Estimating wind fields using drones in a network

MSc thesis

TU Delft

Faculty of Aerospace Engineering

February 2022 - February 2023

Delft University of Technology

TU Delft Delft University of Technology

Challenge the future

The space above and below the message intentionally is left blank.

Estimating wind fields using drones in a network

MSc thesis
This document is written in fulfillment of the
MSc Aerospace Engineering, Control & Simulation.

by

Erik van Baasbank 4554825

Delft, January 27th, 2023
Document Version: 1.0

Committee Members:

Alessandro Bombelli	a.bombelli@tudelft.nl
Jacco Hoekstra	j.m.hoekstra@tudelft.nl
Junzi Sun	j.sun-1@tudelft.nl
Emmanuel Sunil	emmanuel.sunil@nlr.nl

Distribution list
Faculty of Aerospace Engineering, TU Delft
AOAP Department of Royal NLR

Table of Contents

I List of Symbols and Abbreviations

I Scientific Paper

II Preliminary Report

Introduction

Research Methodologies

Literature Review

Preliminary Results

Experiment & Flight Test Design

I List of Symbols and Abbreviations

Abbreviations

ABL	- Atmospheric Boundary Layer
BPNN	- Back Propogation Neural Network
EKF	- Extended Kalman Filter
ESC	- Electronic Speed Controller
FPS	- Frames per second [f/s]
GPS	- Global Positioning System
IMU	- Inertial Measurement Unit
LESO	- Linear Extended State Observer
LIDAR	- Light Detection And Ranging
LR	- Learning Rate
LSTM	- Long- Short Term Memory
MAF	- Moving Average Filter
MEMS	- Micro Electro-Mechanical Systems (IMU)
METSIS	- METeo Sensors in the Sky
ML	- Machine Learning
MOI	- Moment of Inertia
MPM	- Meteo Particle Model
MUAS	- Multi-Rotor unmanned Aircraft System
MSc	- Master of Science
NLR	- Royal Netherlands Aerospace Centre
NN	- Neural Networks
PWM	- Pulse Width Modulation
RPM	- Rotations per Minute
RMSE	- Root Mean Square Error
SESAR	- Single European Sky ATM Research
SMART	- Specific, Measurable, Attainable, Relevant and Timely
SMC	- Sequential Monte Carlo
TU	- University of Technology
UKF	- Unscented Kalman Filter
U-Space	- Unmanned Airspace
VLL	- Very Low Level Flight
V&V	- Verification & Validation
XGBOOST	- Extreme Gradient Boosting

Subscripts

$\emptyset_{E}$	- Earth frame
$\emptyset_{B}$	- Body frame
$\emptyset_{PF}$	- Particle filter
$\emptyset_{DET}$	- Deterministic, using only kinematic equations
$\emptyset_{REF}$	- Reference Measurements
$\emptyset_{EST}$	- Estimations

SYMBOLS

a, b	- Width, breadth, height, depending on axis [m]
C_D, C_d	- Drag coefficient [N]
f	- Frequency [Hz]
f	- Frame [-]
g	- Gravitational constant, 9.81 [m/s/s]
$g.c.$	- GPS ground course
m	- Mass [kg]
l	- Length [m]
P	- Permeability constant [-]
R	- Radius [m]
s	- Distance [m]
S, A	- Frontal surface area $[m^2]$
T	- Thrust [N]
$T.$	- Transpose of a matrix or vector
v	- Speed (not to be confused with: tilt angle ν)
ϕ	- Roll angle, around x-axis $[rad, \circ]$
θ	- Pitch angle, around y-axis $[rad, \circ]$
ψ	- Yaw angle, around z-axis $[rad, \circ]$
ν	- Tilt angle as measured from the euclidean inertial Zenith
ρ	- Air density constant on day of flight $[kg/m^3]$

Part I

Scientific Paper

Estimating hyperlocal wind fields with on-board sensors on quadcopters

Erik van Baasbank[1],
Supervisors: Junzi Sun[1], Jacco Hoekstra [1], and Emmanuel Sunil[2]

[1]*Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, the Netherlands*
[2]*National Aerospace Laboratory, Anthony Fokkerweg 2, 1059 CM Amsterdam, the Netherlands*

***Abstract*—Charting hyperlocal wind using a drone is a challenge of increased attention as it unlocks potential in a variety of fields. In context of the METeo Sensors In the Sky project, this study proposes a method to estimate the magnitude and direction of wind using a quadcopter in hover and cruise without a dedicated wind sensor. Only on-board sensors are used, with no knowledge of thrust and rpm. A deterministic method models drag experienced by the drone classically as a quadratic function of true airspeed, and estimates wind by deducting the estimated true airspeed with the GPS ground speed. Additionally, a particle filter is implemented and compared to the deterministic method. To validate the proposed methods, a series of verification flights is conducted in which the drone is flown straight into the wind, perpendicular to, and away from the wind. The results show that the proposed method can estimate wind for various ground speeds and altitudes. The root mean square error ranges between 0.3-2.0 m/s and 5-35 degrees in most scenarios with high true airspeeds. In most cases, the particle filter shows a slight improvement over the deterministic method, at the cost of reduced adaptivity to wind changes (gusts).**

Keywords: quadcopters, sensorless, particle filter, state-estimation, wind estimation

NOMENCLATURE

$\emptyset_B$	Body frame (subscript)
$\emptyset_E$	Earth frame (subscript)
$\emptyset_{DET}$	Deterministic (subscript)
$\emptyset_{EST}$	Estimation (subscript)
$\emptyset_{PF}$	Particle Filter (subscript)
$\emptyset_{REF}$	Reference measurements (subscript)
ν	tilt angle [deg, rad]
ϕ	Roll angle, around x-axis [deg, rad]
ψ	Yaw angle, around z-axis [deg, rad]
ρ	Air density constant on day of flight 1.2406 [kg/m^3]
θ	Pitch angle, around y-axis [deg, rad]
C_D	Drag coefficient, quadratic [-]
f	Frequency [Hz], friction coefficient [-]
g	Gravitational constant of 9.81 [m/s/s]
$g.c.$	GPS ground course, around z-axis [deg, rad]
h	Altitude [m]
k	Drag coefficient, linear [-]
m	mass [kg]
pwm	Pulse Width Modulation (1000-2000) [µs]
S	Surface area [m^2]
T	Thrust [N]
$T.$	Transpose of a matrix or vector
v	speed [m/s] (not to be confused with tilt ν)
ESC	Electronic Speed Controller
IMU	Inertial Measurement Unit
METSIS	Meteo Sensors in the Sky
MPM	Meteo Particle Model
ODE	Ordinary Differential Equation
RPM	Rotations Per Minute
SMC	Sequential Monte Carlo

I. INTRODUCTION

The marketplace for drones is growing rapidly and offers promising solutions in a wide variety of sectors, ranging from offshore wind farm inspections to military purposes [1], [2]. In operation, drones are affected by winds, impacting battery performance and thereby range significantly [3], [4], [5]. Therefore, it would be beneficial for a drone operator to have precise knowledge of wind magnitude and direction among common drone routes, such that the operator can plan its trajectory tactically. Current options for checking wind include websites, although these only go up to a resolution of roughly 3 km as they get information from ground-based measuring stations. Research around wind estimation on-board of drones aims to provide a more detailed (i.e. hyperlocal) wind map of an area.

For this reason, the METeo Sensors in the Sky (METSIS) project has been set up at Royal NLR. As a suggestion to fill this "gap in U-Space" [6], the wind data is gathered during routine jobs and gets sent to the operator such that it can plan its trajectory accordingly to avoid excessive headwinds. This study aims to explore and implement the best method to estimate those wind fields accurately without dedicated sensors such as anemometers or pitot tubes.

A. Previous work

Previous work towards successful wind estimation on-board of drones is often classified as direct (using a dedicated wind sensor), indirect (only on-board sensors) or hybrid (combination).

Considering the direct method, numerous studies [7], [8], [9], [10], [7] provide a comprehensive overview of different types of wind sensors on-board of drones. Notably, [11] explore the use of pitot tubes on fixed-wing aircraft, but those are impractical in this context as quadcopters operate at low speeds [12] [13], are subject to inconsistent flight directions [14] and do not provide a constant and predictable angle of attack which the pitot sensor requires. Fusing drone data with wind sensor data [9] (i.e. the hybrid method) would not increase the cost-effectiveness and scalability as a sensor is still required, thus this option is left unattended in this paper.

This paper instead builds upon the knowledge of indirect methods. [15] put the focus on gust estimation, while others assume steady-state wind and model the derivative of wind speed to be zero [16]. This simplifies state estimation at the cost of slower convergence time and reduced fidelity.

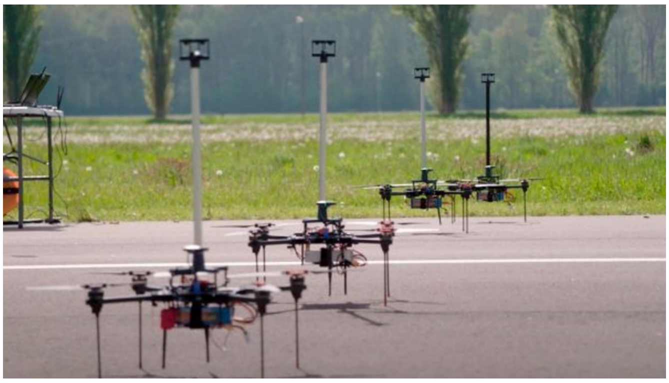


Fig. 1. Foxtech Hover 1 drones used in this research, with wind sensor mounted on top at a distance from rotor air flows. The proposed methods in this paper estimate wind without the use of the shown sensors.

A subgroup of indirect wind estimation on drones is the direct mapping, where a mapping between tilt and the final estimate of v_w is found. This is first established by [14] (who are considered the first to estimate wind on-board of drones). Similar methods have been performed by [17], [18] and [12]. Since most methods rely on extensive wind tunnel tests, this will not be the method of preference.

Another sub-group of the indirect method resolves the drag term into true airspeed, and from there finds the wind speed by means of the triangle method. This commonly established method is at the heart of the proposed methodology in subsection II-C and entails that the difference in true airspeed and ground speed must be attributed to the wind speed.

[19] uses a set of momentum theory equations but requires knowledge of RPM which we do not have for this research. Similarly, [20] resolves the true airspeed but presents a way to model the drag force as linearly dependent on speed, instead of quadratic. This is then followed by [13], who find the drag force (or in their study, ”drag acceleration”) to be linearly dependent on speed, rather than quadratic. This approach is followed by e.g. [13], [21] and later [16], who achieve successful results for wind speeds up to 7 m/s. It has not been documented precisely, however, how well wind estimates are for higher true airspeeds, which is what this paper focuses on.

In terms of filtering, Sylvie Schafer implements a linear Kalman filter (and addresses it well) in [15] and uses an existing MATLAB model to set up the model for the quadcopter. [16] use a filter to address the changing bias in the gyros, and similarly, [13] use a linear extended state observer (LESO) to estimate the drag-term acceleration, which essentially adds an observation-correction term to the equations in state space form, and tries to minimize this term. [22] implements an extended Kalman filter as well as an unscented Kalman filter, but uses a set of equations that requires detailed thrust parameters which are not available.

B. Novelties

In continuation of wind estimation on-board of drones, this paper will try to answer the following research question: ”How can on-board states and intelligent filtering on the Foxtech Hover 1 be used to measure 2D wind speeds with an accuracy of 0.5 m/s and 5 degrees?” This is researched by considering the drone as a point mass and putting the focus on creating reliable performance models (e.g. thrust, drag) from limited data. In many of the methods in the above-referenced literature, drone-specific performance models and parameters are often expected to be known or found using expensive equipment like wind tunnels. For this research, only frameworks and methodologies which could extend to all drones and be available to all operators worldwide are presented. In summary, this research introduces only performance models which are constructed according to the following three pillars of design. The choices presented in this paper should lead to accurate (1), scalable (2), and cost-effective (3) outcomes for METSIS.

Additionally, this paper will implement a particle filter to improve wind estimation, to which to the best of the author’s knowledge has not been implemented before in this exact context on an individual drone. The particle filtering has been used to estimate wind fields in an interpolating application (Meteo Particle Model) by [23], and for model predictive control on a quadcopter [24], but never before to estimate wind on individual quadcopters.

II. Methodology

The methodology for this research is presented below. This is split up into four subsections. First, essential prior knowledge is presented. Second, it is shown how the performance models of the drone (i.e. the sub-models) are created. Third, the equations behind the deterministic method are displayed. Fourth and last, the particle filter approach is shown, which includes the discretization of equations and presents the settings used to run the filter.

A. Basics

A.1 Reference Frames

The reference frame for this research used is Front-Right-Down (Body) and North East Down (Earth/Inertial), as displayed in Figure 2.

The rotation matrix used throughout is displayed in Equation 1, which takes inputs from the body frame and converts them to an inertial earth frame.

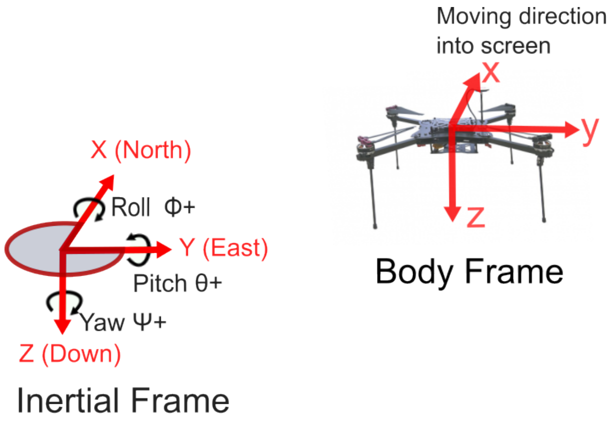


Fig. 2. Earth/Inertial and Body reference frame used.

$$R_B^E(\theta,\phi,\psi) = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - c\phi s\psi & c\psi s\theta c\phi + s\phi s\psi \\ c\theta s\psi & c\phi c\psi + s\theta s\phi s\psi & -s\phi c\psi + s\theta c\phi s\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{1}$$

Where all angles are in- and output as radians. Going from inertial to body frame is done by taking the transpose of matrix R: $R_E^B = [R_B^E]^{T.}$.

A.2 *Magnitude and direction of the wind*

It should be emphasized that the states in this paper can be described in 3D, meaning the vector v_w for example is represented as in Equation 2.

$$v_w = \begin{bmatrix} v_{w,x} \\ v_{w,y} \\ v_{w,z} \end{bmatrix}_E \tag{2}$$

The E signifies wind is presented in the earth frame, where X is Northing and Y is Easting. From here the 2D (or 3D) magnitude can be computed by following Equation 3.

$$|v_w| = v_{w,x}^2 + v_{w,y}^2 \tag{3}$$

In line, the wind direction is estimated using Equation 4.

$$w.d._{origin} = \arctan\left(\frac{V_{w,x}}{V_{w,y}}\right) - 180^\circ \tag{4}$$

Where 180 ° is subtracted to know the origin angle of the wind instead of the destination angle of the wind. Note that $V_{w,x}$ is in the numerator considering this is defined as North in the coordinate system.

A.3 *Software and Hardware*

This paper focuses on the wind estimating capabilities of quadcopters, and so the Foxtech Hover 1 Quadcopter is selected as a representative study.

As a flight controller, the Cube Orange flight controller has been selected. This provides the user with the input variables used in this thesis, shown in the overview in the observation vector in subsubsection II-D1. Here, the inertial measurement units (IMUs) provide unfiltered acceleration in the body frame. The pulse width modulation (PWM) is a power switching technique that provides a value between 1000 and 2000, which determines the voltage fed to the electronic speed controller (ESC) per rotor. In this thesis, as rotational dynamics are not the point of focus for this research, it is assumed that all four rotors create an equal amount of thrust such that the PWM can be averaged over the four rotors.

A.4 *Setting up reference measurements*

As a source of reference, an ultrasonic anemometer (Anemoment Trisonica Mini) is selected. As wind speeds can vary significantly depending on the altitude, the selected sensor is fitted onto a hovering drone. This way, the sensor can be brought to hover at the same altitude as the estimating drone, as is shown in Figure 3.

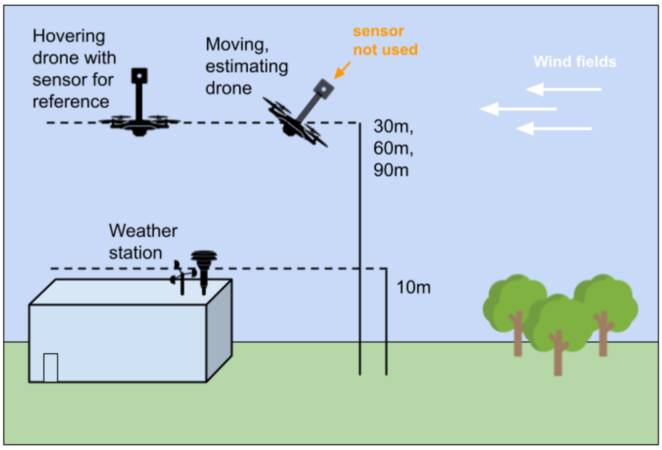


Fig. 3. Overview of estimating drone, together with reference drone and weather station

As a cross reference, a Gill Instrument MetStream is installed on the roof of the test site at an altitude of roughly 10m. Accounting for the expected wind speed difference due to altitude (computed by the Helmann law as in [25]), these measurements serve as a ballpark estimate to prove that the hovering drone with anemometer can be used as a reference. The results of both wind measurements are displayed in Table I.

	Weather Station	**Drone with Reference Sensor**
Altitude [m]	10	30
Avg wind speed [m/s]	4.1	6.5

TABLE I
SHOWING THE MEASURED DIFFERENCE IN TWO REFERENCE SENSORS. NOTE THAT THE ESTIMATES ARE IN THE SAME BALL-PARK AFTER A HEIGHT CORRECTION ($\Delta v_w = 2.4m/s$ DIFFERENCE EXPLAINED BY HELLMAN'S LAW (ASSUMING α=0.30).

B. *Performance Models*

For the Foxtech Hover 1 parameters such as thrust coefficients and drag coefficients are not specified, limiting the methodology approaches which can be performed. Therefore, this paper suggests methods for setting up performance models which should result in sufficient accuracy to estimate wind fields but should also be accessible and affordable for any drone operator across the world intending to participate in U-Space, to be in line with the design principles presented in Section I.

An overview of the performance models required for the selected methodology is presented in Figure 4.

B.1 Drag coefficient

Of the lower blocks in Figure 4, first the determination of the drag coefficient is discussed. For that, a quadratic relationship with airspeed is modeled as in [26], shown in Equation 5.

$$C_d = \frac{2D_E}{\rho v^2 S} \quad (quadratic) \tag{5}$$

As hinted at in earlier literature, for lower true airspeeds a drag model with a linear relationship with speed works better, and therefore Equation 6 has been used on the same set of data. The results coming from this second, linear approach are not the point of focus for this thesis and are presented in the subsection IV-A.

$$k = \frac{D_E}{v} \quad (linear) \tag{6}$$

In Equation 5, D_E is the drag experienced under a controlled environment (which in Figure 4 is written as T - ma). a_{res} is the (low-passed) resultant acceleration measured from the on-board IMUs, and S is the surface area determined later in this section. To fill in Equation 5, an approach similar to that in [13] has been taken, where 25 straight-line flights have been flown. This is performed indoors, since then the absence of wind should make that $v_{gnd} = v_{tas}$. To find a representative C_D for all yaw angles, yaw has been varied ($\psi = [0, 15, 30, 45, 60, 75, 90]$). Since the test was performed in the absence of GPS, the speed and pitch could not be commanded. Instead, for each of the indicated yaw angles, the drone operator was asked to perform a few slow cruise lines and a few fast lines of cruise. In post-processing, speed was found by the camera, and pitch and roll were read out from the Cube Orange. This showed speeds between 5 and 18 m/s, and pitch angles between 1 and 15 °. From there, the computed C_D's are grouped (i.e. rounded off) into "buckets" of speed and then averaged to find the drag coefficients as summarised in Table II.

Drag Coefficient		found by param est	Used for Lowest RMSE
(quadratic)	$C_{D,x}$, $C_{D,y}$	0.14	0.55
(linear)	k_x, k_y	0.17	0.55

TABLE II
DRAG COEFFICIENTS ESTIMATED FOR THE FOXTECH HOVER 1

In the absence of a LIDAR or indoor motion tracking system, an alternative speed-capturing method is presented where estimates of the instantaneous ground speed are made by a slow-motion camera. By counting the frames, and knowing the location of the drone at two points across the screen, the speed can be estimated by following $v = \frac{s}{t} = \frac{s}{\frac{\Delta f}{FPS}}$.

The test shows that 0.10-0.30 is the range the drag coefficient remains within roughly, which seems reasonable in range comparing it to peers (e.g. [13] with 0.15-0.25). From these estimations, adjustments to the C_D which result in lower RMSEs have been proposed, as is discussed in subsection III-B.

B.2 Thrust

Nearly all methodology for setting up performance models requires knowledge of thrust, which the Foxtech hover 1 does not provide. Therefore, this paper presents an alternative method to obtain thrust estimates. A mapping from PWM to thrust is made, by using a Series 1580 Test Stand. This results in a transfer function which is described in Equation 7.

$$T_{rotor} = 0.0226\,[PWM_{avg}] - 27.01 \tag{7}$$

$$\text{for } 1350 \leq PWM_{avg} \leq 1800$$

In this formula, the PWM signal is averaged, since rotational dynamics are excluded in this methodology.

$$PWM_{avg} = \frac{1}{4}\sum_{n=1}^{4}(PWM)_n \tag{8}$$

Note that the result is a linear relation, which means it is only valid for PWM values between 1350 and 1800. This is justified as the rotor speeds of a drone remain relatively constant throughout operation [16]. Analyzing the log files for the Foxtech Hover 1 specifically, the PWM seems to remain within said range, even when performing aggressive jump maneuvers (from 1 to 8.5m).

Notably, the tests show that increasing and decreasing PWM continuously has an effect on the offset of the curve. Without the influence of wind, this offset is found to be small (roughly 0.5%) and under the influence of a 4 m/s wind flow, this offset is found to be up to 2 N (roughly 10%), which is significant in this model. To mitigate these effects, for this reason, research is restricted to measuring wind in (quasi) steady conditions, meaning one should not make measurements of wind when PWM is increased or decreased. It is found that the thrust values from tests with discrete PWM increments provide an average of the tests where PWM is in- and decremented continuously.

B.3 Resultant forces

The resultant forces are found by reading out the Cube Orange's on-board IMUs and converting them to the earth frame via the rotation matrix. In the deterministic method, a low-pass filter is used on this data. In the particle filter, this is not done to allow the filter to process the original noise.

B.4 Frontal area

The frontal area is modeled by making frontal photos and counting the pixels of the drone next to a reference (preferably spherical) object of known size. The photos have been taken from a distance far enough to minimize the distorting lens eye effect and thereby mimic the incoming free flow. Repeating

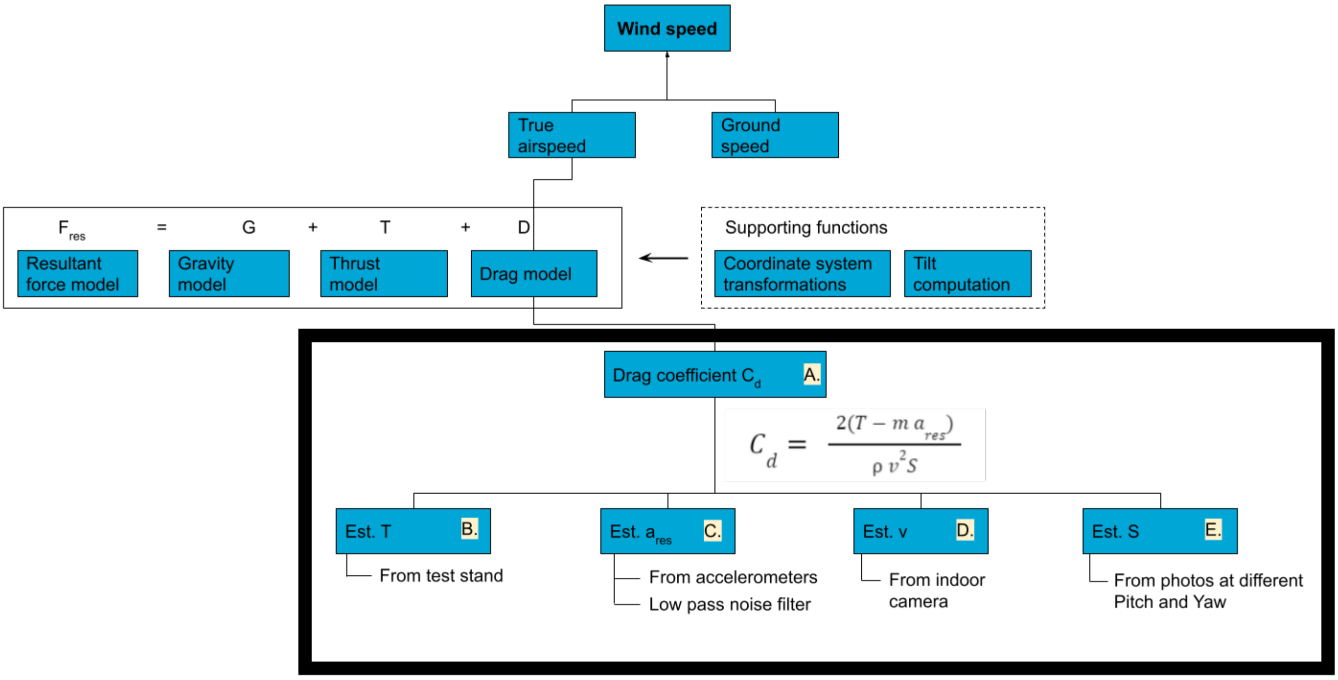


Fig. 4. A tree showing the performance models which need to be identified (thrust, drag, frontal area, resultant force

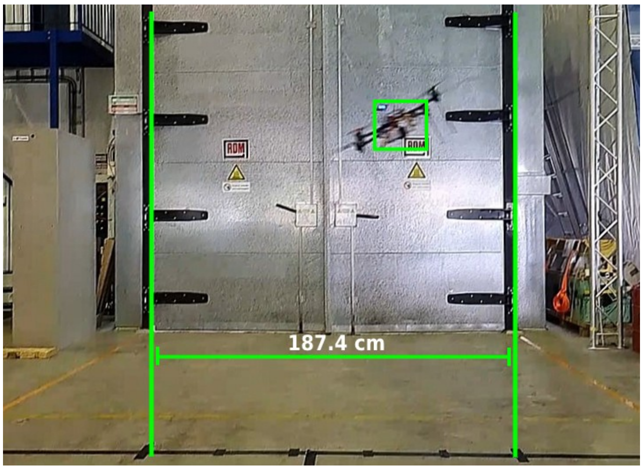


Fig. 5. Indoor drag coefficient test, drone passing the speed trap with video recorded at 480 fps.

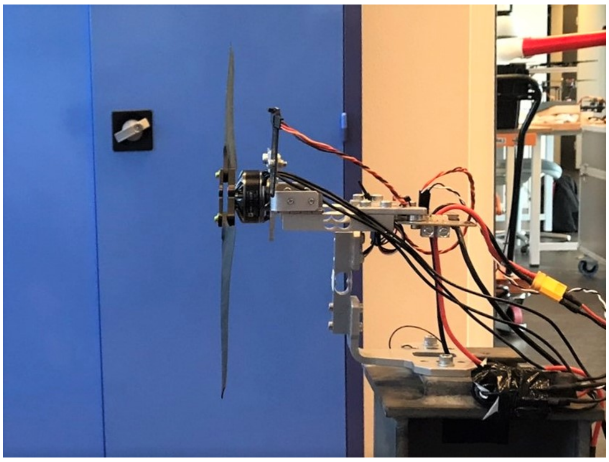


Fig. 6. Thrust set-up shown, using a Series 1580 Test Stand

this procedure for a variety of pitch and yaw angles, an interpolation function of the pitch against (yaw averaged) frontal area is found, as seen in Figure 9. Note that the area of the propellers is included in this surface area estimation.

