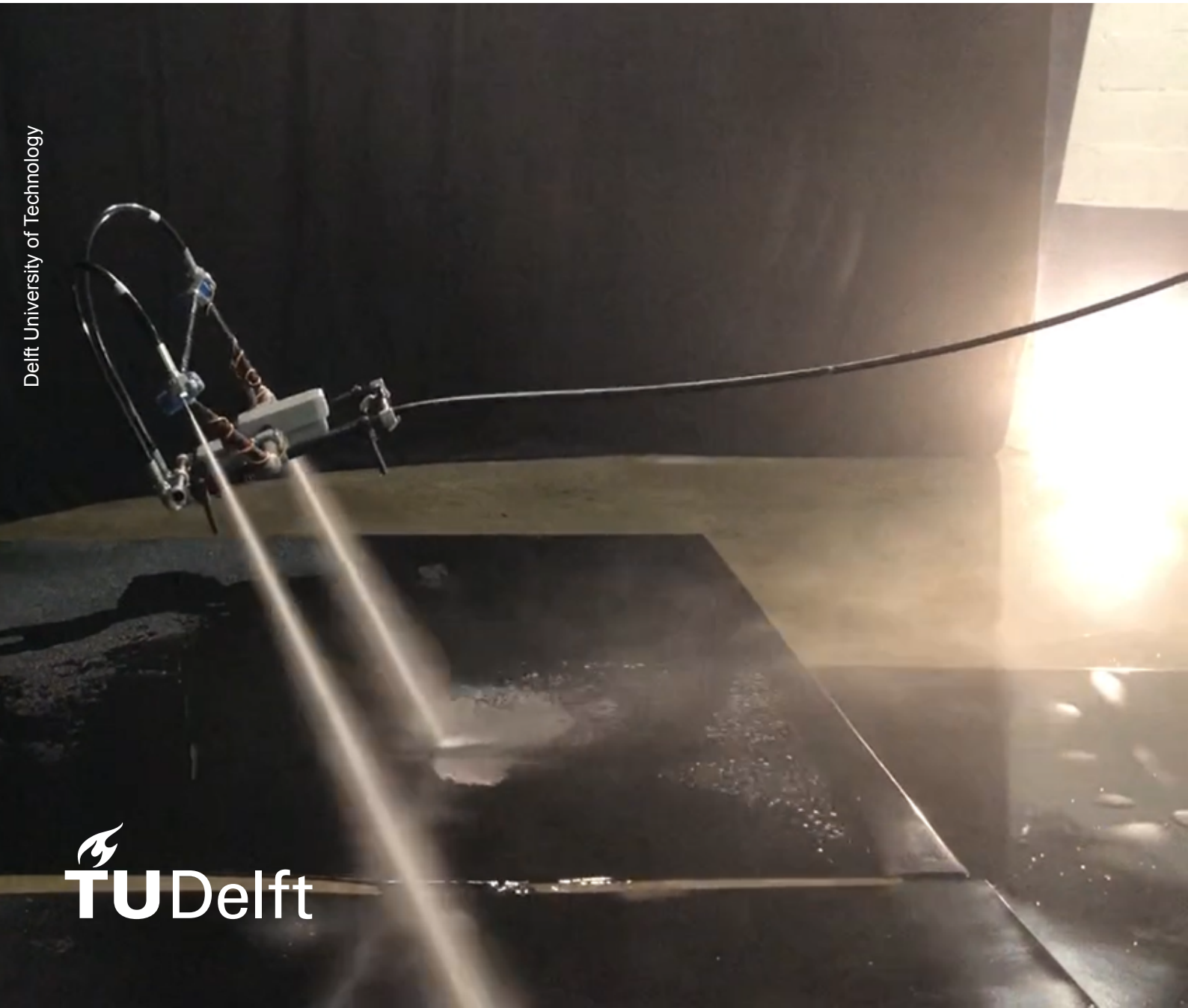
Stability control and positional water jet placement for a novel tethered unmanned hydro-propelled aerial vehicle using real-time water jet detection

Thesis Report

Xander F. van Beurden



Delft University of Technology

TU Delft

Stability control and positional water jet placement for a novel tethered unmanned hydro-propelled aerial vehicle using real-time water jet detection

by

Xander F. van Beurden

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at Delft University of Technology,
to be defended publicly on Thursday February 2, 2023 at 13:00.

Student number:	4377001	
Project Duration:	January, 2022 - February, 2022	
Thesis committee:	Prof. dr. G.C.H.E. de Croon,	TU Delft, Chairman
	Dr. ir. C. De Wagter,	TU Delft, Supervisor
	ir. O.L. van Dijk,	Actiflow, Supervisor
	Dr. A. Menicucci,	TU Delft, External examiner

Cover Image: Hydro-propelled drone during test flight

TU Delft

Preface

First of all, I would like to thank my supervisors, Christophe de Wagter and Oskar van Dijk for their guidance in conducting this research. It was a pleasure to work closely together with my supervisors and brainstorm about possible solutions to the various problems that were encountered along the way. They have allowed me to come up with creative solutions and explore problems on my own. Their constructive feedback significantly improved the overall quality of my work.

