Politecnico
di Torino

Corso di Laurea Magistrale
in Ingegneria Aerospaziale

A.A. 2021/2022
Sessione di Laurea aprile 2022

Multi-rotor UAS for precision crop-spraying: developing an adaptive guidance algorithm to minimize spray drift in a wind environment

Relatore
Giorgio Guglieri
Correlatori
Ncoletta Bloise
Manuel Carreño Ruiz

Candidato
Luca Scagnellato
matr. 275710

Contents

Abstract

In recent years, modern precision agriculture techniques led to a rise in the use of Unmanned Aircraft Systems (UASs) for many different purposes, such as land survey, plant health monitoring and crop spraying. These vehicles are particularly suited for the use over rough and mountainous terrains or dispersed plots of land, that traditional land vehicles may find difficult to reach or too time consuming. UASs can also benefit the farmers as they can be, and often are, automated in their operation, further increasing productivity while reducing the exposure to potentially hazardous chemicals. Politecnico di Torino, in collaboration with SEASTAR Wind Tunnel (in the Environment Park of Turin), started the research on ”New technical and operative solutions for the use of drones in Agriculture 4.0” through a grant linked to the PRIN 2017 (*Progetti di ricerca di Rilevanza Nazionale*, Research projects of national interest), a program of the Italian Ministry of Education, University and Research. This led to the preliminary design of a drone for crop spraying on vineyards, as this cultivation is characterised by uneven, sloped terrain, it presents the ideal scenario to evaluate the precision of Plant Protection Products (PPP) deposition and is widely cultivated throughout Italy.

The thesis presented is the continuation of the previous work. The aim was to develop a guidance and control algorithm for a multi-copter that would be able to minimize PPP particles drift, taking into account the wind conditions. In order to do so, a particle drift model was developed as well, based on an experimental campaign in a wind tunnel.

Chapter 1 introduces the work, reviewing the state of the art for this application. In Chapter 2, the methodology for the development of the drift model is presented. Then, in chapter 3 the multi-rotor platform, its working principles and dynamics are discussed. The guidance and control system and its integration with the drift model can be found in chapters 4 and 5. All results are then presented and discussed in chapter 6, while chapter 7 draws the conclusions and explores possible future developments.

Chapter 1

Introduction

In recent years, modern precision agriculture techniques led to a rise in the use of Unmanned Aircraft Systems (UASs) for many different purposes, such as land survey, plant health monitoring and crop spraying (1)(2)(3). This is because the technology is mature enough to allow for affordable and ready-to-use equipment, which can be used in conjunction with conventional machines to operate in a wide array of different crops. This vehicles bring many advantages to the farmers, as they can automate or optimize more and more farm operations. For example, the plant health assessment used to be done in person by an expert, an agronomist or an experienced farmer himself, who had to walk in the field under observation and look for clues of pests, diseases and overall plant strength. In this way, the assessment can be done only on a small sample of the total number of plants, leaving a margin of uncertainty over the reliability of the method. With the introduction of small-sized aerial vehicles, this operation can be conducted on the whole field, using multi-spectral sensors that can analyze the emission spectrum of a plant and automatically detect any anomaly or need for a specific treatment. This operation itself evolved through the years, and what was once done by a trained professional UAS pilot is now almost fully automated. This is thanks to the evolution of the sensors suite that can be fitted to a vehicle. The most simple multi-rotors may have only brackets for an external camera to be fixed upon, that has to be manually mounted and taken down for the image analysis and post-processing, but the most advanced, state-of-the-art commercially available products reached a technological level that was unthinkable just a few years ago. As an example DJI, one of the leaders in the industry, has an specific agricultural lineup of UASs that can be equipped with a six sensors camera for image acquisition, omni-directional radars for obstacle avoidance, a LIDAR for terrain mapping and a Real Time Kinematic (RTK) system for GPS positioning enhancement.

These sensor suite, together with an "intelligent" guidance system, allows the UAS to conduct *autonomous* operations and the authorities provide the necessary regulations to do so safely. In 2019 EASA (European Union Aviation Safety Agency), which is the central regulatory agency in Europe to which all national agencies refer to, set the standards for unmanned vehicles operations with the Commission Implementing Regulation (EU) 2019/947 (4) and the Commission Delegated Regulation (EU) 2019/945 (5), together with the Acceptable Means of Compliance (AMC) and Guidance material (GM). These are applicable from the 30^{th} December 2020, and provide all the information needed to certificate and operate these kind of vehicles across all the European Union.

Autonomous vehicles, that are able to fully operate without the pilot's intervention, may fall into the *specific* or *certified* category, depending on the operative scenario. For agricultural purposes, the vehicle will fly far from populated areas so, apart from specific cases, it could be classified into the *specific* category that doesn't require any National Aviation Authority (NAA) authorization, assuming that the vehicles will operate in one of the Standard Scenarios (STS) and is labeled with class identification label C5 or C6 (which apply to vehicles with a mass $> 25kg$). If this is not the case, the NAA requires a *risk assessment of intended operation* conducted using the *SORA* methodology, described in the AMC, identifying means to mitigate the risk and comply with the operational safety objectives, opening the market to all kinds of aerial operations.

The technological evolution of electronics, and in particular of the lithium-polymer batteries with their very high energy density, also allows the increase of motor power and overall autonomy, which in turn introduced the possibility to manufacture larger and heavier multi-rotors that can have a satisfactory payload capacity to perform crop-spraying operations that were once the exclusively done by land vehicles. As an example, the largest UAS that DJI Agriculture offers, the *Agras T30* has a maximum take-off weight of $76.5kg$ with a useful payload of $30kg$ and can cover 16 hectares an hour. This is thanks to a $10kg$, $29000mAh$ battery that can provide up to $11kW$ of continuous power. However, most of these vehicles are engineered to operate over large targets, such as fruit trees, and lack the precision needed for smaller and narrow plants treatment, like vines.

It is for this reason that this thesis aims to develop a system that can achieve a higher level of precision crop-spraying; the project was brought forward with a vineyard as the target crop, as the vines are usually arranged in long and narrow rows and where the use of commercially available UASs would not be possible in an efficient manner. Moreover, this kind of culture is often grown on sloped and harsh terrain in scattered plots of land, making an aerial vehicle the ideal candidate for the task, as was previously done in (6). There are many pests and plant diseases that negatively affect the vine (7): these include molds such as *Plasmopara viticola*, *Erysiphe necator*, *Botrytis cinerea*, fungi (generally described as *Grapevine trunk diseases, GTDs*) and insects that directly affect the plant by eating it or are vectors for diseases, like *Planococcus ficus*, *Empoasca vitis* and *Scaphoideus titanus ball.* All these has to be monitored and specific products have to be used for each one of them.

In order to increase the spray precision and minimize the particle drift, it's important to be able to characterize and predict the particles behavior when subject to the rotors' downwash (8) and the atmospheric wind (9), which is why a spray drift model was developed before even beginning the design of the guidance and control algorithm. This phase is described in chapter 2. Then, in chapters 3 and 4, the multi-rotor platform and its control system architecture and operation are reported. The particle drift model can be useful only if the operating conditions are known: that's why in chapter 5 a wind estimation algorithm is conceived, in order to integrate it into the guidance system and achieve the sought after precision. Finally, chapters 6 and 7 present the simulations results, draw the conclusions and discuss possible future developments of the project.

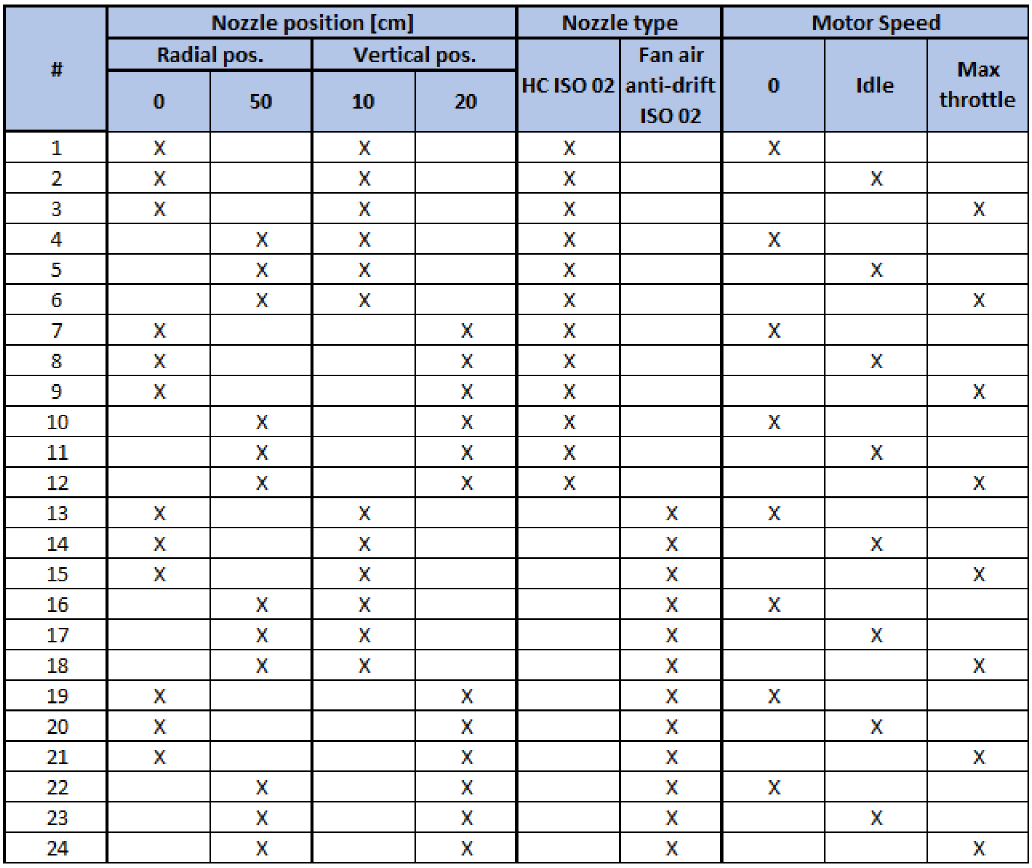
Chapter 2

Drift model

The creation of a drift model is crucial for the following sections, as it is the mathematical base that drives the guidance algorithm. Other experimental work based its research on smoke flow visualization techniques (10), used a numerical approach (11), (12) or used a direct measurement of droplets deposition (9). Here, a different approach was used: the flow is illuminated by a laser light and pictures are taken and later analysed to extrapolate the necessary information

2.1 Dataset

The necessary pictures for the model development were taken in the SEASTAR (Sustainable Energy Applied Sciences, Technology and Advanced Research) wind tunnel in the Environment Park in Turin, Italy. The complete experiment setup and description can be found in (13), and the wind tunnel and the vehicle used for the tests are shown in figure 2.2. Lateral pictures of the spray cone were taken, by changing the wind speed, the throttle, the nozzle and its position. Two different types of nozzles were used: a Fan Air Ceramic ISO 02 80° anti-drift nozzle and a Hollowcone Ceramic ISO 02 80°, made by ARAG (14). Each nozzle was studied in four different positions under the rotor, with three different wind speeds and at three different throttle positions, as explained in figure 2.1. The *radial position* value indicates the position where the nozzle is placed with respect to the rotor center, expressed as the rotor's radius percentage. The *vertical position* value indicates the nozzle distance from the rotor plane, in *cm*.



#	Nozzle position [cm]				Nozzle type		Motor Speed		
	Radial pos.		Vertical pos.		HC ISO 02	Fan air anti-drift ISO 02	0	Idle	Max throttle
	0	50	10	20					
1	X		X		X		X		
2	X		X		X			X	
3	X		X		X				X
4		X	X		X		X		
5		X	X		X			X	
6		X	X		X				X
7	X			X	X		X		
8	X			X	X			X	
9	X			X	X				X
10		X		X	X		X		
11		X		X	X			X	
12		X		X	X				X
13	X		X			X	X		
14	X		X			X		X	
15	X		X			X			X
16		X	X			X	X		
17		X	X			X		X	
18		X	X			X			X
19	X			X		X	X		
20	X			X		X		X	
21	X			X		X			X
22		X		X		X	X		
23		X		X		X		X	
24		X		X		X			X

Figure 2.1: Test matrix

As these tests are repeated for wind speeds of $0, 2$ and $3m/s$, the total number of pictures taken is 72

In addition to that, eight more pictures were taken from a frontal point of view so to be able to characterize the spray cone in three dimensions and not limited to the lateral plane.

2.2 Image analysis

With the data from the wind tunnel tests, an empirical model of the spray swath can be created, based on the pictures taken. This is done in order to predict the particles trajectory and dispersion, so that the drone's guidance and control algorithm can take this into account in order to spray the vines more precisely and exploit the drift due to the wind to target the plants. All the work was conducted in the MATLAB environment.

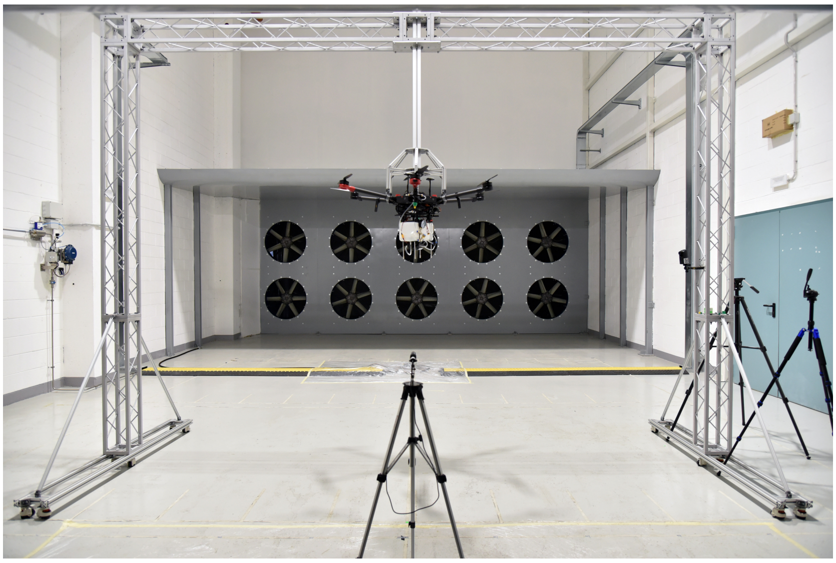


Figure 2.2: SEASTAR Wind Tunnel and the vehicle used during the tests

2.2.1 Spray cone trajectory

As the number of tests is relatively small, the analysis could be done manually and the pictures could be studied one by one to extrapolate the spray trajectory. Instead, this process was automated so to have consistent data that is not influenced by a subjective analysis and that can be recreated or expanded in the future through some more tests or CFD analysis. The following image well explains the workflow of this part of the work.

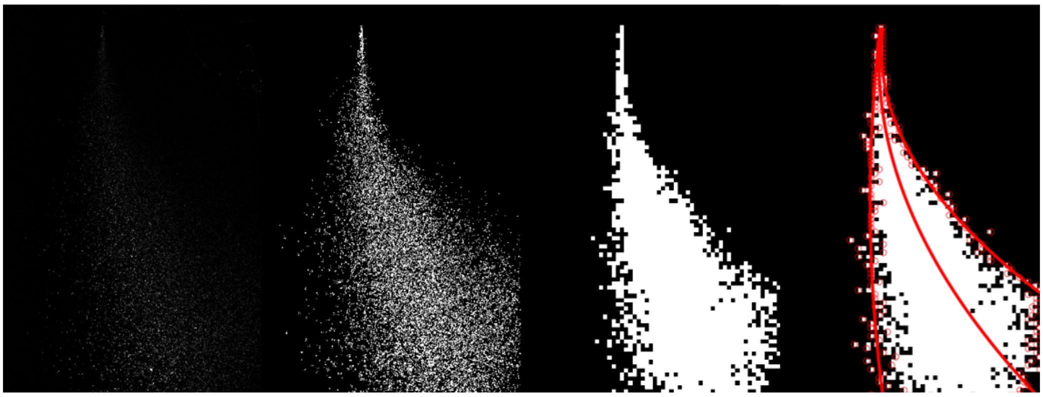


Figure 2.3: Image processing workflow.

The RGB photo (which is the one taken during the tests) must be processed. First, it is converted into a grayscale image so to be able to obtain a binary image by applying an intensity threshold (the left-most image in figures 2.4 through 2.21). This threshold is low as the light reflected from the droplets is very faint. Then, the binary image is *block-processed* so a more solid image can be used for the spray trajectory definition. With this technique the image is divided into small squares and the number of white pixels (which corresponds to droplets) is counted; if the percentage of white pixels is greater than a given threshold, the square in question is considered to be part of the spray cone. Otherwise, it is discarded and completely colored in black (the central image in figures 2.4 through 2.21). This technique allows to eliminate stray droplets that could interfere with the subsequent analysis. In this phase all the points needed for the trajectory fitting are defined: the *block-processed* photo is divided in a number of horizontal lines, and the right- and leftmost white pixel of each line are memorized as the limits of the spray cone.

Next, a least-squares curve fitting is applied to the points found to approximate the spray shape (the right-most image in figures 2.4 through 2.21). Once these curves are defined, the centerline is located in the middle of those two. A second degree polynomial is used for the fitting. The polynomial is defined using the vertical axis as ordinates, so when evaluating it the distance from the nozzle can be used as input and the output is the spray swath. It is important to include in the analysis a way to recognise where in the photo the spray region is outside of the picture itself. Here, a routine is implemented and it counts if two or more consecutive data points are located on the photo's border; if that's the case, all the following are discarded from the curve fitting.

Once all coefficients are defined, they are transformed to take the nozzle position as the origin and starting point for the spray. All curves can then be described in the form:

$$x = ay^2 + by \tag{2.1}$$

Where x is the horizontal displacement from the nozzle and y is the vertical distance from it. These coefficients are used in the following section.

As can be seen from figure 2.3, the raw, non-processed image is quite dark and the droplets are almost imperceptible. That's why in the following images, that present the results, it is not included.