The result is displayed in Equation 9.

$$S = 0.0089 \cdot \nu + 0.039 \tag{9}$$

where the tilt (in degrees) is computed by Equation 10.

$$\nu = \arctan \sqrt{\tan(\theta)^2 + \tan(\phi)^2} \tag{10}$$

The above models can be used to estimate the drone's performance characteristics at any time during flight, allowing us to follow the methodology presented in subsection II-C and subsection II-D.

C. *State estimation – deterministic method*

With the performance models as described in subsection II-B, this subsection describes the system of equations used to estimate the wind. Focus is put on the deterministic method, meaning it follows a set of kinematic equations with no stochastic methods involved. This makes this method computationally efficient, and easiest to implement.

This kinematic approach will implement linear dynamics, considering e.g. [27] does not show significant improvement (table 9) when comparing the in- and exclusion of rotational dynamics.

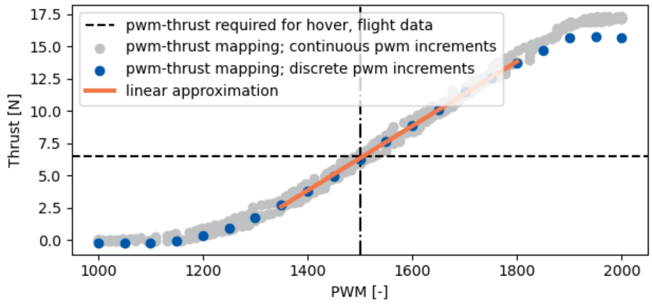


Fig. 7. Thrust mapping from using a Series 1580 Test Stand, with discrete (blue) and continuous (grey) increments of PWM. PWM generally remains around the PWM indicated with the dotted vertical line, where the linear fit forms a good approximation of the two tests.

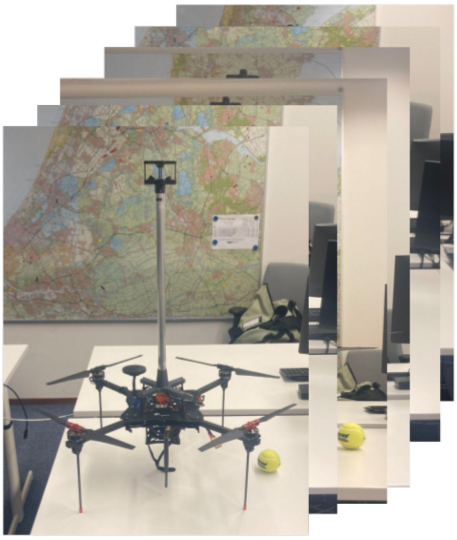


Fig. 8. Frontal area example from photos, taken at 0, 15, 30 and 45° pitch.

Working back from the end goal, the wind is found via the triangle method, which is shown in Equation 11.

$$v_w = v_{gnd} - v_{tas} \tag{11}$$

Where v_{gnd} is observed directly from GPS data on the Cube Orange flight controller. v_{tas} however should be estimated and can be found by resolving Equation 12 or Equation 57. In this paper, the focus on the quadratic relation will be held. In Section IV a subsection will be dedicated to a comparison to the results found with the linear relation.

$$v_{tas} = \sqrt{\frac{2 \cdot D_E}{\rho S C_d}} \quad (quadratic) \tag{12}$$

Where v_{tas} is the true airspeed. On the right-hand side of the equation, ρ is computed from weather data on the testing location at Marknesse. The surface area S and the drag coefficient C_D (modeled as a constant) are found from the performance models discussed in subsection II-B.

Consider Figure 10. The magnitude of D_E is found by a summation of forces in the earth frame, shown in Equation 13.

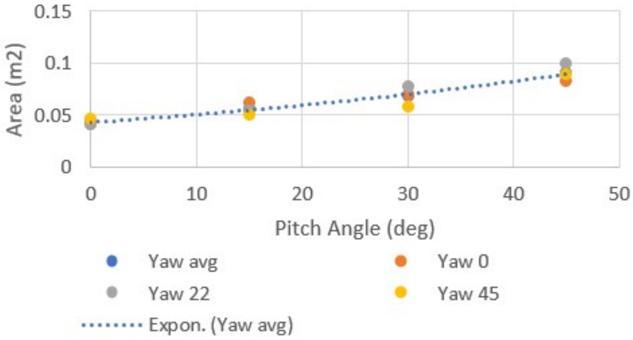


Fig. 9. Frontal area from photos, summarised and interpolated.

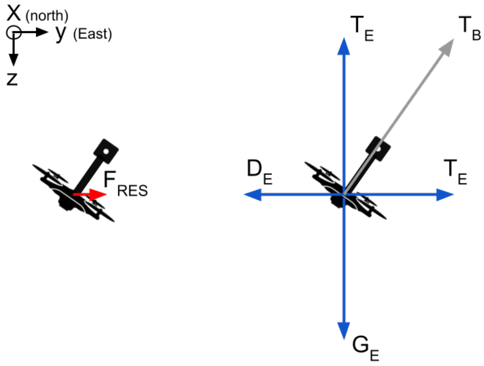


Fig. 10. Simplified Free Body Diagram of drone in cruise into the wind.

$$F_E = G_E + T_E + D_E \tag{13}$$

This is valid if we keep all signs of the performance models in the coordinate system from Figure 2. Rewriting this summation gives:

$$D_E = F_E - G_E - T_E \tag{14}$$

Remember that subscripts B and E refer to body and earth, respectively. Rewriting some of the variables to the body frame is more practical, considering measurements on-board the drone are performed in this frame. Therefore:

$$D_E = R_B^I \cdot F_B - G_E - R_B^I \cdot T_B \tag{15}$$

Such that from here F_B and T_B and G_E are the missing variables. From here,

$$F_B = m \cdot a_B = m \cdot [a_x, a_y, a_z]_B^T. \tag{16}$$

Where a_B is read out from the on-board IMUs. As described in subsection II-B, the IMUs acceleration values are averaged and low-pass filtered (only in the deterministic script. In the particle filter, the data is not filtered). Thrust is found by Equation 17.

$$T_B = [0, 0, -4 \cdot T_{rotor}]_B^T. \tag{17}$$

Where T_{rotor} is found by Equation 7, determined during the performance modeling in subsection II-B.

$$G_E = [0, 0, m \cdot 9.81]^{T\cdot} \tag{18}$$

Note that in the coordinate frames presented in Figure 2, gravity has a positive sign and T_E a negative sign, considering positive is defined as downwards.

Be aware that the equations for thrust and surface are computed by performance models as established in subsection II-B.

D. State estimation – Particle Filter

The deterministic method excels in simplicity and computational speed, however, has limitations for a number of reasons. First, the estimation of the drag coefficient is uncertain as elaborated on in subsection IV-A. Second, the PWM to thrust mapping is an oversimplified one, as it does not take into account factors such as wind and the drop in voltage throughout flight. Lastly, it is difficult to know the exact biases and noise levels of the on-board sensors, such as those of the IMUs. All of the above uncertainties are not adapted to in the deterministic model, and for that reason, this paper implements a particle filter. This provides state estimates, but also eliminates noise and quantifies the uncertainty for each state.

This paper addresses a regularized Sequential Importance Resampling (SIR) particle filter as proposed by [28] with re-sampling steps as suggested by [29]. Particle filters are non-parametric, recursive Bayes filters, where non-parametric means the data is not required to fit a normal distribution (but can be of any unknown distribution). This makes the filter a suitable candidate, considering wind is highly stochastic, especially under the influence of obstacles (i.e. buildings, trees) which are to be found at the low altitudes drones operate in. The particle filter works by generating a large set of possible states in which the system can be, and denotes each of those states as a particle. The probability of these states being the true state is referred to as the weight, where a higher weight means a higher chance of being used and reconsidered for the next round of computations (re-sampled). Generally, particle filters are found to work best on low-dimensional problems, meaning the number of observations should be higher than the number of dimensions. In this case, therefore at all times, the data should have more than 10 observations (since we have 10 independent input states). The steps taken to convert this methodology into a particle filter are shown below.

D.1 States and observable states

First, there is determined what states are required for the computations in this research. These are displayed in vector X.

$$X = \begin{bmatrix} x_t & y_t & v_{x,w,t} & v_{y,w,t} & v_{x,tas,t} \\ v_{y,tas,t} & v_{x,gnd,t} & v_{y,gnd,t} & a_{x,t} & a_{y,t} \\ pwm_t & \theta_t & \phi_t & \psi_t & C_{D,t} \end{bmatrix}$$

Note that in this representation, $C_{D,t}$ is added as a state, meaning it will be estimated recursively (as opposed to it being an assumed constant in the deterministic method). As initial guesses, values from subsection II-B are used. The observed states are represented in Vector Y.

$$Y = \begin{bmatrix} x_t & y_t & v_{x,gnd,t} & v_{y,gnd,t} & a_{x,t} & a_{y,t} \\ pwm_t & \theta_t & \phi_t & \psi_t & & \end{bmatrix}$$

This formulation implies that with the information available, $v_{gnd,t}$, $v_{tas,t}$ and $C_{D,t}$ are un-observable and thus should be deduced from our system of equations.

D.2 State equations

To make use of the particle filter, the equations from subsection II-C need to be written as ordinary differential equations (ODEs). To compute all states for each time step Δt, these ODEs are discretized, resulting in the following set of equations. First, the state equations are shown.

Position:

$$x_t = x_{t-1} + v_{x,gnd,t-1} \cdot \Delta t \tag{19}$$

$$y_t = y_{t-1} + v_{y,gnd,t-1} \cdot \Delta t \tag{20}$$

Ground speed:

$$v_{x,gnd,t} = v_{x,gnd,t-1} + a_{x,t-1} \cdot \Delta t \tag{21}$$

$$v_{y,gnd,t} = v_{y,gnd,t-1} + a_{y,t-1} \cdot \Delta t \tag{22}$$

In the above equations, the acceleration is found by the kinematic relation as expressed below in Equation 23 and Equation 24.

$$a_{x,t} = \frac{(T_{x,t} + D_{x,t} + G_{x,t})}{m} \tag{23}$$

$$a_{x,t} = \frac{(T_{x,t} + D_{x,t} + G_{x,t})}{m} \tag{24}$$

Where gravity components x- and y are zero as the computations are performed in the inertial frame of reference. In accordance with the thrust performance model, the thrust equation is discretized to Equation 25 and Equation 26.

$$T_{x,t} = R_{z2x} \cdot 4 \cdot (0.0226 \cdot PWM_{avg} - 27.01) \tag{25}$$

$$T_{y,t} = R_{z2y} \cdot 4 \cdot (0.0226 \cdot PWM_{avg} - 27.01) \tag{26}$$

The drag term in the earth frame is stated as in Equation 27.

$$D_{x,t} = \frac{1}{2}\rho v_{x,tas,t}^2 \cdot S_t \cdot C_{d,t} \tag{27}$$

$$D_{y,t} = \frac{1}{2}\rho v_{y,tas,t}^2 \cdot S_t \cdot C_{d,t} \tag{28}$$

Note that the results presented in Section III follow these quadratic relations to speed. An alternative, a linear drag model is considered for a few cases and compared. This is further explored in Section IV.

Continuing on the above formulas, the true airspeed is found by following the triangle method:

$$v_{x,tas,t} = v_{x,gnd,t} + v_{x,wind,t} \tag{29}$$

$$v_{y,tas,t} = v_{y,gnd,t} + v_{y,wind,t} \tag{30}$$

Also note that the surface area (including the pole) as from subsection II-B is discretized as shown in Equation 9.

$$S_t = [0.0089 \cdot \nu_t + 0.039] \tag{31}$$

Which includes the area of the propellers. In this equation, pitch and roll are combined to tilt (i.e. inclination) which can be approximated as by Equation 32.

$$\nu_t = \arctan(\sqrt{\tan\theta_t{}^2 + \tan\phi_t{}^2}) \tag{32}$$

D.3 Observation equations

The states from the Y-vector are deduced from the on-board sensors (i.e. the Pixhawk Cube Orange). These are observed and therefore go from timestep t to timestep t. First, Equation 33 converts latitude and longitude to x (Northing) and y (Easting) coordinates, as this is required to ensure the position update equations work properly.

$$x_t, y_t = longlat2xy(lat_t, long_t) \tag{33}$$
$$\hat{x}_t = x_t + \zeta \tag{34}$$
$$\hat{y}_t = y_t + \zeta \tag{35}$$

Similarly, ground speed is deducted as shown below.

$$\hat{v}_{x,gnd,t} = v_{gnd,t} \cdot \cos(g.c.) \tag{36}$$
$$\hat{v}_{y,gnd,t} = v_{gnd,t} \cdot \sin(g.c.) \tag{37}$$

Then, the accelerations are converted from the body to the earth frame, by multiplying with the rotation matrix shown in Equation 1.

$$\hat{a}_{x,t} = [R_{x2x} \cdot a_{x,t}] + [R_{y2x} \cdot a_{y,t}] + [R_{z2x} \cdot a_{z,t}] + \zeta \tag{38}$$
$$\hat{a}_{y,t} = [R_{x2y} \cdot a_{x,t}] + [R_{y2y} \cdot a_{y,t}] + [R_{z2y} \cdot a_{z,t}] + \zeta \tag{39}$$

The states $p\hat{w}m_t$, $\hat{\theta}_t$, $\hat{\phi}_t$, $\hat{\psi}_t$ are directly observed from the on-board flight controller, as shown by Equation 40 to Equation 43.

$$p\hat{w}m_t = pwm_t + \zeta \tag{40}$$
$$\hat{\theta}_t = \theta_t + \zeta \tag{41}$$
$$\hat{\phi}_t = \phi_t + \zeta \tag{42}$$
$$\hat{\psi}_t = \psi_t + \zeta \tag{43}$$

D.4 Evolution equations

The particle filter makes use of evolution equations, which show how the states evolve over time. To control this rate of change, parameter alpha is introduced. This parameter helps determine a logical set of possible states for the system (i.e. the particles). These particles are in turn used to compute the other states and are compared to the measurements, in an attempt to estimate the true state. For the non-observable states, this results in the following set of equations.

$$v_{x,gnd,t} = \alpha \cdot v_{x,gnd,t-1} + \zeta \tag{44}$$
$$v_{y,gnd,t} = \alpha \cdot v_{y,gnd,t-1} + \zeta \tag{45}$$
$$v_{x,wind,t} = \alpha \cdot v_{x,wind,t-1} + \zeta \tag{46}$$
$$v_{y,wind,t} = \alpha \cdot v_{y,wind,t-1} + \zeta \tag{47}$$

And for the directly observable states:

$$pwm_t = \alpha \cdot pwm_{t-1} + \zeta \tag{48}$$
$$a_{x,t} = \alpha \cdot a_{x,t-1} + \zeta \tag{49}$$
$$a_{y,t} = \alpha \cdot a_{y,t-1} + \zeta \tag{50}$$
$$a_{z,t} = \alpha \cdot a_{z,t-1} + \zeta \tag{51}$$
$$\theta_t = \alpha \cdot \theta_{t-1} + \zeta \tag{52}$$
$$\phi_t = \alpha \cdot \phi_{t-1} + \zeta \tag{53}$$
$$\psi_t = \alpha \cdot \psi_{t-1} + \zeta \tag{54}$$

Where α is the model parameter and ζ is a number drawn from a normal distribution around zero, in which σ is the square root of the variance, computed by the observation variables and applying a least squares regression:

$$\alpha = \frac{\sum_{t=1}^{n} \tilde{x}_{t-1} \cdot \tilde{x}_{t-1}}{\sum_{t=1}^{n} \tilde{x}_{t-1}^2} \tag{55}$$

$$\sigma^2 = Var(x_t - \alpha \cdot \tilde{x}_{t-1}) \tag{56}$$

Similar to [23]. These equations are applied on a full flight set of 300 seconds and are only valid for the observed variables.

Bearing in mind the trade-off between computational speed and accuracy, the number of particles is set to 50000. Also, note that the Δt = 0.1. In overview, the parameters are listed in Table III:

D.5 Noise levels

For the noise covariance matrix, it is assumed the noises of the states are independent so only diagonal entries are present. When the states converge too quickly, noises are enlarged, and when the states do not converge, noise levels are brought down. In the end, the results in this paper are attained following the noise levels as stated in Figure 11.

Settings for the SIR particle filter
(with Δt = 0.1, N = 50000)

	α	σ
x, y	1.000, 0.985	1.000, 1.000
vxw', vyw'	1.0, 1.0	0.080, 0080
vxtas', vytas'	1.000, 1.000	0.059, 0.059
vxgnd, vygnd	0.998, 0.999	0.020, 0.020
ax, ay	0.999, 0.997	0.317, 0.317
pwm	1.0	250
theta, phi, psi	0.998, 0.980, 1.0	0.020, 0.009, 0.001
C_D' / k'	1.000	0.041

TABLE III
SETTINGS FOR THE PARTICLE FILTER (WITH ' INDICATING NON-OBSERVABLE STATES)

$$\sigma = \begin{bmatrix} 0.8 & 0.8 & 1 & 1 & 1.50 & 1.70 \\ 1.75 & 1.85 & 0.35 & 0.30 & 0.35 & \end{bmatrix}$$

Fig. 11. Vector displaying the diagonal entries of the noise covariance matrix.

III. EXPERIMENT DESIGN & RESULTS

This section is structured into five parts. First, the experiment design is shown in detail, including its independent variables for which the methodology presented is tested. From that, the drag coefficient is tuned to a constant for which the results are most in line with the reference. Third, the results from the deterministic state estimation are presented. Then, the results of the particle filtering script will be addressed. Fifth and last, the deterministic method and the particle filter will be compared by the hand of two examples, one being in hover and one being in cruise.

Throughout the section, there will be reflected upon the World Meteorological Organization (WMO) requirements for wind measurement. These state that an accuracy for wind speed measurements better than 0.5 m/s (for v_w<$5m/s$), and better than 10% (for v_w>$5m/s$) is considered sufficient for practical wind measurements [30]. Throughout, the RMSE is used as a suggestive indicator of accuracy.

A. Experiment Design

The model is validated by comparing it with real-world data from flight experiments. The only data available for this specific type of quadcopter is from last year's outside flight experiment, where there were very low wind speeds (0-3 m/s).

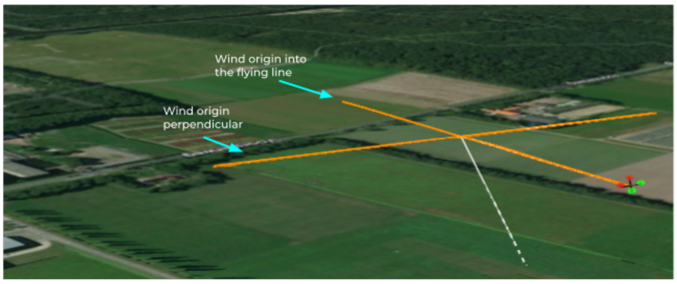


Fig. 12. Two example lines at 30m, into the wind and perpendicular to the wind.

This year, a new flight experiment is performed at the NLR drone center in Marknesse, with patterns as shown in Figure 12. The direction of the cross shape is pointed toward the wind direction origin, which allows the drone to experience the highest true airspeed (towards) as well as the lowest true airspeed (away) for a given ground speed. This means that into the wind, the tilt angle is on average 12°, which corresponds to a high thrust in the xy pane and thus a high true airspeed. On the contrary, away from the wind, the tilt angle is on average 2° for the same ground speed. In that case, there is very little thrust on the xy pane and therefore, in steady-state conditions, the true airspeed is smaller.

Over the course of three hours of testing, there was a moderate breeze varying between 8 m/s and 1 m/s, while the direction gradually shifted from 190° to 250°. The skies were clear, with temperatures around 11°C and humidity of 79%.

Independent Variables on experiment day

ground speed	[m/s]	[0, 5, 7, 9]
altitude	[m]	[30, 60, 90]
direction w.r.t wind	[-]	[into, perp., away from]

TABLE IV
INDEPENDENT VARIABLES ON TESTING DAY, WITH WIND SPEEDS BETWEEN 2-8 M/S FROM 190-250°.

Note that in subsection III-C and subsection III-D, for fair comparison the RMSEs in the tables have been computed by considering the same time length for each line (10 seconds of cruise), where each RMSE in the tables corresponds to one line of flight.

B. Drag coefficient adjustments

Before stating the results of the experiment flight, an exploration of the correctness of the C_D is performed. With the C_D as found in the performance modeling section ($C_D = 0.14$), a large offset in magnitude is found. It is likely that some (aerodynamic) effects are not modeled, and so a sensitivity analysis has been performed to see the effect of varying the C_D on the magnitude of the estimation. The outcome is shown in Figure 13.

Considering the C_D only scales the vectors (and does not distort the shape), the wind direction estimation remains unaffected. This is displayed in Figure 14.

For the 30m case, the difference between the lowest and the highest C_D results in an offset of 5.62 m/s and 0.77 deg. To check whether the effect of C_D is consistent, the same analysis is performed at an altitude of 60m. For the same set of C_Ds, this shows a difference of 5.52 m/s and 0.82 deg, meaning the influence on magnitude and direction is similar for different altitudes.

From analyzing these results, it is found that when a $C_D = 0.55$ was set instead, this always results in an RMSE under 1 when taking a 10-second window of cruise (RMSE 0.90 analyzed between T=20 and T=30). Therefore, from this point on, the C_D will be fixed to 0.55. Potential reasons for this mismatch are discussed in Section IV. With a similar analysis, for hover conditions, a C_D of 1.70 was found to give the lowest overall RMSE, and thus this value will be used from this point on.

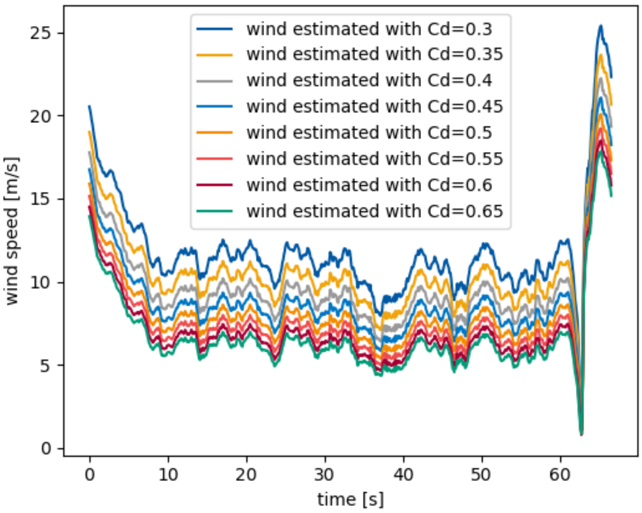


Fig. 13. Sensitivity analysis for C_D during the 5 m/s flight at 30m altitude: the magnitude of wind estimation decreases upon increasing the C_D, and the shape remains unaffected. The lowest RMSE in steady state (T=15 to T=55, where pitch remains constant) is found for C_D = 0.55.

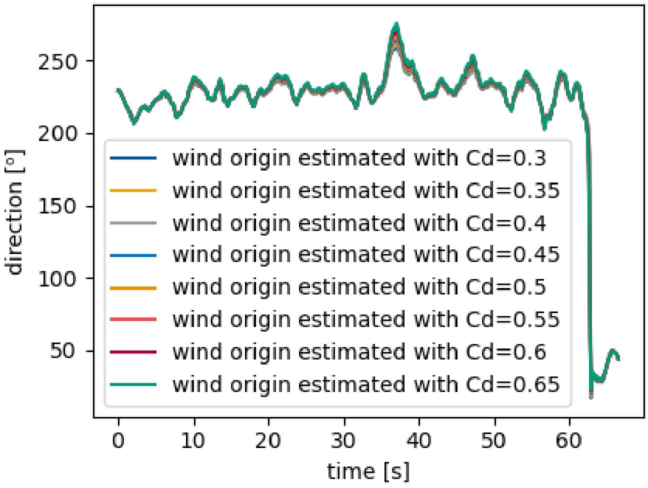


Fig. 14. Sensitivity analysis for C_D during the 5 m/s flight at 30m altitude: the wind direction estimation remains unaffected by a change in C_D.

C. Results: deterministic script

Third in this section, the results of the deterministic method are presented. This is done by the help of RMSE tables in which one independent variable is changed, to see its effect.

For each of these tables, eight rows of information are shown. First, the average wind speed as measured by the drone (Wind speed est avg by drone), and the average wind speed as measured by the reference sensor (Wind speed ref avg) are displayed. Second, the WMO requirements are recapped for those specific wind conditions. Third, the attained mean absolute percentage error (MAPE) for magnitude, the root mean square error (RMSE) for magnitude, and the RMSE for direction estimation are shown. This way, there can be observed for each of the flights whether WMO requirements are met.

C.1 Varying altitudes

To analyze the effect of altitude on the estimation, a hover test has been performed at varying altitudes, where the reference drone is used both for estimating and for its reference sensor data. The results of this are displayed in Table V. The table shows that there is no clear sign of an in- or decreases in RMSE depending on altitude.

		Deterministic Method		
Altitude	[m]	30	60	90
wind speed avg by drone	[m/s]	6.14	5.12	5.38
wind speed avg by ref	[m/s]	5.37	4.97	6.43
MAPE req. by WMO	[%]	10	10	10
RMSE req. by WMO	[m/s]	0.54	0.50	0.64
RMSE req. by WMO (direction)	[°]	5	5	5
MAPE to reference signal	[%]	12.9	3.3	19.8
RMSE to reference signal	[m/s]	0.87	0.35	1.10
RMSE to reference signal (direction)	[°]	13.6	4.7	13.8

TABLE V
Deterministic method: RMSE upon varying altitude (at hover condition)

C.2 Varying directions

To analyze the effect of varying direction, the same line of 5 m/s cruise has been performed in various directions. Flight lines are performed into the wind, perpendicular to the wind, and away from the wind. The results of this are displayed in Table VI.

		Deterministic method		
Direction w.r.t wind	[-]	into	perp	away
wind speed avg by drone	[m/s]	6.3	3.61	6.11
wind speed avg by ref	[m/s]	5.99	3.84	5.68
MAPE req. by WMO	[%]	10	10	10
RMSE req. by WMO	[m/s]	0.60	0.50	0.57
RMSE req. by WMO (direction)	[°]	5	5	5
MAPE to reference signal	[%]	9.4	35.7	86.1
RMSE to reference signal	[m/s]	0.74	1.36	3.01
RMSE to ref signal (direction)	[°]	13.9	34.5	80.7

TABLE VI
Deterministic method: RMSE upon varying direction w.r.t wind (at altitude 30m, ground speed 5 m/s)

There is a slight increase in error for flying perpendicular to the wind as opposed to into the wind. Flying away from the wind however shows a drastic increase in RMSE (3.01 m/s, 80.7° for a 5 m/s flight and 3.7 m/s and 131 ° for a 9 m/s) flight, making this unpractical. This can be attributed to the fact that the true airspeed is very small in this case (e.g. a 6 m/s of wind and 5 m/s ground speed leaves 1 m/s true airspeed), for which the quadratic model is not suited, showing one of its limitations. A suggestion is to use the linear drag relationship for low airspeeds, as is further explored in subsection IV-A.

C.3 Varying ground speeds

Lastly, the effect of varying speeds has been analyzed. Here, there is flown with 5, 7 and 9 m/s , all into the wind for a fair comparison. The results are stated in Table VII.

