is under the influence of two specific forces: drag and thrust. Hence if the total specific force and the thrust is known 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). Then the air speed around the body can be inferred from the previously computed drag using an appropriate fluid friction 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 8.

Table 8: Dynamic Model estimation algorithm summary.

Step	Description	Needed Input	Section
1	Compute thrust	Rotor speed	3.5.2
2	Compute drag	Thrust and Accelerometer measurements	3.5.3, 2.3.7 and 3.5.5
3	Compute air speed	Drag	3.5.4
4	Compute wind speed	Air speed and drone speed	2.7

### 3.5.2 Computing thrust

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

$$F_T^b = \begin{bmatrix} 0 \\ 0 \\ -\rho b(\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2) \end{bmatrix} \quad (41)$$

Where  $\rho$  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 [8]. 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:

$$F_T^b = \begin{bmatrix} 0 \\ 0 \\ -\bar{\rho}\bar{b}\bar{\eta}^2 \end{bmatrix} \quad (42)$$

Where  $\bar{b}$  is the drone thrust constant and  $\bar{\eta}$  is the overall rotor speed, which will be defined in (43). Using a linear least-square regression, one can estimate  $\bar{b} = 4.9 \cdot 10^{-7} [N/RPM^2]$ . 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 [8], 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:

$$\bar{\eta} = \frac{1}{2} \sqrt{\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2} \quad (43)$$

Indeed, if (43) is used in (42):

$$F_T^b = \begin{bmatrix} 0 \\ 0 \\ -\bar{\rho}\bar{b}\frac{1}{4}(\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\bar{\rho}b(\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2) \end{bmatrix} \quad (44)$$

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:

$$F_T^b = \begin{bmatrix} 0 \\ 0 \\ -\bar{\rho}\bar{b} \left( \frac{1}{2} \sqrt{\eta_{RF}^2 + \eta_{LF}^2 + \eta_{LB}^2 + \eta_{RB}^2} \right)^2 \end{bmatrix} \quad (45)$$

### 3.5.3 Computing drag

Using (31) the drag force can be written as:

$$\mathbf{F}_D^b = m\mathbf{f}^b - \mathbf{F}_T^b \quad (46)$$

Moving to the local-level frame:

$$\mathbf{F}_D^l = m\mathbf{C}_b^l \mathbf{f}^b - \mathbf{C}_b^l \mathbf{F}_T^b = rot(\mathbf{q}_b^l, m\mathbf{f}^b) - rot(\mathbf{q}_b^l, \mathbf{F}_T^b) \quad (47)$$

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.

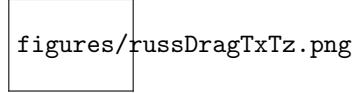


Figure 4: Wind tunnel setup description of [8].

### 3.5.4 Computing air speed

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:

$$F_{D,j} = \frac{1}{2} \rho A_j C_j V^2 = \rho K_j V^2 \quad (48)$$

Where  $\rho$  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  $V$  the air speed magnitude. As for the trust coefficient,  $K_j$  need to be measured or estimated. Here again the data from [8] will be leveraged. The data contains force measurements at a given air speed magnitude  $V_R$ , for various rotor speeds and for various pitch angles. From this work one can compute a model relating drag to air speed incidence angle and rotor speed:

$$F_{D,j} = D_R(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB}) \quad (49)$$

Where  $D_R$  is detailed in Section 3.5.5,  $\gamma$  is the wind incidence angle, see Figure 4. Note that the incidence angle will determine the orientation of the  $j$ -axis.

Hence for a given data point in [8], the following relation can be written using (49) in (48):

$$D_R(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB}) = F_{D,j} = \rho R K_j V_R^2 \quad (50)$$

Where,  $\rho_R$  is the air density experienced during the experiments in [8]. Isolating  $K_j$ :

$$K_j(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB}) = \frac{D_R(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB})}{\rho_R V_R^2} \quad (51)$$

Note that  $K_j$  depends on the wind incidence angle and rotor speeds. Now, since  $K_j$  is a constant (for a given wind incidence and rotor speed), (51) can be used to substitute  $K_j$  in (48):

$$F_{D,j} = \rho \frac{D_R(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB})}{\rho_R V_R^2} V^2 \quad (52)$$

Isolating  $V$ :

$$V = \sqrt{V_R^2 \frac{\rho_R}{\rho} \frac{F_{D,j}}{D_R(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB})}} \quad (53)$$