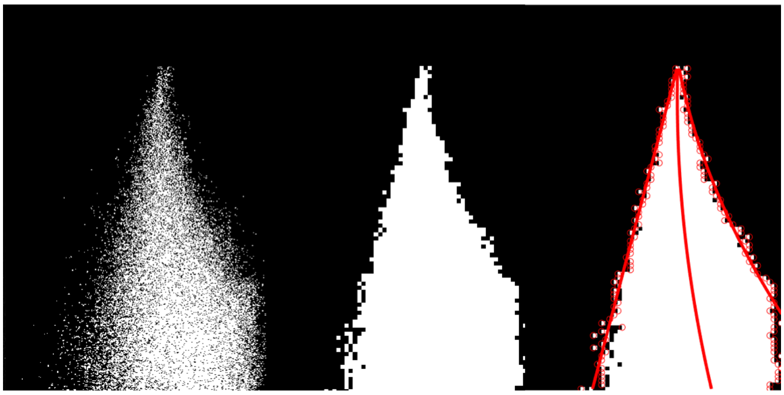


Figure 2.4: FAN nozzle, $V = 0m/s$, $rpm = 0$

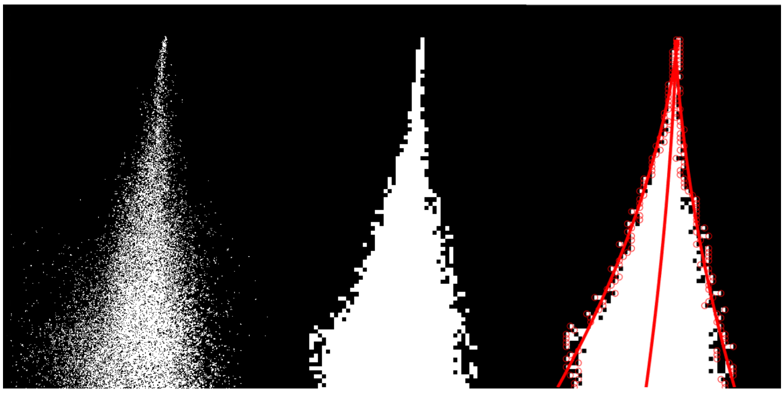


Figure 2.5: FAN nozzle, $V = 0m/s$, $rpm = 1590$

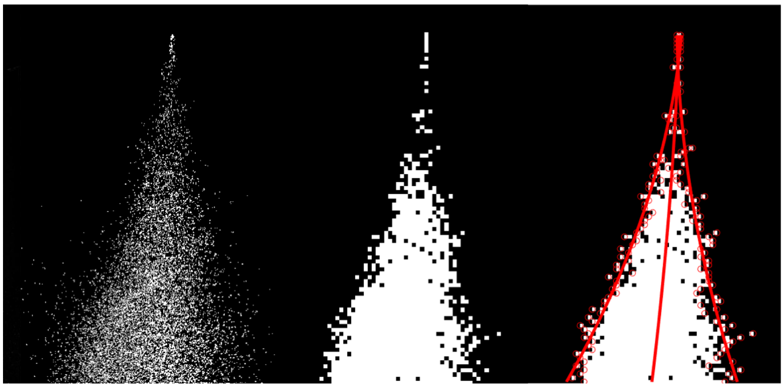


Figure 2.6: FAN nozzle, $V = 0m/s$, $rpm = 5100$

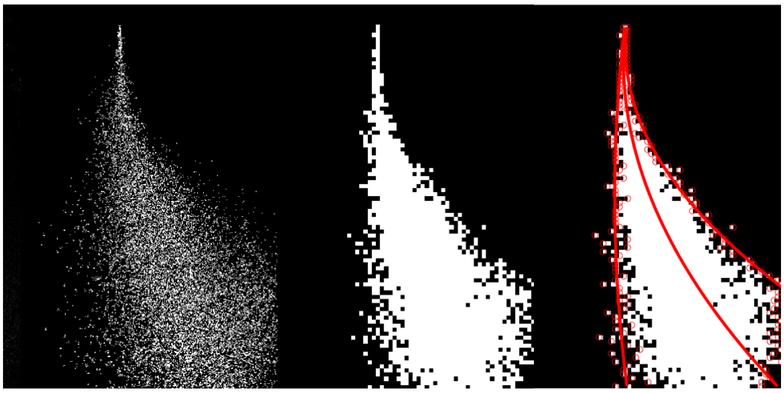


Figure 2.7: FAN nozzle, $V = 2m/s$, $rpm = 0$

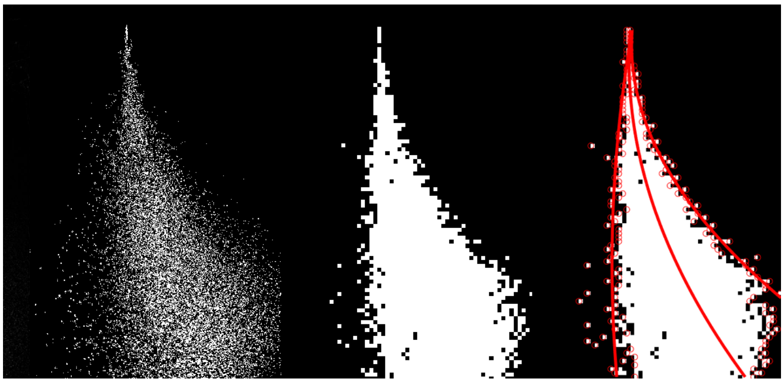


Figure 2.8: FAN nozzle, $V = 2m/s$, $rpm = 1590$

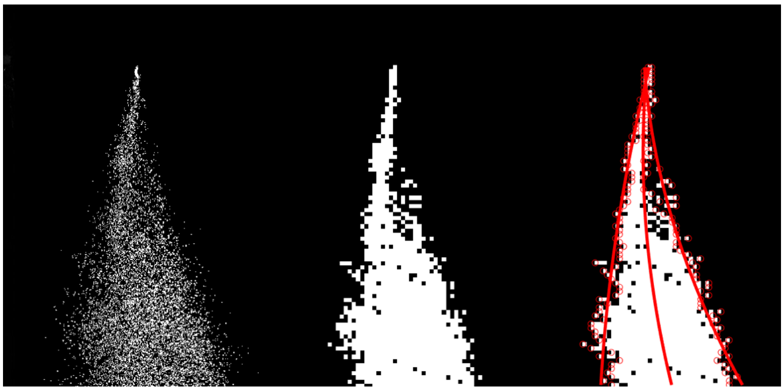


Figure 2.9: FAN nozzle, $V = 2m/s$, $rpm = 5100$

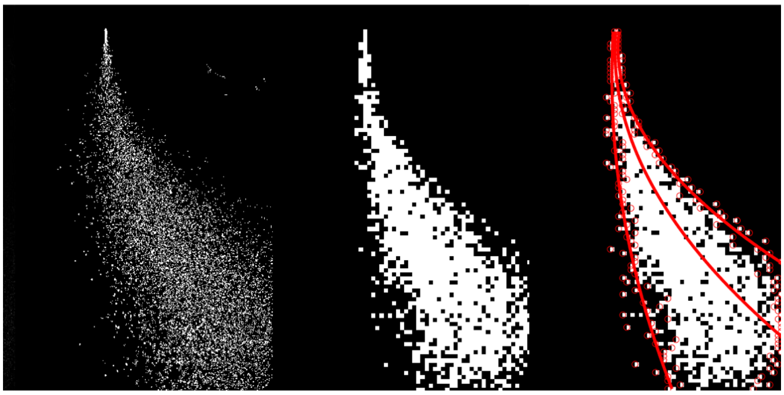


Figure 2.10: FAN nozzle, $V = 3m/s$, $rpm = 0$

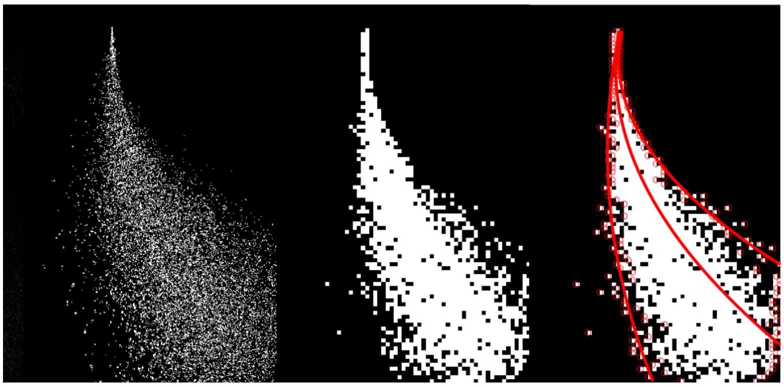


Figure 2.11: FAN nozzle, $V = 3m/s$, $rpm = 1590$

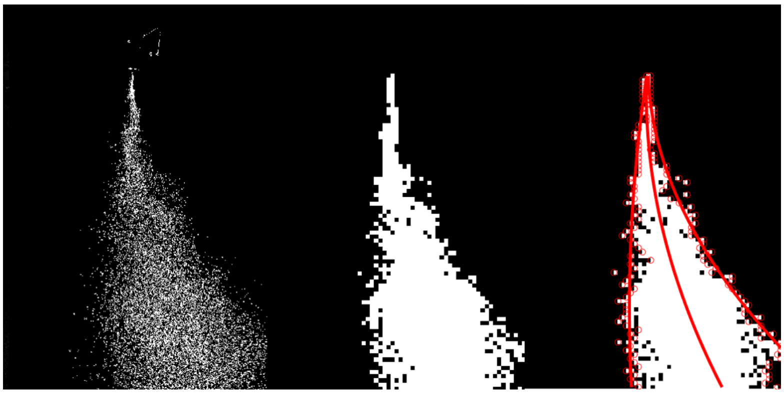


Figure 2.12: FAN nozzle, $V = 3m/s$, $rpm = 5100$

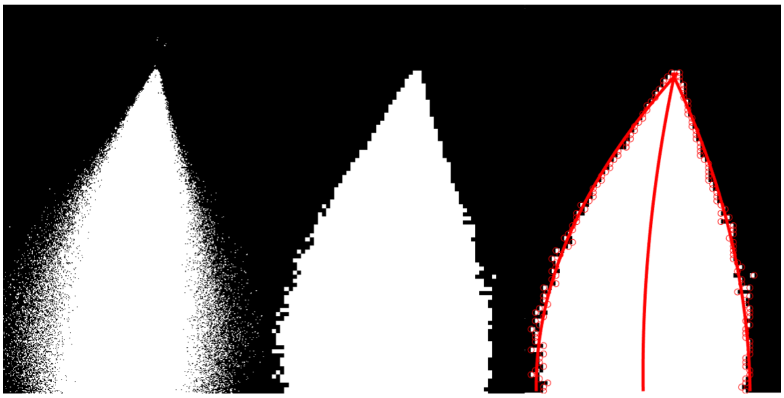


Figure 2.13: HC nozzle, $V = 0m/s$, $rpm = 0$

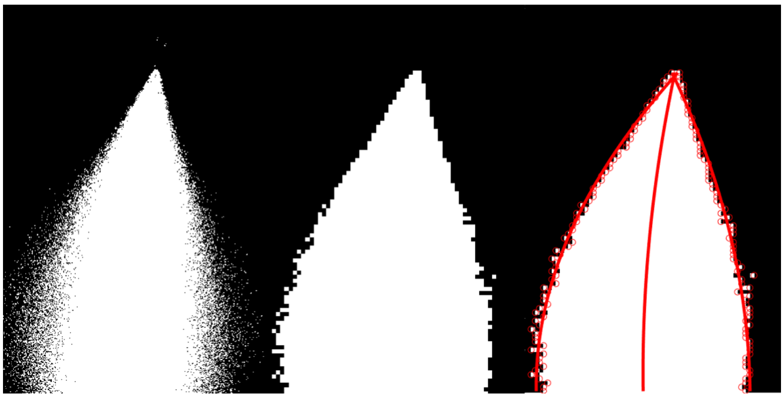


Figure 2.14: HC nozzle, $V = 0m/s$, $rpm = 1590$

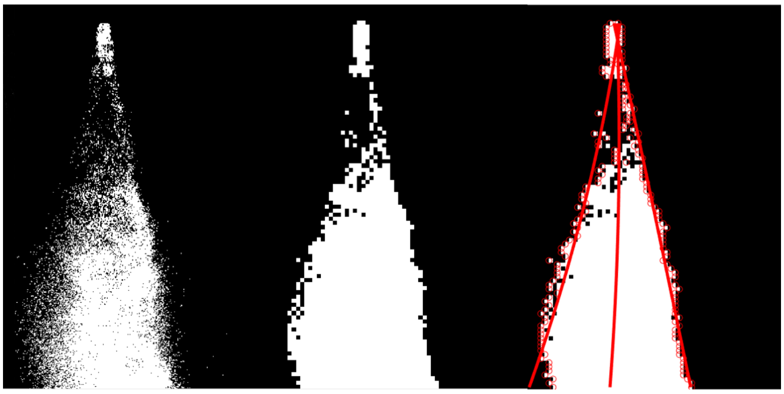


Figure 2.15: HC nozzle, $V = 0m/s$, $rpm = 5100$

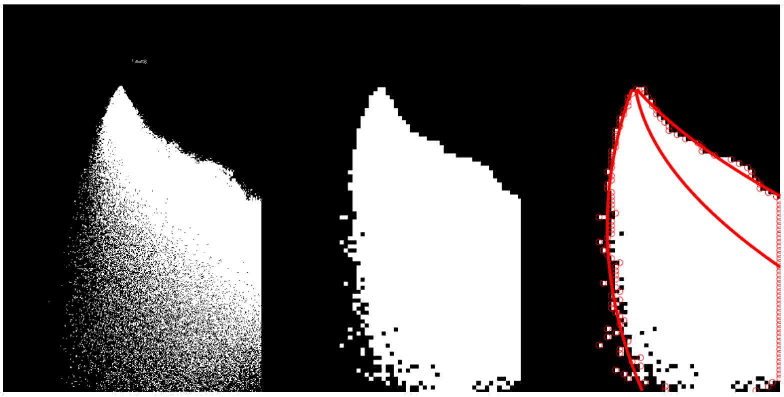


Figure 2.16: HC nozzle, $V = 2m/s$, $rpm = 0$

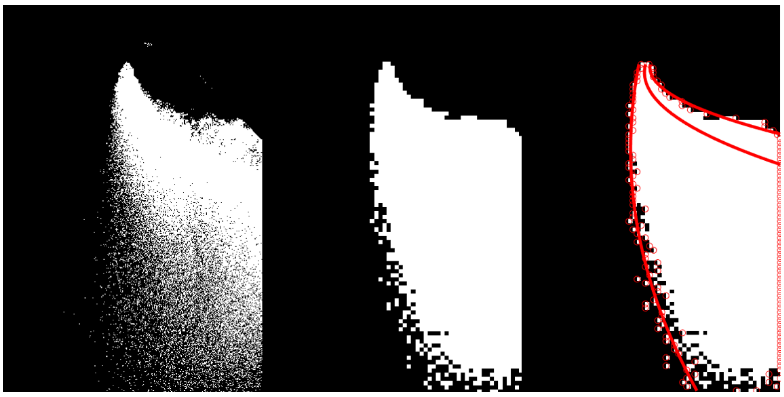


Figure 2.17: HC nozzle, $V = 2m/s$, $rpm = 1590$

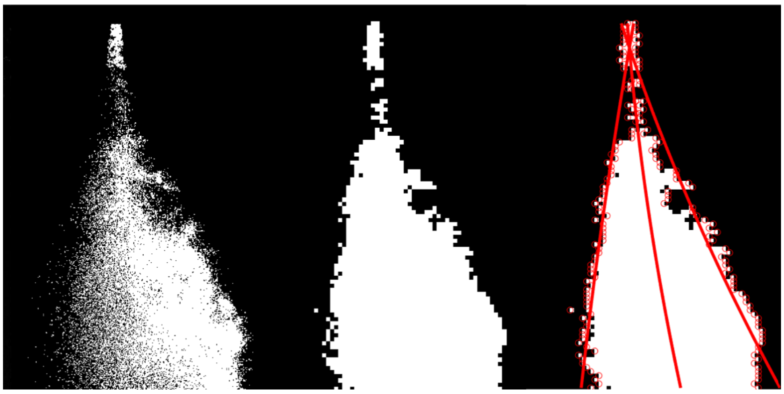


Figure 2.18: HC nozzle, $V = 2m/s$, $rpm = 5100$

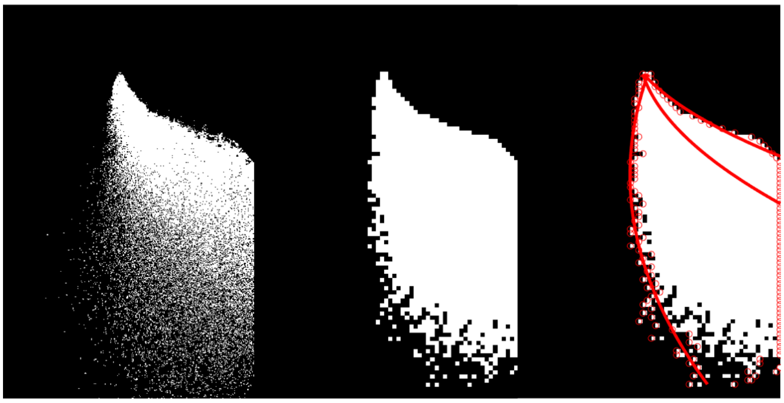


Figure 2.19: HC nozzle, $V = 3m/s$, $rpm = 0$

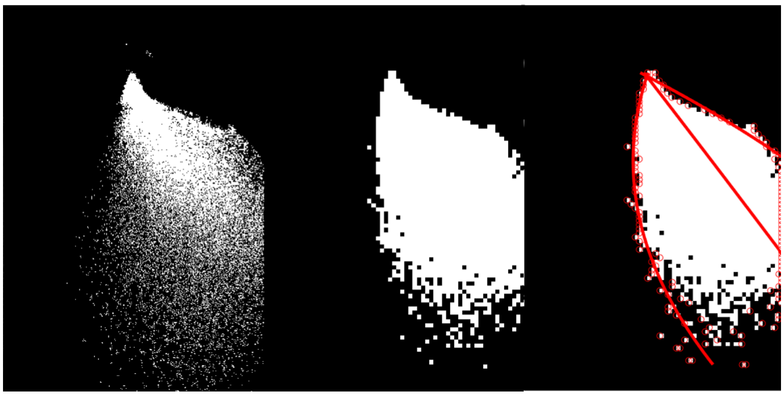


Figure 2.20: HC nozzle, $V = 3m/s$, $rpm = 1590$

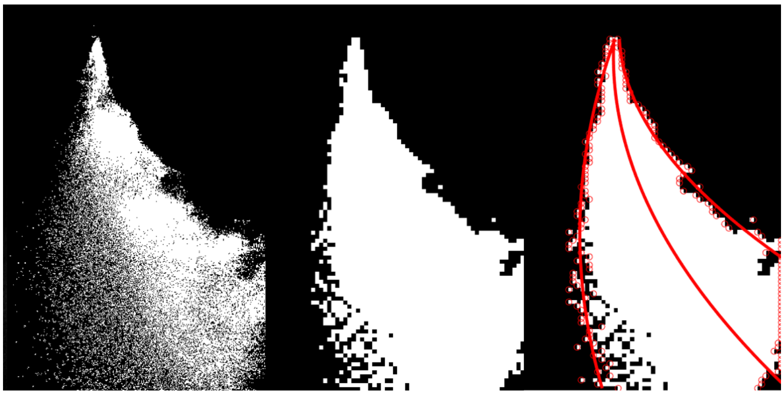


Figure 2.21: HC nozzle, $V = 3m/s$, $rpm = 5100$

As can be seen from the comparison between the images above, it is evident how the *hollowcone* nozzle produces much finer, and so lighter, droplets, that are subject to the wind drag much more than the ones produced by a *FAN* nozzle. In fact, in the images pertaining the *HC* nozzle, the spray trajectory presents more variability when the wind speed and *rpm* increase. In order to make the drift more predictable, it is then advised to use a *FAN* nozzle for this application, and the model derived from it will be used in all simulations.