Although the 9 m/s case does not show improvement over the 7 m/s case, the RMSE is in a lower order than for the

Ground speed	[m/s]	Deterministic method 5	7	9
wind speed avg by drone	[m/s]	6.11	6.06	6.41
wind speed avg by ref	[m/s]	5.68	5.86	5.93
MAPE req. by WMO	[%]	10	10	10
RMSE req. by WMO	[m/s]	0.57	0.59	0.59
RMSE req. by WMO (direction)	[°]	5	5	5
MAPE to reference signal	[%]	86.1	5.7	7.5
RMSE to reference signal	[m/s]	3.01	0.38	0.51
RMSE to reference signal (direction)	[°]	31.4	8.6	8.1

TABLE VII
Deterministic method: RMSE upon varying ground speed (at altitude 30m, into the wind)

Altitude	[m]	Particle Filter 30	60	90
wind speed avg by drone	[m/s]	6.27	5.89	5.46
wind speed avg by ref	[m/s]	5.37	4.97	6.43
MAPE req. by WMO	[%]	10	10	10
RMSE req. by WMO	[m/s]	0.54	0.50	0.64
RMSE req. by WMO (direction)	[°]	5	5	5
MAPE to reference signal	[%]	9.1	15.6	17.6
RMSE to reference signal	[m/s]	0.72	0.94	0.98
RMSE to ref. signal (direction)	[°]	13.4	6.9	17.2

TABLE VIII
Particle filtering method: showing the effect of increasing altitude on RMSE

5 m/s case. This hints at a decreasing RMSE for increased ground speeds in the same direction into the wind, confirming the hypothesis that the quadratic relationship is best suited for higher true airspeeds. This is shown in Figure 15. Also, note that the RMSE for direction estimation for 7 m/s and 9 m/s is much lower than that of 5 m/s.

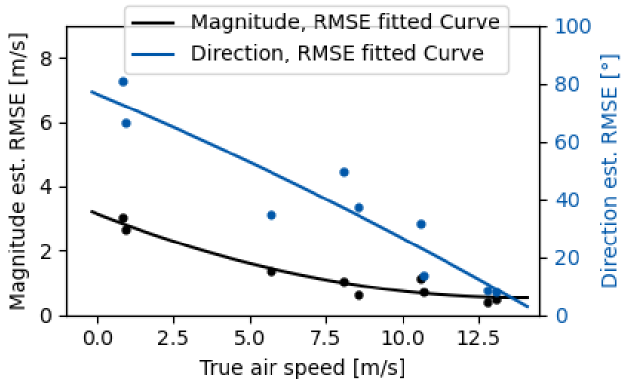


Fig. 15. Overview of the RMSEs for flights performed in cruise at 30m altitude. By flying at higher true airspeeds, a lower RMSE is observed.

Linking the results from the deterministic tests to practical use, into the wind at 7 m/s as well as at 9 m/s are scenarios for which the WMO requirements are met. For the other scenarios, the RMSE is slightly above this value, or far off in case of flying away from the wind.

D. Results: particle filter

Fourth, the particle filter results are displayed under the same flight conditions, i.e. on the same dataset. Similarly, the effects of varying the altitudes and directions with respect to the wind and ground speed are presented.

D.1 Varying altitudes

Similar to the preceding subsection, the same hover test has been performed at varying altitudes. The results of this are displayed in Table VIII. The results show a slight increase in RMSE for increased altitude, although it can be considered insignificant when compared to the effect of changing direction or ground speed. Note that the RMSEs are close to but not low enough to meet the desired accuracy according to [30].

Notably, the particle filter has very similar RMSEs to the deterministic method, not improving or degrading the performance significantly as opposed to subsection III-C. The particle filter however shows more stable behavior (lower responsivity) when looking at one of the examples in e.g. Figure 16.

D.2 Varying directions

To analyze the effect of varying directions, the same data set of 5 m/s flights in cruise has been performed in various directions, similar to how it is varied in subsection III-C. This is done into the wind, perpendicular to the wind, and away from the wind. The results of this are displayed in Table IX.

Direction w.r.t wind	[-]	Particle Filter into	perp	away
wind speed avg by drone	[m/s]	5.88	3.41	3.69
wind speed avg by ref	[m/s]	5.99	3.84	5.68
MAPE req. by WMO	[%]	10	10	10
RMSE req. by WMO	[m/s]	0.60	0.50	0.57
RMSE req. by WMO (direction)	[°]	5	5	5
MAPE to reference signal	[%]	8.9	17.8	54.6
RMSE to reference signal	[m/s]	0.54	0.55	2.01
RMSE to reference signal (direction)	[°]	13.5	11.7	27.6

TABLE IX
Particle filtering method: RMSE upon varying direction (at altitude 30m, $v_g = 5$ m/s)

In line with expectations, it is found that the particle filter produced the lowest RMSE into the wind, as it then encounters the highest true airspeed. When tested perpendicular to the wind, there is a slight increase in RMSE. Finally, analyzed away from the wind, there is a significant increase in error.

D.3 Varying ground speed

Lastly, the effect of varying ground speeds has been analyzed. Here, there is flown with 5, 7, and 9 m/s, into the wind each time for a fair comparison. The results are stated in Table X. Although tested for only three ground speeds, there appears to be a clear trend of decreasing RMSE for an increased ground speed.

Linking this to practical use, into the wind at 9 m/s seems to be the only scenario in which the WMO requirements are met [30]. For the other scenarios, the RMSE is slightly above this value, and far off in case of flying away from the wind. Also, it should be emphasized that the filter requires close initial guesses for the non-observable states in order to converge.

		Particle Filter		
Ground speed	[m/s]	5	7	9
wind speed avg by drone	[m/s]	3.69	6.91	5.63
wind speed avg by ref	[m/s]	5.68	5.86	5.93
MAPE req. by WMO	[%]	10	10	10
RMSE req. by WMO	[m/s]	0.57	0.59	0.59
RMSE req. by WMO (direction)	[°]	5	5	5
MAPE to reference signal	[%]	54.6	14.8	6.9
RMSE to reference signal	[m/s]	2.01	1.17	0.44
RMSE to reference signal (direction)	[°]	31.4	18.0	9.8

TABLE X
PARTICLE FILTERING METHOD: RMSE UPON VARYING GROUND SPEED (AT ALTITUDE 30M, INTO THE WIND)

E. Comparison: deterministic and particle filter

In this subsection, both the deterministic method and the particle filter are shown alongside, by means of an example in hover and an example in cruise condition.

E.1 Comparison: hover scenario

The first scenario to be analyzed is hovering at 30m altitude. RMSEs found for both are presented in Table V and Table VIII. This example is shown in Figure 16 and Figure 17.

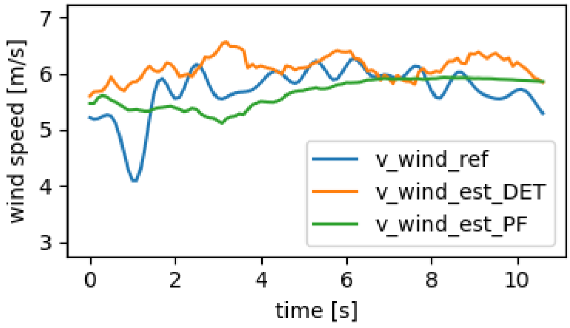


Fig. 16. 10 seconds of hover at 30m altitude, estimating wind magnitude using the deterministic method (DET) and the particle filter (PF).

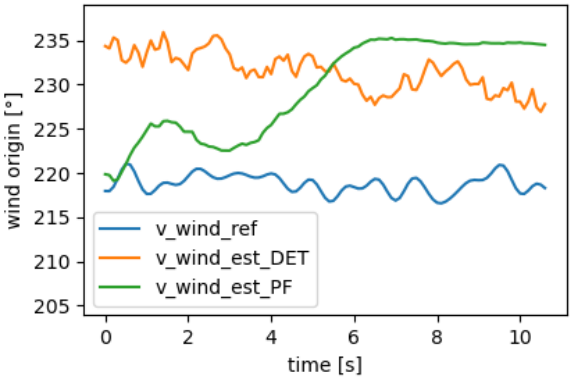


Fig. 17. 10 seconds of hover at 30m altitude, estimating wind direction using the deterministic method (DET) and the particle filter (PF).

When comparing the deterministic method to the particle filter signal, the particle filter has a slightly lower RMSE in magnitude and a slightly higher RMSE in terms of direction. This can be confirmed by comparing Table V and Table VIII. It must be noted that the computation time required for the particle filter is much higher than for the deterministic method (in the order of 10x). From this, there can be said that the slight benefit does not outweigh the added computational strain. Also, under the current settings, the filter does not track the signal very well as it is unresponsive to changes in magnitude.

E.2 Comparison: cruise scenario

A line of flight of cruise is analyzed accordingly. The outcome for the deterministic method and the particle filter is shown in Figure 18. Similar RMSE's are found for the deterministic method and the particle filter, as can be seen by comparing Table VII and Table X. Similar to hover, the particle filter does not improve performance significantly.

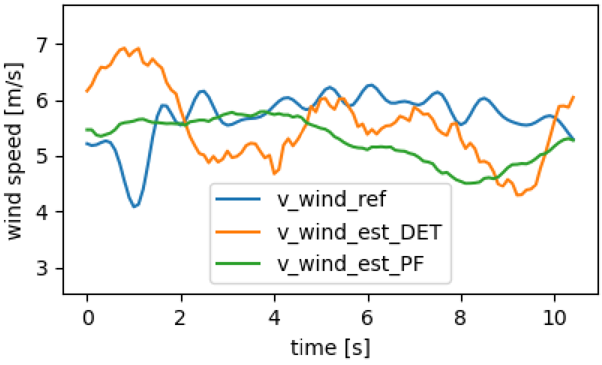


Fig. 18. Example of 10 seconds of wind magnitude estimation while in cruise (v_{gnd} 5 m/s at 30m), using the deterministic method and the particle filter.

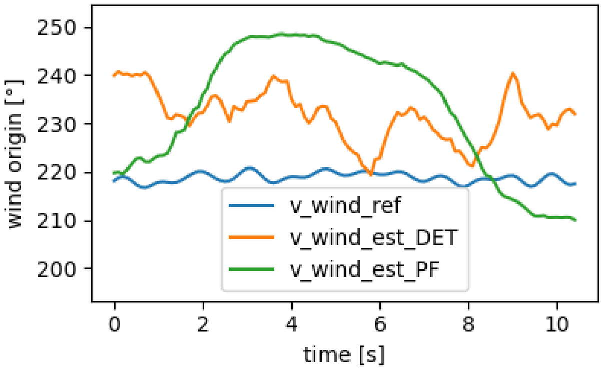


Fig. 19. Example of 10 seconds of wind direction origin estimation while in cruise (v_{gnd} 5 m/s at 30m), using the deterministic method and the particle filter.

Interestingly, the particle filter's magnitude seems to be in line with the magnitude of the reference data although the direction estimation is not. This means that although the magnitude is correct, the x- and y-magnitude components are not computed in proportion. When verifying this hypothesis, it shows that the X component gets overestimated and the Y component gets underestimated slightly.

E.3 Towards practical use

Comparing the deterministic method and the particle filtering method in terms of practical use, one method does not provide strong accuracy benefits over the other. The particle filter offers the advantage of addressing the uncertainty by providing upper- and lower limits for wind, which could be a useful figure for operators. On the downside, the filter is currently not optimized for computational speed, and so it can not be used effectively in real-time. Instead, it can only be deployed on short periods of the strict cruise for practical use.

IV. Discussion & Recommendations

Following the experiment and its result, a critical view on the results from Section III is presented. This is followed up by a set of recommendations for future work.

A. Discussion

In this subsection, the significance and some interesting findings related to the results section are discussed.

A.1 Drift flight analysis

In one of the experiment flights, the quadcopter is set to altitude hold and is let to drift (or coast) away with the wind for 110 seconds, reasoning that the quadcopter's terminal ground velocity should approximately become equal to the wind velocity it is carried with. From about 30 seconds onwards, it converges to an estimation of the wind magnitude. In a 10-second window, an RMSE of 1.82 m/s and 51.1 degrees is found. In practical use, if space and time allow for it, this method could be used to get preliminary estimates and use those as an initial estimate for the estimation scripts.

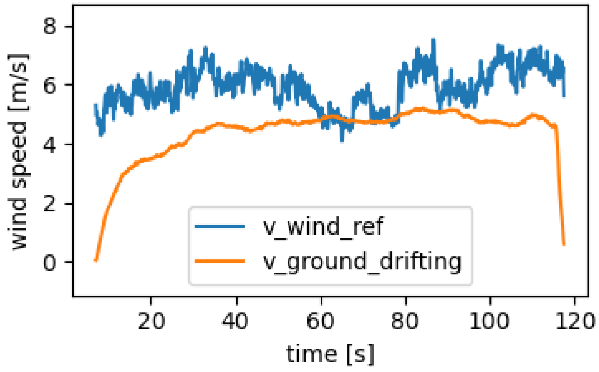


Fig. 20. Full line of drift, estimating wind magnitude as the ground speed.

The fact that the drifting quadcopter underestimates the wind is attributed to the quadcopter's weight and complex shape, not allowing it to reach full speed. The fact that the wind direction detection is off is unexpected, considering this is estimated by observing the ground course over the course of a 200m line of drift, which is a range the GPS should be able to track very well. As there is still a discrepancy with the reference sensor's indicated wind direction, this might hint at a compass calibration offset.

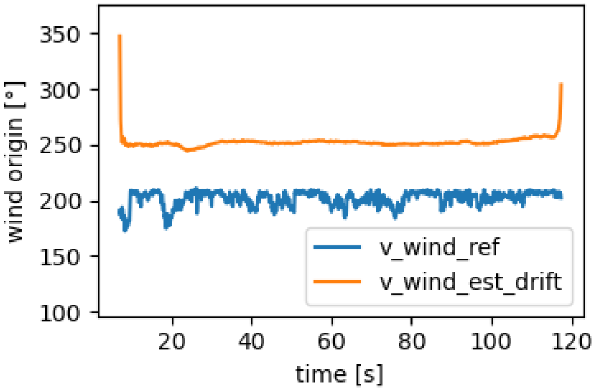


Fig. 21. Full line of drift, estimating wind direction from the ground course.

A.2 Particle filtering

The results of the particle filter show potential for improving state estimation in this context. In most cases, there is proper convergence for the observable states and less convergent behavior for the un-observable states (See hover at 30m example in Figure 25 in the appendix). This run is analyzed with noise levels which are estimated from prior flight data and are then adjusted slightly. For the un-observable states, including wind, noise levels required tweaking as well. Higher process noise for wind means better response to gusts at the cost of higher uncertainty and worse convergence. Here it must be noted that a non-converging wind state is congruent with expectations considering the wind keeps changing slightly over time.

In contrast to papers like [14], [21] and [13], this paper attempts to find the thrust performance model from the PWM-thrust mapping. One interesting result from the particle filtering is that the script only works if the noise levels on the PWM state are set high, meaning the wind is not estimated properly when PWM converges. As thrust is the only state coupled to the PWM, this hints at the fact that the PWM- thrust model could be oversimplified, i.e. that the true thrust value is not in line with the computed thrust value.

A.3 Quadratic vs. linear drag models

The flight lines which generally have resulted in low RMSEs are flights with higher true airspeeds (larger than 10 m/s, by combining wind speed and ground speed) and a drag modeled as quadratic with speed. The expected limitations of this model are that accuracy will drop for lower true airspeeds, which is where most literature today focuses on (e.g. [13], [20]). These authors make use of a linear relation with speed, as is shown in Equation 57.

$$v_{tas} = \frac{D_E}{k} \qquad (linear) \tag{57}$$

Which in the particle filter (as an ODE, discretized) becomes:

$$D_{x,t} = k \cdot v_{x,tas,t} \quad (58)$$

$$D_{y,t} = k \cdot v_{y,tas,t} \quad (59)$$

This drag model is implemented and tested for the two most extreme scenarios in terms of true airspeed (into the wind at the highest ground speed and away from the wind at the lowest ground speed). The results are shown in Figure 22 and Figure 23 and are in line with previous research, which suggests that the linear model as in Equation 57 is better applicable for lower v_{tas}.

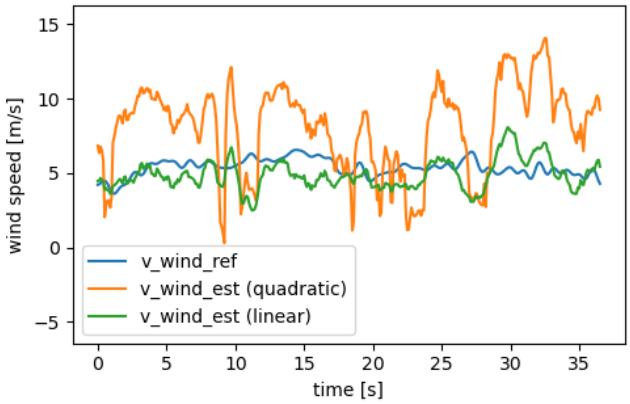


Fig. 22. Flight away from the wind, resulting in a low v_{tas}. Notice that the linear drag model (green) outperforms the quadratic drag model (orange).

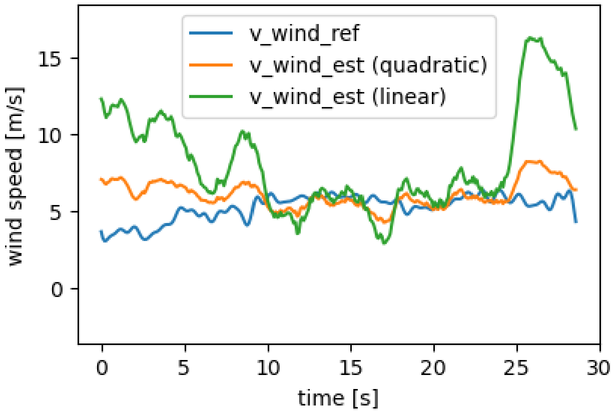


Fig. 23. Flight into the wind, resulting in a high v_{tas} ($\approx$ 13 m/s). Notice that the linear model (green) underperforms as opposed to the quadratic drag model (orange).

From these graphs, it can be confirmed that for high true airspeeds, generally, the quadratic model is more suited, and for low true airspeeds the linear model is better suited.

A.4 A discussion on C_D

The preliminary C_D analysis performed in Section III shows a mismatch between the estimated C_D (0.14) and the C_D which gave the overall lowest RMSE (0.55). This mismatch could be attributed to a number of factors. First, the indoor drag coefficient set-up consisted of lines of flight which might not have been enough to establish a quasi-steady flight, considering the IMUs still showed accelerations in the earth frame. Additionally, the indoor speed estimation is instantaneous and it is difficult to match the recorded speed with the exact time at which the quadcopter passed the camera (since there is no GPS data available indoors), so this brings uncertainty. The mismatch might also be attributed to other assumptions or effects which are not modeled for and should be included in the constant C_D (making it more logical to name it differently, similar to how [16] describes in its approach). Factors could include a different air density than expected, the roughness of the drone, or the permeability of the rotor area [31].

A.5 Outcome to the METSIS project

In this research, a number of methods are considered for use in the context of the METSIS project [32]. For each of these, there is assessed whether the solution is accurate enough, but also whether the required performance models are constructed adhering to the three design principles presented in Section I.

The use of the deterministic method results in RMSEs as low as 0.38 m/s and 5 degrees and is at this stage considered the most promising. It works in cruise, is computationally light, and is easiest to implement. It does require a test flight to adjust the C_D to the proper value. In the near future, it is recommended that there is found under which true airspeed it is better to have a linear relationship with speed, and make the drag model adopt to that.

The particle filter results in RMSEs as low as 0.44 m/s and 28 degrees in parts of strict cruise. With current noise levels, it converges but takes away some of the tracking abilities. Also, it performs much worse when the quadcopter's states are changing, i.e. sudden jumps in acceleration and deceleration are generally not dealt with successfully. This method at this stage requires more refinement and noise tuning before practical use. As currently it is not running in real-time, computational optimization is required to make this a more suitable candidate for the final use in on-board wind estimation for METSIS.

The drift method addressed in this section can be an interesting way to find the wind direction but underestimates the wind magnitude. Also, it requires the most time as well as physical space to deploy, making this method less suitable for METSIS, as it should be able to estimate wind during regular operations.

The use of sensors, as well as alternative methods (machine learning [3] or optical flow [33]) are discarded early on in this research, as discussed in the introduction.

An overview of the considered methods and their proposition to the METSIS project is presented in Table XI in the appendix.

A.6 Reviewing the reference sensor used

In this research, the reference sensor was assumed as a reference. One should be critical of the reference it is being measured to, however, considering the following was found during the experiments.

- The wind sensor used (Trisonica Mini) measures imperfectly. Wind tunnel tests show that for a wind tunnel set-up with 6 m/s wind and a fixed clamp, the sensor still has an interquartile range (IQR) of approximately 0.25 [32].
- The sensor found wind direction did not match the direction detected by the drifting quadcopter. The fact that the estimating quadcopter was closer to the reference measurements than to the drifting quadcopter could suggest that both the reference quadcopter and the estimating quadcopter have an offset in the true north.
- When converting the anemometer data, the pitch, roll, and yaw angle in the reference analysis are assumed constant (while in fact, this is fluctuating slightly $\sigma = 1.14°$).
- Although the sensor is on a 50cm pole, there still might be an influence of the rotors, which especially becomes pronounced under an angle, as then the rotor flow has a component in the x- and y- pane, where it interferes with the actual wind.

B. Recommendations

More work is suggested on the topics included in the following (non-exhaustive) list.

- More extensive testing under different particle filter settings is encouraged. Also, optimizing the algorithm for speed would help in making this application more practical. Currently, due to timing inefficiency, the particle filter should only be used on short snippets of flight data. To find the correct moment to engage the filter, simple logic could be applied to ensure the quadcopter is in quasi-steady flight.
- When determining the drag coefficient from indoor data, an indoor capturing system could help better capture the speed of the quadcopter, leading to a more accurate estimate of the C_D.
- In terms of the linear and quadratic modeling of drag, it is encouraged to perform a more extensive study on what drag model is best suited for which range of airspeeds. This could result in a threshold speed at which to switch drag model, or perhaps in a cross-over region of speed, where a (weighed) combination of a quadratic and linear term could be used.
- In terms of performance models, as a next step from here it is suggested to repeat the presented methodology presented in this paper on a variety of quadcopter types to see how well the methodology transfers to a different quadcopter.
- Of the performance models, the PWM-thrust model is an important one to improve. Concretely, one could lower the intercept of the chosen interpolation function by an amount depending on the wind-inflow. In order to make this quantitative, it is suggested to perform thrust tests again at a variety of wind speeds, above the speeds currently tested for (0 and 4 m/s). This could increase model fidelity and also extend the working of the proposed methods to flight parts such as take-off, landing, and turns.
- Considering future experiment designs, more variation in experiments is encouraged. Particularly, the focus should be on designing experiments in which the quadcopter encounters a range of true airspeeds, as that seems to have the largest influence on the RMSE. As this implementation suggests success at higher true airspeeds, it is suggested to perform experiments on days with stronger winds. Once the methodology in this paper has been verified more extensively, it is encouraged to test out the combination of the network part [32] with this methodology for sensorless wind estimation.

V. Conclusion

The aim of this study was to investigate how wind magnitude and direction can be estimated with a quadcopter using only on-board sensors. The quadcopter to be used had limited data available and so performance models and parameters were found using in-house tests. From there, two methods for wind estimation were proposed. First, a deterministic method was constructed from a kinematic approach and models drag as quadratic with speed. Second, a particle filter was implemented to improve the estimates and address uncertainty. The two methods scripts were validated by performing an experiment flight, both in hover and cruise. From analyzing the flight lines, the C_D which gave the best results is adjusted to from the one expected to work by testing (0.55 instead of 0.14). However, once this adjustment is done, this is kept constant during all flights to produce the results they do in this paper. The outcome of this research can be concluded in a set of main findings as described below.

- The deterministic method is able to estimate wind with an RMSE of between 0.3 and 3.7 m/s in terms of magnitude, and 4-35 degrees in terms of direction. It works for hover, as well as for different cruise speeds (5 m/s, 7 m/s, 9 m/s, and different altitudes (30 m, 60 m, 90 m). The WMO suggests an accuracy of 0.5 m/s and 5 degrees, which is met in some of the test cases (e.g. hover and cruise with true airspeeds above 10 m/s) but not met in other cases (e.g. low true airspeeds). Thus, more work is required to ensure the estimations work under a wider variety of conditions UAVs in U-Space are expected to operate in.
- The particle filter lowers the RMSE in most test cases slightly but insignificantly. With the current noise level estimates, there is slow convergence (order of 10-20 seconds) and low awareness of changes in wind speed. It is quite resilient to initial guess offsets for states which have observations, but prone to initial estimates for the states that have no observations, so concretely it is important to have reasonable initial values of $v_{w,x}$ and $v_{w,y}$.

In terms of opportunities, the outcomes support the previous work performed with a positive outlook for the METSIS project and show the potential for it to be a promising solution for the U-space weather information service.

Acknowledgements

I would like to express my deep gratitude to Dr. Emmanuel Sunil and Dr. Junzi Sun, my research supervisors, for their genuine enthusiasm, encouragement and constructive feedback during this research work.

I would also like to thank Professor Jacco Hoekstra for advising me during the milestones of this thesis work, and extend my appreciation to the Faculty of Aerospace Engineering and all (assistant) professors within which I was able to spar with. I would also like to extend my thanks to Royal NLR for facilitating this research and offering me resources, especially Jan-Willem van Doorn in conducting the experiments to verify & validate the work presented in this paper. Also, the Living Lab is thanked for supporting this extension of research financially.

Finally, I wish to thank my friends, family and especially my parents for their support and encouragement throughout my study.

References

[1] Single European Sky ATM Research 3 Joint Undertaking (EU body or agency), *European drones outlook study: unlocking the value for Europe*. LU: Publications Office of the European Union, 2017.

[2] K. Kabbabe Poleo, W. J. Crowther, and M. Barnes, "Estimating the impact of drone-based inspection on the Levelised Cost of electricity for offshore wind farms," *Results in Engineering*, vol. 9, p. 100201, Mar. 2021.

[3] S. Allison, H. Bai, and B. Jayaraman, "Wind estimation using quadcopter motion: A machine learning approach," *Aerospace Science and Technology*, vol. 98, p. 105699, Mar. 2020.

[4] L. Palazzetti, "Routing Drones Being Aware of Wind Conditions: a Case Study," June 2021.

[5] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Comprehensive Energy Consumption Model for Unmanned Aerial Vehicles, Based on Empirical Studies of Battery Performance," *IEEE Access*, vol. 6, pp. 58383–58394, 2018. Conference Name: IEEE Access.

[6] J. Pérez Castán, V. Gomez Comendador, A. Cardenas-Soria, D. Janisch, and R. Valdés, "Identification, Categorisation and Gaps of Safety Indicators for U-Space," *Energies*, vol. 13, p. 608, Jan. 2020.

[7] W. Thielicke, W. Hübert, U. Müller, M. Eggert, and P. Wilhelm, "Towards accurate and practical drone-based wind measurements with an ultrasonic anemometer," *Atmospheric Measurement Techniques*, vol. 14, pp. 1303–1318, Feb. 2021. Publisher: Copernicus GmbH.

[8] P. Bruschi, M. Piotto, F. Dell'Agnello, J. Ware, and N. Roy, "Wind Speed and Direction Detection by Means of Solid-state Anemometers Embedded on Small Quadcopters," *Procedia Engineering*, vol. 168, pp. 802–805, Jan. 2016.

[9] C. A. Wolf, R. P. Hardis, S. D. Woodrum, R. S. Galan, H. S. Wichelt, M. C. Metzger, N. Bezzo, G. C. Lewin, and S. F. de Wekker, "Wind data collection techniques on a multi-rotor platform," in *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 32–37, Apr. 2017.

[10] N. Vasiljević, M. Harris, A. Tegtmeier Pedersen, G. Rolighed Thorsen, M. Pitter, J. Harris, K. Bajpai, and M. Courtney, "Wind sensing with drone-mounted wind lidars: proof of concept," *Atmospheric Measurement Techniques*, vol. 13, pp. 521–536, Feb. 2020. Publisher: Copernicus GmbH.

[11] A. Rautenberg, M. Graf, N. Wildmann, A. Platis, and J. Bange, "Reviewing Wind Measurement Approaches for Fixed-Wing Unmanned Aircraft," *Atmosphere*, vol. 9, Oct. 2018.