However, this relation cannot be used directly to compute the 3D air speed vector. Indeed, in [8], the  $j$ -axis, is always in the included in the  $xz$ -plane<sup>4</sup> of the body frame. Which is not true in the general case where wind may cause the air speed vector to live outside of the  $xz$ -plane. Furthermore, in [8], non-zero roll angles are not considered, which are needed in the general case where the drone is allowed to move freely. 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. In this frame, since the tilt frame will orient itself such that there is no "apparent roll", if the air speed vector is contained in the  $xz$ -plane, then the situation is equivalent to the experiment found in [8]. 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  $y$ -tilt-axis, then the situation is also equivalent the experiment found in [8] but with an incidence angle of zero. Hence for an arbitrary  $j$ -axis, the problem can be decomposed by projecting air speed and drag force onto the  $xz$ -tilt-plan and onto the  $y$ -tilt-axis. Figure 5 illustrates this projection.

<sup>4</sup>The plane spanned by the  $x$ -axis and  $z$ -axis of the body frame

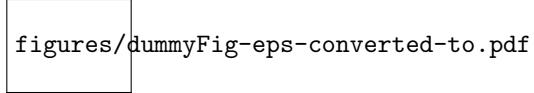


Figure 5: Drag projection in the tilt frame.

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

$$\mathbf{F}_D^t = \text{rot}(\mathbf{q}_b^t, \mathbf{F}_D^b) \quad (54)$$

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

$$V_{txtz} = \sqrt{V_R^2 \frac{\rho_R}{\rho} \frac{F_{D,txtz}}{D_R(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB})}} \quad (55)$$

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

$$V_{ty} = \sqrt{V_R^2 \frac{\rho_R}{\rho} \frac{F_{D,ty}}{D_R(0, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB})}} \quad (56)$$

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

$$\mathbf{V}^b = \text{rot}\left(\mathbf{q}_t^b, \begin{bmatrix} V_{txtz} \cos(\gamma) \\ V_{ty} \text{Sgn}(F_{D,ty}) \\ V_{txtz} \sin(\gamma) \end{bmatrix}\right) \quad (57)$$

### 3.5.5 Computing drag from force data

In [8], 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 [7] and in [9]. The final expression, once updated to correspond to this work's notation reads:

$$D_R(\gamma, \eta_{RF}, \eta_{LF}, \eta_{LB}, \eta_{RB}) = \tilde{F}_x \cos(\gamma) - (\tilde{F}_z + F_{T,z}) \sin(\gamma) \quad (58)$$

Where  $\gamma$  is the incidence angle,  $\tilde{\mathbf{F}}^b$  is the measured force acting on the body and  $\mathbf{F}_T$  the trust force which can be estimated as described in Section 3.5.2.

## 3.6 Software overview

### 3.6.1 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/meirkilian/WEMUAV>.

### 3.6.2 Data flow

sw data flow file format in appendix

### 3.6.3 Architecture

build such that it could be run with matlab runtime sw dependency graph link to github

## 4 Results

### 4.1 Data Sets

### 4.2 Direct wind estimation from tilt

Arthur's results

### 4.3 Direct wind estimation from dynamical equation

### 4.4 Filtered wind estimation using EKF

with various drag coef model

### 4.5 Optmized wind estimation using DN

with various drag coef model

## 5 Discussion

- 5.1 Method performance comparison
- 5.2 Method tradeoff
- 5.3 Applicability to meteorological research

## 6 Conclusion and future work

- time series via autonomous work... (but not open category drone in eu regulation due to LOS operation)
- further validation, other atmospheric conditions (temp, pressure, humid, altitude, latitude...) for example svalbard, lidar wind profile, extreme environment lab balloon... - does drag produce torque ? could be part of estimation

## References

- [1] P. P. Neumann, “Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit,” en, p. 11, 2015.
- [2] M. Khaghani, “Vehicle Dynamic Model Based Navigation for Small UAVs,” en, Ph.D. dissertation, EPFL, 2018.
- [3] Y. Stebler, “Modeling and Processing Approaches for Integrated Inertial Navigation,” en, Ph.D. dissertation, EPFL, 2013.
- [4] “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](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).
- [5] “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](https://dl.djicdn.com/downloads/phantom_4_rtk/20200721/Phantom_4_RTK_User_Manual_v2.4_EN.pdf).
- [6] “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.
- [7] 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](https://folk.ntnu.no/richahan/Publications/2020_Garreau_Thesis.pdf).
- [8] C. Russell, J. Jung, G. Willink, and B. Glasner, “Wind Tunnel and Hover Performance Test Results for Multicopter UAS Vehicles,” en, p. 20,
- [9] 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).