Another consideration that can be made from the observation of these results is the high impact of the rotor's downwash on the spray drift. In fact, as the *rpm* rises, the atmospheric wind magnitude decreases with respect to the wake velocity, and so the spray cone isn't deformed as much and the particles are deposited on the ground closer. This is particularly evident if, for example, figure 2.10 and 2.12 are compared. When the rotors are shut off, the drift is very apparent and the majority of the particles fall outside of the image's bounds. Conversely, when the rotor reaches its maximum speed the droplets' trajectory is almost vertical and the drift is minimal. Of course this is an intuitive principle and that's why the nozzle was placed under the rotor in the first place. It is however important to remember this as the UAS' mass decreases during the mission, and so does the rotors' velocity as they need to provide less thrust. This means that as the mission progresses, the spray drift increases and the overall spray system's precision is degraded. This was also the result found in (8), where a statystical approach was used to find a correlation between the airspeed along the vertical direction and the droplet distribution and penetration characteristics. It highlights the aforementioned facts that, in order to obtain a more precise and predictable spray trajectory, it is necessary to maximize the rotors' downwash and keep the vehicle at an optimum vertical distance from the to-be-treated plant. In fact, flying too low and close to the plant can introduce updraft due to the wake interaction with the ground (15) (16), but as the height rises the particles quickly disperse so a balance must be found. As the branches and leaves interact with the wake, this balance is different for each nozzle/plant combinations and so has to be found empirically.

2.3 Spray model

The objective of the spray model creation is to obtain an estimation of what the ground footprint of the spray cone is, under different wind and rotor velocity conditions. In order to do so, the spray cone trajectories gathered in the previous section have to be correlated to the conditions in which the photos were taken. Of the 72 pictures taken, half are pertinent to the FAN nozzle and the other half to the HC nozzle. Each nozzle is placed in four different positions (as explained in the previous section), so that there are nine different pictures for each nozzle position. Furthermore, 8 frontal pictures were taken, and they will help to estimate the three-dimensional shape of the spray cone. A correlation between these pictures must be found in order to have a model that can evaluate the swath width and position as a function of:

- distance from the ground;
- relative wind speed;
- rotors RPM;

for each nozzle and for each of the four nozzle positions. As the test matrix shows, for each throttle position (still rotor, idle and maximum throttle), there are three wind speed values (0, 2 and 3 m/s). First, a linear regression on the coefficients pertaining each throttle position gives three different linear relationships that describe the spray cone as a function of wind speed, all in the form:

$$x = (av + b)y^2 + (cv + d)y \tag{2.2}$$

Where x is the horizontal displacement from the nozzle, y is the vertical distance from it and v is the wind speed.

A second regression is operated on these coefficients in order to obtain the final formula that can be easily evaluated to investigate the spray cone shape:

$$x = ((aRPM+b)v+(cRPM+d))y^2+((eRPM+f)v+(gRPM+h))y \tag{2.3}$$

The same analysis can be conducted for the frontal images, but the variables are only RPM and circuit pressure. By joining together the two models, a 3-D visualization of the spray can be generated for a given height from the ground, wind speed, RPM and circuit pressure.

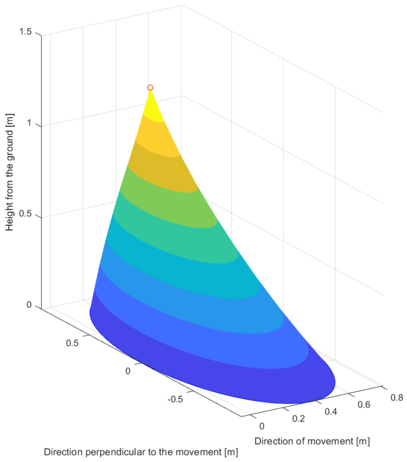


Figure 2.22: Spray visualization in 3-D. Windspeed=3 m/s, RPM=5000 (FAN nozzle)

2.4 Validation of spray model

In order to verify the mathematical model presented above, the results it gives must be compared to the original photos taken during the wind tunnel tests. It is expected not to be totally accurate, as for each nozzle position only nine data points are available. To increase accuracy, more tests should be performed; alternatively, CFD simulations could be useful to stretch the model limits. However, as shown during the field tests and in (11) (12) (9) (8), there are many factors that have an impact on the flow (such as the wind disruption given by plants, the drone pitch and roll angles, the real and not the wind-tunnel-controlled atmosphere,...), so increasing the model's theoretical precision would not yield the same precision in a real-world test.

2.4.1 Comparison between the model results and the original images

The model is used to produce a virtual spray cone shape, that is then compared to the processed image seen in the section before.

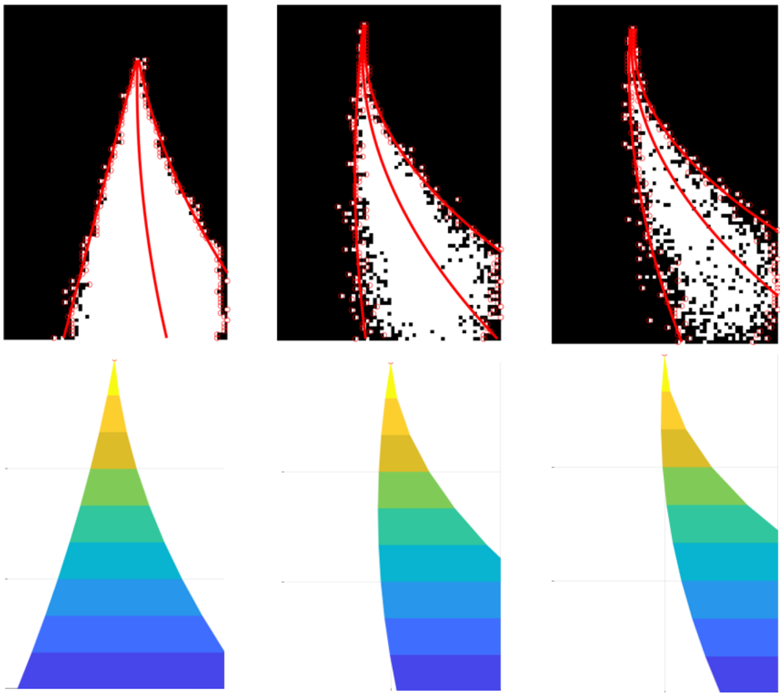


Figure 2.23: *rpm = 0*, increasing wind speed $(0, 2 \text{ and } 3m/s)$

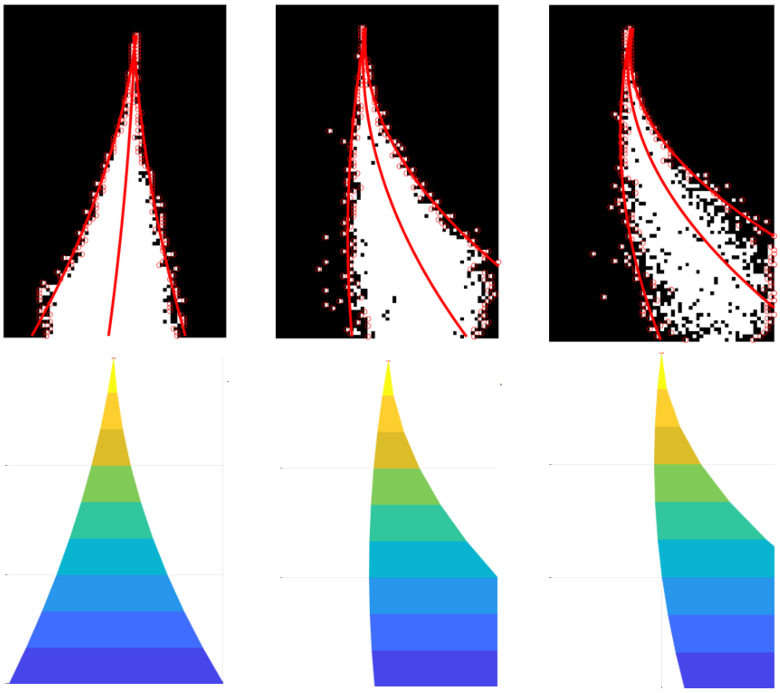


Figure 2.24: *rpm = 1590*, increasing wind speed $(0, 2 \text{ and } 3m/s)$

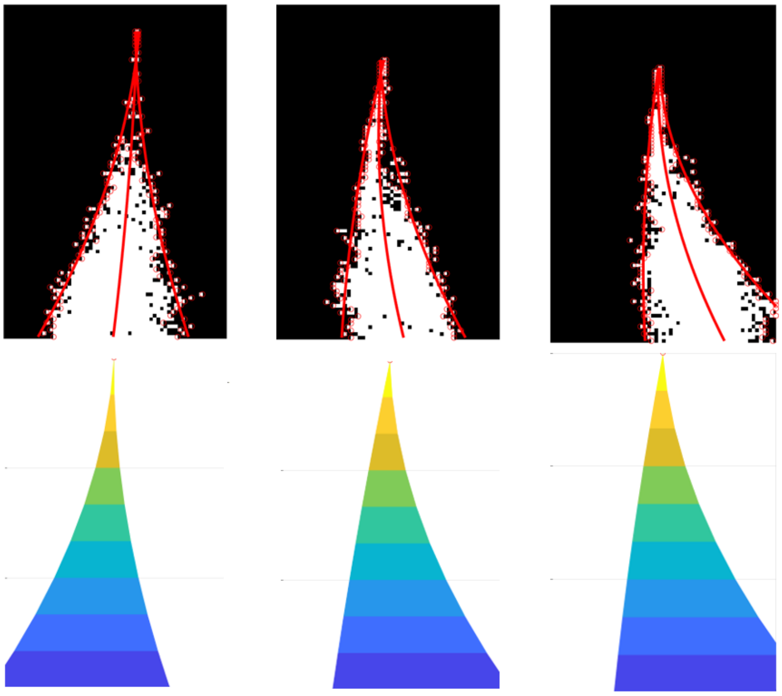


Figure 2.25: *rpm = 5100*, increasing wind speed $(0, 2$ and $3m/s)$

As figures 2.23 2.24 and 2.25 show, the model accurately reproduces what the original photos captured. However, uncertainty is still present for values of wind speed and *rpm* different than the ones depicted, as there is no experimental data to compare the model to.

2.4.2 Correlation between drift and particles momentum

The model can be compared to some simple simulations done in the MATLAB environment. This tool allows for the analysis of the particles' trajectory beyond the image limits and some considerations can be done. In order to launch the simulation, some assumptions on the particles dimensions and the environment are made, based on the work done in (17). In this case, a constant $3m/s$ wind is set in the direction specified in figure 2.27, while the particles are spread in the spray cone with an illustrative normal distribution (mean $\mu = 0$ and variance $\sigma = 0.234$), all with an initial velocity of $15m/s$. The droplets size also follows a representative normal distribution with $\mu = 300\mu m$ and $\sigma = 6 * 10^{-5}$, as depicted in figure 2.26, and they are considered to be spherical. In this environment, with low Reynolds number

($Re \approx 100$), the coefficient of drag can be approximated (18) as

$$C_D = Re/24$$

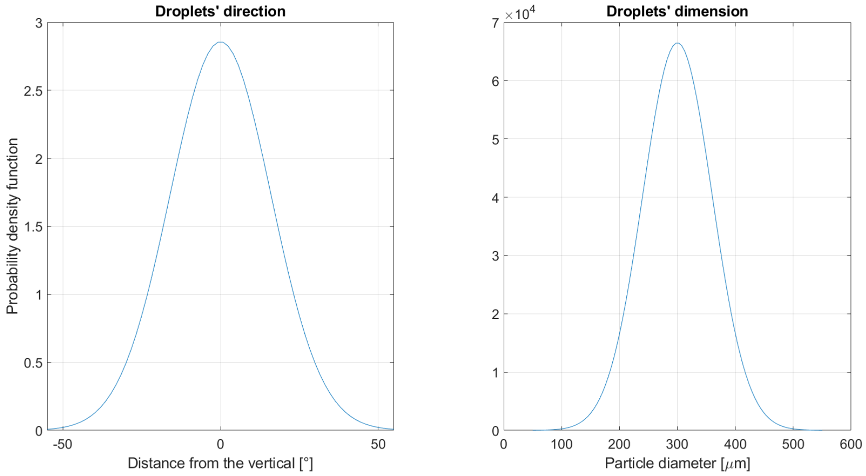


Figure 2.26: Droplets' direction and size distribution

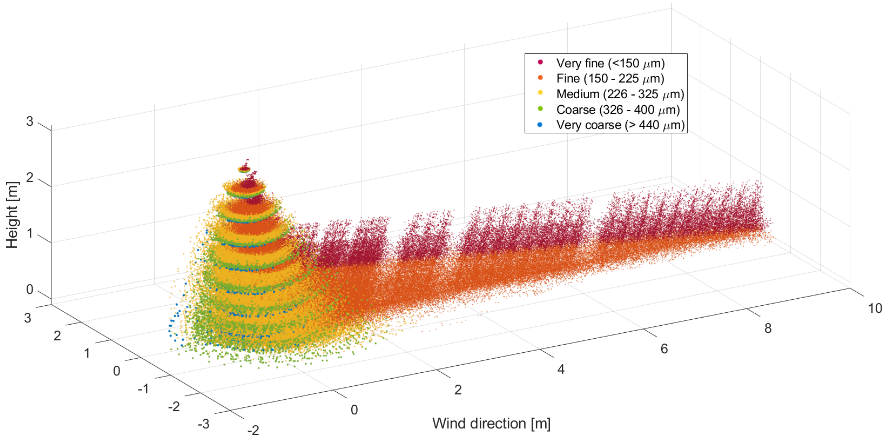


Figure 2.27: Particle trajectory simulation

Figure 2.27 follows the trajectory of some droplets and depicts their position in consecutive instants. As can be noted, the heavier particles (*very coarse, coarse and medium*, in blue, green and yellow respectively) follow the path captured by the photographic spray model and the spray cone has the same qualitative shape. However, the lighter particles are carried by the wind much farther and deposit on the ground very far from the nozzle. This behaviour couldn't be captured by the photos as the camera field of view is much narrower than distances involved, but has to be expected. The wind effect on *very fine* and *fine* drops is orders of magnitude greater than on the heavier ones, and this is due not only to their dimension but also on their mass. The volume of a sphere, and so its mass, increases with the cube of the diameter, while the drag is a linear function of it:

$$D = \frac{1}{2}\rho S C_D V^2 = \frac{1}{2}\rho V^2 \pi \frac{d^2}{4} \frac{24\mu}{\rho V d} = 3V\pi\mu d \tag{2.4}$$

Where μ is the dynamic viscosity of the air and d is the droplet diameter.

This means that comparing the smallest particle ($d = 140\mu m$) and the largest ($d = 400\mu m$), in a $3m/s$ wind stream the drag to weight ratio is respectively 16 and 2. This translates in the smaller particle being easily carried by the wind, while the larger one falls to the ground almost immediately.

Such drift, however, doesn't invalidate the precision of the photographic model discussed before, as the majority of the drops (in terms of volume or weight) follows the trajectory that can be observed through the photos. Instead, it highlights two correlated facts:

- in order to achieve better spraying precision and reduce drift, it is better to use a nozzle that produced coarser particles, as they are the least subject to the wind effects. This was seen while analyzing the photos taken in the wind tunnel as well, but this confirms that hypothesis (as is also suggested in (19), (20));

- it is very difficult to completely eliminate drifting particles that will deposit in off-target areas and that could contaminate the soil. The particles' dimension can increase only so much in order to be effectively sprayed and not poured onto the plants, and so finer drops are a necessary by-product of the nebulisation process that happens in the nozzle. However, as discussed before, these stray droplets are only a small portion of the total PPP sprayed and can be neglected while planning the spraying operations.

2.5 Future improvements

Even if this model is a good reference to characterize the spray particles in a wind environment, it's still limited in its scope. In fact, it is able to describe the trajectory only in the closest proximity of the nozzle, as the reference images only frame an area of about 2 × 1.5m and so no information of the spray trail can be obtained. In order to better describe the phenomenon, wider images should be taken, but this presents its own challenges. In fact, to capture such a large area the camera should be position farther away, which may not be possible in a small wind tunnel and would significantly reduce the droplets definition. Another option would be to use an ultra-wide lens, but it would distort the peripheral parts of the image, which should then be further post processed in order to restore the real proportions. The third option would be to use several different cameras positioned along the wind stream. This solution would increase the overall experimental campaign cost and would again need a post-processing phase just to correctly merge the different pictures taken into one. The reliance on pictures for the model characterization main limit, though, is the frontal visualization of the spray cone. In fact, the laser beam used to visualize the droplets was fitted straight underneath the nozzle; this allows for a good highlighting of the cone aperture in the half-meter close to the nozzle. However, as the particles are pushed away from the wind, they cannot be visualized anymore in the bottom part of the image, and the result is a distorted cone shape. For this reason, only the top half of the frontal photos were used for the interpolation.

Another model shortcoming is the lack of particles size and distribution in the jet. The nozzle manufacturer provides and estimate for the droplets' size range, but the pictures can't give any information on the distribution along the stream. Of course, it is expected that the wind has the least effects on bigger and heavier particles and that smaller and lighter ones are carried away farther by the wind, but only more advanced techniques could give a numeric description of the phenomenon. For this reason, once the spray ground footprint is calculated, some strong assumptions on the spatial distribution of particles have to be made.

Furthermore, this work doesn't take into account the vehicle's tilt angle, as all pictures were taken while the UAS was horizontal with respect to the ground. This shouldn't introduce large errors when compared to the reality as while the quad rotor is operating it won't tilt more than a few degrees, but it still is one of the model's limitations.

For all these reasons, it is recommended that this picture-based model is used only to have an approximate estimate of the spray trajectory, but in order to precisely characterize the entire flow the used of CFD techniques is advised for future works.

Chapter 3

The multi-rotor system

In this chapter the working principle of a multi-rotor vehicle is described, as well as the mathematical modeling of its dynamics. In particular, a quad-rotor is taken as a reference but all notions can be applied to vehicles with more rotors.

3.1 Flight and control principles

Considering the UAS as a rigid body, its movement can be described through six degrees of freedom. These motions are the forward and backward, lateral, upward and downward (where the centre of gravity is taken as a reference to describe the overall motion in the three-dimensional space) and the rotations around the pitch, roll and yaw axis. These degrees of freedom are controlled through the propeller rotational speeds and the thrust and torque that derives. As the UAS only has four motors and propellers, which are the inputs to the dynamical system, and six outputs (the six degrees of freedom), it is an *underactuated* system. In fact, as all propellers are pointed in the vertical direction, it isn't possible to directly control the longitudinal and lateral movement. However, as is further described, these two movements are indirectly controlled by rotating the vehicle around the pitch and roll axis. The quad-rotor can be flown in two different configurations:

- the X configuration, where all rotors contribute together to the pitch and roll rotations, so a frontal or lateral rotor can't be identified.
- the *cross* configuration (fig 3.1, where there are two rotors that control the pitch rotation and two for the roll rotation. This is the configuration selected for this application because the nozzles can be placed under

rotors 1 and 3 that present no offset with respect to the vine row, thus minimizing the spray deviation from the row itself.

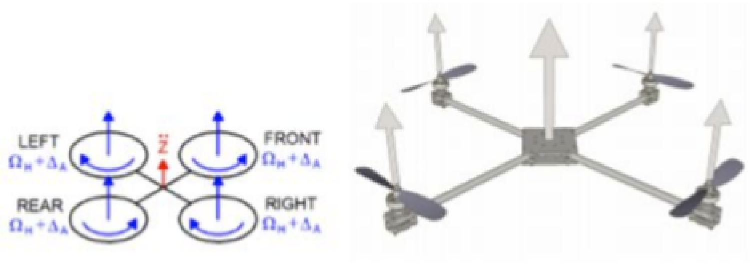


Figure 3.1: Quad-rotor *cross* configuration and spin direction

It must be noted that the configuration selected has no effect on the flight dynamics, as only the resultant thrust and torque are relevant.