[12] G. W. Donnell, "Wind Characterization Using Onboard IMU of sUAS | 2018 Atmospheric Flight Mechanics Conference." Archive Location: world.

[13] B. Luo, M. Zeng, and Q.-H. Meng, "A Wind Estimation Method with an Unmanned Rotorcraft for Environmental Monitoring Tasks," *Sensors*, vol. 18, p. 4504, Dec. 2018. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[14] P. P. Neumann and M. Bartholmai, "Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit," *Sensors and Actuators A: Physical*, vol. 235, pp. 300–310, Nov. 2015.

[15] S. Schafer, J. F. Horn, and J. Cooper, "Flight test measurement of ship airwake disturbances using small-scale rotorcraft," *Annual Forum Proceedings - AHS International*, vol. 4, no. January, pp. 2681–2695, 2015.

[16] G. Hattenberger, M. Bronz, and J.-P. Condomines, "Estimating wind using a quadrotor," *International Journal of Micro Air Vehicles*, vol. 14, p. 17568293211070824, Jan. 2022. Publisher: SAGE Publications Ltd STM.

[17] R. T. Palomaki, N. T. Rose, M. v. d. Bossche, T. J. Sherman, and S. F. J. D. Wekker, "Wind Estimation in the Lower Atmosphere Using Multirotor Aircraft," *Journal of Atmospheric and Oceanic Technology*, vol. 34, pp. 1183–1191, May 2017. Publisher: American Meteorological Society Section: Journal of Atmospheric and Oceanic Technology.

[18] M. Marino, A. Fisher, R. Clothier, S. Watkins, S. Prudden, and C. S. Leung, "An Evaluation of Multi-Rotor Unmanned Aircraft as Flying Wind Sensors," *International Journal of Micro Air Vehicles*, vol. 7, pp. 285–299, Sept. 2015. Publisher: SAGE Publications Ltd STM.

[19] M. Simma, H. Mjøen, and T. Boström, "Measuring Wind Speed Using the Internal Stabilization System of a Quadrotor Drone," *Drones*, vol. 4, p. 23, June 2020.

[20] T. Luukkonen, "Modelling and control of quadcopter," p. 26.

[21] L. Sikkel, G. Croon, C. De Wagter, and Q. Chu, "A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost MEMS IMUs," pp. 2141–2146, Oct. 2016.

[22] N. Y. Ko, I. Choi, G. Song, and W. Youn, "Three-Dimensional Dynamic-Model-Aided Navigation of Multirotor Unmanned Aerial Vehicles," *IEEE Access*, vol. PP, pp. 1–1, Nov. 2019.

[23] J. Sun, H. A. Blom, J. Ellerbroek, and J. M. Hoekstra, "Particle filter for aircraft mass estimation and uncertainty modeling," *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 145–162, Aug. 2019.

[24] K. Shimada and T. Nishida, "Particle Filter-Model Predictive Control of Quadcopters," Oct. 2014.

[25] F. Bañuelos-Ruedas, C. Angeles-Camacho, and S. Rios-Marcuello, "Analysis and validation of the methodology used in the extrapolation of wind speed data at different heights," *Renewable and Sustainable Energy Reviews*, vol. 14, pp. 2383–2391, Oct. 2010.

[26] J. D. Anderson, *Fundamentals of aerodynamics*. McGraw-Hill, 2011.

[27] J. González-Rocha, C. A. Woolsey, C. Sultan, and S. F. J. De Wekker, "Sensing Wind from Quadrotor Motion," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 4, pp. 836–852, 2019. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G003542.

[28] C. Musso, N. Oudjane, and F. Gland, "Improving Regularised Particle Filters," (New York, NY), pp. 247–271, Springer New York, 2001. Book Title: Sequential Monte Carlo Methods in Practice.

[29] J. S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].

[30] WMO, "Measurement of meteorological variables - Chapter 5: Measurement of surface wind."

[31] J. M. Cano, "Quadrotor UAV for wind profile characterization," *undefined*, 2013.

[32] E. Sunil, R. Koerse, S. van Selling, J.-W. Doorn, T. Brinkman, and J. Sun, "METSIS: Hyperlocal Wind Nowcasting for U-space," Dec. 2021.

[33] H. W. Ho, G. C. de Croon, and Q. Chu, "Distance and velocity estimation using optical flow from a monocular camera," *International Journal of Micro Air Vehicles*, vol. 9, pp. 198–208, Sept. 2017. Publisher: SAGE Publications Ltd STM.

[34] A. Teimourian and D. Firouzbakht, "A PRACTICAL METHOD FOR DETERMINATION OF THE MOMENTS OF INERTIA OF UNMANNED AERIAL VEHICLES," Sept. 2013.

Appendix A: Plots and figures

State Estimation - Example analysis on Particle Filter

This appendix shows the output of running the particle filter on an example case of hover at 30m altitude. To start with, Figure 24 shows all states as observed by the on-board sensors.

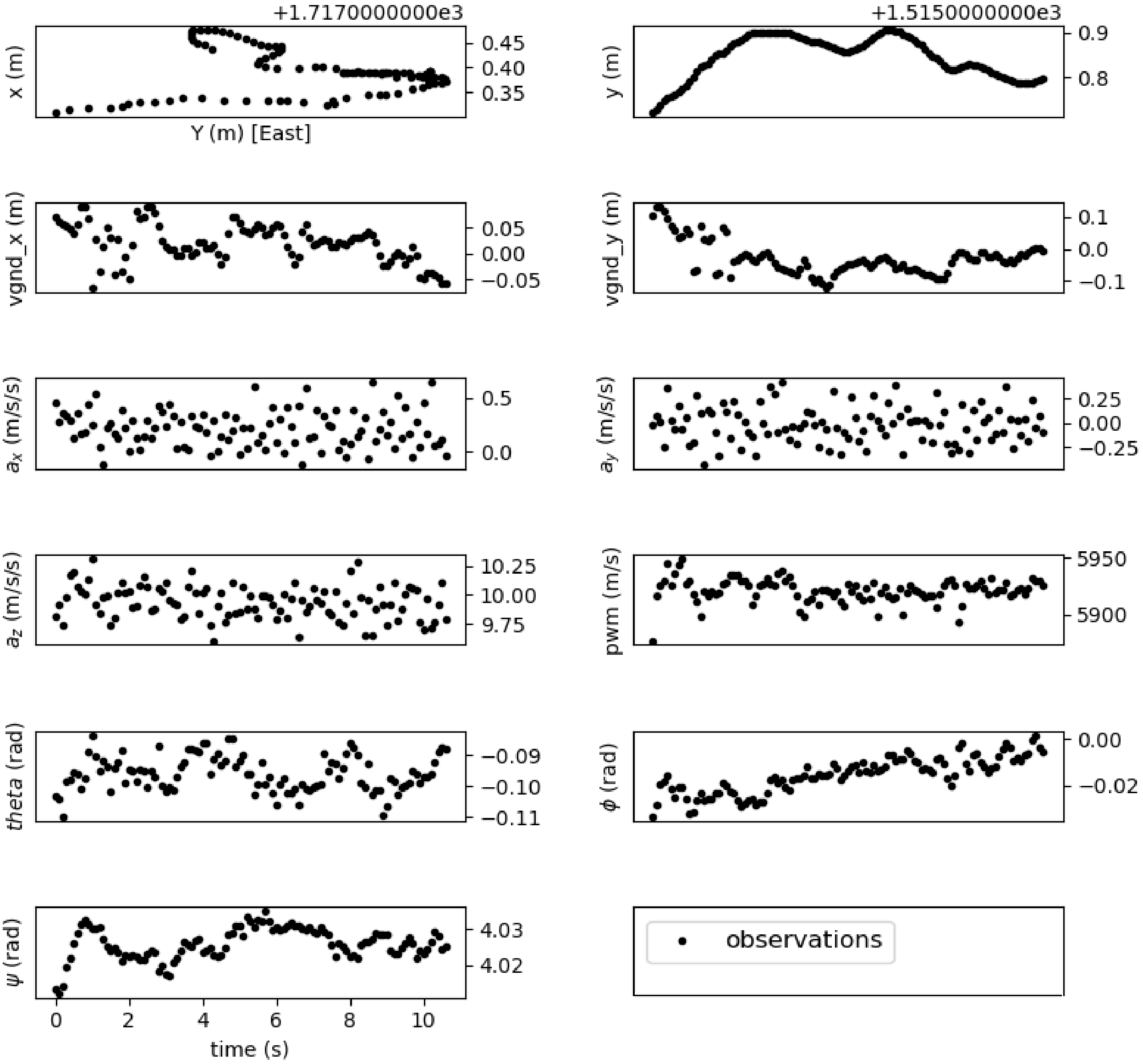


Fig. 24. Example of hover at 30m. Shows all the states observed during hover, before any processing.

Accordingly, Figure 25 shows all of the states estimated after a run.

Recommendations to METSIS

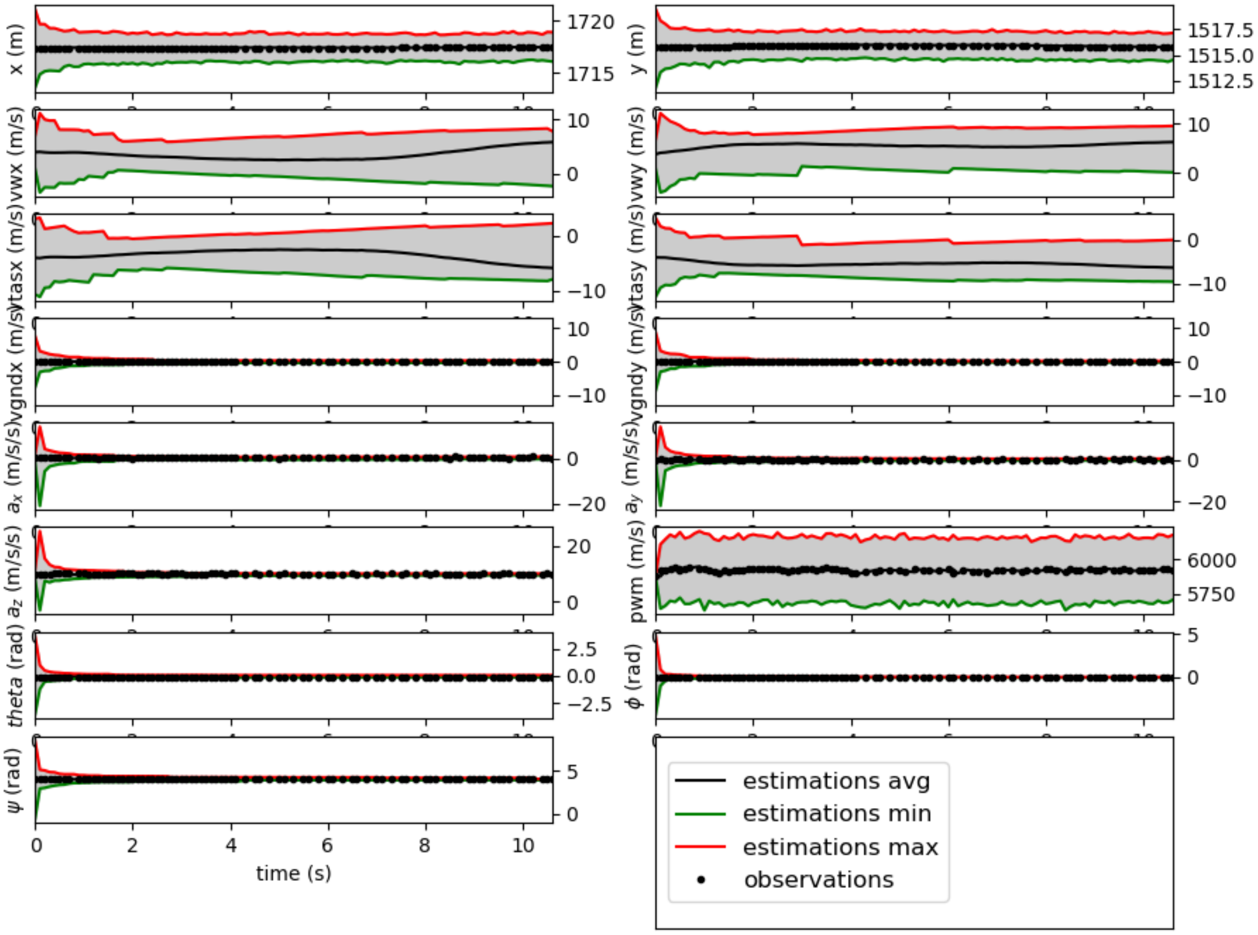


Fig. 25. Example of hover at 30m. Shows the particle with the average value (black), the particle with the highest outcome (red), and the particle with the lowest possible outcome (green) for each state. The states do not converge fully, but that is not expected considering the wind is varying slightly constantly.

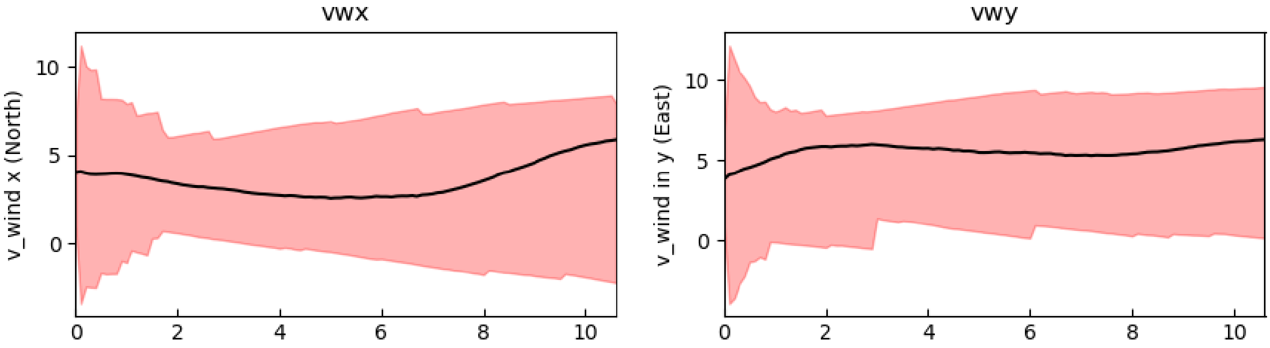


Fig. 26. Example of hover at 30m. A closer look at the estimated wind magnitude in the x- and y directions shows that the estimates for wind do not fully converge, and that wind in x- the direction tends to converge less with the current setting.

Method	RMSE	Advantage	Disadvantage	Suitability to METSIS
Sensor (Anemometer)	0.3-3.0	Straight forward	Added weight, cost, works only in hover	-
Drifting	1.82 m/s, 36°	Excellent dir. estimation	Slow convergence, impractical near obstacles slow adaptivity	-
[redacted] (Quadratic, Linear)	0.3-3.7 m/s, 5-36°	Works in cruise, Easy to implement	Requires different C_D in hover and cruise	++
Particle Filtering	0.3-3.7 m/s, 5-36°	Self estimates C_D for different flight conditions	Requires good initial guesses, inattentive to gusts, slower	+

TABLE XI

Overview of explored methods, final recommendations towards METSIS project. *data not generated, from literature [7], [8], [9], [10]

Other scripts used

A set of utility scripts have been written, which are not integrated with the main methodology's script but help in parameter estimation for the performance models. In the following list, the links redirect the reader to the GitHub repositories.

- Indoor Speed Estimation: https://github.com/erikbaas/speed_trap
  This script is used to estimate speed indoors.
- Frontal Area determination https://github.com/erikbaas/frontal_area_detector
  This script is used to determine the frontal area of the drone by automatically detecting the outline of the drone, using opencv.
- Wind field generator (Low Altitude Dryden gusts) https://github.com/erikbaas/windfield_generator
  This script is used as a gust simulator, which helps create realistic inputs for the Meteo Particle Model.
- Excel MOI Tool https://github.com/erikbaas/compute_moment_of_inertia
  During performance model tests, the moment of inertia had been computed by an approach similar to [34]. The equations used are found in the preliminary report. This excel sheet helps the operator use those equations to find the moment of inertia by assuming the quadcopter consists of a set of simplified shapes.
- Excel Drag coefficient https://github.com/erikbaas/waypoint_automator
  This script is used to compute the drag coefficient indoors.
- Thrust Curve Determination https://github.com/erikbaas/thrust_curve_from_pwm
  The thrust curves have been derived from csv handed data.
- Waypoint Automator https://github.com/erikbaas/waypoint_automator
  This script was used to speed up the waypoint creation. Since on testing day, the script was subject to the direction of the wind, it was useful to have 360 versions of flight lines ready (one for each degree of wind direction). This script asks for the change per file and then creates copies of waypoint files, alters them, and saves them.

Part II

Preliminary Report

II Table of Contents

III List of Figures and Tables

IV List of Symbols and Abbreviations

Abbreviations

ABL	- Atmospheric Boundary Layer
BPNN	- Back Propogation Neural Network
EKF	- Extended Kalman Filter
GPS	- Global Positioning System
IMU	- Inertial Measurement Unit
LESO	- Linear Extended State Observer
LIDAR	- Light Detection And Ranging
LR	- Learning Rate
LSTM	- Long- Short Term Memory
MAF	- Moving Average Filter
MEMS	- Micro Electro-Mechanical Systems (IMU)
METSIS	- METeo Sensors in the Sky
ML	- Machine Learning
MOI	- Moment of Inertia
MPM	- Meteo Particle Model
MUAS	- Multi-Rotor unmanned Aircraft System
MSc	- Master of Science
NLR	- Royal Netherlands Aerospace Centre
NN	- Neural Networks
PWM	- Pulse Width Modulation
RPM	- Rotations per Minute
RMSE	- Root Mean Square Error
SESAR	- Single European Sky ATM Research
SMART	- Specific, Measurable, Attainable, Relevant and Timely
TU	- University of Technology
UKF	- Unscented Kalman Filter
U-Space	- Unmanned Airspace
VLL	- Very Low Velocity Flight
V&V	- Verification & Validation
XGBOOST	- Extreme Gradient Boosting

Symbols

a, b	- Width, breadth, height, depending on axis [m]
B (subscript)	- Body frame
C_D, C_d	- Drag coefficient [N]
E (subscript)	- Earth frame
f	- Frequency [Hz]
f	- Frame [-]
FPS	- Frames per second [f/s]
g	- Gravitational constant, 9.81 m/s/s
M	- Mass [kg]
l	- Length [m]
P	- Permeability constant [-]
R	- Radius [m]
s	- Distance [m]
S, A	- Frontal surface area [m^2]
T	- Thrust [N]
$T.$	- Transpose of a matrix or vector
υ	- Speed (not to be confused with: tilt angle (ν)
ϕ	- Roll angle, around x-axis
θ	- Pitch angle, around y-axis
ψ	- Yaw angle, around z-axis
ν	- Tilt angle as measured from the euclidean inertial Zenith

V Executive Summary

This preliminary report is written in fulfilment of the MSc thesis as part of completion to the MSc Aerospace Engineering, track Control & Operations (Spec. Control & Simulation). It outlines the preliminary results, as well as the planning for the following months up to the green light.

The thesis is meant as a contribution to METeo Sensors In the Sky (METSIS), which is a proof of concept which tries to show the use and feasibility of having drones in network performing wind measurements. The research question is stated as follows.

"How can onboard states and intelligent filtering be used on the Foxtech Hover 1 to measure 2D wind speeds with an accuracy of 0.5 m/s and 5 deg?"

The research objective of this thesis is to find out whether wind speed and direction by means of the indirect method can provide the same or more accurate results as by the previously implemented direct method on the Foxtech Hover 1 quadcopter. The sub goals essentially boil down to making the methodology as accurate, scalable and cost effective as possible, bearing in mind the outcome of this thesis should become a standard for a multitude of drone types and sizes.

Three methods, namely one using sensors (direct method), using no dedicated wind sensors (indirect method) and a hybrid form are addressed, and from there it follows that the indirect method will be the focus of this thesis. By means of linear kinematic analysis, the drag term is found from which the true airspeed is estimated. This true airspeed is then deducted from the GPS speed vector in order to find an estimate of the wind speed and direction (this is called the triangle method). The method should reach an accuracy of 0.5 m/s and 5 deg, and thereby be an improvement to the current anemometers used, in terms of accuracy, cost and scalability.

In summary, nearly all papers with kinematic approaches agree to this method, however the way to model drag forces (and from there resolve to true airspeed) vary. While some papers consider drag force quadratic with speed (classical), others find that for lower speeds a linear relation with speed yields better results. For this thesis, currently the focus is on the classical method, as that has not been explored fully and might be an improvement for higher speeds of operation. Since all is done is post processing, in later stages the designed model could be extended or replaced.

One of the motivations for this (applied) research is the societal impact this could have on the many sectors, including healthcare, surveillance and also consumer deliveries. The outcome contributes to finding a standard for all drones in the future, such that wind can be measured at any time, on any drone, during any (routine) job performed within U-Space.

1
INTRODUCTION

The drone epidemic is inevitably evident. In Europe alone, the market associated with unmanned aircraft is expected to rise up to EUR 35B by 2035 [4]. The marketplace for drones is growing rapidly and is a promising solution for many problems in a wide variety of sectors ranging from off shore drone inspections to military to leisure purposes [5].

On the road to reaching the full potential of these markets there are obstacles ahead, too. For one, drones are lightweight and thereby affected by winds, which impacts battery performance and safety [6]. Therefore, it would be beneficial for a drone operator to have good knowledge of wind among common drone routes, such that the operator can plan its trajectory tactically.

For this exact reason, the METeo Sensors in the Sky (METSIS) project has been set up at NLR and TU Delft. The project aims to detect and offer a way to prevent the impact of wind for drones, providing a service to U-Space [4]. It works according to a three step principle [1]. First, the drones measure wind as accurately as possible during their routine jobs (which is the focus of this thesis). Second, the data gets sent to the Meteo Particle Model (MPM), where the wind fields over an urban- or industrial area gets computed and mapped, as introduced in [7]. Thirdly, this data gets sent to the operator, as such it can plan its trajectory accordingly, i.e. avoid excessive headwinds, thereby increase range and decrease safety hazards.

The network functionality of METSIS has been tested in a small swarm of 4 drones by [1]. The network functioned as desired, although the wind measurements on the drones do not yet meet the required accuracy recommended by the WMO [8]. For that reason, this thesis has been set up. It aims to explore and implement the best method to estimate wind fields accurately without dedicated sensors such as anemometers or pitot tubes.

Since many of discussed operations are expected to be performed with quadcopters [9], this thesis will restrict itself to quadcopters. More specifically, this thesis uses the Foxtech Hover 1 Quadcopter to demonstrate the wind estimating capabilities. The advantages of this drone are its high maximum speed of 20 m/s, and sampling rates which should be sufficient for our observation purposes. Also, the drone should provide more reliable data than aggressive and smaller drones, which usually have lower-cost sensors. The downsides to the use of this drone are that it does not measure wind estimates automatically, that it does not provide information on thrust or rpm, and that little to no parameters are listed online. Therefore, a testing day, including an indoor flight tests has been performed at Marknesse to get an estimate on these parameters. These results are listed in chapter 4.

This research will design according to three design principles, namely accuracy, cost-effectiveness and scalability. That is the reason that no wind tunnel tests have been performed, as the METSIS is geared towards finding scalable methods of parameter estimation, which operators or manufacturers should be able to do without expensive wind tunnel tests. The accuracy does not need to be perfect but it does need to be accurate enough to outweigh the benefits of placing heavy and bulky sensors, such as anemometers.

The remainder of this report sets out as follows. First, in chapter 2, the research question and research objectives of the thesis are discussed, and broken down into tangible and workable problems. Second, in chapter 3 an extensive literature study has been performed, where the direct, hybrid and indirect method are discussed in depth. In chapter 4, all preliminary results are presented, ranging from the approach of state estimation to a detailed description of the parameter estimation. In chapter 5, as a follow up to this preliminary report there is explained what flight test could be performed in early November to validate the found algorithm. In chapter 6, a project management perspective is taken on the project, including work packages and tools to keep track of the performed work. Code and larger charts are placed- or referenced to in the appendices.

2 RESEARCH METHODOLOGIES

This chapter consists of discussion on two main parts: the research question and the research objective, both split up into sub-questions and sub-objectives for more tangible and focused work approach.

2.1. RESEARCH QUESTION(S)

For this research, the main research question is formulated in a is stated as follows.

"How can onboard states and intelligent filtering on the Foxtech Hover 1 be used to measure 2D wind speeds with an accuracy of 0.5 m/s and 5 deg?"

To ensure this question is of significant contribution to the development of science, and of impact to the METSIS project, this question is written in a SMART manner. It is specific in the sense that the required accuracy is clearly stated. It is Measurable, as in early November a flight test can be performed to validate the found results. Earlier papers tackle this issue in different manners and attain accuracy's within this range, so it is considered Attainable too. It is Relevant in in that it would solve a puzzle piece in the METSIS project and on bigger scale be of value to U-space. Lastly, it is Timely as the scope of this thesis is clearly defined.

To give holistic answer to the above question, sub-questions have been set up.

Sub-questions

1. Can we prove that the indirect method is most feasible as opposed to the direct- and hybrid method?
2. Will linear kinematics provide us with wind estimates within this required accuracy? If not, will the inclusion of rotational dynamics do that?
3. Can we implement a filter (particle, kalman etc.) to reduce the error in wind estimates?
4. How can we ensure that the proposed solution is scalable to a multitude of drones?
5. How can we ensure that the proposed solution is cost-effective?
6. How can we ensure that the proposed solution is verified and validated properly?
7. What are the limitations to the selected approach? How well does the algorithm work in hovering, static and dynamic situations (e.g. accelerating and decelerating)? How do we test this successfully?
8. What alternative methods are there and are they worth exploring? E.g. can we use a ML algorithm to improve accuracy measurements in hovering conditions?

2.2. RESEARCH OBJECTIVE

The sub-questions in section 2.1 provide guidance in conducting this experiments in this thesis, but to answer these questions, they are divided up into research objectives in this section.

Goals

The objectives more broadly state what the expected outcomes of this thesis are. The main research objective of this thesis is:

> "To achieve an accuracy of 0.5 m/s and 5 deg by implementing linear kinematic analysis for the Foxtech Hover 1".

Where linear refers to the motion of the drone, as opposed to linear vs. non-linear equations.

Sub-goals

To connect the research objective to the literature review and experiment design, the research objective can be divided up into sub-goals which will form the basis of the tasks outlined in the planning in Appendix A. These are explained and summarised in this subsection.

State estimation and filtering

The first set of sub-goals essentially find out whether the indirect method can be implemented on the Foxtech Hover one drone. In practical terms, this means implementing the kinematic equations required and with that find out if the desired accuracy can be attained. If not, the plan is to implement a filter investigate whether augmenting the data filter, i.e. adding sensor information to this algorithm can make the results more accurate.

1. Find out what information on board is necessary to find wind estimates within the required range within the chosen approach
2. Find out up to what extent the kinematic approach get us above the requirements?
3. Find out if the inclusion of rotational dynamics improves the results
4. Find out if a particle filter can be used to improve the results and eliminate noise and bias?
5. Find out whether a low-cost sensor is viable and would improve the results of the implemented algorithm (i.e. explore the feasibility of the Hybrid method).

Parameter estimation

The third sub-goal is to find out what parameters and models need to be estimated, and ensure accessibility to any drone operator.

1. Estimate drag coefficient by flying indoor lines, without the availability of GPS
2. Estimate the frontal area by taking photos with a reference object
3. Estimate the thrust continuously, as a function to either PWM or RPM
4. Estimate the moments of inertia Ixx, Iyy, Izz and Ixy, in case we include rotational dynamics.
5. Ensure that the chosen parameter estimation method is scalable enough for a variety of operators and drone types.

Verification and Validation

One of the sub-goals is to find out how to save valuable testing time An example of such is setting up wind field simulations to mimick realistic weather scenarios. The simulation tool which will be set up, containing a constant wind and a gust component, will be a quantifiable outcome of this sub-goal.