## 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 software<sup>5</sup> 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 6 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 6, 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].

---

<sup>5</sup>At least partly. Source code for version 4 could not be found, but version 3 can be found on [GitHub](#)

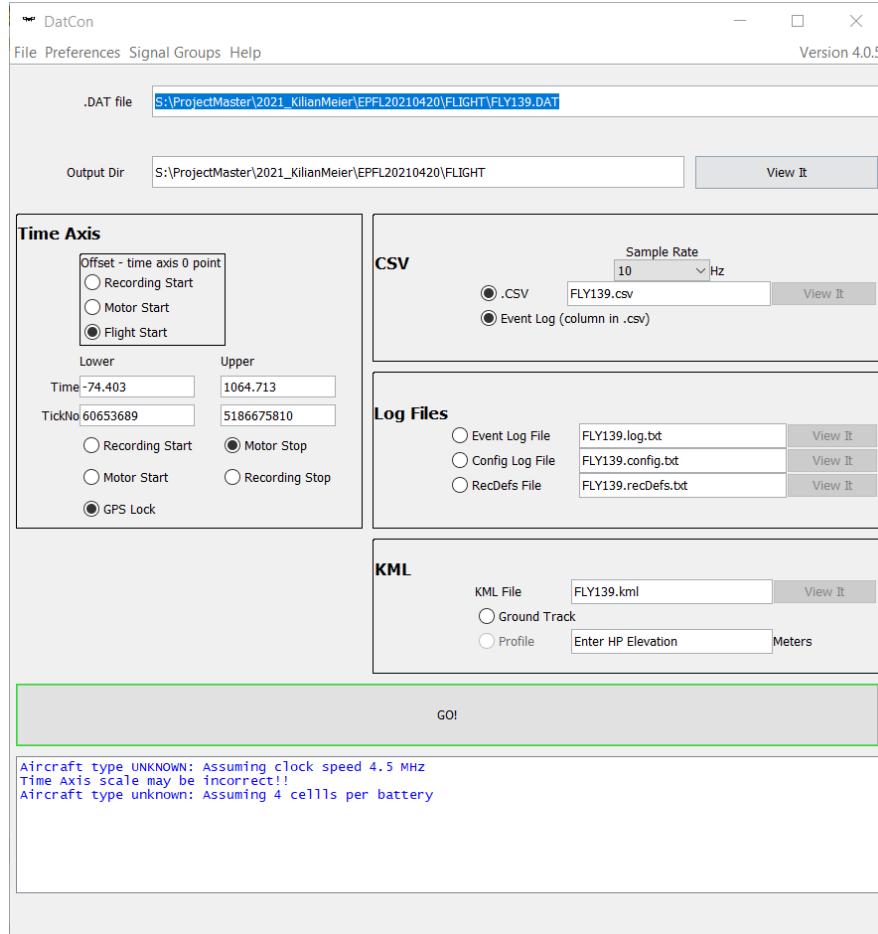


Figure 6: Snapshot of running DatCon software.

### 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 Impact of drone flights on wild life

General recognized impact of drone flights on wild life, distinction between fixed wing and quad (noise, concurrence ...)

Relevance in this work: wind measurement in remote area which are also more wild

Concretely phatom vs goeland

Mitigation technique:

operationally: avoid sensitive times and areas (nesting), escape manoeuvres (ebee example)

engineering: noise reduction, blade guards

acknowledgment

| <https://www.cambridge.org/core/journals/environmental-conservation/article/drones-as-a-threat-to-wildlife-youtube-complements-science-in-providing-evidence-about-their-effect/E433B815520AE5EE10C9168A5CEEEFA8>

<https://royalsocietypublishing.org/doi/full/10.1098/rsbl.2014.0754>

## C 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  $\Omega_{ie}^e$  is constant.
3. The angular velocity of the earth with respect to the inertial frame  $\Omega_{ie}^l$  is zero.
4. The local frame transport rate  $\Omega_{el}^l$  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.