3.1.1 Vertical motion

To control the upward and downward motion, it is sufficient to balance all propellers rotational speeds in order to obtain a uniform thrust that balances, exceeds or is less than the vehicle weight so to hover, move up or down respectively. Note that the propellers cannot all spin in the same direction, but must be alternated as shown in fig. 3.1. This is because otherwise the reaction torque would make the UAS spin in the opposite direction, like an helicopter would do without the tail rotor. That is also why there cannot be a multi-rotor with an odd number of rotors, as it would not be possible to balance and cancel the reaction torque.

3.1.2 Pitch rotation

In level flight, the front and rear rotors increase or decrease their rotational speed, while the lateral are kept the same. If, for example, the UAS has to pitch forward, the rear rotor will increase its thrust while the front one will decrease. This does not change the overall thrust but creates a torque that makes the vehicle rotate. This is how the forward/backwards motion is achieved, as the resulting thrust component is not aligned to the gravity vector anymore but has a component in the horizontal plane, which then moves the vehicle.

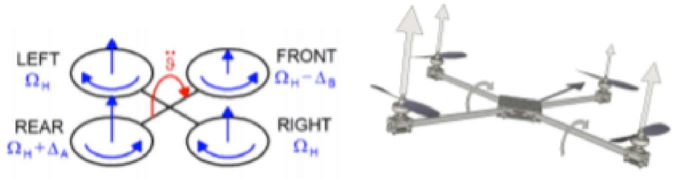


Figure 3.2: Pitch control

3.1.3 Roll rotation

The same consideration can be made as for the pitch rotation, but in this case the lateral rotors come into action. This also makes the lateral movement possible.

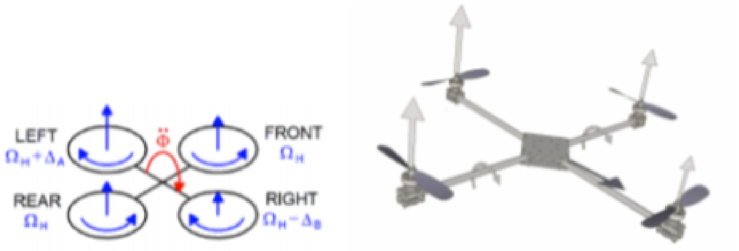


Figure 3.3: Roll control

3.1.4 Yaw rotation

To control the rotation about the vertical axis, the control system does not rely on the thrust difference but instead it exploits the rotors torque. For example, to rotate the quad-copter clockwise, the clockwise-rotating propellers speed will decrease, while the counter-clockwise ones will increase. This results again in the same overall thrust so not to affect the vertical motion, but the reaction torque isn't balanced and a yaw rotation happens.

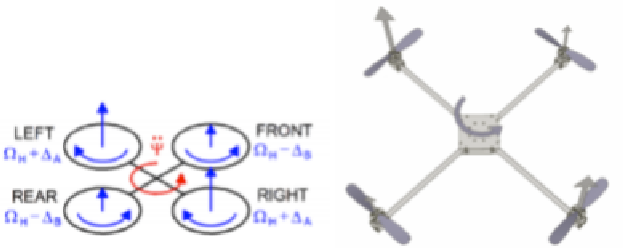


Figure 3.4: Yaw control

3.2 Motion equations

Once the basic architecture and working principles are defined, a mathematical model of the UAS must be defined in order to simulate its dynamics.

3.2.1 Reference frames

Before any mathematical modeling, the reference frames in which the machine operates must be defined. The classical description of a body moving in a three-dimensional space makes use of two different frames: a *body-fixed* one and an *inertial* one. An inertial reference frame is one where Newton's first law applies, so if a body isn't subject to any forces it remains in its state of rest or uniform motion. Here, the *NED* frame (*North-East-Down*) is taken as a fixed, inertial reference. Its axis will be referred to as X_{NED}, Y_{NED} and Z_{NED}.

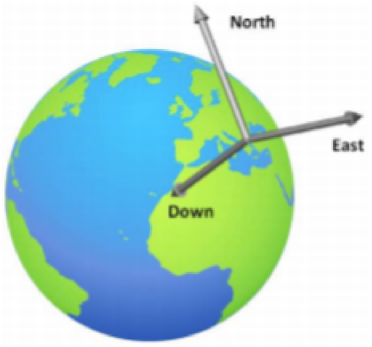


Figure 3.5: NED reference frame

Theoretically this isn't an inertial reference, as it is fixed to Earth which rotates about its axis and around the Sun. A body which is stationary in this frame is subject to a centrifugal force and the Coriolis force; however, these effects are negligible if compared to the force and thrust that the vehicle develops, so the *NED* frame can be approximated as inertial (21).

The *body-fixed* frame is centred in the centre of gravity of the quad-copter and rotates with it. The x_b-axis points forward, the y_b-axis to the right and the z_b-axis points down.

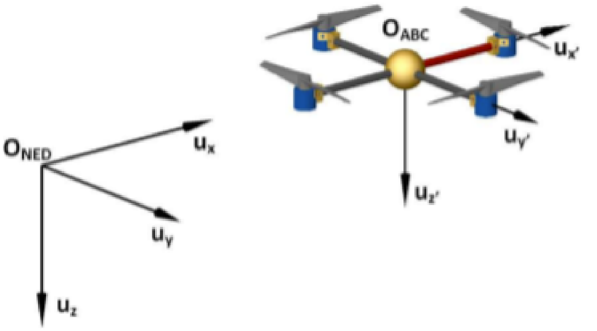


Figure 3.6: Aircraft Body Coordinates

3.2.2 Euler Angles

As the frames are defined, a way to describe the relative orientation must be found. This is usually done through Euler angles, which identify a standardized sequence of three elementary rotations to pass from one frame to another, in this case from the *body-fixed* frame to the *inertial* one. Each rotation can be described by a matrix of sines and cosines. The first matrix rotates the body frame about its X-axis:

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{pmatrix} \tag{3.1}$$

Then, a second rotation is performed around the Y-axis:

$$R_y(\theta) = \begin{pmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{pmatrix} \tag{3.2}$$

Last, the third rotation makes the two frames coincide

$$R_z(\psi) = \begin{pmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

The complete rotation matrix can be calculated by multiplying the three elementary rotation matrices:

$$R_b^{in}(\phi, \theta, \psi) = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) \tag{3.4}$$

$$= \begin{pmatrix} c(\theta)c(\psi) & s(\psi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\psi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{pmatrix} \tag{3.5}$$

As this matrix is square and orthogonal, it can be transposed to pass from *NED* to the *body-fixed* reference.

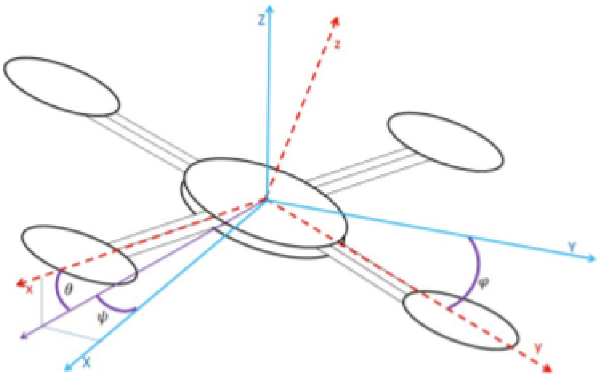


Figure 3.7: Euler angles

For a more thorough explanation, refer to (22), (23) and (24).

3.2.3 Dynamic model

In this section, the kinematic and dynamic equations for the UAS are obtained, but can be applied to any rigid body. First, the state variables must be defined:

- X = the position along X_{NED}
- Y = the position along Y_{NED}
- h = the position along $-Z_{NED}$, i.e. the altitude
- u = the body frame velocity along x_b
- v = the body frame velocity along y_b
- w = the body frame velocity along z_b

- $\phi, \theta, \psi =$ the Euler angles
- $p =$ the roll rate about x_b
- $q =$ the roll rate about y_b
- $r =$ the roll rate about z_b

Note that the altitude variable h is opposite to the *NED* axis Z_{NED}. For this reason the last row of the matrix R_b^{in} is multiplied by -1, to maintain coherence:

$$R_b^{in} = \begin{pmatrix} c(\theta)c(\psi) & s(\psi)s(\theta)c(\psi)-c(\phi)s(\psi) & c(\psi)s(\theta)c(\psi)+s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi)+c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi)-s(\phi)c(\psi) \\ s(\theta) & -s(\phi)c(\theta) & -c(\phi)c(\theta) \end{pmatrix} \tag{3.6}$$

The first kinematic relation links the inertial velocities to the *body-frame* velocities:

$$\frac{d}{dt}\begin{pmatrix} X \\ Y \\ h \end{pmatrix} = R_b^{in}\begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{3.7}$$

Then, a relationship between the Euler angles rates and the *body-frame* rates can be found:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{pmatrix}\begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{3.8}$$

With this last relationship, all six kinematics equations are defined. To complete the system, the six dynamics equations are needed. Let v be the velocity vector of the quad-rotor in the *inertial* reference frame. By applying Newton's second law of motion we can write:

$$m\frac{dv}{dt} = F \tag{3.9}$$

where m is the mass and F is the resultant force vector. Then the Coriolis equation gives:

$$m\frac{dv}{dt} = m(\frac{dv}{dt} + \omega \times v) = F \tag{3.10}$$

where ω is the angular velocity vector in the inertial frame. This expression is inconvenient, as the rate gyros on board measure the angular velocities in the body frame and the forces are easily computed in that frame as well. For this reason, equation 3.10 can be re-written in body coordinates:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{3.11}$$

where $(f_x, f_y, f_z)^T$ is the resultant force vector in body coordinates. This vector can be further expanded, dividing it into the thrust component, the gravity component and the wind drag component. As the thrust is aligned with z_b, it can be simply written as:

$$\frac{1}{m}\begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \tag{3.12}$$

where F is the total thrust. Note that z_b points to the ground, hence the negative sign. To express the gravity vector in body coordinates, the rotation matrix calculated before (3.4) can be used. First it has to be transposed:

$$R_{in}^{b} = R_{b}^{inT}$$

The gravity vector then becomes:

$$R_{in}^{b}\begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \begin{pmatrix} -mgsin(\theta) \\ mgcos(\theta)sin(\phi) \\ mgcos(\theta)cos(\phi) \end{pmatrix} \tag{3.13}$$

In order to write into equations the wind effects, first some assumptions have to be made:

- the total drag force is proportional to the square of the apparent wind's magnitude;
- the drag coefficient is a constant;
- the resultant drag force can be computed as the sum of the drag forces along the body axis:

  $$D = \frac{1}{2}\rho A_x C_{Dx} w_x^2 + \frac{1}{2}\rho A_y C_{Dy} w_y^2 + \frac{1}{2}\rho A_z C_{Dz} w_z^2 \tag{3.14}$$

  Where w is the apparent wind speed and A is the cross section. This approximation was made as there is no available aerodynamic model of the vehicle, but it will be needed for the wind estimation algorithm in chapter 5.

The resultant applied force vector is:

$$\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} -mgsin(\theta) \\ mgcos(\theta)sin(\phi) \\ mgcos(\theta)cos(\phi) \end{pmatrix} + \begin{pmatrix} -sign(w_x)D_X \\ -sign(w_y)D_y \\ -F - sign(w_z)D_z \end{pmatrix} \tag{3.15}$$

Where D_x, D_y, D_z are the drag components of equation 3.14.

Regarding the rotational motion, Newton's second law can be applied again:

$$\frac{dH}{dt} = M \tag{3.16}$$

where H is the angular momentum and M is the resultant applied torque vector. Again, the Coriolis equation yields:

$$\frac{dH}{dt} = \frac{dH}{dt} + \omega \times H = M \tag{3.17}$$

For the same reason as equation 3.11, it is convenient to write 3.17 in body coordinates. However, an important assumption is made: the mass distribution is assumed to be symmetrical around the body axis, so that the inertia matrix is diagonal:

$$J = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \tag{3.18}$$

The Coriolis equation in body coordinates can then be computed:

$$\begin{aligned}
\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} &= \begin{pmatrix} 1/J_x & 0 & 0 \\ 0 & 1/J_y & 0 \\ 0 & 0 & 1/J_z \end{pmatrix} \left[\begin{pmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{pmatrix} \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \right] \\
&= \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_\phi \\ \frac{1}{J_y} \tau_\theta \\ \frac{1}{J_z} \tau_\psi \end{pmatrix}
\end{aligned} \tag{3.19}$$

The complete cinematic and dynamic system then becomes:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c(\theta)c(\psi) & s(\psi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\psi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ s(\theta) & -s(\phi)c(\theta) & -c(\phi)c(\theta) \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \begin{pmatrix} -gsin(\theta) \\ gcos(\theta)sin(\phi) \\ gcos(\theta)cos(\phi) \end{pmatrix} + \frac{1}{m} \begin{pmatrix} -sign(w_x)D_X \\ -sign(w_y)D_y \\ -F - sign(w_z)D_z \end{pmatrix}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_\phi \\ \frac{1}{J_y} \tau_\theta \\ \frac{1}{J_z} \tau_\psi \end{pmatrix}$$

So, by integrating the last two set of equations, the values needed to integrate the first two set can be calculated, thereby closing the problem resolution.

In the preliminary design phase, the vehicle's mass was set to $m = 25kg$. As there isn't a CAD model available, the moment of inertia are estimated as follows:

- As the vehicle is symmetrical:

  $$J_x = J_y = \frac{2}{5} m_c r^2 + \frac{2 m_a b^2}{3} + 2 m_{mot} b^2$$

  $m_c = 22kg$ is the core's mass (central structure, tank, electronics), $m_a = 0.5kg$ is the single motor's arm mass, $m_{mot} = 0.25kg$ is the single motor's mass, $r = 0.2m$ is the core's radius (as the core is considered a sphere) and $b = 0.75m$ is the arm's length. This results in $J_x = J_y = 0.82075kgm^2$

- $I_z = \frac{2}{5} m_c r^2 + \frac{4}{3} m_a b^2 + 4 m_{mot} b^2 = 1.2895kgm^2$

3.3 Propulsive system

An important system to adequately simulate is the propulsive one. As will be explained later and as shown in figure 3.10, each rotor receives in input an angular velocity from which a thrust and torque value are derived. A simplification could be made and the system dynamics could be neglected, considering the relationship between the input signal and the developed thrust and torque as instantaneous. However, even if the response time of this system is much faster than the rigid-body dynamics of the UAS, it still introduces a retard that could affect the overall stability of the complete system. For this reason, a complete dynamic model of the rotors is developed. This is also useful to estimate the power consumption and so the overall autonomy and maximum mission duration. In order to accurately model the propulsive system, a commercial motor with available data (25) was chosen as a reference; this is the T-motor P80 III KV100, coupled with a G30*10.5" Carbon Fiber propeller. The parameters that can be used for this model are reported in the following table:

Throttle %	RPM	Current [A]	Thrust [g]	Torque [Nm]
40	1719	4.97	2850	0.97
42	1783	5.54	3108	1.06
44	1849	6.07	3331	1.13
46	1913	6.65	3571	1.22
48	1981	7.29	3859	1.3
50	2040	7.99	4114	1.39
52	2125	8.80	4439	1.5
54	2204	9.80	4748	1.61
56	2292	10.89	5143	1.75
58	2386	12.20	5515	1.88
60	2459	13.11	5886	1.99
62	2531	14.39	6306	2.14
64	2610	15.56	6649	2.25
66	2683	17.00	7061	2.39
68	2755	18.22	7434	2.51
70	2836	20.15	7886	2.67
76	3030	24.34	9074	3.07
82	3223	29.34	10215	3.45
88	3397	34.75	11500	3.88
94	3569	40.33	12603	4.28
100	3747	46.99	14011	4.78

Table 3.1: Motor parameters

As a direct relationship between angular velocity and thrust (and torque) can be directly derived from this data, the propeller/air interaction is not simulated, for example using a finite blade-element scheme. As the vehicle motion is very slow (maximum $3m/s$ by design), the propulsive characteristics are very similar to the ones measured in the table. As can be seen from figure 3.8, the relationship between thrust/torque and angular speed is quadratic. A constant thrust coefficient c_F and a constant torque coefficient c_M can then be calculated inverting the following equations:

$$F = \frac{1}{2}\rho c_F \omega^2 \tag{3.20}$$

$$M = \frac{1}{2}\rho c_M \omega^2 \tag{3.21}$$

By averaging for all values of thrust and torque in table 3.1, the final values are computed:

$$c_T = 1.43288 \times 10^{-3} [m^4]$$

$$c_M = 4.95217 \times 10^{-5} [m^5]$$

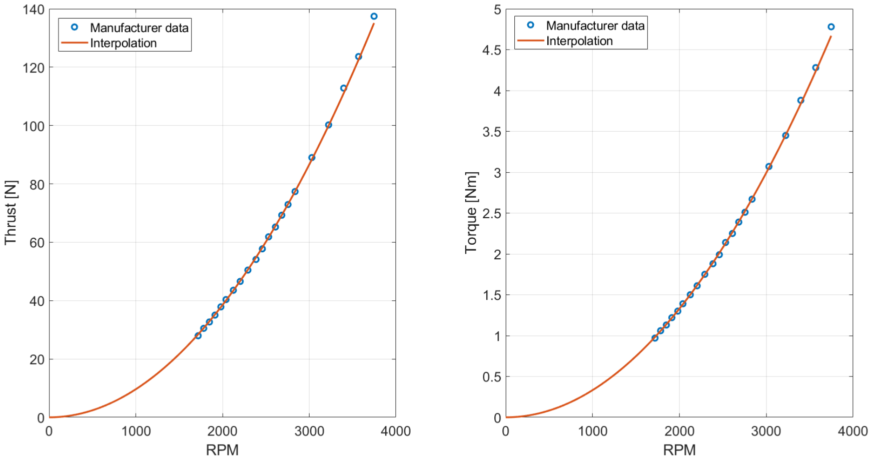


Figure 3.8: Thrust and torque as a function of motor speed

In the model, the ESC (Electronic Speed Controller) is simulated as well, although there is no commercial component data available. A simple PID controller (discussed in detail further on) is used to emulate this component. The motor manufacturer only specifies that the ESC used to control the motor (T-Motor FLAME 80A 12S) has a maximum output of 120A for 10 seconds (26); this value is used to set the maximum output from the PID controller. Furthermore, a relationship can be found that relates the torque produced to the input current:

$$T = c_1 I^{c_2} \tag{3.22}$$

Where $T[Nm]$ is the motor torque, $I[A]$ is the input current and c_1, c_2 are two constant coefficients ($c1 = 0.3322$, $c2 = 0.6932$)

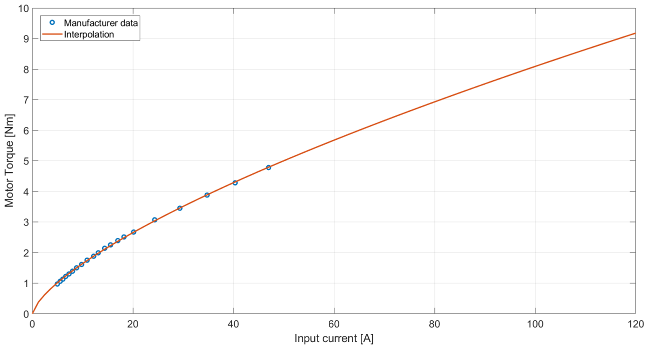


Figure 3.9: Motor torque as a function of the input current

Once all the data is gathered, the propulsion system model can be implemented in Simulink:

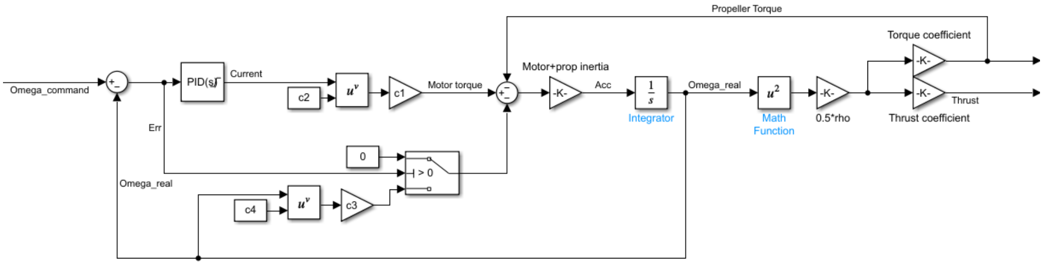


Figure 3.10: Simulink block scheme of the propulsive system

The model is very simple in its operation: it takes as input an angular speed signal from the control system (which be explained in the next chapter), which is compared to the actual angular speed of the rotor generating an error signal. The error is fed to the PID controller that regulates the current going into the motor. From equation 3.22, the produced torque is calculated. Then, the aerodynamic breaking torque is calculated using eq. 3.21 and the different between the two is the resultant torque that accelerates or breaks the propeller rotation. Using eq. 3.20 the thrust is computed, which together with the propeller torque is fed into the quad-rotor dynamic model.