1. Recreate realistic wind in x, y and z direction by creating a 3D wind tunnel simulation. This can be used as input to generate wind for the MPM, but also to make a simulation of the drone's dynamics and expose it to these fields.

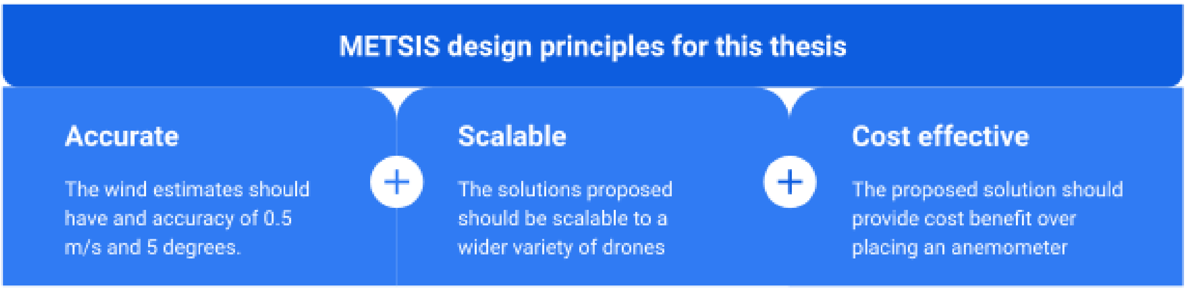
2. Verify that the axis system conventions and rotational matrix are implemented properly by connecting the drone without battery and conducting a calibration flight.
3. Implement the wind estimates in real-time rather than post processing. This can be done by either down streaming the data at a sufficient rate and analysing “packages” of states data using an (existing) publish and subscribe method, or by implementing a filter which is able to run in real time, as filters assume all historical data is stored in the previous (n-1) and current (n) step.
4. Use the flight experiment to validate whether the algorithm estimates the wind magnitude properly.
5. Use the flight experiment to validate whether the algorithm estimates the wind direction properly.

2.3. DESIGN PRINCIPLES

In summary, when contemplating solutions to the problems at hand, during all steps in this research there should be considered whether they are in line with the three principles displayed in Figure 2.1.

Design principles for this thesis

1. Accurate
2. Cost effective
3. Scalable and accessible



METSIS design principles for this thesis

Accurate

The wind estimates should have and accuracy of 0.5 m/s and 5 degrees.

+

Scalable

The solutions proposed should be scalable to a wider variety of drones

+

Cost effective

The proposed solution should provide cost benefit over placing an anemometer

Figure 2.1: Schematic displaying the three design principles which serve as technical requirements throughout the design of the proposed solution.

As an example, the use of wind tunnels will provide accurate estimates, but is not cost effective nor accessible to a wide variety of operators, making it score low on cost and scalability. These three pillars will often be referred to throughout the remainder of this report.

3
Literature Review

This chapter provides an overview which covers the relevant parts of the research field of this thesis. Starting top level (what is U-Space and how can it benefit from METSIS?) and working a way down (what kind of filtering techniques are commonly used?), this chapter will provide the reader with a comprehensive view on what has been established both historically, what recent developments are, and what research gaps can be filled.

3.1. Wind as a challenge for drones

As hinted at in the introduction, The drone marketplace is growing rapidly and is a promising solution for many problems in a variety of sectors. One of those is last mile package delivery, where it is estimated that up to 95% cost can be saved when switching from conventional trucks to quadcopters [10]. Another fields of application of interest is scientific research and meteorology, where many sources comment on the potential to contribute to valuable data in the ABL [11].

Table 9 from Pérez Castán *et al.* [4] acknowledges wind as a potential hazard and considers strong winds a "gap in the U-Space services". Palazzetti [12] experimentally shows the power consumption can more than triple when facing headwind of 15 m/s, both with and without payload. Abeywickrama *et al.* [13] shows that even for a wind speed of 1 m/s flying 15 km/h head or tailwind can make a noticeable difference in required power (W). In turn, this means reduced battery and thus affects the time of flight. This calls for a method to avoid headwinds at all cost.

3.2. METSIS project

Briefly summarised, the METSIS project aims to help drone operators and autonomous drones to plan their trajectories such that they are least exposed to wind. This essentially happens in a three-step plan, which is displayed in figure Figure 3.1.

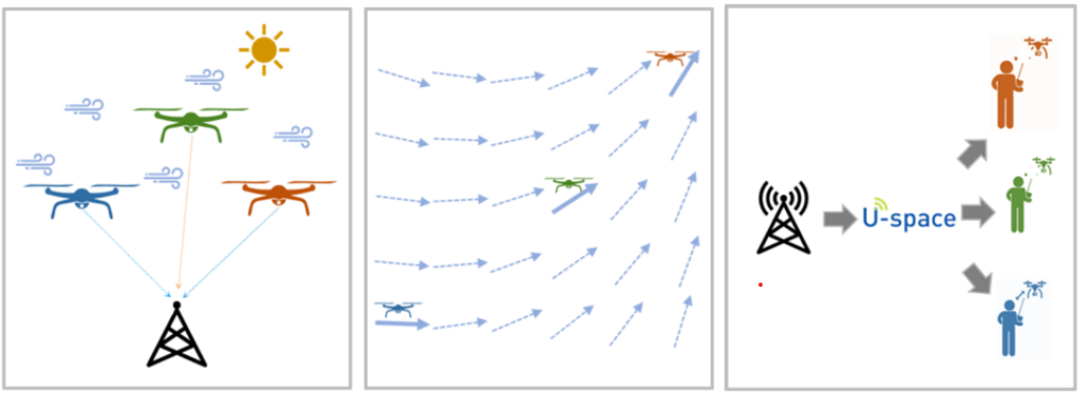


Figure 3.1: METSIS working principle from [1], where drones measure instantaneous wind (left) to send to a centralised Meteo Particle Model (mid), which can then be broadcast to operators (right).

This service will be a U2 or U3 implementation into U-SPACE, and is of best use to drones operating in high density areas, as they can benefit from more frequent wind updates and thereby more accurate interpolation. This in turn results in better trajectory planning. Example applications would include packages deliveries within cities, or frequent inspections.

In terms of altitudes of operation, [14] displays interest in wind mapping in all of the Atmospheric Boundary Layer (up to roughly 1 km), due to meteorological purposes. Since most drones in future day-to-day operations are expected to operate at very low level (VLL) [15], one can restrict the wind mapping to a ceiling height of roughly 120-150m [15]), [9], [4]. Also, to be of use one does not have to measure at different latitudes. Instead, this can be solved in MPM, e.g. by introducing Atmospheric stability α is used to extrapolate between heights. 1/7 is normal for neutral atmospheric stability (similar to [16].)

[15] discusses the possibility of dynamic creation of geofences. Extending on this application, advisory (i.e. cautionary) zones could be created to protect drones from headwinds, such that drone operators need less effort interpreting wind data. No Fly zones could even be implemented to keep create traffic lanes in which there is always a tailwind, as such to increase efficiency of traffic whilst minimising chance of collision. METSIS could hereby help the ATM in the decision making process when creating, updating and deleting these Geo Fences. Dynamic geofencing is however a U3 feature and thus not top priority.

For more information on U-Space and the data flow going into the Meteo Particle Model, consider [1], [17].

3.3. RESEARCH GAP

This section addresses the research gap. Concisely summarised that boils down to the following list.

- There currently is no research paper which addresses the issue of wind measurements in the context of a project similar to that of METSIS. This is however hinted at in future reference work, by e.g. Cano [18] and Marino *et al.* [19].
- Very few of the methods in literature are described as such that the methodology is universal to a variety of drones, but rather applicable to the specific type of drone. In many existing papers, a lot of parameters are expected to be known, or found using expensive equipment. For METSIS research, only frameworks and methodologies which extend to all drones and be available to all operators world wide are presented. This is bearing in mind the expectation that a whole variety of drones will be allowed to enter the airspace, rather than one type.
- Current options for checking wind include websites such as:
  https://www.uavforecast.com/ and
  https://www.windfinder.com,
  although these only go up to a resolution of roughly 5-3 km, using just fixed wind measuring poles. This thesis and METSIS in general should provide a more detailed (i.e. hyperlocal) wind map of an area.
- All code presented in this thesis will be entirely open source and written in Python (as opposed to commercial software presented in papers, such as MATLAB, AUSPEX, CIFER, etc.).
- This thesis will try to implement a particle filter to improve wind estimation of a single drone, which to the best of the author's knowledge has not been implemented before in this exact context.

3.4. DIRECT, HYBRID AND INDIRECT METHOD

This section classifies the wind estimation methods as direct, indirect and hybrid and explores which one is most suited for the METSIS project.

Note that during the exploration of methods in this chapter, the implementation of choice is not commented on, as that is presented in chapter 4.

DIRECT MEASURING METHOD

In literature, the direct method refers to methods where there is a sensor used directly to measure the wind.

In terms of sensor selection, Thielicke *et al.* [20] notes that K-vane sensors are outdated, that hot wire anemometers are very accurate but tend to be fragile and expensive. The paper explores the use of miniature anemometers but attains low accuracy using these as opposed to normal sized anemometers. Lastly, anemometers like the one used for METSIS up to now are found heavy and large.

As another alternative to classic anemometers, Bruschi *et al.* [21] explores the use of solid state anemometers (MEMS), but concludes that the influence of rotors towards the speeds measurements is too significant to be of use.

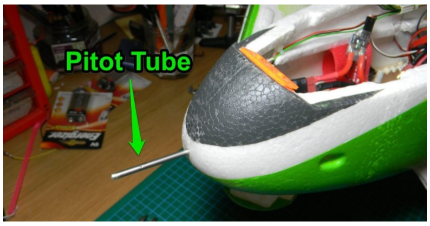


Figure 3.2: Fixed wing aircraft have the advantage of placing pitot tubes in free-stream velocity air.

Sensor placement

In addition to sensor selection, [20] provided an in depth study on where to optimally place them, and suggests that 2.5 times rotor diameter should be kept as a minimum distance from the rotors towards the wind sensors. The results from [1] however suggest that even this is not sufficient to reach the desired accuracy as stated in section 2.2, even when bearing in mind placement tips from the specs sheet of the Trisonica Anemometer [22].

Also, according to [16], it is recommended to keep the drone under 10 degrees pitch and roll, to avoid fast roll and pitch rates, and to lower speeds to improve wind measurement accuracy (< 2 m/s).

Considering the problem of sensor measurements while the drone is in rotated state, both [20] and [23] suggest placing the sensors (miniature sensors and LIDAR sensors, respectively) on a gimbal. For METSIS, this is not in line with the third cost and scalability aspects as in section 2.3, however, and thus these are discarded.

Fixed wing vs quadcopters

Note that fixed wing aircraft have a strong advantage in having a constant inflow of wind at a reasonably ranged and anticipated angle of attack, making it suitable for sensors such as pitot tubes [24].

Moreover, the sensors can be placed in tactical positions where the propellers have little influence on the airflow towards the pitot tube [11]. For that reason, as implemented in Kroonenburg et al, delivers what is considered state of the art wind estimation on UAVs [25].

For quadcopters, however, pitot tubes are generally considered impractical, as quadcopters operate at low speeds (< 12 m/s) [11] [26], and are subject to inconsistent flight directions [25]. Alternatively, hardware such as gimbals would be needed to compensate the drone's inclination angle and turn the pitot tube nearly in flight direction [25].

This thesis will from here on restrict research to implementing wind measurements on rotor-crafts and more specifically quadcopters, as METSIS assumes these to be the most suitable candidates for typical applications in U-Space.

Figure 3.3: Image of the Foxtech Hover 1, used for testing days

An overview of all considered sensors is found in Table 7.1 in Appendix A.

HYBRID MEASURING METHOD

In [27] describes what will be refered to as the Hybrid method, where the sensor information is fused with the state estimates. This would however not increase our cost effectiveness and scalability, thus this option is considered a last resort in case the indirect method under-performs.

An alternative in terms of sensors is using Pitot tubes, like [24] have done. This provides some additional complexities to this application, however, such as the earlier mentioned reduced accuracy of using Pitot tubes when flying at lower speeds. Also, the Pitot tube is only accurate up to a small range of angle of attack, after which the wind flow gets distorted, rendering the sensors useless.

Driessen *et al.* [28] use an Extended Kalman Filter to include optical flow as well as sonar data, in addition to the onboard sensors (making this approach Hybrid), in estimating velocity, position and attitude. The adding of these sensors would increase the complexity of the system.

INDIRECT MEASURING METHOD

The indirect methods refer to those that do not use a dedicated wind sensor, but rather induct them solely from states of the vehicle. Donnell [11] claims the indirect method (without sensors) is better than direct method (with sensors), although it lacks the ability to detect sudden changes in wind direction.

Other literature seems to make the same distinction. Where some papers focus on gust estimation

[29], others focus more on steady state wind and even model the derivative of wind speed to be zero. An example is [30], where $\dot{v}_w = 0$, which by following the triangle principle means $\dot{v}_{tas} = \dot{v}_{gps}$, which can help in simplifying state estimation at the cost of slower convergence time, as found in [30].

To guide the reader through the remainder of this section, an overview of commonly used techniques is shown in Figure 3.4.

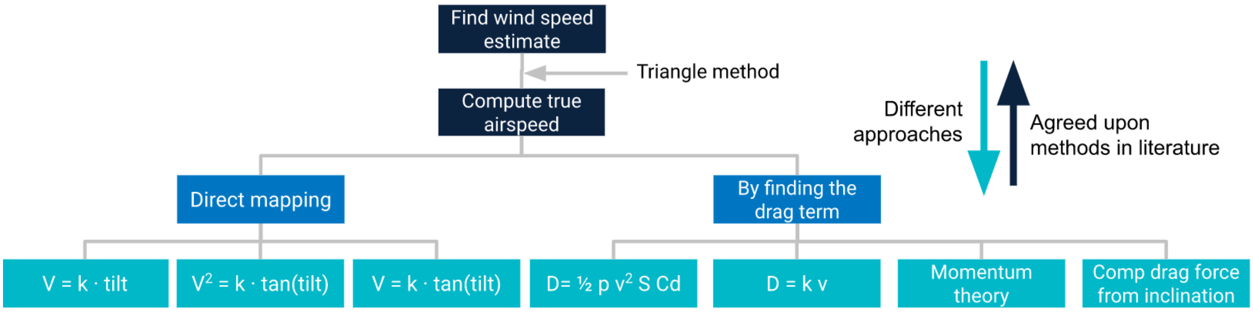


Figure 3.4: Subclassification of current agreed upon (top) and varying (down) indirect methods in literature, regarding finding (initial) estimates of wind speed

Following a top-down approach, these strategies will be illustrated in the remainder of this section.

Triangle Method

A major trend in literature is within the indirect method is the triangle method, where the ground speed vector is deducted from the true airspeed vector. This way, what is left is the wind speed vector which caused this discrepancy in the first place.

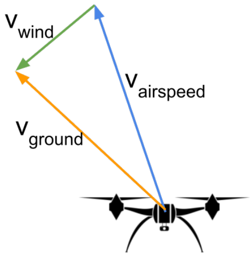


Figure 3.5: Triangle method at the heart of indirect methods

Generally literature agrees upon this triangle method, although there exists a variety of ways to find estimates, of which a large split is in direct mappings against kinematic resolving.

Direct mappings

Interestingly, Neumann and Bartholmai [25] (who are considered pioneers in wind estimation on board of drones), use the classical formula approach to find an estimate for C_d, but from there uses an experimental mapping in a Gottingen-type windtunnel to find a mapping between tilt and the final estimate of v_{tas}. Other papers which use this direct method are [31], [19] and [11]. Since most methods rely on extensive windtunnel tests, this will not be the method of preference.

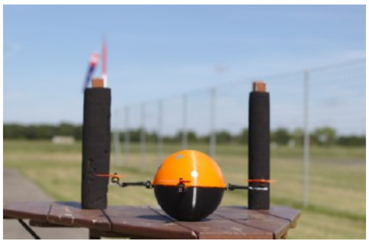


Figure 3.6: Drone with custom spherical body, to simplify drag force estimation

Interestingly, Hattenberger *et al.* [30] uses the tilt method but attempts to make the computations slightly easier by imposing restrictions on the drone. Namely, this drone is roughly symmetrical (see Figure 3.6), making it that the drag force is very similar under different tilt angles. Although interesting for meteorology, for this thesis, however, the drone type for experiment day is set, and it would be out of scope to make changes to the mechanical layout of the drone in use. Also, it would go against the principle of scalability to "demand" future drone operators to fly with a spherical body, since this thesis' solution should be applicable to a variety of drones, working on a variety of applications. Lastly, although the drag force is more similar in multiple directions, it is expected *higher* overall as it introduces additional mass as well as frontal area, making the drone inefficient for every-day use.

Resolving the Drag term

Wang et al. finds the drag force (or in fact, "drag acceleration") to be linearly dependent on speed, as first presented by Luukkonen [32]. This relation is shown in Equation 3.1

$$D = m \cdot a_{drag} = kv \tag{3.1}$$

This approach is then followed by e.g. Hattenberger *et al.* [30], and earlier by Sikkel *et al.* [33], who explain that this relationship is due to the rotor drag, which prevents the drone from "accelerating indefinitely".

Sikkel finds this constant k empirically, where Hattenberger further further disects this constant k into $k = mg \cdot c_\alpha$ where c_α is found by drawing a regression line in Equation 3.2

$$\tan \alpha = c_\alpha \cdot v \tag{3.2}$$

Tested for different speeds v in front of a WindShape speed generator.

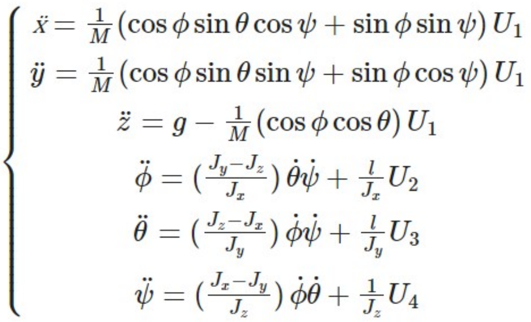
This assumption is only said to work for relatively low speeds [26], as that is where hover drag from momentum theory seems to dominate [34]. It has not been documented precisely, however, above what speeds the quadratic relationship with speed tends to take over.

Momentum Theory

An example of a paper using blade momentum theory is[35]. It follows a set of 10 equations, of which the most essential one is the resolving of the drag term:

$$D_{horiz,induced,B} = -TC_d(v_x + v_y) \tag{3.3}$$

In which T is thrust, C_d is drag coefficient and v is airspeed, all in the body frame.



$$\begin{cases} \ddot{x} = \frac{1}{M}\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right)U_1 \\ \ddot{y} = \frac{1}{M}\left(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi\right)U_1 \\ \ddot{z} = g - \frac{1}{M}\left(\cos\phi\cos\theta\right)U_1 \\ \ddot{\phi} = \left(\frac{J_y - J_z}{J_x}\right)\dot{\theta}\dot{\psi} + \frac{l}{J_x}U_2 \\ \ddot{\theta} = \left(\frac{J_z - J_x}{J_y}\right)\dot{\phi}\dot{\psi} + \frac{l}{J_y}U_3 \\ \ddot{\psi} = \left(\frac{J_x - J_y}{J_z}\right)\dot{\phi}\dot{\theta} + \frac{1}{J_z}U_4 \end{cases}$$

Figure 3.7: Commonly used equations of motion describing drone dynamics. Image from [2]

$$T = 8\rho A_{props} v_{vert,induced,z,B} U_{abs} \tag{3.4}$$

The parameters needed involve knowledge of power and rotational speeds, which can all be determined from bench tests. For this thesis, currently this system of equations remains unsolvable considering the Power to RPM mapping is not available from the testing day.

Note that all of these equations are performed in the body frame until the true airspeed is found, after which it is converted to inertial frame.

DIRECTION OF THE WIND

Can be modeled according to Equation 3.5 as suggested by Palomaki et al [31], which only works when yaw angle can be set to zero customly. If not the case for the Foxtech Hover 1, slightly more complex formulations such as in [36] or [37] can be considered.

$$\lambda = \arctan\left(\frac{-\sin\phi\cos\theta}{\cos\phi\sin\theta}\right) + 180\,\text{deg} \tag{3.5}$$

3.5. FILTERING METHODS

For a better look on noise filtering, different authors consider different filters, of which some are presented below.

Nearly all papers find resultant acceleration by reading out the imu's on board. Due to its inherently noisy output, most IMU and gyroscope data are led through a low-pass filter before being used. Also, in many of the discussed papers, the resulting wind estimates are subjected to a Moving Average Filter (MAF), with time values between 1 and 20 seconds.

KALMAN FILTER

Sylvie Schafer implements a linear Kalman filter (and adresses it well) in [29], but uses an existing MATLAB model to set up the model for the quadcopter. Since this thesis is expected to be modeled in python, this would require a set of equations of motion to serve as a model to compare to the observations. For one, Xiang *et al.* [38] and Luo *et al.* [26] show the use of linear kinematic equations, while [39] and [40] are examples of papers which include rotational dynamics as well. As an example of these commonly used set of equations of motion is displayed in Figure 3.7 from [2].

Baoubi [40] provides equations of motion similar to that of Figure 3.7 but relies heavily on RPM as well as on other undocumented parameters such as drag friction coefficients and solid friction constants, which are not to our disposal.

Hattenberger *et al.* [30] acknowledge the (changing) bias terms of the gyros and so they identify their covariance matrices, which are then used as starting point in the Kalman filter implementation.
Ko *et al.* [41] implements an extended kalman filter, as well as an unscented kalman filter, but uses detailed thrust parameters and angular rates which are not available.

Luo *et al.* [26] uses a linear extended state observer (LESO) to estimate the drag-term acceleration, which essentially adds a observation-correction term to the equations in state space form, and tries to minimise that, making its working similar to the Kalman filter. For this thesis, the author will focus on implementing a similar type of observer, called the Particle filter, which will be the focus of attention starting in September. This optimal filter is applicable to non linear systems and should not only estimate the wind speeds and direction, but also quantify the uncertainties in the measurements [42].

POST FILTERING IN METSIS

After the wind has been measured and processed on board of the drone, its data will be sent to the Meteo Particle Model (MPM). In here, adjustments can still be made to improve the accuracy of the final wind map, such as interpolation, rejection criteria, and another filter which contains a model for how wind evolves throughout the city (rather than for true airspeed per drone, as hinted at in the remainder of this thesis), and uses that to improve its measurements.

3.6. PARAMETER ESTIMATION METHODS

Methods vary from boldly assuming values up to measuring them in controlled environments. This section gives an overview of each the methods found in literature, and addresses in why and how these parameters are used.

DETERMINING MOMENT OF INERTIA

This thesis will try to limit itself to linear dynamics, considering [39] does not show significant improvement (table 9) when comparing the in- and exclusion of rotational dynamics. In order to best prepare for the possible inclusion of rotational dynamics, as presented in Figure 3.7 for example, it is of interest to determine moment of inertia. Two methods are addressed briefly.

$$I = \frac{mgD^2T^2}{16\pi^2 h} \tag{3.6}$$

Teimourian and Firouzbakht [43] and Koken [3] do this by suspending a bi-filar pendulum and measuring the time of oscillation, and then filling in Equation 3.6.

Mendes *et al.* [44] does this by modeling the parts as boxes, cylinders and slender rods, and also takes the opportunity to compare the two method described above. The paper states an accuracy within 1% for the geometric method making this an appropriate option for the Foxtech Hover 1, of which the results are displayed in chapter 4.

DETERMINING DRAG COEFFICIENT

In determining drag coefficient, there are basically two streams in literature, similar to that in Figure 3.4.

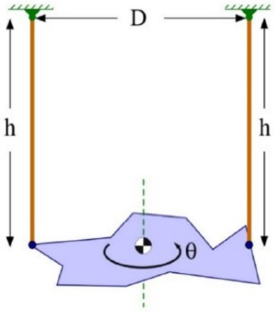


Figure 3.8: Pendulum set up describing parameters in Figure 3.8. Image from [3]

- **Drag coefficient** C_d, as defined in Fundamentals of Aerodynamics by Anderson [45]
- **Drag coefficient k**, loosely defined constant, which takes into account other variables as well.

Which are explained more in detail below.

Drag coefficient C_d
Starting from the "classical" drag equation,

$$D = \frac{1}{2}\rho v^2 S C_d \tag{3.7}$$

Rewriting this for C_d becomes:

$$C_d = \frac{2D}{\rho v^2 S} = \frac{2T}{\rho v^2 S} \tag{3.8}$$

An example of a paper which uses this classical approach is Felismina *et al.* [46] and Neumann et al, [25]. In the above case, there is assumed T = D, meaning the drone is in perfect cruise condition. As the drone operator has not been able to achieve perfect cruise conditions within the limited indoor space (flight path of roughly 15 m), the resultant acceleration term should be included, and so the equation accounted for acceleration becomes:

$$C_d = \frac{2(T - ma_{res})}{\rho v^2 S(v)} \tag{3.9}$$

Where T is T in the moving direction, which is in this case the inertial x direction. Similarly, a is in the moving direction (x) and v as well.

Drag coefficient k instead of C_d
This is the group which loosely defines an empirical constant as the drag coefficient, thereby lumping surface area and dynamic pressure into one constant.

Within this group, there is a split in papers which consider C_d as quadratic with speed as in Equation 3.10, and linear as in Equation 3.11, with slight variations.

$$D = kv^2 \tag{3.10}$$

$$D = kv \tag{3.11}$$

From these equations, experiments are performed, and then k is found via a regression. Interestingly, what Simma *et al.* [35] does, as well as Sikkel *et al.* [33], is tweak the drag coefficient *after* a number of test flights, and seeing for which value the estimates correspond best with the reference values for wind of those flights.

DETERMINING THRUST

A few options are found throughout literature. A first approach is to assumed the drone is steady and does not make any vertical motion (i.e. drag in z is zero), such that gravity cancels out the thrust exactly. Neumann and Bartholmai [25] uses this approach and states $T(= D) = mg\tan v$, meaning drag (and thrust) on the horizontal plane is based solely on the mass of the drone and the tilt angle v.

[38] uses a similar approach but extends this method to a varying not assume constant altitude. Instead, the paper allows for acceleration in z byy incoorporating IMU data, as in Equation 3.12.

$$T = \frac{(a_z + g)m}{\cos\theta\cos\phi} \tag{3.12}$$

Sikkel *et al.* [33] uses a similar approach, where the IMUs output in the z direction is measured in the body frame (and multiplying this by the drones mass), arguing $T = m \cdot a_{z,B}$.

In [26], the author uses momentum theory and the conservation of energy to find a relation between thrust and rotational speeds of the rotors, as in Equation 3.13.

$$T = k\sum_{n=1}^{4} RPM_n^2 \tag{3.13}$$

Where k is the thrust coefficient which should be determined experimentally in a force bench, or assumed. Since there is no knowledge of RPM data (it would require an external sensor, which means added complexity), this makes this method impractical. The authors do propose a method in which the current voltage is compared to the baseline voltage by means of $U^2/U^2_{baseline}$ (i.e. the voltage required for hovering), and that provides an estimate of the acceleration due to thrust without knowledge of the thrust coefficient, so this can in fact be considered in the future.

DETERMINING FRONTAL AREA

A building block of performing the kinematic approach is to know the frontal area at all times, as it affects the drag force. Different approaches are taken in literature.

[26] makes one photo and from there assumes a constant frontal area. [18] instead makes photos at three different angles, and from there creates an interpolation function.

Marino *et al.* [19] assumed the drone as if a flat disk and computes the surface of this flat disk using a width · height formula, taking into account its tilt into the moving direction.

Intelligibly, Hattenberger *et al.* [30] use a special drone, designed in a spherical case, such that the frontal area (and, as the authors argue, thereby the drag force) does not change too much.