The first-order differential equation that regulates the motor's dynamics, that is represented graphically in 3.10, can be written as:

$$\dot{\omega} = \frac{c_1 I^{c_2} - \frac{1}{2}\rho c_M \omega^2}{J_M} \tag{3.23}$$

Where J_M is the motor's moment of inertia. A second block can be seen contributing to the overall torque acting on the propeller shaft: it is used to simulate the back-emf and in fact it outputs a non-zero value only when the error signal is negative, meaning that the propeller should be slowing down and that no input current is coming from the controller. As the manufacturer doesn't provide data to simulate this effect, the parameters inserted are an educated estimate.

The output current is also logged and integrated in time to calculate the mission power consumption, as will be shown discussing the results.

3.4 Spray system

The spray system modeling is very simple, as it should only simulate the change in mass. This module receives in input a spray plan which is generated prior to the mission beginning, that provides a value for the PWM duty cycle that controls the nozzle opening. This value is calculated based on each plant need, so it's linked to the UAS position, as will be discussed in the Wind Estimation chapter.

From the PWM duty cycle value, a flow rate is calculated and integrated in time so to obtain the overall mass change, which itself is fed to the dynamic model. The variation in mass changes the vehicle moment of inertia, too. However, as the tank is very close to the centre of mass, this variation isn't accounted for.

The much more complex physical system scheme is shown in figure 3.11 for completeness.

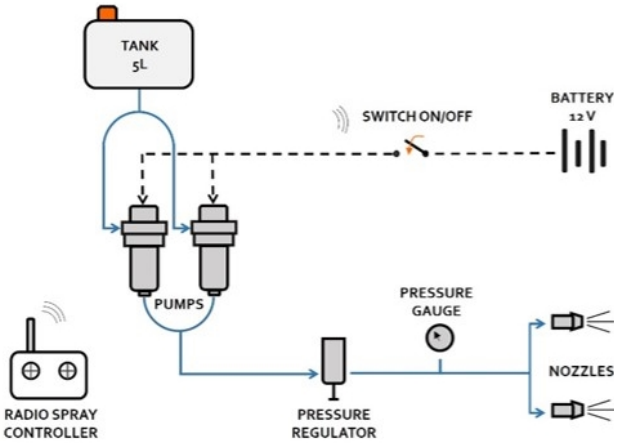


Figure 3.11: Spray system functional scheme

Chapter 4

Guidance and control system

The guidance and control system employs an architecture commonly used for this application, where there are two nested loops: the outer one controls the vehicle position and outputs the input signal for the inner one that controls the attitude. Here, a decentralized strategy (fig 4.2) is implemented, where all control loops are separated and do not interact with each other. This is less precise than an decoupling controller strategy (fig 4.1), that accounts for the effects of each command on all other axis through a decoupling gain matrix that modifies the system's transfer function. For example, when the UAS pitches, the front and rear rotors torque do not match, so a yawing moment arises. With a decoupling controllers strategy, the yawing moment would be accounted for and the pitching command would include a compensation for the yawing moment, whereas with a decentralized strategy the yaw controller acts separately to correct the error. The reduced accuracy is balanced by a simpler design, which is the reason why this strategy is chosen.

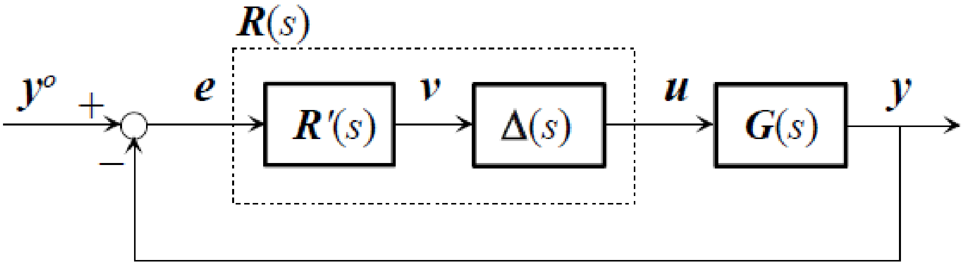


Figure 4.1: Decoupling controller strategy

In the figure above, it can clearly be seen that each variable has its own controller that acts independently and doesn't interact with all other variables.

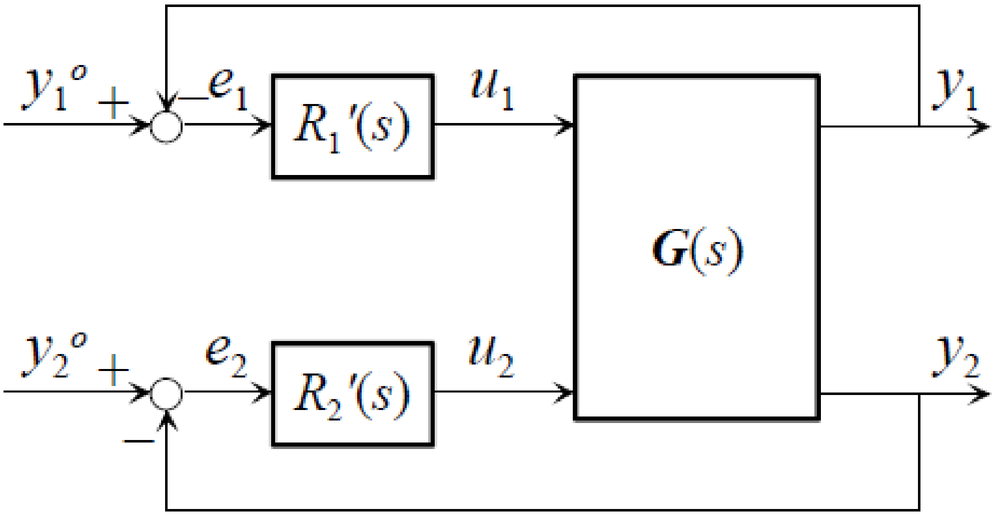


Figure 4.2: Decentralized strategy

Here however, the controller with the transfer function $R(s)$ (which is a matrix) has multiple inputs and multiple outputs. The complexity of this system derives from the fact that the so-called *decoupler* $\Delta(s)$ has to be designed so to make $G(s)\Delta(s)$ diagonal. The controller $R'(s)$ can then be designed as diagonal, so the loop transfer function $G(s)\Delta(s)R'(s)$ will be diagonal as well, realizing the variables decoupling as was intended. For a more thorough mathematical explanation, see (27).

There are six controllers in total, one for each degree of freedom, plus one for each motor that regulate the current input.

In a real-world application, the control system relies on data from sensors, which are affected by noise and uncertainties that should be accounted for when designing the system. However, as the focus of this work is about analyzing and correct the effects of wind and not designing application control system, the sensors were not modeled and the position and attitude data are taken directly from the output of the vehicle dynamics simulation.

4.1 PID controllers

In order to realize the quad-rotor control system, PID regulators were chosen as controllers. Even if they are less sophisticated and robust (28) than other type of regulations (such as LQR or H-infinite) they are very reliable and easy to tune. For this reason, they find applications across all industries and systems.

Considering e (the error between the controlled variable and the reference signal) as the input to the controller and u as the exit, in the time domain we can write:

$$u(t) = K_p e(t) + K_I \int e(\tau)d\tau + K_D \frac{d}{dt}e(t) \tag{4.1}$$

By applying the Laplace transform, the same equation can be written in the frequency domain:

$$U(s) = (K_P + K_I \frac{1}{s} + K_D s)E(s) = H(s)E(s) \tag{4.2}$$

Where $H(s)$ is the controller transfer function:

$$H(s) = K_P + K_I \frac{1}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s} \tag{4.3}$$

Strictly speaking, the regulator introduces a pole and two zeros in the feedforward loop. K_P, K_I and K_D are respectively the proportional, integral and derivative gains. The PID controller in fact acts on the error signal in three different ways:

- The K_P term produces a signal proportional to the error, achieving the basic feedback compensation control;
- The K_I term produces a signal proportional to the integral of the error. This is useful in the steady state error compensation;
- The K_D term produces a signal proportional to the error derivative, that help stabilizing fast transients and speeding up the response.

In order to achieve a satisfactory response, these values must be carefully tuned.

4.2 Motor controllers tuning

The propulsive system is an inherently stable system, as it is strongly dampened by the propeller aerodynamic breaking torque. For this reason, the controller can be a simple PI, without the need for a derivative action. The gain values are found by using a MATLAB script, that through multiple simulations finds the values that provide the satisfactory time-response. This means the shortest possible rise time and a steady-state error of less than 1 %, which was achieved.

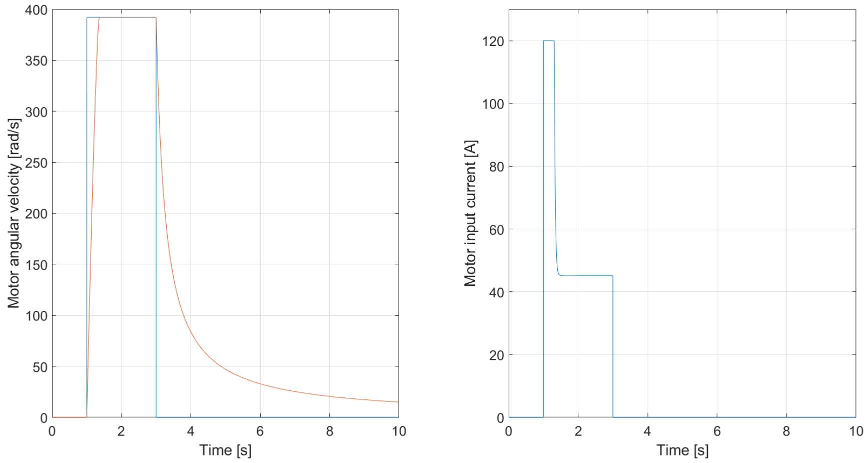


Figure 4.3: Motor step input response

In the figure above, the motors' step input response can be seen. The reference step value is the motor's maximum angular velocity, equal to 392 rad/s or 3750 rpm. As the graph on the right shows, the current reaches the saturation value (120 A) for a brief moment, meaning that the fastest response possible was achieved. The step response graph on the left also shows the motor deceleration, which is orders of magnitude slower than the acceleration.

4.3 Attitude control loop

The attitude control loop takes in input the reference angles given by the outer position loop, and outputs the reference angular speed for each motor.

4.3.1 Attitude loop architecture

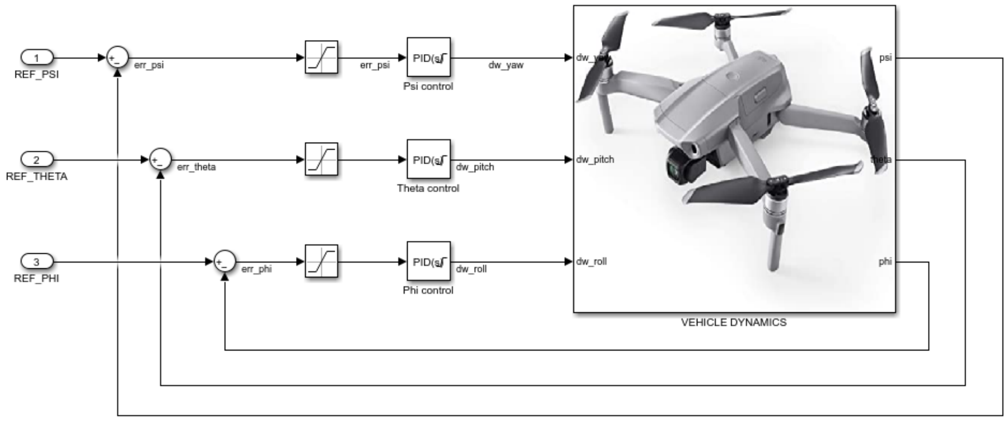


Figure 4.4: Visual representation of the attitude control loop

As can be seen, the reference angles from the position loop are compared to the actual values, and the resulting error signal is fed to the controllers that command the proper $\Delta\omega$ (variation with respect to the signal given by the altitude controller, as explained in the next section) to the motors in order to achieve the required attitude.

4.3.2 Controller tuning and response to inputs

The controllers have again been tuned through the use of a MATLAB script that run multiple simulations until the desired response was achieved. In this phase, the ESCs were already defined so the current saturation was accounted for.

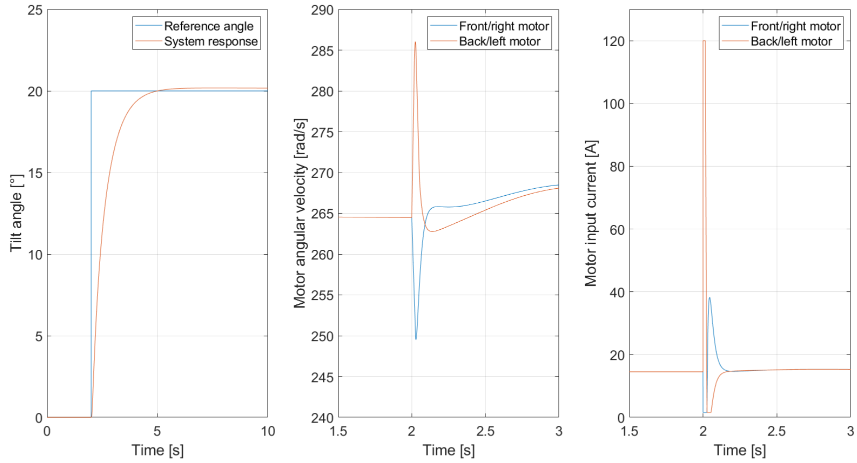


Figure 4.5: θ and ϕ step input response

In figure 4.5, the attitude response, the motors' angular velocity and the input current are shown in this order, relative to a pitch or roll command. As the vehicle is symmetrical, it is sufficient to study either the pitch or roll response. It can clearly be seen that the back/left motor increase their speed after the step input is given at $t = 2s$ while the front/right motor slow down, resulting in a pitching/rolling moment, respectively.

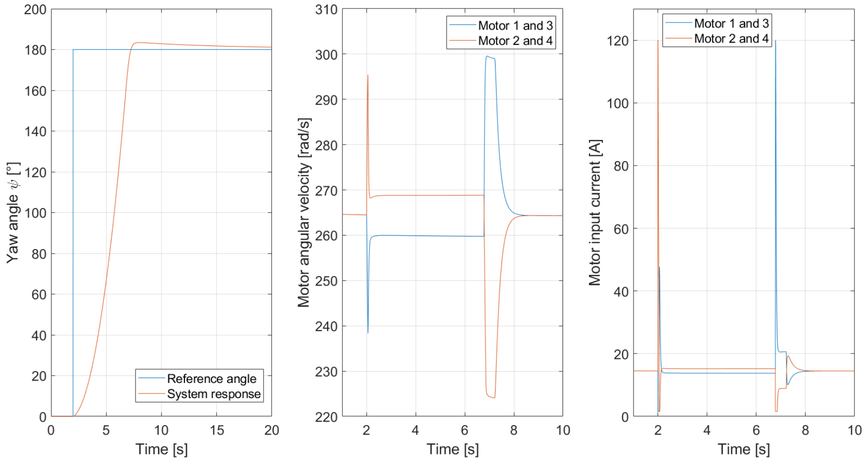


Figure 4.6: ψ step input response

In figure 4.6 the results regarding the yaw angle ψ are shown, with a step input of 180°, simulating a turn-around. The response is quite slow, but this is due to the limited authority given to the yaw controller. In fact, the control authority priority is given to the pitch and roll regulators, that should have enough angular velocity margin to control the UAS position at all times. This can be seen in the second and third graph, where the angular velocity and input currents, apart from the fast initial transitory, are well below the motors' capability.

4.4 Position control loop

The position control loop reads a position input from the path plan which is calculated prior to the mission (as discussed in the next section), and outputs the reference angles for the inner attitude loop

4.4.1 Position loop architecture

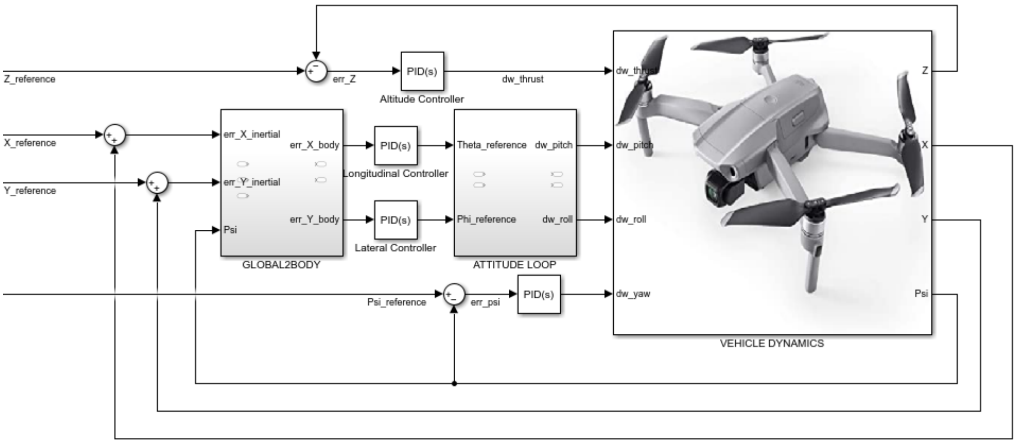


Figure 4.7: Visual representation of the position control loop

From figure 4.7, the loop can be seen comparing the position and altitude reference values to the real ones, and the resulting error is fed to the longitudinal and lateral controllers. The block *GLOBAL2BODY* acts as an intermediary, as it translates the position error calculated in the inertial frame to the equivalent value in the body reference so that the proper pitch and roll commands can be given to the motors. In fact, this block receives as input the heading angle of the UAS. Note that the altitude and yaw controllers are separate from the rest, as the degrees of freedom that they control don't have any effect on position.