In literature ([47], [18]) the propeller area (A_{prop}) is included too, and so this is included as well in the computations for the Foxtech Hover 1. There is also note of a semi-permeability constant P as in fact the area of the spinning disks is not a solid, however the paper gives no sense of what this constant is or what its range should be. Thus, for this thesis this is assumed constant as 1.

Note that for the sources in this section which deviate from the classical C_d determination as in Equation 3.8, these do not need knowledge of the frontal area anymore as this term is "bundled" into the constant k.

3.7. ALTERNATIVE METHODS

Some of the methods which do not fit the above sections, whether for state or parameter estimation, are presented here.

OPTICAL FLOW

At one point there is considered the use of optical flow. Optical flow is, however, mainly of use indoors, when GPS fails to act as a source of position or stabilisation [48]. In this thesis, however, METSIS focuses on outside operations and will thus assume always to have GPS signals in reach. Also, the drone used for testing in this thesis is off the shelve and stable, such that no further improvement of accuracy in pitch or roll is necessary, thus using optical flow as an addition source of pitch and roll is superfluous. In effect, this would only demand for added complexity, as an external system would need be integrated into the data current flow.

Ho *et al.* [49] uses a good camera and flash lights to determine position and velocity, but only estimates vertical velocity, which is not the intent of this research.

MACHINE LEARNING

In terms of wind now-casting, Phan [50] uses XGBoost and thereby outperform BPNN and LR in successfully forecasting wind speed in 5 minute time sequences. Looking more at quadcopters wind estimation, Allison *et al.* [6], trained a Long Short-Term Memory (LSTM) neural nets (NN), such that roll and pitch angles can be inputted and expected wind speeds can be outputted. Herewith the authors claims to outperform the Triangle method. The paper uses a Dryden gust model as well as Large Eddy Simulations to realistically model outside wind fields. The authors claim that the model generalises well with this generated data, although all tests are performed in simulation, thus further testing is necessary which might not fit into the time scope of this thesis.

Applying machine learning as a main approach knows some challenges in terms of data. Whereas stationary towers are active 24/7 (and thus have a lot training- and validation data), this project so far only has flight data from two consecutive days of very similar wind conditions (< 5m/s). The limited training data introduces risk of overfitting to the type of drone as well as the type of weather conditions, which is not in line with the design principles of section 2.3. Therefore, the more deterministic models are preferred as these can be scaled to a wider variety of drones.

3.8. OVERVIEW OF PAPERS IN A TABLE

Note that the papers (with indirect methods) which are referenced to the most are summarised and displayed in Table 7. This can be found in Appendix A.

4 PRELIMINARY RESULTS

This chapter will present the results attained thus far for the MSc thesis. Note that up to now, work focused especially on parameter estimation, as in section 4.2. For more details on the state estimation, a plan is layed out in section 4.1.

Considering many of the exploration is done in chapter 3, this chapter will be written more concisely. Where possible, the author tries to showcase the results first, and give reasoning after. Where in-depth argumentation seems required, there will be referred back to the appropriate section in chapter 3.

4.1. KINEMATIC ANALYSIS

Out of all the methods considered in chapter 3, this section briefly addresses the most suitable one. After consideration, the kinematic force-balance approach is the most straight forward one to reach the desired accuracy. The main principle is shown in Figure 4.1. This section will dive in to depth on the details to demonstrate how this method should give us the desired wind estimation.

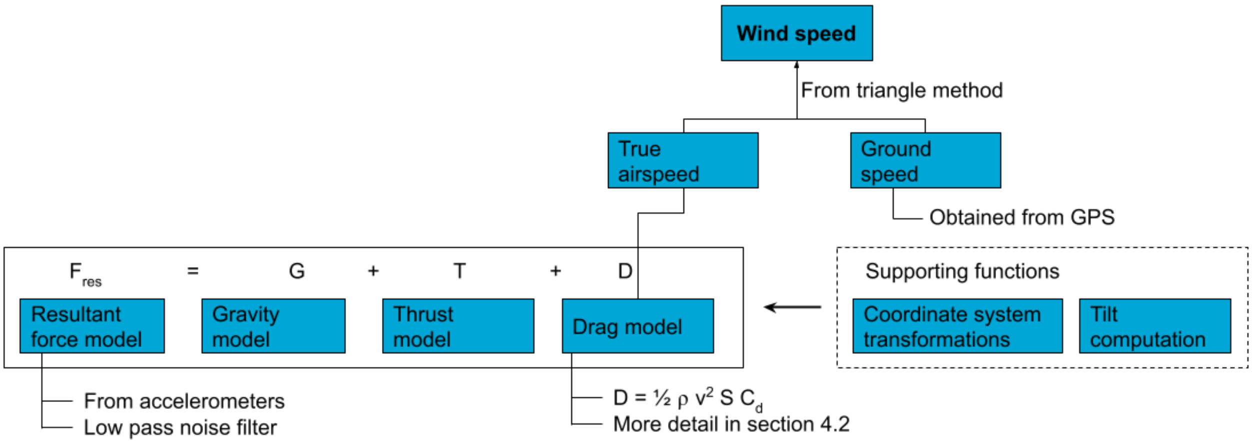


Figure 4.1: Schematic overview of how the selected kinematic approach on finding wind speed

COORDINATE SYSTEMS AND CONVERSION

Essentially this thesis is concerned with the body frame, and the earth frame. This is displayed in Figure 4.2.

The Rotation Matrix used to convert from body to inertial frame is as in Equation 4.14. By transposing the matrix ($R_I^B = R_B^{I,T.}$), one can convert from earth to body frame if required.

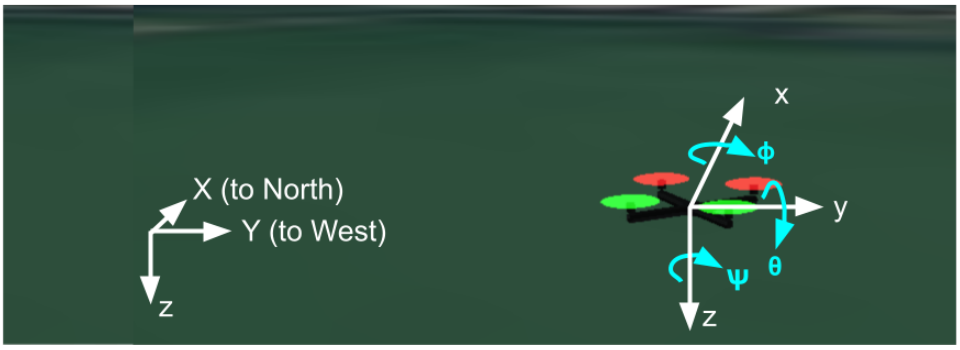


Figure 4.2: Schematic showing the body frame and the earth frame used in this thesis.

$$R_B^I(\theta,\phi,\psi) = \begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\theta\sin\phi - \cos\phi\sin\psi & \cos\psi\sin\theta\cos\phi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\theta\sin\phi\sin\psi & -\sin\phi\cos\psi + \sin\theta\cos\phi\sin\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{4.1}$$

This matrix is validated by inputting mock-up vectors for thrust and seeing what the expected results should be, and comparing that to the output of the matrix. This way, problems in the model with sign conventions and mix-ups in formulas are detected and resolved.

Also note in this report, rather than θ, ϕ, ψ, often v will be referred to as a combination of the three, defined as the tilt angle as in Equation 4.2.

$$v = \arccos \frac{\vec{u}_{xy} \cdot (\vec{e}_\phi \times \vec{u}_\theta)}{|\vec{u}_{xy}| \cdot |\vec{e}_\phi \times \vec{e}_\theta|} \tag{4.2}$$

In which:

$$\vec{e_\phi} = \begin{bmatrix} 0 \\ \cos\phi \\ \sin\phi \end{bmatrix} \tag{4.3}$$

$$\vec{e_\theta} = \begin{bmatrix} \cos\theta \\ 0 \\ -\sin\theta \end{bmatrix} \tag{4.4}$$

$$\vec{u}_{xy} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.5}$$

WORKING PRINCIPLE

Remember that subscript B and E refer to Body and Earth, respectively. Starting from a summation of forces in the earth frame:

$$F_E = G_E + T_E + D_E \tag{4.6}$$

Rewriting some of the variables to body frame is more practical, considering measurements onboard the drone are performed in this frame. Therefore:

$$R \cdot F_B = R \cdot G_B + R \cdot T_B + D_E \tag{4.7}$$

Where D_E is kept in the Earth frame. From here,

$$F_B = m \cdot a_B = m \cdot [a_x, a_y, a_z]_B^T. \tag{4.8}$$

Where a_B is read out from the onboard IMUs. As described in section 4.2, the IMUs acceleration values are averaged and low-pass filtered. Then:

$$T_B = [0, 0, -T]_B^{T.} \tag{4.9}$$

Where T is an obtained model from section 4.2. Note the negative sign in the body frame, considering z is defined positive downwards while thrust is directed upwards in the body frame.

$$G_E = R \cdot G_B = R \cdot [0, 0, 9.80665]_E^{T.} \tag{4.10}$$

And:

$$D_E = \frac{1}{2} \rho v_E^2 S C_d \tag{4.11}$$

Where v_E is the true airspeed which is what should be found. Therefore:

$$v = \sqrt{\frac{2(T - F - G)}{\rho S C_d}} \tag{4.12}$$

Or, in case as suggested in more recent literature, rotor drag does dominate in our flight speed regime, instead of Equation 4.11 we will use Equation 4.13.

$$v = \frac{a_{res}}{k} \tag{4.13}$$

Where k is the earlier defined constant. Both equations will be considered and compared, after which one will be selected. The outcome may not be as black and white though; from literature, it is expected that Equation 4.12 works better at high speed flights, and that Equation 4.13 works better at low speed flights.

It should be emphasized that the above equations are performed in 3D, meaning the output v looks like:

$$v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{4.14}$$

From there, the magnitude of true airspeed can be computed.

$$|v_{tas}| = v_{tas,x}^2 + v_{tas,y}^2 \tag{4.15}$$

From which the triangle method is used, as in Equation 4.16 (and displayed earlier in Figure 3.5).

$$v_{wind} = v_{gps} - v_{tas} \tag{4.16}$$

To find the direction, Equation 3.5 from chapter 3 is used. A plan to eliminate noise and quantify the uncertainty is to implement a particle filter.

4.2. PARAMETER ESTIMATION

Part of the challenge is making the methodology presented in this thesis work on a new type of drone, for which its technical details are often unlisted. For the Foxtech Hover 1, parameters such as moments of inertia, thrust coefficients and drag coefficients are not specified, while they are required for the planned state estimation. Therefore, this section of the preliminary thesis is dedicated to finding estimates of those parameters which are needed for wind estimation.

To give a holistic picture of which parameters need to be estimated, recall Figure 4.1. This summarises the overall approach of the thesis, with data streaming from the bottom to the top to find wind speed. Here, yellow boxed letters A, B, C and D represent parameter tests and how they fit into the main algorithm.

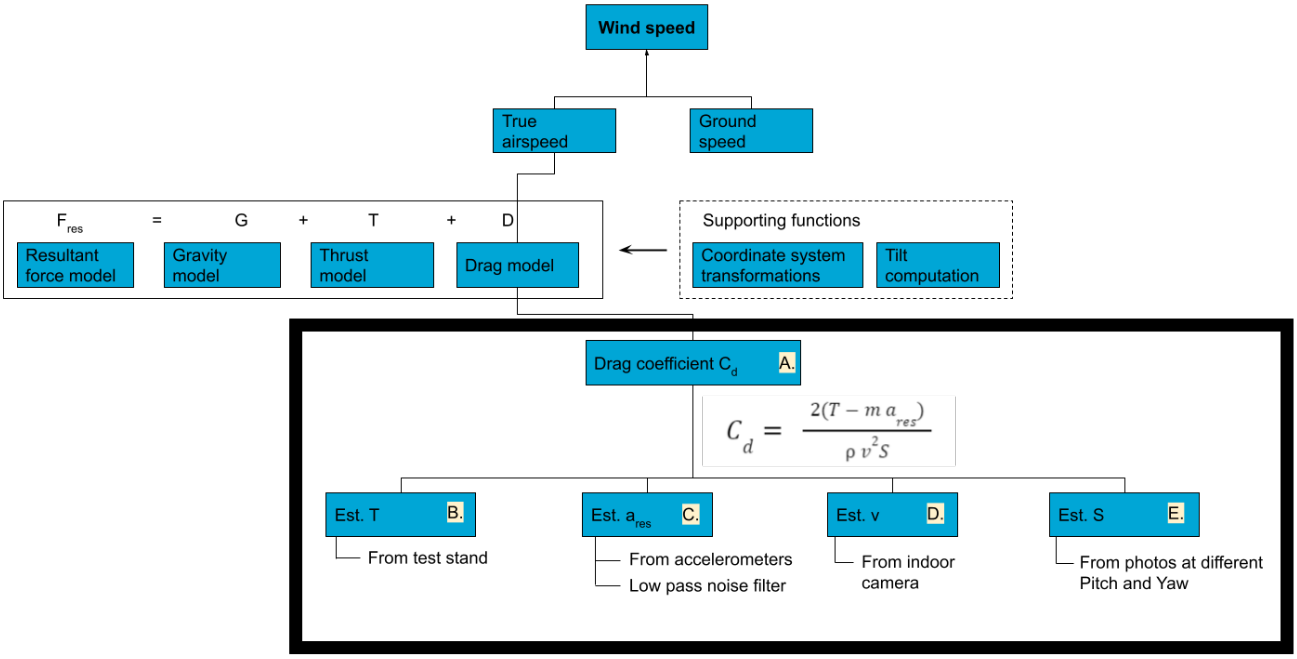


Figure 4.3: Schematic overview of how the parameters will help in estimating wind speeds and how they are modeled. Yellow boxed letters mark the parameters/models which are required for this test

This section will dive into depth into the preliminary outcome of these results for test A, B, C, D and E.

A. DRAG COEFFICIENT DETERMINATION

To determine the drag coefficient, recall section 3.6. From this follows the following equation:

$$C_d = \frac{2(T - ma_{res})}{\rho v^2 S} \tag{4.17}$$

Where T is thrust, determined later in this section, a_{res} is the (low-passed) resultant acceleration measured from the onboard IMUs, and S is the surface area determined later in this section. Note that the Foxtech possesses 3 IMU's, of which the first is the primary IMU and uses a vibration damper. The average of the three IMUs is used for further processing. Also note that the gravity component is ignored, as we only are interested in C_d, x and C_d, y (2D components).

Note that the computations in this section have been performed in the inertial axis frame, meaning components such as thrust and acceleration have been converted from body to inertial frame using Equation 4.14 as transformation matrix.

Drag Coefficient		
Quadratic with speed	$C_{d,x}$, $C_{d,y}$	0.14
Linear with speed	k_x, k_y	0.17

Table 4.1: Drag coefficients currently used for the Foxtech Hover 1

From the equation, several indoor flights have been performed, since then $v_{ground} = v_{airspeed}$. A total of 25 lines have been flown in several yaw, pitch and speeds. Later, this information is grouped (i.e. rounded off) into "buckets" of speed. The procedures is summarised in the following bullets and the results are summarised in Table 4.1.

During flights

1. Set the drone at a starting position and set a yaw angle
2. Fly a number of lines.
    - The more indoor flight room there is, the better the results will be as the drone will be more in trim condition. There will be less noise influence of acceleration measurements.
    - The more identical lines are flown, the more certain the result for a specific configuration will be.
    - The more variation in speeds flown, the clearer the trend will be.
3. Capture the speed by means of camera footage (or an indoor tracking system if available).
4. Increase yaw angle by fixed amount and repeat from step 1.
5. Stop when you have reached 90 degrees yaw

This procedure makes use of the assumption that the drone is symmetrical in x - and y, such that a yaw of 10 degrees gives similar results to that of a yaw of -10 degrees. Therefore, only 0-90 degrees yaw have been considered in this test.

Post processing

1. Group the flown lines into buckets of 1-2 m/s, 3-4 m/s, 5-6 m/s, 7-8 m/s, etc. up to 19-21 m/s (max speed).
2. Average out the yaw of the resultant lines which fall into the buckets
3. Plot the speed against drag coefficient, as shown in Figure 4.5.

The figures confirm that 0.10-0.30 is the range the drag coefficient remains within roughly, considering operating speeds (5 m/s-20 m/s). Note that Cd,z is not computed, but that there is data to compute this should this be necessary later on.

We have not been performing slow flights (1-5 m/s), so should be careful extrapolating the polynomial quadratic trend backwards. Namely, from the literature study in chapter 3 it shows that at lower speeds the Cd tends to behave more linearly to v as opposed to quadratic, and so it would be better to make purposely slow flights in this speed region and find a linear fit.

Although the fit is suggested to be quadratic, the uncertainty in the measurements cause the trend to not be extremely pronounced; n.b. a linear fit would have been believable as well. Therefore it is

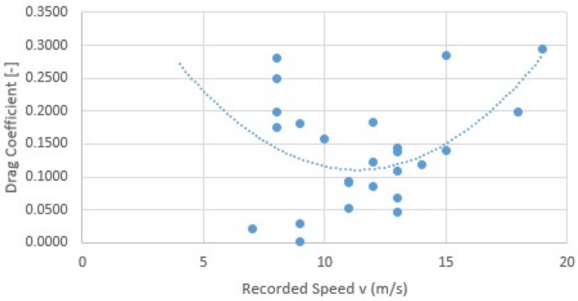


Figure 4.4: Overview of found Cd's, all shown per bucket of 1 m/s speed.

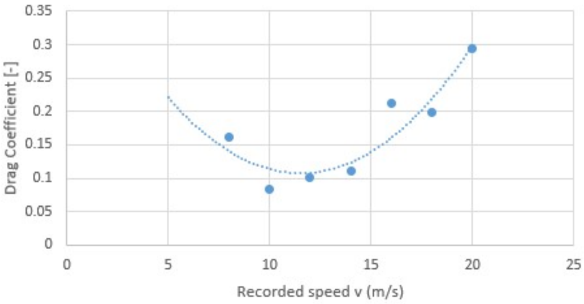


Figure 4.5: Overview of found Cd's, averaged per bucket of 2 m/s speed.

safest to assume a constant drag coefficient for all speeds, which is in this case the average of the 25 flown lines.

B. DETERMINING THRUST

Thrust is one of the main forces in the force balance approach when considering both the x and y axes. The Foxtech hover 1 does not provide information on thrust and so the test bench approach has been taken to map this. Using a Series 1580 Test Stand, a PWM to Thrust (N) transfer function has been found, which is describes in Equation 4.18.

$$T = 4 * T_{rotor} \tag{4.18}$$

Where:

$$T_{rotor} = 0.0226\left[\frac{1}{4}\sum_{n=1}^{4}(PWM)_n\right] - 27.01$$

Where PWM_n is the incoming signal of PWM per rotor, which is summed and then divided by the number of rotors n, as the transfer function is valid for a single rotor.

The results above have been deducted from discrete/static tests, where PWM has been increased in steps of 50, ranging between 1000 and 2000. An example of such a test is seen in Figure 4.6.

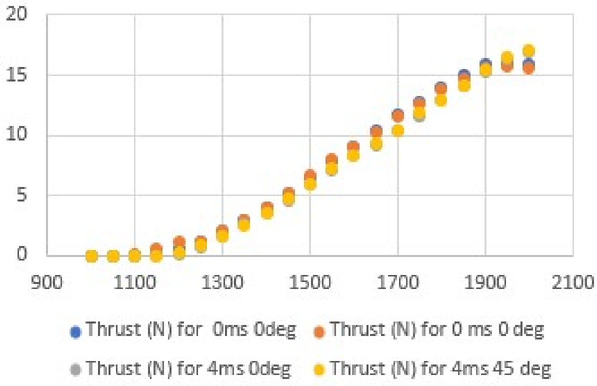


Figure 4.6: Static test overview, without (0 m/s) and with (4m/s) influence of a fan, so emulate wind inflow.

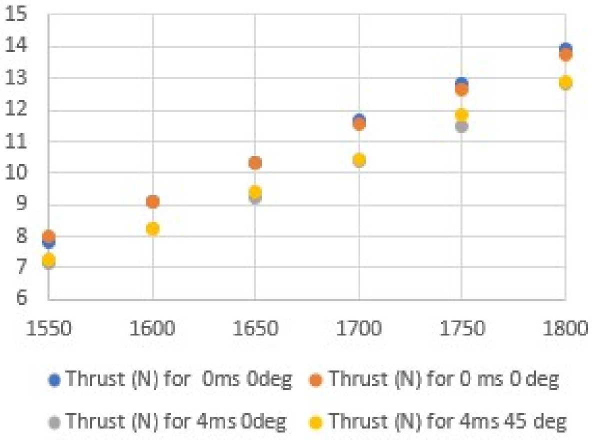


Figure 4.7: Static test overview zoomed in

What can be noted from zooming in in Figure 4.7 is that there is a max discrepancy of roughly 2N for the test with- and without wind influence. This is something to be aware of during state estimation. In code, one could lower the intercept of the chosen interpolation function, depending on the amount of wind-inflow. In order to make this curve's intercept a dynamic value (or even a constant, adjusted value), it is mandate to perform thrust tests again at a variety of air speeds ranging, above the speeds currently tested for (0 and 4 m/s) to be able to act to it appropriately.

Modeling the top end

Notably, the equation in this subsection is chosen as a linear fit, while the curve in Figure 4.6 shows nonlinear behaviour. This is because generally the rotor speeds of a drone remain relatively constant throughout operation [30]. Analyzing the log files for the Foxtech Hover 1 specifically, we note that the PWM always ranges around 1650, and always remains within 1500 and 1800, even when performing aggressive jump maneuvers (from 1 to 8.5m). This means modeling of maximum thrust dynamics is not required, as this only occurs in the area of 1850-2000 PWM. In case outdoor tests in September proof that in windy conditions PWM exceeds 1850 at some point, the plan is to fit the data points with a polynomial of order 3 or 4.

Influence of external windflow (Sweep tests)

In addition to the static test a sweep test has been performed to get an understanding of the dynamic effects of increasing and decreasing the PWM. These can help understand the influence of abrupt changes in the drone's pwm input on the thrust model (incrementing and decrementing PWM). Concisely, the results show a discrepancy between throttling up (lower line) and throttling down (upper line), which will be referred to as the *split effect*. When PWM is decreased continuously, there already is a larger inflow of air than required, and thus this gets added as excess thrust, resulting in a higher thrust for the same PWM.

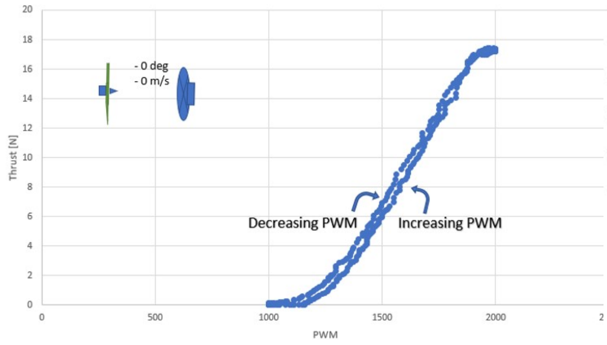


Figure 4.8: Sweep test at 0 m/s and 0 degrees fan influence. Going up in throttle requires more PWM to get the same level of thrust. So, if the PWM before the current PWM was lower, it will produce less thrust.

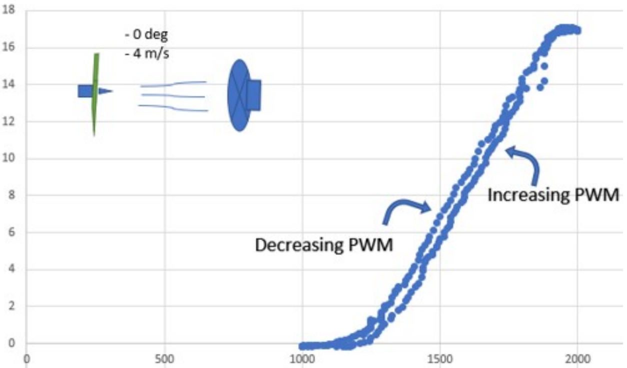


Figure 4.9: Sweep test at 4 m/s and 0 degrees fan influence

From these sweep test in Figure 4.10 it shows that for 4 m/s, 45 degrees inflow, the split effect is most pronounced. The discrepancy can be up to 2 N (roughly 10%) for just 4m/s and therefore should not be ignored in case of dynamic throttling. For this reason, we will restrict the research to measuring wind in (quasi) steady conditions, meaning we do not make measurements of wind when PWM is increased or decreased, as such to impose additional error from these dynamic effects. As seen in Figure 4.11, the static tests are a good representation of the sweep tests, as they are roughly the average of the two trends.

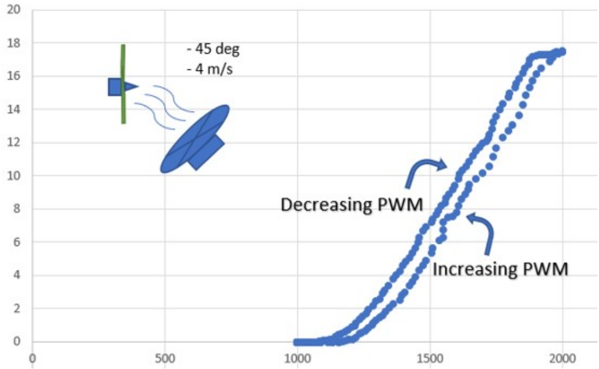


Figure 4.10: Sweep test at 4 m/s and 15 degrees fan influence

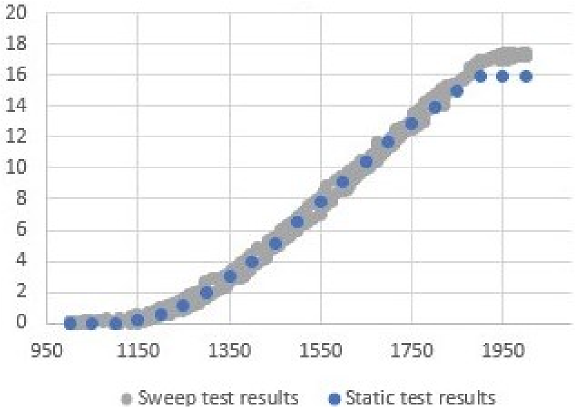


Figure 4.11: Sweep test compared to static test under similar conditions. 0 m/s at 0 deg fan influence.

A suggestion for future work could be to perform sweep tests under a wider variety of inflow speeds, similar to how it is done in [39]. Granted that the sweep tests show a very clear trend under a large variety of wind speeds, we could extend the methodology of wind estimation to the flight parts such as take of, landing, and aggressive turns as we are then able to model these dynamics more precisely.

Influence of voltage drop

The mapping of PWM to thrust is not a direct one, considering thrust is influenced by RPM of the propellors, battery voltage drop, temperature, and load [51]. Correction models can be set up to account for the influence of voltage drop, but interestingly, according to [51]) the PWM-thrust curve gives a better fit than the volt-thrust curve, arguing in favor of pwm-thrust relationship. Note that all bench tests in this section are performed at 25.4 volts, which is assumed representative for the whole flight profile.

C. DETERMINING RESULTANT FORCES

The resultant forces are found by the following steps.

1. Read out the three onboard IMUs, AccX, AccY and Accz in the body frame
2. Average the three IMUs
3. Use a low pass filter on the found result

4. Convert the IMU accelerations to the earth frame using the rotation matrix, shown in Equation 4.14.

The result is the acceleration to x, y and z in the earth frame.

D. Determining speed indoors

To fill in the v component in Equation 3.9, the ground speed is required, however the indoor environments as such do not allow for GPS to capture this. Therefore, to estimate this speed, a custom camera speed trap has been set up. It is an opencv script which detects moving objects and registers how many frames it takes for the drone to pass a known distance. This is then inserted into Equation 4.19.

$$v = \frac{s}{t} = \frac{s}{\frac{\Delta f}{FPS}} \tag{4.19}$$

Where Δf is the amount of frame counted in between passing the first- and the second green vertical line. In between, the distance s is known. During flight tests, the operator focused on staying at a fixed distance from the camera (over the black line on the ground) as such to minimise errors in distortion.