4.4.2 Controller tuning and response to inputs

Like for the motor and attitude controllers, the tuning was done through multiple iterations with a MATLAB script. This was the last phase of the guidance and control system design, in order to then implement the wind correction system.

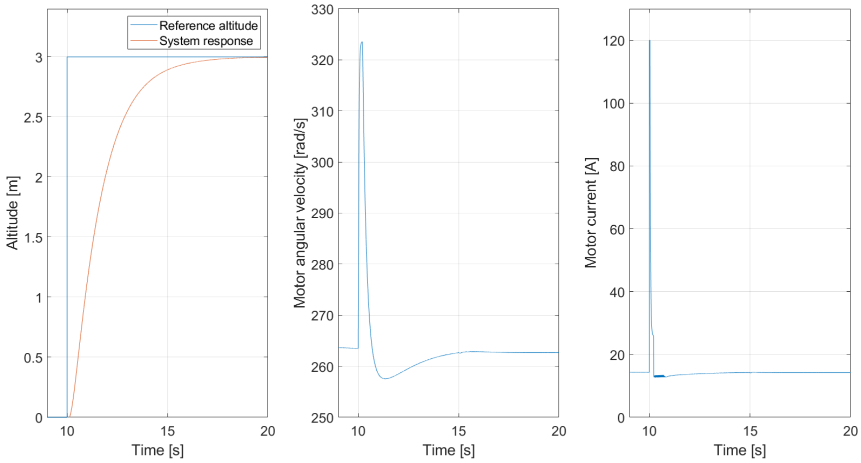


Figure 4.8: Altitude step input response

In figure 4.8, data from only one motor is represented, as the altitude regulator controls all motors in the same manner and their response characteristics overlap.

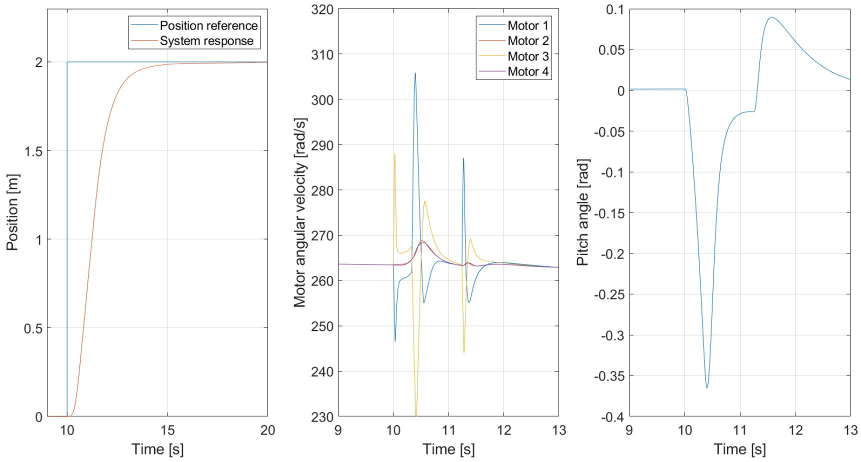


Figure 4.9: Position step input response

The motors behaviour is very clear: at $t = 10$, motor 1 decreases its speed while motor 3 decreases, creating the torque that tilts the vehicle forward; at $t \approx 10.5$ the opposite happens, as the UAS starts tilting the opposite way while motors 2 and 4 both provide a small increase in thrust. After $t = 11$, the pitch is positive (so, for the axis conventions, the quad-rotor is tilted back) in order to decelerate and capture the position reference.

Note that the quad-rotor maximum speed can be set prior to the mission, in order to control the maximum amount of PPP that can be deposited for a given area. In figure 4.10, the pitch angle is shown for different speed settings.

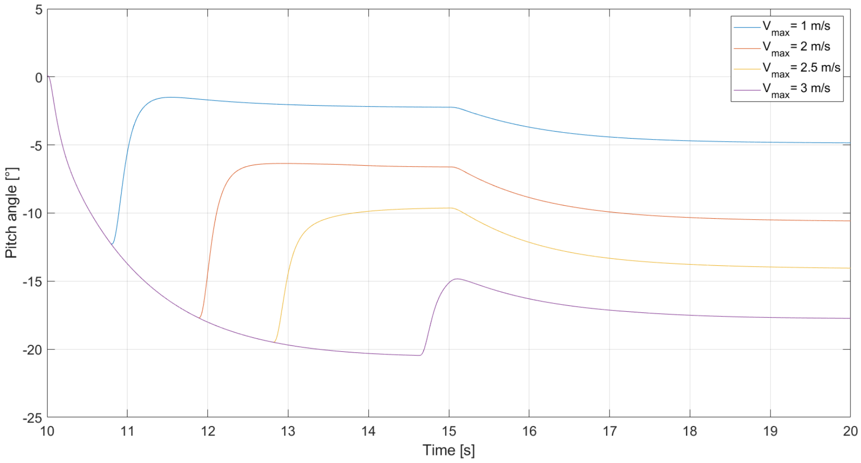


Figure 4.10: Pitch angle at different maximum speed settings

4.5 Summary of controller gains

The controller gains obtained by the response analysis, as well as the output saturation limits, which are used for all simulations are summarized in table 4.1.

Controller	P	I	D	Upper Lim.	Lower Lim.
Motor	3.27	0.51	0.001	120	0
Altitude	39.24	0.97	61	64.04	-132.15
Position (X,Y)	1.03	0.95	0.11	20	-20
Pitch, Roll	8.13	0	9.79	78.48	-78.48
Yaw	13.08	0.09	98.86	78.48	-78.48

Table 4.1: PID controllers' gains and output limits

4.6 Guidance algorithm and path planning

As reported before, the position control loop compares the current position to a reference value that is computed before the mission starts. This phase is called *path planning.* A dedicated algorithm creates a set of time-position values that are fed into the loop, so that the vehicle can track the prescribed trajectory. This phase can be set up manually as a series of waypoints, or it can be done automatically after a survey executed by another aerial vehicle which through techniques of image processing, computer vision and machine learning can recognise the plants most in need of treatment and so calculate the most convenient path. For this work, as the focus is to evaluate the particle drift, the flight plan makes the quad-copter follow all vine rows of a virtual vineyard. Usually, the path plan is calculated and fed straight to the control system; however, in this case it is continuously modified in order to take into account the effects of wind on the spray particles trajectory, as will be discussed in chapter 5.

The waypoints definition is executed as follows. First, the rows length and distance between each other is defined; in an operative scenario, this would mean replicating the real position of the rows by means of a preliminary survey but, in order to graphically illustrate in an effective manner the results, the ”virtual” vineyard used for the following simulations is defined as five $25m$ long rows separated by a gap of $3m$. Then, the row is divided into a preset number of points; each of these points contains the information about the position, the altitude and the heading that the vehicle should acquire at a given instant. The control algorithm reads this data points in sequence, and each point is linked to a time-stamp that indicates when the vehicle should be in the indicated position. The time-stamp definition is strictly linked to the operating speed of the vehicle: if, for example, it is set at $1m/s$, two points that are $1m$ apart from each other will have a time-stamp difference of $1s$ so that the control system can guide the UAS at the proper speed. This is particularly useful when different parts of the vineyard need different amounts of PPP, and the vehicle can slow down if the spray system cannot supply the desired amount at the standard operating speed (so to achieve the desired l/ha deposited) or speed up when, for example, no spraying is required over a specific part of the vineyard or when flying from/back to base.

Finally, the operating altitude is defined on the basis of the plants height, that can be very different from one *cultivar* to another, and some tests are needed to find the optimal vertical distance, as explained before (8)(16). In all the simulations, the UAS flies at an altitude of $3m$.

Chapter 5

Wind estimation

The drift model that has been developed can be used in the guidance algorithm of an agricultural drone in order to minimize the drift and particles dispersion. Two of the three inputs (rotor RPM and distance from the ground) are directly available from the sensors in the drone. Wind speed however must be derived indirectly, as explained in (29), (30) and (31). It is possible to directly measure the relative wind (32), (33), but this would require the installation of additional hardware that would lower the useful payload and increase costs.

5.1 Wind estimation theory

The most common method of wind estimation is based on the wind triangle, that utilizes data from the onboard IMU and GPS. Literature ((29), (30), (31)) suggests to use the method based on the relationship between the drone tilt angle and the relative wind. However, it is reliable only in steady-state flight, such as in hovering. For this reason a more advanced methodology has been developed.

This theory still relies on the knowledge of the effects of wind on the drone motion: it is necessary to conduct tests beforehand in order to establish a precise 3-dimensional drag and thrust model as a function of wind. In this paper, as only the theoretical working principle is explored, it is assumed to be already available.

The working principle is intuitive. First, a reference acceleration value is computed from the current attitude and thrust setting, based on the knowledge of the vehicle approximate dynamic model; this reference acceleration does not take into account the effects of wind on the vehicle. This value is then compared to the real accelerometer reading. Any discrepancy must be caused by the aerodynamic drag, and through that the relative wind speed and direction be evaluated. By comparing it to the speed given by the GPS sensor, the real wind vector is extracted. The mathematical formulation is hereby explained.

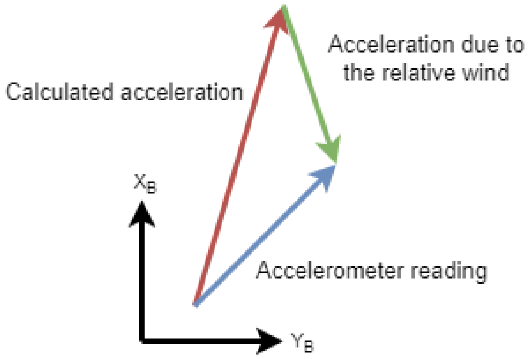


Figure 5.1: Wind correction algorithm basic working principle

First, data from sensors must be gathered, including:

- Euler angles from the IMU (φ, θ, ψ, roll, pitch and yaw);
- 3D GPS speed;
- rotors RPM, from which a value of total thrust can be computed (through a thrust model);
- body frame angular velocities from the rate gyro p, q, r;
- body frame accelerations from the accelerometer a_x, a_y, a_z;

The rotation matrix R can be evaluated from the Euler angles:

$$\begin{pmatrix} cos(\theta)cos(\psi) & sin(\psi)sin(\theta)cos(\psi){-}cos(\varphi)sin(\psi) & cos(\psi)sin(\theta)cos(\psi){+}sin(\varphi)sin(\psi) \\ cos(\theta)sin(\psi) & sin(\varphi)sin(\theta)sin(\psi){+}cos(\varphi)cos(\psi) & cos(\varphi)sin(\theta)sin(\psi){-}sin(\varphi)cos(\psi) \\ sin(\theta) & -sin(\varphi)cos(\theta) & -cos(\varphi)cos(\theta) \end{pmatrix} \tag{5.1}$$

This matrix represents the rotation from body frame to inertial. It is orthogonal, so its inverse is its transposed. This property come in handy as the body frame velocities (u, v, w) have to be calculated from the GPS reading:

$$(u, v, w)^T = R'(\dot{X}, \dot{Y}, \dot{Z})^T$$

The reference accelerations can then be evaluated:

$$a_{x,ref} = rv - qv - gsin(\theta)$$

$$a_{y,ref} = pw - ru + gcos(\theta)sin(\varphi)$$
$$a_{z,ref} = qu - pv + gcos(\theta)cos(\varphi) - T/m$$

These values can be compared to the real accelerometer reading. For this work, is is assumed that a quadratic relationship exist between drag and wind speed. The apparent wind can then be evaluated as:

$$wind_{apparent} = \sqrt{\frac{2(a - a_{ref})m}{(\rho C_D A)}} \tag{5.2}$$

The final wind estimation value is:

$$wind_{estimated} = wind_{apparent} + (u, v, w)^T$$

As these values are expressed in the body frame coordinates, they have to be multiplied by the rotation matrix R so to obtain a wind estimation in the inertial frame of reference.

5.2 Wind environment

In literature, many empirical and simulation models are found, such as the Weather Research and Forecasting model (34), (35) that mainly focus on large scale phenomena but fail to describe local conditions. Also, in the Simulink environment there are pre-set blocks to simulate the atmospheric wind phenomena, like the Dryden and Von Karman wind turbulence model. However, these can be very useful to simulate the upper atmosphere but can't be used to accurately represent the near-ground wind environment, especially in mountainous areas where it's disturbed by obstacles such as the plants themselves. For this reason, the wind is simply modeled as white noise, whose intensity and frequency characteristics are derived from an analysis of different sources (36) (37) (38) (39), including the aforementioned models. Anyway, whatever the model used, the wind estimation will be filtered by a low pass filter, so the wind model does not have to exactly represent the real conditions as the higher frequency components won't be considered in the guidance system design.

5.3 Results

In order to test the reliability of the method, simulations were conducted introducing errors in the thrust and drag coefficient estimation. This simulates a real-world environment where it is not possible to have a perfect model of the vehicles characteristics. In particular, a white noise of 10% of real the value is added to C_d and the estimated thrust, as shown in figure 5.2. Furthermore, errors were introduced into the sensors signals, and their values are reported in table 5.1

Noise power and frequency					
	C_D	Thrust	GPS	Rate gyro	Accelerometer
Power	10 % of C_D	10 % of thrust	1e-2 [m]	1e-5 [rad/s]	1e-5 [m/s^2]
Frequency	10 Hz	10 Hz	100 Hz	100 Hz	100 Hz

Table 5.1: White noise power and frequency for different variables

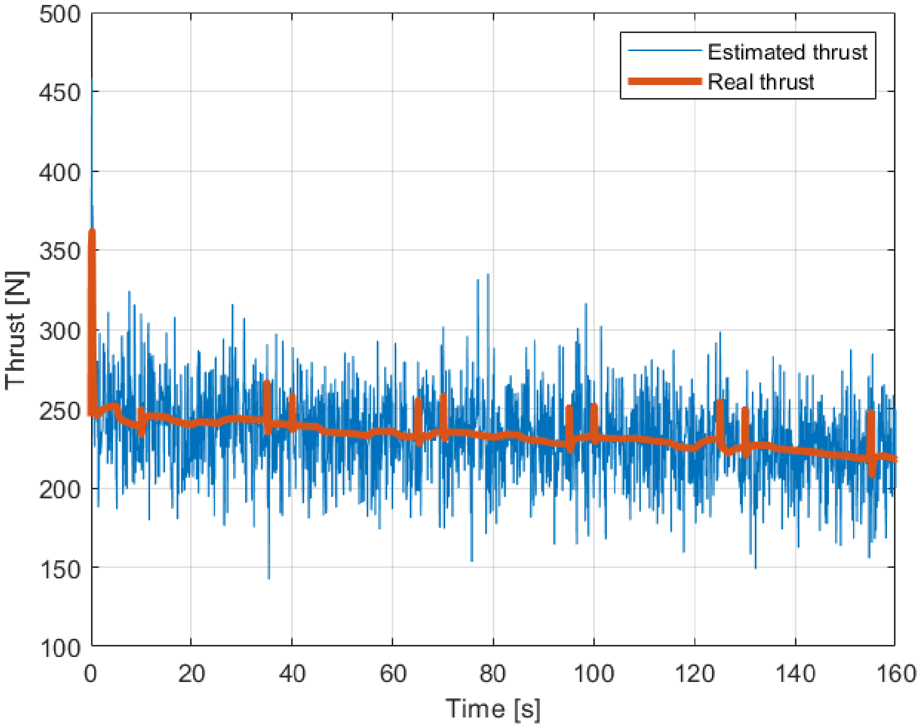


Figure 5.2: Real and estimated thrust

Even with this relevant uncertainty the algorithm is still able to effectively estimate the wind. It was run at 10 Hz and then a low-pass filter with a cut-off frequency of 5 Hz was applied in order to smooth the signal. However, as the thrust mainly affects the accelerometer reading on the vertical axis while C_d contributes to the reading on the horizontal plane, as the uncertainty on the former is greater than the one on the latter, the vertical axis wind speed estimation is much less precise than the horizontal plane components.

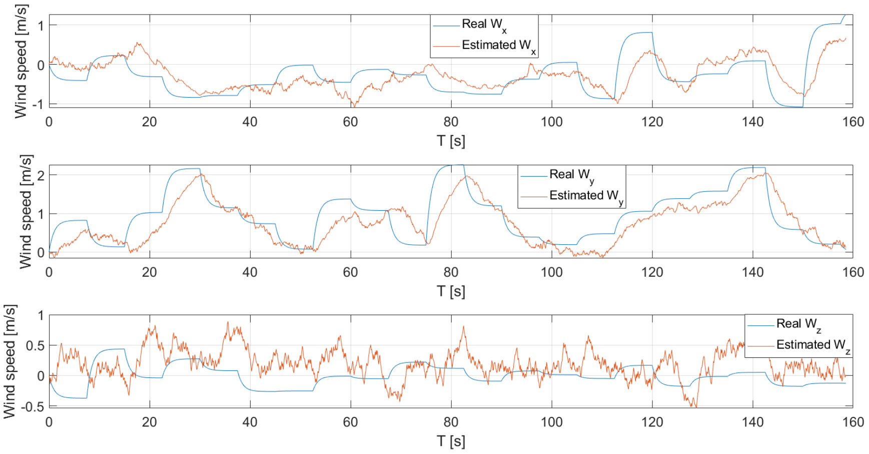


Figure 5.3: Comparison between real and estimated wind components on the three axis

This does not compromise the correct operation of the system, as the relevant wind components are in the horizontal plane and the estimation is still in an acceptable range and yields satisfactory results, as will be discussed later.

5.4 Adaptive guidance algorithm and spray controller

Once a reliable relative wind measure is available, the spray model can be used to correct the trajectory and the spray planning of the drone in real time. The model, given the height from the ground, the rotor RPM and the relative wind speed and direction, gives the position of the spray footprint's centroid relative to the drone. The correction acts in two ways: it modifies both the path and the timing of the spray. The path correction only accounts for the lateral deviation (with respect to the vine row) of the spray cone, while the timing correction assures the longitudinal compliance to the spray plan. The spray plan is made in advance, before the flight itself, and provides the spray system with the PWM intensity (to regulate the flow) together with a time stamp, which is linked to the drone position. It is essential that the timing is modified because, if the longitudinal correction was made only through the positioning of the drone, it would swing back and forth if wind bursts were present and would also increase battery discharge and shorten the maximum flight time.

This system also ensures that no PPP is wasted, as it is able to detect if the UAS is in a position where the majority of the droplets will not reach the target plant and will drift away from it. This is particularly useful when there's a sudden change in the wind magnitude and direction, as it shuts off the pump giving the vehicle the time necessary to re-position itself.

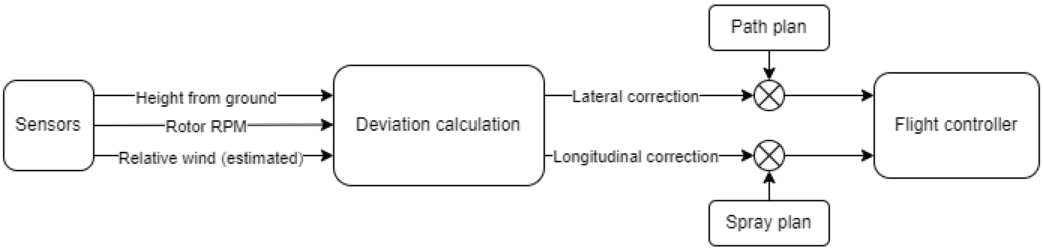


Figure 5.4: Adaptive guidance simplified scheme

Figure 5.4 graphically represents the working principle of the system: the *Deviation calculation* module calculates the longitudinal and lateral correction, based on reading from sensors and the wind estimation results. These corrections are then added to the position read from the path plan and to the spray plan, obtaining the overall needed correction.