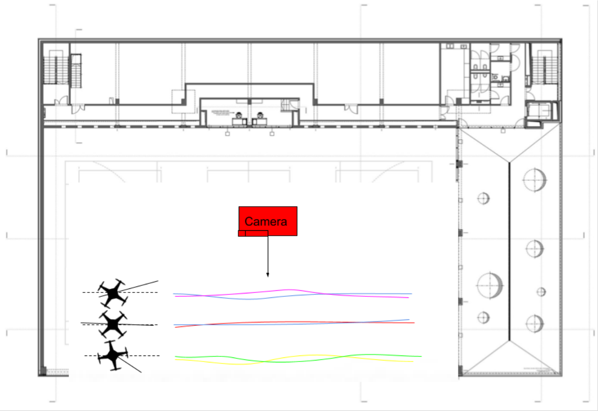


Figure 4.12: Schematic showing the drone passing the speed trap, schematically and conceptually (top view).

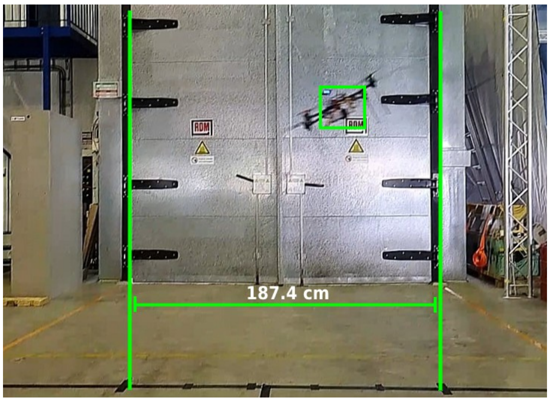


Figure 4.13: Drone passing the speed trap, video recorded at 480 fps.

The videos used are recorded by a smartphone in slow-motion such that there is 480 fps to divide by, rather than 30 fps, leading to a higher accuracy.

For the slowest flight, a speed of 4 m/s has been recorded, whereas for the most aggressive flight, a speed of almost 19 m/s is found, which is in line with the maximum speed of 20 m/s specified for the drone.

The data is matched by looking at the point at which the pitch angle changes aggressively, since that means the drone had to break, in order to not hit the wall at the end of the indoor hall. The half a second before that steep decrease in pitch is usually constant in pitch and PWM, and that represents the part which is flown in Figure 4.13.

E. DETERMINING FRONTAL AREA

The frontal area is modeled as two parts, as displayed in Equation 4.20. How both of these terms are estimated continuously is discussed in this section.

$$A = A_{bare}(v) + A_{prop}(v) \tag{4.20}$$

To determine the bare area (A_{bare}), a Python script is created which detects the object with color differentiation and counts the amount of pixels of a still photo of the drone. When placed next to a reference object of know size, by means of simple a cross tab it can estimate the frontal area. Repeating this procedure for a variety of pitch and yaw angles results in the values as displayed in Table 4.1. From the data, an interpolation function of pitch against (yaw averaged) frontal area is found, as seen in Figure 4.16. This results in the A_{bare} term as shown in Equation 4.21.

A theoretical approach to include propeller area A_{prop} has been taken, where the propeller area is assumed as a disk. The influence of pitch is accounted for by using basic trigonometry, explaining the sine term in Equation 4.22.

$$A_{bare}(v) = 0.043e^{0.0164v} \tag{4.21}$$

$$A_{prop}(v) = 4\pi R^2 P \sin(v) \tag{4.22}$$

Where v is the tilt angle, which is in this case assumed assumed to be similar to pitch.

From chapter 3, it is assumed a solid disk and this constant is set to 1.0.

Currently, the average of all yaw angles is taken, and the interpolation function shown in Figure 4.16 is used in the main algorithm. There, the tilt angle is assumed as if it is the pitch angle (because the yaw and thereby the tilt is averaged in this approach). Alternative implementations are discussed in section 3.6.

To mimick the incoming free flow, the photo has been taken from a distance (roughly 2m) to minimise the lens eye effect,. Later the photos have been cropped for easier processing.

Currently, the results for different yaw angles (and thereby different tilt angles) are averaged. Alternative implementations could however be to convert Table 4.4 into a pandas 2D interpolation table, which inputs both yaw and pitch and outputs the interpolated bare frontal area. We see no trend in

Estimated frontal area - computation

			Pitch Angle (deg)	
			15	
			Tennisball	Foxtech
Yaw angle	0	Px	22456	399345
		Area	0.003473	**0.06177**

Figure 4.14: Crosstab displaying how the reference object is used to compute the area of the Foxtech (number in bold).

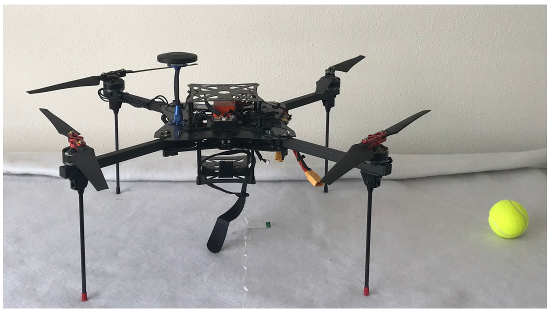


Figure 4.15: Example of frontal area image of yaw = 0 deg and pitch = 15 deg, with reference object of known size, both with a (fairly) neutral background to allow for color detection.

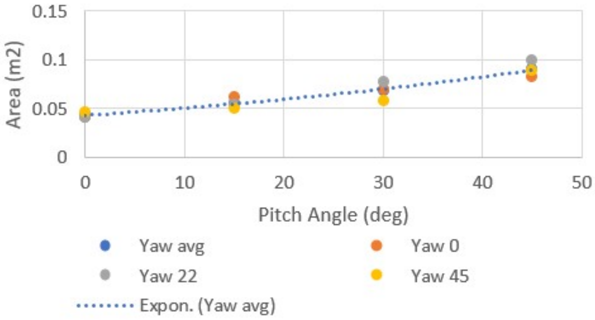


Figure 4.16: Bare frontal area at different yaw angles, scattered and interpolated.

Estimated frontal areas - overview

		Pitch Angle (deg)			
		0	15	30	45
Yaw Angle (deg)	0	0.0415	**0.0618**	0.0702	0.0835
	22.5	0.0408	0.0550	0.0783	0.1005
	45	0.0471	0.0499	0.0588	0.0890

Figure 4.17: Overview of all found results

changing the yaw angle, however, (i.e. the area roughly stays the same), for it is more logical to consider pitch (or tilt) as the only varying function.

Determining Moment of Inertia

For the current approach, we are following a linear kinematic approach, i.e. rotational dynamics are not included. Nevertheless, rotational dynamics are very common in drone modeling ([2], [39]), the drone will require knowledge of the moments of inertia, and so in preparation for that (and considering the low effort to determine it) this subsection describes how moments of inertia are found.

To determine the moments of inertia (MOI), from section 3.6 a geometric approach has been taken. First, the MOIs of each individual component have been computed. Second, their parallel axis term has been added. Lastly, the MOIs are rotated 45 degrees around the z-axis, since that is representative of the true "front", i.e. flying direction of the drone.

Results: MOI of Foxtech Hover 1

Without pole	Ixx	0.03962	Iyy	0.03962	Izz	0.06885	Ixy	0.0003374
With pole	Ixx	0.1964	Iyy	0.1964	Izz	0.06892	Ixy	0.0003374

Table 4.2: Moment of inertia following from the geometric method

The four rotors (red), as well as the pole on top of the drone (green), have been modeled as cylinders.

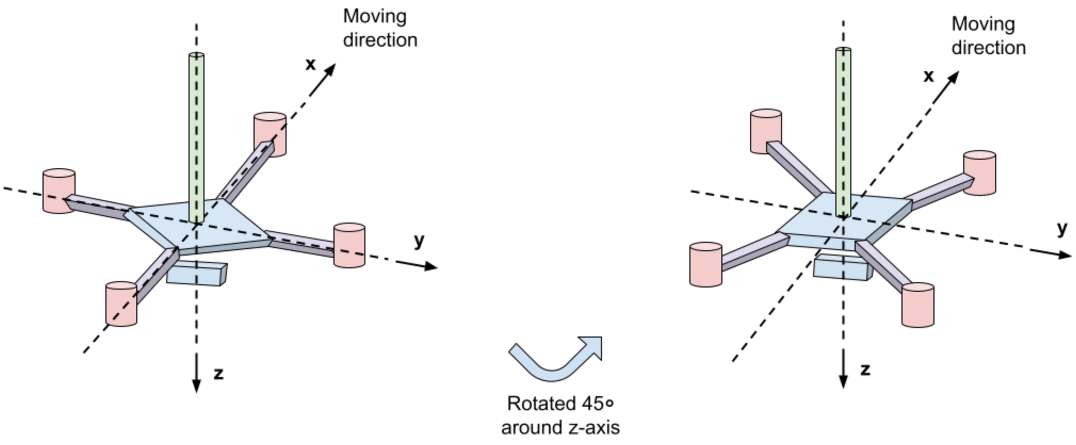


Figure 4.18: The Foxtech hover 1 simplified into blocks and cylinders. Then rotated around its z-axis to compute final MOI.

The equations to calculate their MOI around their respective axes are shown in Equation 4.23 and Equation 4.24.

$$I_{cyl,central.axis} = \frac{1}{2} M_{cyl} r_{cyl}^2 \tag{4.23}$$

$$I_{cyl,central.diam} = \frac{1}{4} M_{cyl} r_{cyl}^2 + \frac{1}{12} M_{cyl} l_{cyl}^2 \tag{4.24}$$

The central plate (blue), the battery (blue) and the legs (purple) have been modeled as boxes. This can be summarised in Equation 4.25.

$$I_{box} = \frac{1}{12} M_{box} (a_{box}^2 + b_{box}^2) \tag{4.25}$$

Where a and b are the dimensions as if looking from the principal axis. Then, the parallel axis theorem has been applied to each of the individual components, as displayed in Equation 4.26.

$$I = \sum_{k=1}^{n} (I + Md^2)_k \tag{4.26}$$

Where n is the amount of parts modeled. Then, Ixy has been computed (Equation 4.27), after which a 45 degrees rotation over the z-axis has been performed as in Equation 4.28.

$$I_{xy} = I_k + Axy \tag{4.27}$$

$$I_{xy,rotated} = \frac{I_x - I_y}{2} \sin 2\psi + I_{xy} \cos(2\psi) \tag{4.28}$$

And for I_x and I_y these rotations follow:

$$I_{x,rotated} = I_x \cos^2\psi + I_y \sin^2\psi - I_{xy} \sin 2\psi \tag{4.29}$$

$$I_{y,rotated} = I_x \sin^2\psi + I_y \cos^2\psi + I_{xy} \sin 2\psi \tag{4.30}$$

Where ψ is the amount to rotate around z.

This methodology has been converted into an Excel sheet found in Appendix B, where the user is asked to insert mass, dimensions and offsets of centres of gravity. From there, it will compute the moments of inertia as shown in Table 4.2.

Comparing the results in Table 4.2 to that of the Parrot Bebop (which has an I_x of roughly 0.0010) the results seem realistic, considering the Bebop is approximately $\frac{1}{2}$ smaller in size (quadratic) and a factor of 7 times lower in mass, while the MOI is roughly 40 times less. Also, comparing it to Mendes *et al.* [44], it is in line with expectations that the Izz is roughly double the value of Ixx and Iyy.

Gonzalez-Rocha et al. [39] use a HobbyKing Quantum Nova quadrotor of comparable size and finds I's of 0.033, 0.037 and 0.064 for Ixx, Iyy and Izz respectively.

Computation	**Equation used for determining MOI**	
Cylinder	$I_{cyl,centralaxis}$	$= \frac{1}{2} M_{cyl} r_{cyl}^2$
	$I_{cyl,centraldiameter}$	$= \frac{1}{4} M_{cyl} r_{cyl}^2 + \frac{1}{12} M_{cyl} l_{cyl}^2$
Box	I_{box}	$= \frac{1}{12} M_{box} (a_{box}^2 + b_{box}^2)$
Parallel axis theorem	I	$= \sum_{k=1}^{n} (I + Md^2)_k$
Rotated inertias	I_{xx}	$= I_x \cos^2\psi + I_y \sin^2\psi - I_{xy} \sin 2\psi$
	I_{yy}	$= I_x \sin^2\psi + I_y \cos^2\psi + I_{xy} \sin 2\psi$
	I_{zz}	$= I_{zz}$ (same as before)
	I_{xy}	$= \frac{I_x - I_y}{2} \sin 2\psi + I_{xy} \cos 2\psi$

Table 4.3: Overview of all MOI equations applied to Figure 4.18, where ψ is the rotation around z, in this case 45 deg.

4.3. TESTING FRAMEWORKS

This section briefly addresses the tools that can be used to verify and validate the methods proposed in this preliminary report.

ANALYTICAL MODEL

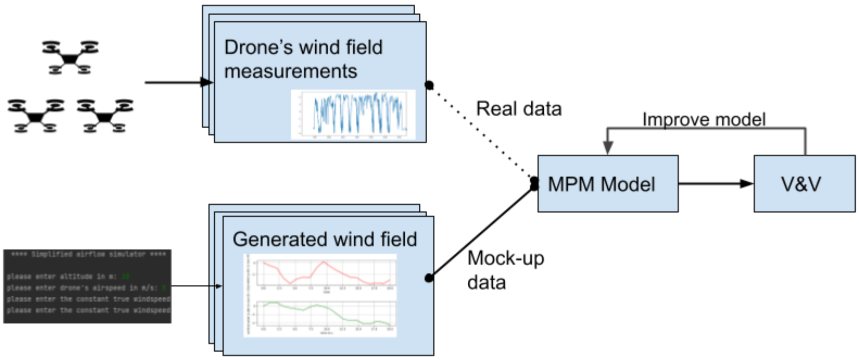
To verify that the model is implemented well into Python, some exemplary inputs are taken and written down on paper. These will be analyzed with a handheld calculator, where-after the outputs will be compared to that of the Python model to detect any coding errors. This will serve as verification source.

The input data should be data from representative parts of flight, i.e. parts where the flight is relatively steady and does not increase- or decrease throttle. Flight data from past year servers well for this purpose, since it has straight lines. Data from next year will provide for even better verification (see chapter 5), as it will contain flights directly into- and away from the wind.

WIND FIELD GENERATOR

For validation purposes, the first point of testing could be a virtual wind tunnel, which has been designed in Python. It uses the Dryden model (adapted for low altitudes) to create realistic gusts for a given speed, and superimposes that onto a constant wind from a selected speed and heading to form a representation of real wind. [37] has served as a starting point for this program.

Initially this is meant to test the wind detection algorithm, but in order to do that one would need a high fidelity plant model (with a fidelity higher than the wind detection model). Since this is unknown for the Foxtech Hover 1 (and out of scope to focus on), this is left as a future recommendation. The wind field generator can, however, also be used as reference "measured" values from the drone to go into the MPM software, to test that further, as is shown in Figure 4.19.



```
**** Simplified airflow simulator ****

please enter altitude in m: 10
please enter drone's airspeed in m/s: 5
please enter the constant true windspeed
please enter the constant true windspeed
```

Figure 4.19: Wind field generator, showing how it can emulate the drones in a network when ran in concurrency with slightly different values.

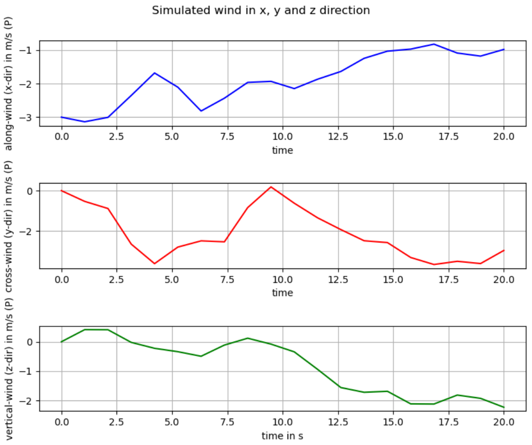


Figure 4.20: Continuous wind field generator output, with wind fields shown per axis.

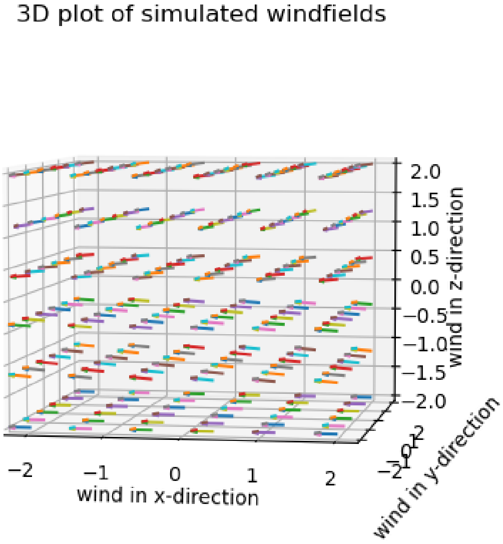


Figure 4.21: Continuous wind field generator output, with wind fields shown in a 3D display.

Similarly, this model can be a meaningful contribution to TU Delft's BlueSky [52], since currently the wind implementation is a discrete model rather than continuous. The model will be written in Python and is fully object oriented, such that later, potentially it can be merged into BlueSky.

MISSION PLANNER

Mission planner helps in verification in the sense that you can connect data and that the following can be tested.

- Test the data linkage of the radio before making actual flights
- Verify that the signs in the coordinate system are as expected (by tilting the drone to a side and seeing whether the outputted angle is positive or negative)
- Practice the online data retrieval, either via the "web-scraping method" as in [17] or via Mission Planner.
- Practise the synchronization of state-estimated wind speed and sensor-recorded wind speeds

Note that one can not get log or bin files from the drone, since this only activated when the drone is armed, and that only happens when the batteries are in place. This is not allowed until testing day.

FLIGHT DATA FROM PAST YEARS

The most valuable asset in terms of V&V is flight data from past year. Namely, from here the flight data is logged on an sd card, together with the wind data from wind sensor of the drone in hover. Using this Trisonica wind sensor's data as validation source, we can see if the estimated wind speeds of the other three drones are in line.

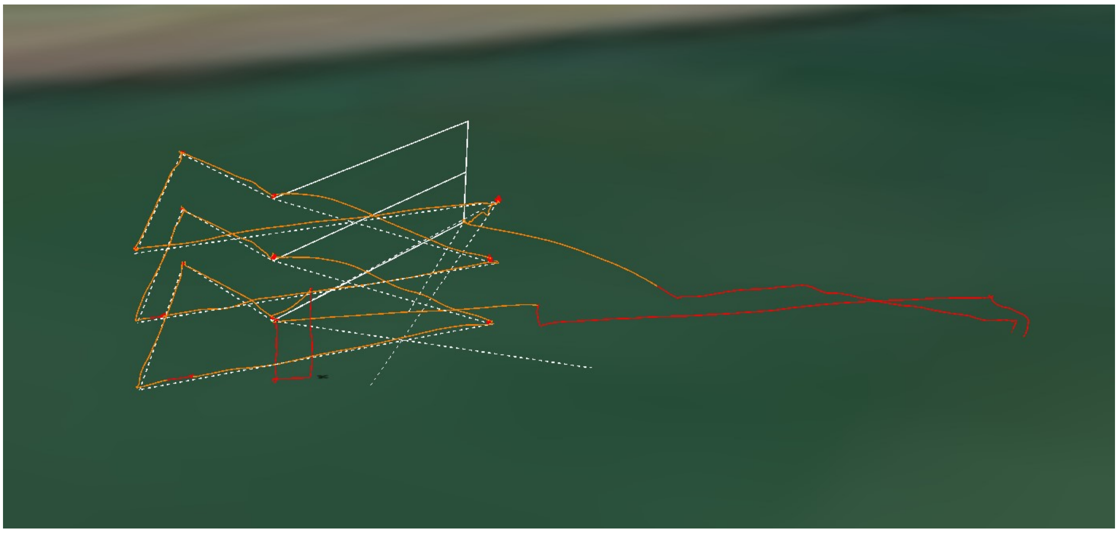


Figure 4.22: Flight data from past year, flown in a triangle at three altitudes.

In addition to pas year's data, new data will be generated as discussed in chapter 5.

5
Experiment and flight test design

5.1. Flight plan

As a follow up to section 4.3, a new flight test will be performed in early November 2022. This way, new data is generated which will serve for validation purposes. Some differences from this test to the year before and the reasoning for it are described in the following list.

Novelty and rationale for new flight plan

- This flight plan will fly in a box rather than a triangle. This is done to make verification easier. For one, flying straight into the wind means you have one less axis to worry about (as your y-wind component will roughly be zero), making it easier to validate the conversion of body to inertial frame. Secondly, the box will mean that the drone flies straight into the wind and straight away from the wind, and from that we can better verify whether the drone works in these two most "extreme" conditions (which is interesting considering the pitch will be different while the true airspeed is the same). Lastly, the square could serve as a validation source for the drag coefficient, considering [11] estimated drag by flying in boxes, arguing that the bias of the wind is “averaged out” since you flying opposing directions considering wind.
- This flight plan will, include a few tests under 7m/s and a few tests above 10m/s. The reason is that papers hint that drag acts linear to speed at low speeds and quadratic at high speeds. To confirm this (and see whether we can adopt to this), it is wise to at least have some data of higher velocity speeds. Examples could be 5 and 15 m/s.
- This flight plan will, in addition to a hovering flight, contain a moment of loitering/casting where the drone is hovering but not held to its x and y position. This will cause it to "drift" off, and that should tell us something about the wind speed, according to [34]. Past year’s flight plan did not have such a drift.
- This year will fly one drone with an anemometer pole, and the other without. This will tell whether the algorithm works as expected for different configurations and mass, which is expected.
- This flight the wind most likely will be stronger, considering the test takes place in early October when weather tends to get more turbulent around the flight location (Marknesse, in the Netherlands). This will result in additional validation data, as opposed to that of previous year, which had low wind speeds (< 5 m/s).
- This year, the focus will not be on the connectivity between the drones, so one drone for measuring and one drone for reference will be sufficient. Therefore, also only one operator is required.

Consider that in Figure 5.1 each of the yellow lines define a roughly 100 m distance, which should give the drone 20 seconds enough to get to cruise at a speed of 5 m/s.

Note that in addition to the yellow lines, the red marker represents a second drone which is used for reference measurement. As this drone remains in hover, it is expected that it will remain relatively stable, providing a good basis for the wind sensor to function on, thus yielding a reliable reference.

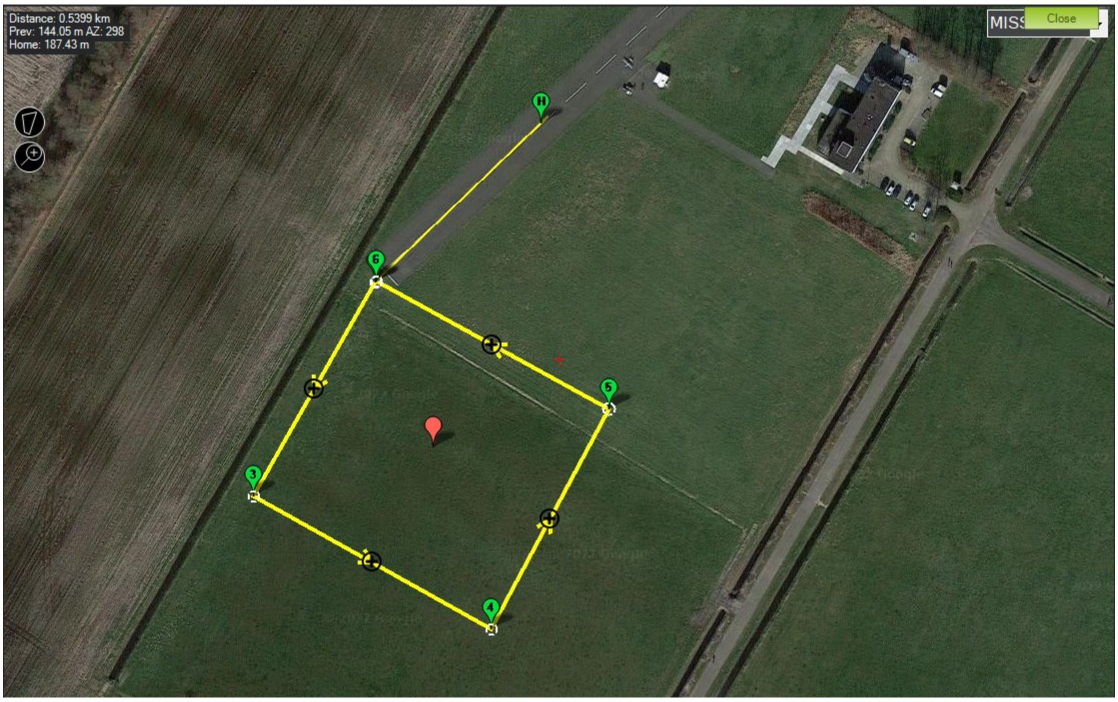


Figure 5.1: Exemplary flight plan at Marknesse location. Subject to change as the square is supposed to be exactly perpendicular to the wind direction.

5.2. DATA FLOW AND HANDLING

Currently, the data is logged onto an SD card and from there, wind is determined in post processing. This is shown in the "post flight" part of Figure 5.2. Looking more closely, the log files originating from the SD card are decrypted from bin/log to a MATLAB structure using an open source tool (ArduPilog). From there, a self-made MATLAB script converts the data variables we need to a csv file. This is where Python script is able to read it and process the data into wind estimates.

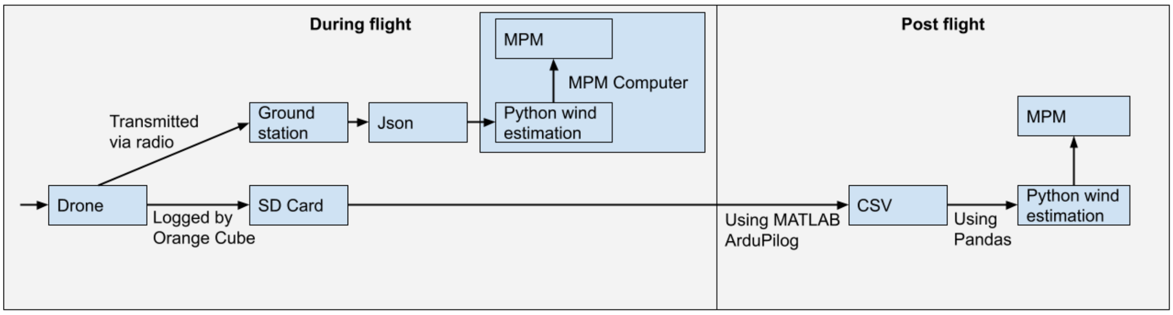


Figure 5.2: Data flow for post processing and for online processing.

For a more insight on how data flows from the drone of the groundstation, consider the technical report of METSIS [17].

5.2.1. DATA RATES AND SAMPLING TECHNIQUES

Note that during the MATLAB conversion to csv, some data is up-sampled (using Matlab's interp1D()) and some is down-sampled (using MATLAB's downsample()), as these states are recorded in different frequencies. The IMU records at 17 Hz, while the angles from pitch and PWM are recorded in 8.5 Hz. The lowest frequency of recording is that of the GPS, with 4.25 Hz. For now, this is up sampled to 8.5 Hz by means of linear interpolation, but plans are to experiment with down sampling the true airspeed once found to this lowest frequency (4 Hz).

6
Project Management & Work packages

6.1. Work packages

In essence, the work packages are split up in the following work packages.

Work packages

- Project management work package (2 h/w)
- Kinematic approach work package (10 h/w)
- Filtering work package (16 h/w)
- V&V Work package (20 h/w)

6.2. Milestones and Meetings

An overview of upcoming milestones and meetings is displayed in Table 6.1.

Date	Milestone
March 17th	Kick off
April 15th	Comparing three best methods
June 9th, 10th	Parameter tests in Marknesse
Aug 25th	Hand in Preliminary Report
Sep 2nd	Present Preliminary Presentation
Nov 1st	Flight test in Marknesse
Nov 18th	Green light and hand in draft paper
Dec 15th	Hand in final version of paper
Dec/Jan	Defense

Table 6.1: Important dates and milestones for the thesis, summarized in a table

6.3. Logbook

To keep track of progress, a logbook has been set up and maintained over the past months. An example of a month is shown in Figure 7.2.

For a more detailed description of the work in the past few months, consider the full logbook, which is handed in together with this preliminary report.

Also, note that a lot of progress is discussed in the Weekly and Official meetings, of which the slides will also be made available upon request.

6.4. Focus per week for the coming weeks

To better visualise the points of attention globally, consider Figure 7.1 in the Appendix.

7
CONCLUSION

This preliminary report outlines the literature research, preliminary results and plan of approach for the coming months on working towards completion of the MSc thesis. As a contribution to METeo Sensors In the Sky (METSIS), the thesis tries to improve wind estimates without using dedicated sensors, answering the following research question.

"How can onboard states and intelligent filtering be used on the Foxtech Hover 1 to measure 2D wind speeds with an accuracy of 0.5 m/s and 5 deg?"

The sub goals are stated in chapter 2 and essentially come down to making the methodologies presented in this thesis as accurate, scalable and cost effective as possible.

To make the methods more accurate, one could increase complexity of the model (e.g. by including rotational dynamics) in case desired accuracy is not reached, similar to how [39] has an increasing-complexity approach. In case results are noisy (which is expected), a particle filter will be introduced for better observations. Comparing recent papers, the state of the art accuracy is between 0.3-1.5 m/s RMSE, making this a challenging but feasible target. To make the methods scalable, the parameter tests are performed as such that any type of drone (operator) could execute them. For example, no windtunnel tests are performed as it is expected flight data under controlled environment can provide good enough estimates of parameters. To make this approach cost effective, the use of sensors will be avoided, saving weight and cost.

Later, chapter 3 shows that in literature, many papers agree upon kinematic approach towards finding the drag term, but vary on how to resolve this term to find true airspeed. While some papers consider drag force quadratic with speed (classical), others find that for lower speeds a linear relation with speed yields better results, and others use momentum theory. This thesis will focus on implementing linear kinematic analysis to find the drag term, and resolving it into true airspeed by using the classical drag formula approach. This should yield an estimate for true airspeed, which is then deducted from the GPS speed vector in order to find an estimate of the wind speed and direction (i.e. the triangle method). Papers suggest different kinds of filters and observers, but have not implemented a particle filter for this purpose.

Later in chapter 4, section 4.1 explains the main method planned and section 4.2 shows the parameter estimation results. In brief, drag coefficients are estimated from indoor flight, moment of inertia is calculated analytically, and frontal area and speed are estimated by using camera imagery.

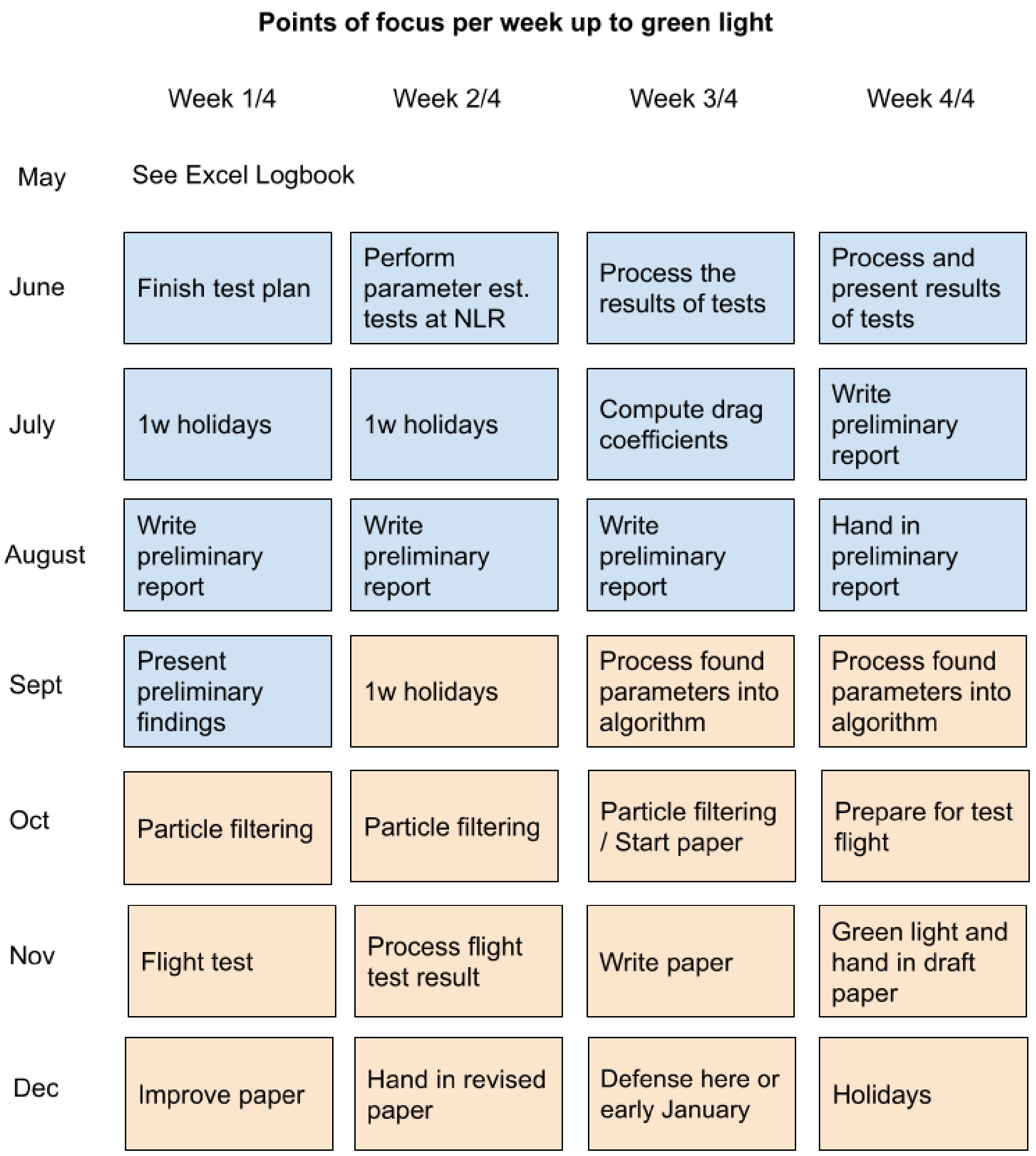
Lastly, chapter 5 shows the flight plan which can serve as verification for this thesis.

The estimation script is in progress but is expected to work in hover as well as cruise conditions, but not work where aggressive manoeuvres are performed. Namely, when pwm is increased or decreased, a large discrepancy in the thrust model is found, as presented in section 4.2. A suggestion to thrust curves under more wind conditions is made to be able to extend the model to other parts of flight.

One of the motivations for this research is the societal impact this could have on the many sectors, including healthcare, surveillance and also consumer deliveries. The outcome might contribute to finding a standard for all drones in the future, such that wind can be measured at any time, on any drone, during any (routine) job performed within U-Space. In turn, this could contribute to a more efficient drone ecosystem.

Appendix A: Tables and Figures

Weekly planning

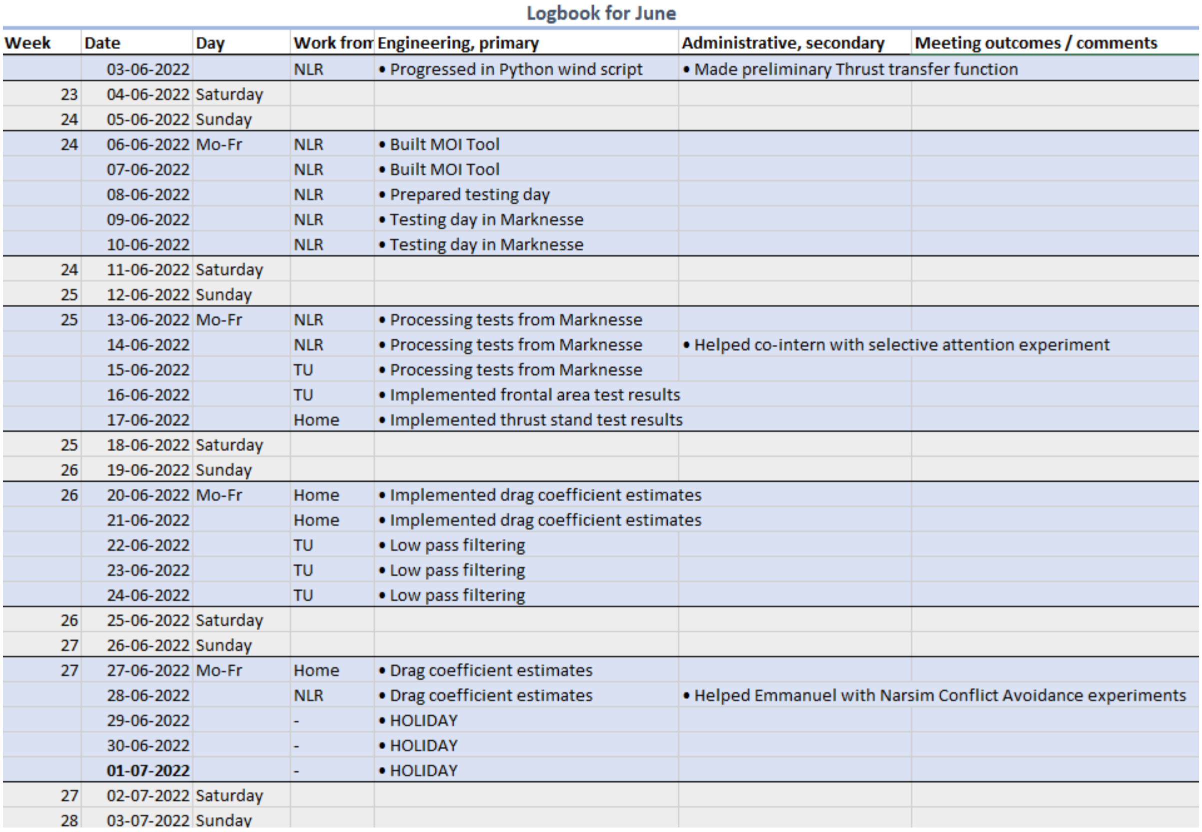


Points of focus per week up to green light

	Week 1/4	Week 2/4	Week 3/4	Week 4/4
May	See Excel Logbook			
June	Finish test plan	Perform parameter est. tests at NLR	Process the results of tests	Process and present results of tests
July	1w holidays	1w holidays	Compute drag coefficients	Write preliminary report
August	Write preliminary report	Write preliminary report	Write preliminary report	Hand in preliminary report
Sept	Present preliminary findings	1w holidays	Process found parameters into algorithm	Process found parameters into algorithm
Oct	Particle filtering	Particle filtering	Particle filtering / Start paper	Prepare for test flight
Nov	Flight test	Process flight test result	Write paper	Green light and hand in draft paper
Dec	Improve paper	Hand in revised paper	Defense here or early January	Holidays

Figure 7.1: Weekly planning up to defense

LOGBOOK EXAMPLE



Logbook for June

Week	Date	Day	Work from	Engineering, primary	Administrative, secondary	Meeting outcomes / comments
	03-06-2022		NLR	• Progressed in Python wind script	• Made preliminary Thrust transfer function	
23	04-06-2022	Saturday				
24	05-06-2022	Sunday				
24	06-06-2022	Mo-Fr	NLR	• Built MOI Tool		
	07-06-2022		NLR	• Built MOI Tool		
	08-06-2022		NLR	• Prepared testing day		
	09-06-2022		NLR	• Testing day in Marknesse		
	10-06-2022		NLR	• Testing day in Marknesse		
24	11-06-2022	Saturday				
25	12-06-2022	Sunday				
25	13-06-2022	Mo-Fr	NLR	• Processing tests from Marknesse		
	14-06-2022		NLR	• Processing tests from Marknesse	• Helped co-intern with selective attention experiment	
	15-06-2022		TU	• Processing tests from Marknesse		
	16-06-2022		TU	• Implemented frontal area test results		
	17-06-2022		Home	• Implemented thrust stand test results		
25	18-06-2022	Saturday				
26	19-06-2022	Sunday				
26	20-06-2022	Mo-Fr	Home	• Implemented drag coefficient estimates		
	21-06-2022		Home	• Implemented drag coefficient estimates		
	22-06-2022		TU	• Low pass filtering		
	23-06-2022		TU	• Low pass filtering		
	24-06-2022		TU	• Low pass filtering		
26	25-06-2022	Saturday				
27	26-06-2022	Sunday				
27	27-06-2022	Mo-Fr	Home	• Drag coefficient estimates		
	28-06-2022		NLR	• Drag coefficient estimates	• Helped Emmanuel with Narsim Conflict Avoidance experiments	
	29-06-2022		-	• HOLIDAY		
	30-06-2022		-	• HOLIDAY		
	01-07-2022		-	• HOLIDAY		
27	02-07-2022	Saturday				
28	03-07-2022	Sunday				

Figure 7.2: Example of logbook in Excel of the month June, showing points of focus per day

Overview of sensors considered

	Sensor type	Advantage	Disadvantage	Explored in e.g.	Suitability for quadcopters
1	Ultrasonic anemometer	High range	Ineffective at large angles of attack	METSIS	Medium
2	Miniature anemometer	Lightweight	Lower accuracy, requires gimbal .	Thielicke,	Low
3	Solid state anemometer (MEMS)	Modern	Very prone to noise Relatively new	Bruschi	Low
4	Hot wire anemometers	Accurate, small, lightweight	Directional, very fragile	Wolf et al	Low
5	(differential) pressure sensors	Factor of 10 cheaper than anemometer	Directional, not for low speeds, requires free stream velocity	Marino	Low
6	K-Vane sensor		Mechanical parts, added complexity, fragile, low accuracy		Low
7	Camera (optical flow)	Low cost, versatility	Difficult to implement, expensive, added complexity	Claybrough, Ho et al	Medium
8	LIDAR	Very accurate	Extremely heavy, very expensive, low spatial resolution	Marino, Donnell	Low
9	No dedicated sensor	No added cost, No added weight	Difficult to implement, not reliable in all flight conditions		High

Table 7.1: Overview of sensors available for use on quadcopters, with their strongest ad- and disadvantage. Note that all of the papers present results between roughly 0.1 and 1.5 m/s RMSE [m/s]

Overview of papers with indirect methods

Author	State estimation	Parameter estimation
Kinematic approaches		
Luo et al.	Use a linear relationship between drag force and speed	Cd estimated from indoor flights (45m)
Canos	Uses a direct tilt approach, a linear equation method and a Kalman filter and compares the three methods (in Simulink)	Cd assumed as if an angled cube Frontal area estimated by photos
Gonzalez et Rocha	Use three versions of their models, based upon aircraft EOM, with increasing complexity.	Cd is estimated from indoor flight (150m)
Xiang et al.	Computes thrust by ignoring drag in z-direction and resolving mg = T. Results in hover far better than while moving.	Assumes many parameters
Bouadi et al	Presents EOM including control inputs Heavily dependent on accurate RPM measurements	
Sikkel et al.	Estimate wind fields but ignore gusts (derivative of wind = 0). Also implement an EKF.	Tunes the drag coefficient by hand depending on flights and recommends online tuning
Luukonen	Studied quadcopter dynamics (so not wind estimation specifically)	One of the first to assume Cd is linear with v
Tilt approaches		
Hattenberger et al	Uses tan(alpha) = k v to relate tilt to wind speed. Converges slowly due to vdot = 0 assumption. Also corrects final offset in estimation by hand.	All parameters lumped into k
Neumann and Bartholmai	Estimates drag force from tilt relationship, and from that relationship finds v. Requires wind tunnel testing.	Conversely, estimate drag coefficient "classically". Surface area is found using 3D software.
Palomaki et al	Relates tilt to v quadratically (v2 = k tan(tilt))	Finds k from flying outside. No windtunnel required.
Marino et al	Uses a theoretical mapping between inclination and drag, and then finds thrust required. The difference is accredited to differential thrust, from which actuator disk theory gives a velocity estimate.	Cd assumed from a flat disk (1.28). Area computed as if flat disk. Extensive wind tunnel tests
Donnel et al	v = k tan(alpha)	k found by flying in boxes outside Verification using LIDAR
Filtering approaches		
Pappu et al	Uses a Kalman filter. See appendix for states and inputs.	Uses expensive commercial software tools to estimate parameters
Yong Ko et al.	Uses an EKF for navigation in between buildings when low gps signal	
Schafer	Uses a Kalman filter to estimate gusts specifically, but already has a MATLAB model	

Appendix B: Code

In the following list, the links redirect the reader to the Github repositories.

- https://github.com/erikbaas/thesis_wind_estimation_drones
- https://github.com/erikbaas/speed_trap
- https://github.com/erikbaas/frontal_area_detector
- https://github.com/erikbaas/windfield_generator
- https://github.com/erikbaas/compute_moment_of_inertia

Bibliography

[1] E. Sunil, R. Koerse, S. van Selling, J.-W. Doorn, T. Brinkman, and J. Sun, *METSIS: Hyperlocal Wind Nowcasting for U-space,* (2021).

[2] D. Park, T.-L. Le, N. V. Quynh, N. K. Long, and S. K. Hong, *Online Tuning of PID Controller Using a Multilayer Fuzzy Neural Network Design for Quadcopter Attitude Tracking Control,* Frontiers in Neurorobotics **14**, 619350 (2021).

[3] M. Koken, *The Experimental Determination of the Moment of Inertia of a Model Airplane,* , 17.

[4] J. Pérez Castán, V. Gomez Comendador, A. Cardenas-Soria, D. Janisch, and R. Valdés, *Identification, Categorisation and Gaps of Safety Indicators for U-Space,* Energies **13**, 608 (2020).

[5] K. Kabbabe Poleo, W. J. Crowther, and M. Barnes, *Estimating the impact of drone-based inspection on the Levelised Cost of electricity for offshore wind farms,* Results in Engineering **9**, 100201 (2021).

[6] S. Allison, H. Bai, and B. Jayaraman, *Wind estimation using quadcopter motion: A machine learning approach,* Aerospace Science and Technology **98**, 105699 (2020).

[7] J. Sun, H. Vû, J. Ellerbroek, and J. M. Hoekstra, *Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model,* PloS One **13**, e0205029 (2018).

[8] WMO, *Measurement of meteorological variables - Chapter 5: Measurement of surface wind,* .

[9] S. E. S. A. R. . J. U. E. body or agency), *European drones outlook study: unlocking the value for Europe* (Publications Office of the European Union, LU, 2017).

[10] T. A. Rodrigues, J. Patrikar, N. L. Oliveira, H. S. Matthews, S. Scherer, and C. Samaras, *Drone flight data reveal energy and greenhouse gas emissions savings for very small package delivery,* Patterns **0** (2022), 10.1016/j.patter.2022.100569, publisher: Elsevier.

[11] G. W. Donnell, *Wind Characterization Using Onboard IMU of sUAS | 2018 Atmospheric Flight Mechanics Conference,* Archive Location: world.

[12] L. Palazzetti, *Routing Drones Being Aware of Wind Conditions: a Case Study,* (2021).

[13] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, *Comprehensive Energy Consumption Model for Unmanned Aerial Vehicles, Based on Empirical Studies of Battery Performance,* IEEE Access **6**, 58383 (2018), conference Name: IEEE Access.

[14] S. Watkins, J. Burry, A. Mohamed, M. Marino, S. Prudden, A. Fisher, N. Kloet, T. Jakobi, and R. Clothier, *Ten questions concerning the use of drones in urban environments,* Building and Environment **167**, 106458 (2020).

[15] V. Alarcón, M. García, F. Alarcón, A. Viguria, Martínez, D. Janisch, J. J. Acevedo, I. Maza, and A. Ollero, *Procedures for the Integration of Drones into the Airspace Based on U-Space Services,* Aerospace **7**, 128 (2020), number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[16] D. Hollenbeck, M. Oyama, A. Garcia, and Y. Chen, *Pitch and Roll Effects of On-board Wind Measurements Using sUAS,* in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)* (2019) pp. 1249–1254, iSSN: 2575-7296.

[17] R. Koerse, *Final thesis: Meteo Sensors in the Sky (METSIS),* , 65.

[18] J. M. Cano, *Quadrotor UAV for wind profile characterization,* undefined (2013).

[19] M. Marino, A. Fisher, R. Clothier, S. Watkins, S. Prudden, and C. S. Leung, *An Evaluation of Multi-Rotor Unmanned Aircraft as Flying Wind Sensors,* International Journal of Micro Air Vehicles **7**, 285 (2015), publisher: SAGE Publications Ltd STM.

[20] W. Thielicke, W. Hübert, U. Müller, M. Eggert, and P. Wilhelm, *Towards accurate and practical drone-based wind measurements with an ultrasonic anemometer,* Atmospheric Measurement Techniques **14**, 1303 (2021), publisher: Copernicus GmbH.

[21] P. Bruschi, M. Piotto, F. Dell'Agnello, J. Ware, and N. Roy, *Wind Speed and Direction Detection by Means of Solid-state Anemometers Embedded on Small Quadcopters,* Procedia Engineering Proceedings of the 30th anniversary Eurosensors Conference – Eurosensors 2016, 4-7. Sepember 2016, Budapest, Hungary, **168**, 802 (2016).

[22] Anemoment, *UAV-Application-Brochure-Dec2020.pdf,* .

[23] N. Vasiljević, M. Harris, A. Tegtmeier Pedersen, G. Rolighed Thorsen, M. Pitter, J. Harris, K. Bajpai, and M. Courtney, *Wind sensing with drone-mounted wind lidars: proof of concept,* Atmospheric Measurement Techniques **13**, 521 (2020), publisher: Copernicus GmbH.

[24] A. Rautenberg, M. Graf, N. Wildmann, A. Platis, and J. Bange, *Reviewing Wind Measurement Approaches for Fixed-Wing Unmanned Aircraft,* Atmosphere **9** (2018), 10.3390/atmos9110422.

[25] P. P. Neumann and M. Bartholmai, *Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit,* Sensors and Actuators A: Physical **235**, 300 (2015).

[26] B. Luo, M. Zeng, and Q.-H. Meng, *A Wind Estimation Method with an Unmanned Rotorcraft for Environmental Monitoring Tasks,* Sensors **18**, 4504 (2018), number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[27] C. A. Wolf, R. P. Hardis, S. D. Woodrum, R. S. Galan, H. S. Wichelt, M. C. Metzger, N. Bezzo, G. C. Lewin, and S. F. de Wekker, *Wind data collection techniques on a multi-rotor platform,* in *2017 Systems and Information Engineering Design Symposium (SIEDS)* (2017) pp. 32–37.

[28] S. P. H. Driessen, N. H. J. Janssen, L. Wang, J. L. Palmer, and H. Nijmeijer, *Experimentally Validated Extended Kalman Filter for UAV State Estimation Using Low-Cost Sensors,* IFAC-PapersOnLine 18th IFAC Symposium on System Identification SYSID 2018, **51**, 43 (2018).

[29] S. Schafer, J. F. Horn, and J. Cooper, *Flight test measurement of ship airwake disturbances using small-scale rotorcraft,* Annual Forum Proceedings - AHS International **4**, 2681 (2015).

[30] G. Hattenberger, M. Bronz, and J.-P. Condomines, *Estimating wind using a quadrotor,* International Journal of Micro Air Vehicles **14**, 17568293211070824 (2022), publisher: SAGE Publications Ltd STM.

[31] R. T. Palomaki, N. T. Rose, M. v. d. Bossche, T. J. Sherman, and S. F. J. D. Wekker, *Wind Estimation in the Lower Atmosphere Using Multirotor Aircraft,* Journal of Atmospheric and Oceanic Technology **34**, 1183 (2017), publisher: American Meteorological Society Section: Journal of Atmospheric and Oceanic Technology.

[32] T. Luukkonen, *Modelling and control of quadcopter,* , 26.

[33] L. Sikkel, G. Croon, C. De Wagter, and Q. Chu, *A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost MEMS IMUs,* (2016) pp. 2141–2146.

[34] ArduPilot, *Wind Estimation in multicopters with Dr Paul Riseborough,* (2021).

[35] M. Simma, H. Mjøen, and T. Boström, *Measuring Wind Speed Using the Internal Stabilization System of a Quadrotor Drone,* Drones **4**, 23 (2020).

[36] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, *Autonomous Gas-Sensitive Microdrone: Wind Vector Estimation and Gas Distribution Mapping,* IEEE Robotics Automation Magazine **19**, 50 (2012), conference Name: IEEE Robotics Automation Magazine.

[37] P. Abichandani, D. Lobo, G. Ford, D. Bucci, and M. Kam, *Wind Measurement and Simulation Techniques in Multi-Rotor Small Unmanned Aerial Vehicles,* IEEE Access **8**, 54910 (2020), conference Name: IEEE Access.

[38] X. Xiang, Z. Wang, Z. Mo, G. Chen, K. Pham, and E. Blasch, *Paper: Wind field estimation through autonomous quadcopter avionics,* in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)* (2016) pp. 1–6, iSSN: 2155-7209.

[39] J. González-Rocha, C. A. Woolsey, C. Sultan, and S. F. J. De Wekker, *Sensing Wind from Quadrotor Motion,* Journal of Guidance, Control, and Dynamics **42**, 836 (2019), publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G003542.

[40] H. Bouadi and M. Tadjine, *Nonlinear observer design and sliding mode control of four rotor helicopter,* Int Journal of Eng. and Applied Science **3** (2007).

[41] N. Y. Ko, I. Choi, G. Song, and W. Youn, *Three-Dimensional Dynamic-Model-Aided Navigation of Multirotor Unmanned Aerial Vehicles,* IEEE Access **PP**, 1 (2019).

[42] J. Sun, H. A. Blom, J. Ellerbroek, and J. M. Hoekstra, *Particle filter for aircraft mass estimation and uncertainty modeling,* Transportation Research Part C: Emerging Technologies **105**, 145 (2019).

[43] A. Teimourian and D. Firouzbakht, *A PRACTICAL METHOD FOR DETERMINATION OF THE MOMENTS OF INERTIA OF UNMANNED AERIAL VEHICLES,* (2013).

[44] A. Mendes, E. van Kampen, B. Remes, and Q. P. Chu, *Determining moments of inertia of small UAVs: A comparative analysis of an experimental method versus theoretical approaches,* in *AIAA Guidance, Navigation, and Control Conference* (American Institute of Aeronautics and Astronautics, Minneapolis, Minnesota, 2012).

[45] J. D. Anderson, *Fundamentals of aerodynamics* (McGraw-Hill, 2011).

[46] R. Felismina, M. Silva, A. Mateus, and C. Malça, *Study on the aerodynamic behavior of a UAV with an applied seeder for agricultural practices,* (Rome, Italy, 2017) p. 020049.

[47] *Understanding the physics of drones - Building drones around Beagle Bone Blue,* (2021).

[48] M. Claybrough and F. Defaÿ, *Stabilization of an unmanned aerial vehicle using real-time embedded motion estimation,* (2012) pp. 2189–2194.

[49] H. W. Ho, G. C. de Croon, and Q. Chu, *Distance and velocity estimation using optical flow from a monocular camera,* International Journal of Micro Air Vehicles **9**, 198 (2017), publisher: SAGE Publications Ltd STM.

[50] Q. T. Phan, *Applied Sciences | Free Full-Text | A Hybrid Wind Power Forecasting Model with XGBoost, Data Preprocessing Considering Different NWPs | HTML,* .

[51] Crazyfly, *PWM to Thrust | Bitcraze,* .

[52] *BlueSky - The Open Air Traffic Simulator,* (2022), original-date: 2015-04-01T17:13:37Z.