More specifically, the algorithm operates in the following way. First, the vineyard surface is divided into a grid, and each point in the grid is assigned a spray system flow rate, according to the spray plan. The wind correction algorithm then gives the control system the lateral correction needed to move the vehicle in the right position, and then compares the expected position of the droplets deposition to the grid defined before. If the ground footprint is inside the target, the pump is turned on and the spraying operations will be performed, otherwise the pump is shut off so to not deposit PPP in off-target areas, as described in figure 5.5.

This figure shows both the lateral correction (which is enforced by modifying the path plan), highlighted by the vehicle's offset with respect to the vine row, and the longitudinal one, which is done by comparing the expected droplets' trajectory to the target area position, shutting the pump off when the particles deposit off-target achieving simultaneously the timing correction and the automatic shut off functionality.

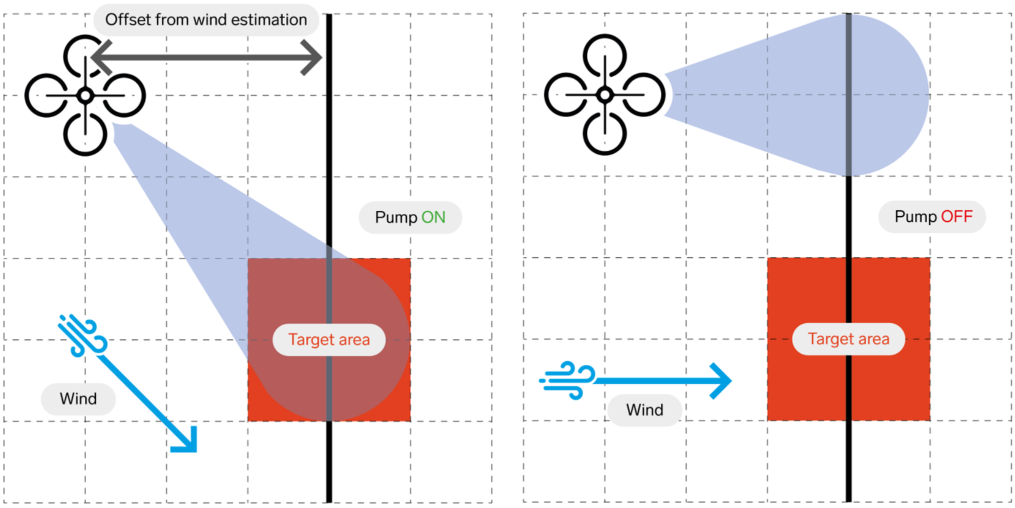


Figure 5.5: Spray correction during operations

Chapter 6

Simulation results

In this chapter, the simulation results are presented and discussed, by analyzing the vehicles dynamics and the effects of the wind correction algorithm functionality on the droplets deposition.

6.1 Vehicle dynamics

In this section, the simulation results of the UAS' flight are presented and analyzed, focusing on the flight dynamics and its response to disturbances. All simulations were performed using the mass and moments of inertia ($M.o.I.$) characteristics calculated in chapter 3.2, that are reported in table 6.1 for convenience.

Mass [kg]	pitch M.o.I. [kgm^2]	roll M.o.I. [kgm^2]	yaw M.o.I. [kgm^2]
25	0.82075	0.82075	1.2895

Table 6.1: Mass and inertia characteristics

As said before, no data is available on the aerodynamic characteristics of the vehicle, so for simplicity the product between surface area A and coefficient of drag C_D is assumed to be a constant: $C_D \cdot A = 5m^2$. As was explained in the previous chapter, it is very important to have an accurate aerodynamic model of the vehicle in order to successfully use the proposed method, but as this was not available at the time of writing, this constant value was used just to prove the system's functionality.

6.1.1 Motors and energy consumption

The results here presented were obtained under the same flight conditions as before ($2m/s$ wind directed eastward with $2m/s$ random gusts), while the vehicle flies at an altitude of $3m$.

First, the motors input current is visualized (figure 6.1). Only motors 1 and 2's data is plotted, as the other two mostly overlap these (as the vehicle is symmetrical). One main pattern is evident: there are spikes at regular intervals, and these coincide with the end of the vine rows, when the vehicle has to rotate 180° on its vertical axis. These high currents are required as the yaw torque, derived from the difference in the four motors counter torque, is minimal with respect to the vertical axis moment of inertia, as will be shown further on.

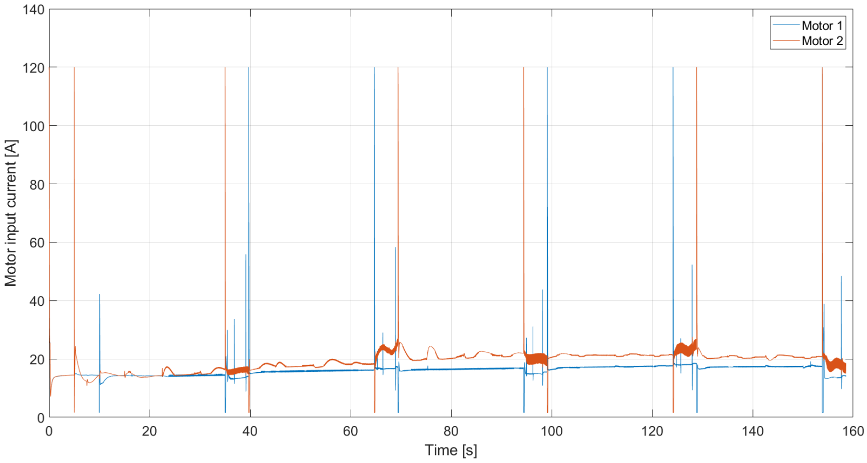


Figure 6.1: Motor input current

By integrating all four input currents, the overall energy consumption for a mission can be estimated. The energy consumption is a very important parameter to monitor, as it is the main factor affecting mission duration and so area coverage capability. This in turn affects the operational costs, so only an energy-efficient UAS that is able to fly over an extensive area without the need to constantly fly back to base could provide a good return on investment, especially in agriculture where the margins of profit are tight. This vehicle positions itself right in the average of the already available agriculture-oriented UASs, as in a single mission with a full payload can fly for almost 10 minutes if equipped with a 10000 mAh Li-Po battery, which is the standard for vehicles of this size (40). The autonomy was evaluated by averaging the energy consumption over 10 different missions under the same wind and payload conditions that were used in section 2 of this chapter ($2m/s$ constant wind with $2m/s$ random gusts), while the spray plan was set so to spray at a constant rate of $1l/min$ in order to fully deplete the expected $10kg$ payload over the course of the mission.

The currents determine the torque acting on the motors' shafts, and in turn they determine the rotational velocity (figure 6.2).

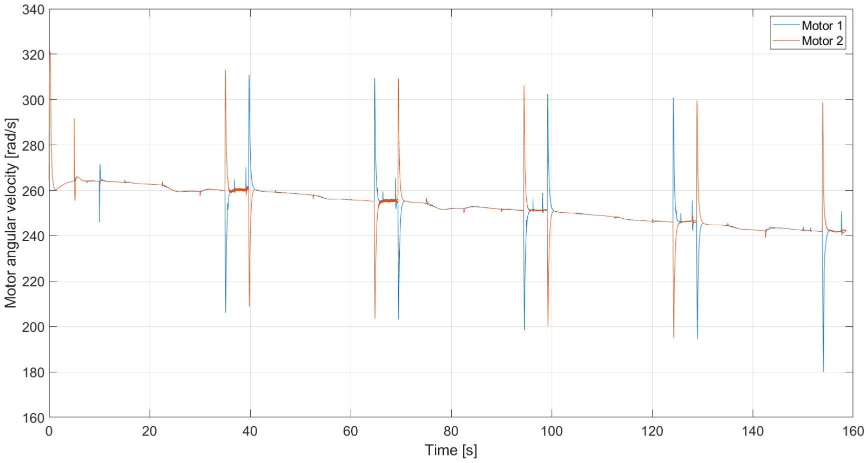


Figure 6.2: Motor angular velocities

The angular velocities follow the same pattern as the currents, with spikes during the rotation about the yaw axis, but highlight another factor: the average speed decreases during the mission, as the PPP is sprayed and depleted, requiring less overall thrust, as will be shown in the next section.

6.1.2 Thrust and torques

As anticipated, the overall thrust is strictly correlated to the vehicle's mass (figure 6.3).

The same spikes as before can be seen, as when the vehicle reaches the end of the row it slows down in order to perform the turn and then accelerates again to the set operating speed. The spike just after $t = 0$ is due to the lift off from the ground to the operating height.

Then, the resulting torques are analyzed (figure 6.4).

This figure highlights what was said before and shows that the yaw torque is very little in magnitude if compared to the others, meaning that the control system, in order to have sufficient authority on this axis, is required to momentarily spin the motors to their maximum speed. Again, the spikes

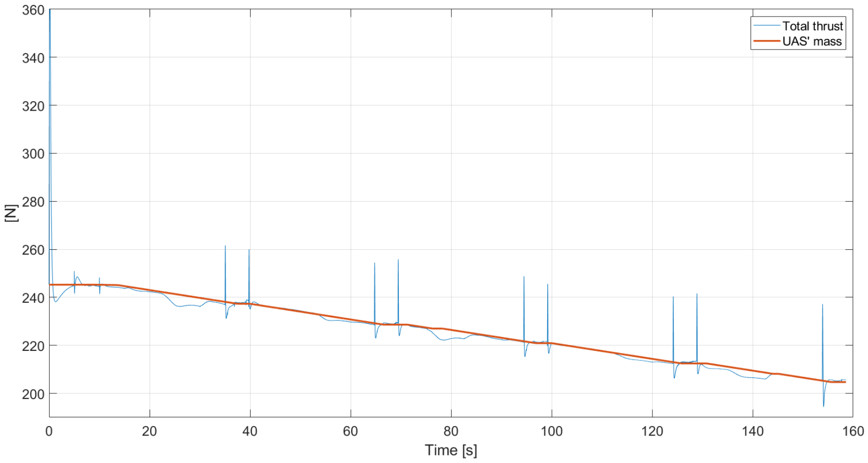


Figure 6.3: Thrust compared to the quad-rotor's mass

during turns are evident, in particular those related to the pitch axis which is responsible for the longitudinal speed of the vehicle.

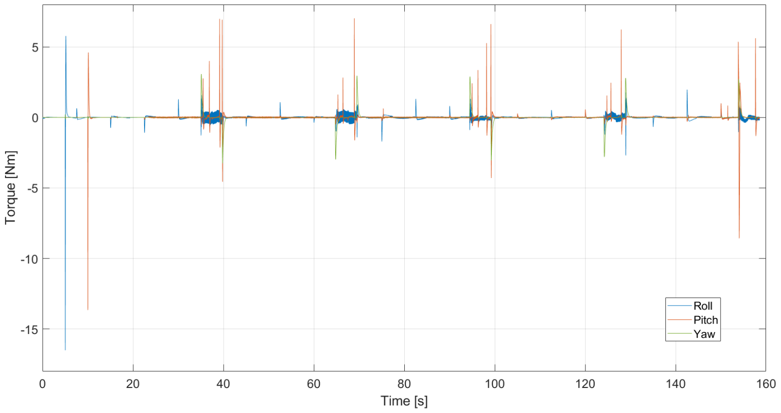


Figure 6.4: Resulting torques

6.1.3 Attitude

Here, the pitch and roll Euler angles during the mission are shown (figure 6.5).

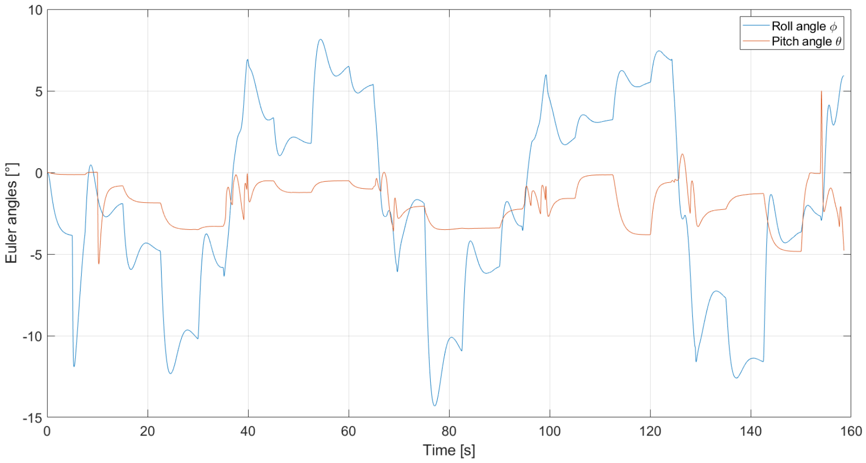


Figure 6.5: Pitch θ and roll ϕ angles

These show great variability due to the wind correction algorithm that constantly changes the vehicle's attitude so to capture the optimal position, but again some patterns emerge from the noise.

First, the pitch angle is always slightly negative (*i.e.* the UAS is leaning forward, as per the axis convention), so to maintain a positive longitudinal velocity. Second, the roll angle shifts from positive (*i.e* right) and negative (*i.e* left) at regular intervals: as the wind is blowing eastward, initially the vehicle tilts left in order to acquire the optimal spraying position and counteract the wind force; then, when it makes a 180° turn, it has to tilt the opposite way to achieve the same result, and so on until the end of the mission.

6.2 Wind correction algorithm and spray drift analysis

In this section, the wind correction algorithm is tested to see if it's able to correct the trajectory in the right way. To demonstrate its effectiveness, both in the lateral and longitudinal directions, two simulations are brought as example: in the first one the wind is set as a constant $2m/s$ eastward (*case 1*), while in the second one it blows at $2m/s$ westward (*case 2*). In both simulations, random-direction $2m/s$ gusts are added and the vehicle flies at an altitude of $3m$.

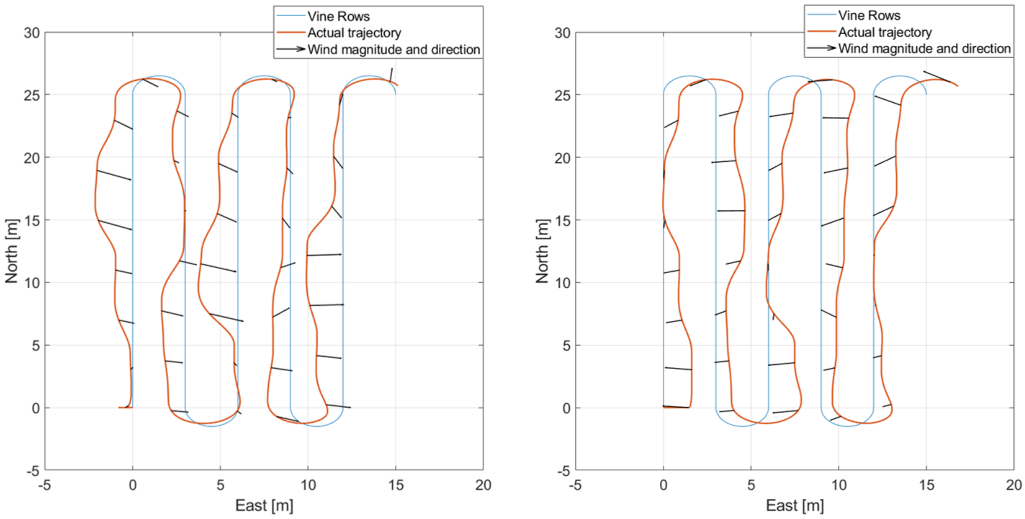


Figure 6.6: Vehicle trajectory compared to the vine rows' position

In figure 6.6, it can be seen that the path correction part of the wind correction algorithm is effective, as it shifts the quad-rotor position in the direction opposite to the wind direction, of a quantity that is proportional to the wind magnitude and that is calculated through the drift model that was created in chapter 2.

The wind correction algorithm also acts on the spray timing, so a comparison between the spray plan and the actual pump activation is needed. Here, in order to better visualize the shift between the two, the wind is set as a constant $2m/s$ breeze directed southward, parallel to the vine rows and the vehicle flies at a constant speed of $1m/s$.

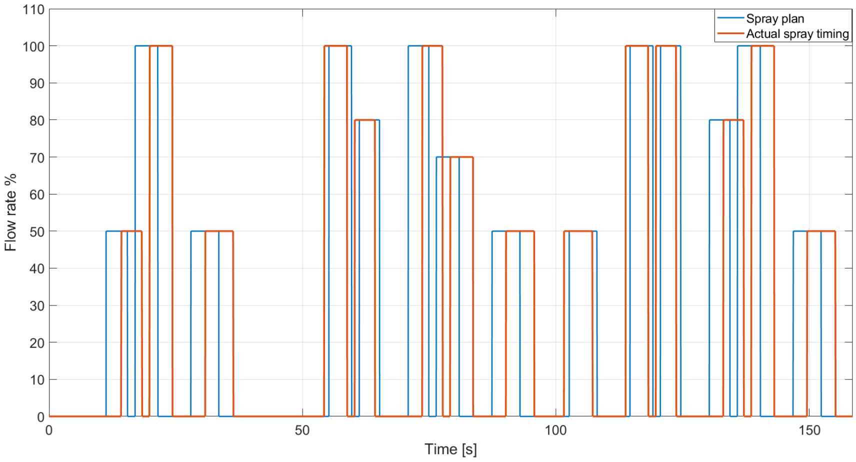


Figure 6.7: Spray plan correction

In the figure above, the timing correction is evident. As the wind is directed southward, the algorithm initially retards the pump activation as the UAS should be further along the row than planned in order to correctly target the plant. Then, when the flight direction changes, the correction anticipates the spraying as the vehicle is slower than the wind, so the droplets are blown in front of it. Notice that when the quad-rotor is moving opposite to the wind direction, the correction is much more significant than when it moves in the same direction, as should be expected as the vehicle speed is added to the wind speed.

As the main goal of this work was to develop a guidance algorithm capable of minimizing the spray drift, the particle deposition on the ground must be analyzed in order to understand the real capabilities of the system. To do this, a two-variable probability density function for the droplets deposition inside the ellipsoid calculated through the spray model has to be hypothesized. It is assumed that 99.7 % of the particles deposit inside of it so, using a normal distribution probability function centered on the centre of the calculated deposition area, the value of the standard deviation σ along each of the two axis is easily computed:

$$Pr(\mu - 3\sigma \leq X \leq \mu + 3\sigma) \approx 99.7\%$$

Meaning that along a distance a between the points $-\frac{a}{2}$ and $\frac{a}{2}$:

$$\frac{a}{2} = 3\sigma \text{ and so } \sigma = \frac{a}{6}$$

This is particularly useful as the spray model outputs not only the ellipse's centre position but also the dimensions of the two axis a and b. Given the probability density function, a grid of points is constructed, and in each simulation time-step it can be evaluated in the grid points, thereby calculating the PPP distribution across the vineyard.

The entire calculation is done in the post-processing phase, based on the simulation results.

As done before, two simulations are presented, under the same wind conditions: a constant $2m/s$ eastward (*case 1*) or westward (*case 2*) wind and $2m/s$ random bursts, at a constant altitude of $3m$. The spray plan is set so the pump is supposed to be always active along the rows' length.

The difference in the particle deposition with the wind correction algorithm active and inactive is very clear: as can be seen in figure 6.8 (bottom figures) the spray ground track is shifted towards the direction the wind is blowing to, with a lot of variability due to the random bursts, and rarely overlaps the vine rows. Also, the north-south component of the wind makes it go beyond the rows' length.

With the wind correction algorithm (middle figures), this doesn't happen. The ground track mostly overlaps the rows, apart from short transients when the wind suddenly changes direction. The timing correction is evident as well, as little to no droplets are deposited beyond the row length. In this plot the automatic pump shutoff functionality is also highlighted, in particular in the bottom of the central row in the middle-right plot: here, the wind shifts direction very suddenly and the UAS has no time to position itself in a favourable position and it determines that most of the PPP will be wasted. Therefore for a brief moment the pump is not active, and so there is a discontinuity on the particle density plot (the darker area).

Having made all these considerations, it is evident that this algorithm is essential for precision agriculture techniques, as otherwise the final PPP distribution is randomly determined only by the wind.

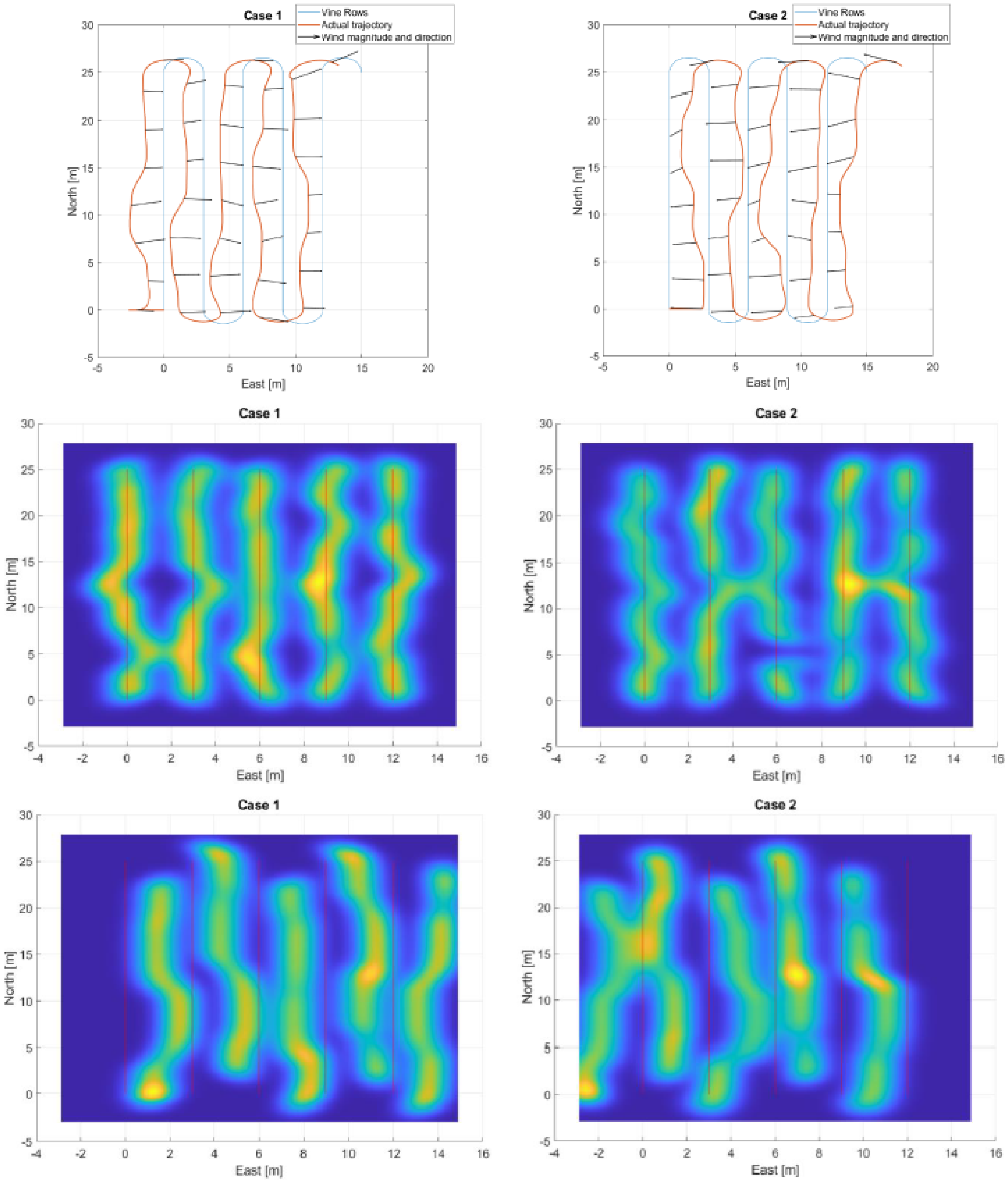


Figure 6.8: Comparison between the deposition with (*middle figures*) and without the correction algorithm (*bottom figures*)

Chapter 7

Conclusions and future developments

7.1 UASs as a plant protection tool

Throughout this work, the potential performances of a UAS for an precision agriculture application have been explained. It is clear that it could introduce a level of precision and automation that more traditional tool cannot achieve, a property that is more and more looked for in modern agriculture. However, a thorough analysis of the benefits vs. costs should be conducted, as this kind of UAS gives high performances but is limited in scope and its price would land in the tens of thousand of euros/dollars.

In fact, as a single unit, it would be only capable of covering a very limited area and delivering a small amount of PPP, making it unsuitable for high-volume, intensive plant treatments where cheaper, traditional machinery would still be a better option. It is though a really viable solution for light treatment in sloped terrain and scattered plots of land, where moving heavy land based machines may be impractical. Were it part of a large integrated system made of many different vehicles interacting with each other, where economies of scale would certainly offset the high purchase price of the system, it could become a viable alternative to current practices, increasing productivity and reducing PPP waste, resulting in an overall more environment friendly agriculture.

In a real-world operative scenario, many different products are sprayed onto the plants in order to tame different diseases and parasites, as explained in (41), for a total of about 750 to $1000 l/ha$. Supposing that accurate surveys (in order to target only the plants that need treatment) and other forms of pests and disease control (42) (7) are implemented, cutting down the required amount to $500l/ha$, in order to deliver such quantities with a payload of about $12.5kg$, the UAS should do at least 40 trips for a single hectare of land. With a spraying capacity of $\approx 2l/min$, that would require at least 4 hours in only flight time, not considering the time required to refill the tank and charge or switch the batteries. These operations, including the time required to go from the operations base to the vineyard, would certainly double the time needed, bringing it up to 8 hours per hectare. This is slightly more than what is required to do the same treatment by hand (4-6 h/ha) and almost an order of magnitude more than what highly mechanized plain viticulture requires (0.75-1 h/ha) (41). However, in steep slope viticulture, only manual work can be employed and it results in a +161% cost difference if compared to plain vitculture (43). This is exactly the sector where such system could be used in, as its autonomous operation would cut costs dramatically and would offset the extra time required to distribute the PPP (as it could operate even when dark), which itself could be reduced if two or more UASs are used simultaneously.

As in Italy the average vineyard size is $2.34ha$ (44) and the treatments operations are conducted only $\approx$ 10 times per year (41), it might not be a cost effective solution for a farmer to purchase the system and use it on his own, but it could be shared or rented as is often done with expensive machinery, making it a viable solution for steep and rough vineyards plant protection treatments.

7.2 Simulation improvements

In summary, this system presents can potentially improve existing technologies for agricultural applications, and with further research and study can be improved to extend its capabilities. The system design was carried out completely through simulations, and as explained throughout the thesis many assumptions and approximations were made, some for the lack of available data or because the information needed could only be got through experiments. This chapter aims to define the framework for the future developments of the simulation tool, so to have a better understanding of the system's real performances and capabilities and build a more accurate and solid model for the eventual implementation and manufacturing.

First of all, as discussed in chapter 2, the wind drift model should be improved to better adapt to a wider range of flying conditions. If the image analysis method is to be followed, the experimental campaign should be conducted on a prototype similar to the proposed quad-copter, as the one used for this work was a very different vehicle and the results were adapted to fit the conceptual design of the final UAS. As discussed before, the spray model could also be augmented through CFD, that can provide information that photography and particle-tracking methods such as *PIV* (Particle Image Velocimetry) can't. Furthermore, it can be used to better predict the particle deposition on the plants, as in this work the system's precision was assessed only by studying the theoretical deposition at ground level, which doesn't fully reflect the real system effectiveness.

CFD can also be used together with wind tunnel experiments for the definition and characterization of the aerodynamics: as explained in chapter 3 and 5, the aerodynamic drag is the most important characteristic to be estimated when running the wind estimation algorithm, and it was assumed to be known. In the dynamic model a very simple, constant drag coefficient was used just do demonstrate the working principle of the system, but in the real world this estimation could be much less precise if a full aerodynamic model wasn't available. The wind drag characterization should investigate the widest possible range of operative conditions, studying the vehicle's behaviour in all attitudes and operating speeds. Computational fluid dynamics could also provide a better insight on the wind conditions close to ground level in a mountainous area, where the plants play an important role in the overall air turbulence as well.

The airflow computational simulation can also be a valid tool helping the assessment of the propellers' performance. In this work, the manufacturer's data was used in the propulsive system modeling, but it was obtained from static tests. In a first approximation, this could suffice to calculate the UAS's dynamics as it will only operate at slow speeds. The next step would be using the consolidated *blade element theory*, which would introduce the relative wind speed in the thrust and torque calculation. However, this would still neglect the interaction of the air with the vehicle's structure that only CFD would be able to evaluate. Furthermore, this simulation could be conducted together with the spray model one, as it introduces the induced turbulence of the rotor in the particle trajectory calculation.

Another important aspect of the propulsion system that could only be approximated was the motors and ESC's behaviour: the manufacturer only provides data obtained in static tests, but no information is available on the internal dynamics of acceleration and deceleration transients. This data could be easily obtained through a series of bench test, that could better describe the electric characteristics of these vital components and provide a more accurate model of the whole motor/propeller assembly.

Regarding the dynamic model, the quad-rotor's mass and moments of inertia were assumed and approximated, respectively. This is because a precise (or even approximate) CAD model wasn't available, as neither was a physical prototype over which measurements could be taken. This is even more relevant as the UAS's mass decreases during the mission, modifying its center of mass' position and the moments of inertia. In order to develop a design of this kind, all components should have been defined, but as this thesis was brought forward in the preliminary design phase, very little information was consolidated enough to start such procedure.

The definitions of the components could also benefit the guidance and control architecture's definition and operations. A realistic sensors and electronics model is in fact very important in the assessment of the system's performance and precision, as the errors introduced by the sensors deeply affect the whole control system ability to accurately drive the vehicle in its mission. Even if this operation of error introduction was partially done in the wind estimation algorithm in chapter 5, the real sensor performance could be radically different and introduce errors in this estimation, on which the whole guidance system relies.

Last, if this quad-copter was enter service and start operating on crops, it would be most effective if part of a fleet of vehicles conducting the same operation. This would of course improve productivity and system availability, making operations possible 24/7, but it would need a dedicated system to optimize all vehicles movements and mission planning.

In conclusion, even if this work well describes the working principle of the system, a lot of work is still to be done in order to perfect and refine it, translating the theoretical principles into reality and maybe, one day, see it operating in a real scenario, an endeavour that surely must be taken by a larger team of engineers, scientists and students.

Bibliography

[1] D. Tsouros, S. Bibi, and P. Sarigiannidis, “A review on uav-based application for precision agriculture,” *Information*, vol. 10, no. 11, pp. 1–26, 2019.

[2] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, “A compilation of UAV applications for precision agriculture,” *Computer Networks*, vol. 172, p. 107148, 2020.

[3] X. He, “Rapid development of unmanned aerial vehicles (uav) for plant protection and application technology in china,” *Outlooks on Pest Management*, vol. 29, no. 4, pp. 162–167, 2018.

[4] [Online]. Available: https://www.easa.europa.eu/document-library/regulations/commission-implementing-regulation-eu-2019947

[5] [Online]. Available: https://www.easa.europa.eu/document-library/regulations/commission-delegated-regulation-eu-2019945

[6] D. Sarri, L. Martelloni, M. Rimediotti, R. Lisci, S. Lombardo, and M. Vieri, “Testing a multi-rotor unmanned aerial vehicle for spray application in high slope terraced vineyard,” *Journal of Agricultural Engineering*, vol. 50, no. 1, pp. 38–47, 2019.

[7] [Online]. Available: https://ec.europa.eu/eip/agriculture/en/publications/eip-agri-focus-group-diseases-and-pests

[8] Y. Zhan, P. Chen, W. Xu, S. Chen, Y. Han, Y. Lan, and G. Wang, “Influence of the downwash airflow distribution characteristics of a plant protection uav on spray deposit distribution,” *Biosystems Engineering*, vol. 216, pp. 32–45, 2022.

[9] Y. Zhang, Y. Li, Y. He, F. Liu, H. Cen, and H. Fang, “Near ground platform development to simulate uav aerial spraying and its spraying test under different conditions,” *Computers and Electronics in Agriculture*, vol. 148, pp. 8–18, 2018.

[10] L. S. v Zong-nan Chen and C. Wen, “Experimental investigation of the flow structure over a delta wing via flow visualization methods,” *Journal of visualized experiments, JoVE*, no. 134, 2018.

[11] S. Wen, J. Han, Z. Ning, Y. Lan, X. Yin, J. Zhang, and Y. Ge, “Numerical analysis and validation of spray distributions disturbed by quad-rotor drone wake at different flight speeds,” *Computers and Electronics in Agriculture*, vol. 166, p. 105036, 2018.

[12] M. E. Teske, D. A. Wachspress, , and H. W. Thistle, “Prediction of aerial spray release from uavs,” *Transactions of the ASABE*, vol. 61, no. 3, pp. 909–918, 2018.

[13] N. Bloise, M. C. Ruiz, D. D’Ambrosio, and G. Guglieri, “Wind tunnel testing of remotely piloted aircraft systems for precision crop-spraying applications,” *2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor). IEEE*, 2021.

[14] [Online]. Available: https://www.aragnet.com/VediMacro/EN/246/products/

[15] S. Aich, C. Ahuja, T. Gupta, and P. Arulmozhivarman, “Analysis of ground effect on multi-rotors,” *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, 2015.

[16] A. Matus-Vargas, G. Rodriguez-Gomez, and J. Martinez-Carranza, “Ground effect on rotorcraft unmanned aerial vehicles: a review,” *Intelligent Service Robotics*, vol. 14, pp. 99–118, 2021.

[17] D. Nuyttens, K. Baetens, M. De Schampheleire, and B. Sonck, “Effect of nozzle type, size and pressure on spray droplet characteristics,” *Biosystems engineering*, vol. 97, no. 3, pp. 333–345, 2007.

[18] W. R. Goossens, “Review of the empirical correlations for the drag coefficient of rigid spheres,” *Powder Technology*, no. 352, pp. 350–359, 2019.

[19] R. Grisso, P. Hipkins, S. Askew, L. Hipkins, and D. Mccal, “Nozzles: Selection and sizing,” *Virginia Cooperative Extension*, vol. 442, 2013.

[20] M. P. Johnson and L. D. Swetnaml, “Sprayer nozzles: Selection and calibration,” *Kentucky Cooperative Extension*, 2011.

[21] C. Jekeli, *Inertial Navigation Systems with Geodetic Applications.* De Gruyter, 2001.

[22] M. Islam, M. Okasha, and M. M. Idres, "Dynamics and control of quadcopter using linear model predictive control approach," *IOP Conference Series: Materials Science and Engineering*, vol. 270, 2017.

[23] P.-S. Tsai, N.-T. Hu, T.-F. Wu, J.-Y. Chen, , and J.-J. Yuan, "Platform for simulating six-rotor unmanned aerial vehicle," *Sensors and Materials*, vol. 33, no. 4, pp. 1313–1332, 2021.

[24] R. Beard, "Quadrotor dynamics and control," *ScholarsArchive*, vol. 1325, 2008.

[25] [Online]. Available: https://store.tmotor.com/goods.php?id=783

[26] [Online]. Available: https://store.tmotor.com/goods.php?id=830

[27] L. Bakule, "Decentralized control: an overview," *Annual Reviews in Control*, vol. 32, pp. 87–98, 2008.

[28] M. Malik and N. Sehgal, "A comparative study of classical and modern controllers," *International Journal of Engineering Research and Technology (IJERT)*, vol. 5, 2017.

[29] 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, 2015.

[30] Y. Song, Q.-H. Meng, B. Luo, M. Zeng, S.-G. Ma, and P.-F. Qi, "A wind estimation method for quadrotors using inertial measurement units," *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2016.

[31] R. T. Palomaki, N. T. Rose, M. van den 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, 2017.

[32] 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*, vol. 8, pp. 54 910–54 927, 2020.

[33] S. Prudden, A. Fisher, M. Marino, A. Mohamed, S. Watkins, and G. Wild, "Measuring wind with small unmanned aircraft systems," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 176, pp. 197–210, 2018.

[34] M. O. Mughal, M. Lynch, F. Yu, B. McGann, F. Jeanneret, and J. Sutton, “Wind modelling, validation and sensitivity study using weather research and forecasting model in complex terrain,” *Environmental Modelling and Software*, vol. 90, pp. 107–125, 2017.

[35] R. L. Roux, M. Katurji, P. Zawar-Reza, H. Quénol, and A. Sturman, “Comparison of statistical and dynamical downscaling results from the wrf model,” *Environmental Modelling and Software*, vol. 100, pp. 67–73, 2018.

[36] G. Forristall, “Wind spectra and gust factors over water,” *Offshore Technology Conference*, 1988.

[37] M. Ochi and Y. Shin, “Wind turbulent spectra for design consideration of offshore structures,” *Offshore Technology Conference*, 1988.

[38] R. I. Harris, “The nature of the wind,” *Proceedings of the seminar held on 18 June 1970 at the Institution of Civil Engineers*, 1970.

[39] T. V. Karman, “Progress in the statistical theory of turbulence,” *PNAS*, vol. 34, no. 11, pp. 530–539, 1948.

[40] L. W. Traub, “Validation of endurance estimates for battery powered uavs,” *The Aeronautical Journal*, vol. 117, pp. 1155–1166, 2013.

[41] G. Imperatore, A. Ghirardelli, L. Strinna, C. Baldoin, A. Pozzebon, G. Zanin, and S. Otto, “Evaluation of a fixed spraying system for phytosanitary treatments in heroic viticulture in north-eastern italy,” *Agriculture*, vol. 11, no. 833, 2021.

[42] I. Pertot, T. Caffi, V. Rossi, L. Mugnai, C. Hoffmann, M. Grando, C. Gary, D. Lafond, C. Duso, D. Thiery, V. Mazzoni, and G. Anfora, “A critical review of plant protection tools for reducing pesticide use on grapevine and new perspectives for the implementation of ipm in viticulture,” *Crop Protection*, vol. 11, pp. 70–84, 2017.

[43] L. Strub and S. Loose, “The cost disadvantage of steep slope viticulture and strategies for its preservation,” *OENO One*, vol. 55, no. 1, 2021.

[44] E. Pomarici, A. Corsi, S. Mazzarino, and R. Sardone, “The italian wine sector: Evolution, structure, competitiveness and future challenges of an enduring leader,” *Italian Economic Journal*, vol. 7, pp. 259–295, 2021.