Next, I want to express my gratitude to Actiflow for allowing me to conduct this research for their company. It has taught me that as a control systems engineer, it is essential to consider various aspects of a project to achieve success. Finding the optimal solution within the limitations of budget, time, performance, and complexity can be challenging. Effective communication with stakeholders and consideration of all factors is crucial.

However, the journey was challenging. The dynamics of a hydro-propelled drone, PX4 flight stack, electronics, and airframe assembly were new to me, so patience was necessary. As I became more familiar with them, the project progressed more quickly. The design trade-off allowed me to gain new insights, and I was able to expand the scope of the research by adding position control to the system. I enjoyed developing the drone and working at the drone arena in Breda. I appreciated the coffee morning routine with Oskar, who was my sparring partner and provided technical assistance with the assembly and repairs of the drone.

I want to express my appreciation to all of my friends, whom I met at the faculty and through extracurricular activities, for their role in making my time at TU Delft fruitful. I am particularly grateful to my family for their unwavering support throughout my studies.

Xander F. van Beurden
Delft, January 2023

Contents

Nomenclature

List of Abbreviations

ADRF	Active disturbance rejection framework
API	Application programming interface
BEC	Battery eliminator circuit
CCW	Counterclockwise
CW	Clockwise
D	Derivative
DDC	Disturbance decoupling control
DoF	Degrees of freedom
FFT	Fast Fourier Transform
FMU	Flight management unit
fps	Frames per second
GPS	Global positioning system
HSV	Hue, saturation, value
I	Integral
IMC	Internal model control
LK	Lucas-Kanade
LQR	Linear quadratic regulator
MoCap	Motion capture
MPC	Model predictive control
NDI	Nonlinear dynamic inversion
P	Proportional
PFL	Partial feedback linearisation
PID	Proportional, integral, derivative
PMB	Power management board
PWM	Pulse-width modulation
PX4	Pixhawk 4
QGC	QGroundControl
R-CNN	Region-based convolutional neural network
RGB	Red, green, blue
ROS	Robot operating system
SMC	Sliding mode control
STOL	Shortrun take-off and landing
TLX	Task loading index
UAV	Unmanned aerial vehicle
UHAV	Unmanned hydro-propelled aerial vehicle
UI	User interface
UWAV	Unmanned water-propelled aerial vehicle
VIO	Visual inertial odometry
VTOL	Vertical take-off and landing
YOLO	You Only Look Once

List of Symbols

(X, Y, Z)	Set of unit vectors in the pressure frame	
(x, y, z)	Set of unit vectors in the body frame	
α	Azimuth angle	[rad]
α_v	Viewing angle of the camera	[rad]
β	Servo bias	[rad]
δ_n	Nozzle eflection	[rad]
$\dot{m}$	Mass flow	[kg/s]
η	Bank-yaw coupling ratio	
η	Percentage overshoot	[%]
$\hat{x}$	Estimation of variable x	
R	Rotation matrix	
μ	Bank angle	[rad]
ν	Yaw angle	[rad]
ω_c	Cutoff frequency	[Hz]
ϕ	Roll angle (Euler)	[rad]
ψ	Yaw angle (Euler)	[rad]
ρ	Density	[kg/m^3]
σ	Full deflection angle of servo	[rad]
τ	Torque	[Nm]
θ	Pitch angle (Euler)	[rad]
ζ	Damping ratio	
A	Cross-sectional area	[m^2]
a	Acceleration	[m/s^2]
a	Catenary constant	[N/kg^1]
C	Actuator command	
d_d	Horizontal distance between camera and drone	[m]
F_x	Force in the x-direction	[N]
F_y	Force in the y-direction	[N]
F_z	Force in the z-direction	[N]

FF	Feedforward gain	
h	Horizontal distance	[m]
$H(S)$	Transfer function in the Laplace domain	
I_{XX}	Moment of inertia around the X-axis	[kg m^2]
I_{YY}	Moment of inertia around the Y-axis	[kg m^2]
I_{ZZ}	Moment of inertia around the Z-axis	[kg m^2]
K	Gain	
L	Length of the catenary	[m]
M_n	Moment of the nozzle	[Nm]
mix_s	Mix command for servo 1 and 2	
$P(x)$	Command of x in percentage	[%]
P_1	X, Y, Z location of point $P1$	[m]
P_2	X, Y, Z location of point $P2$	[m]
R	Rotation matrix	
s	Operator variable in the Laplace domain	
T	Tension force of the hose	[N]
T	Thrust	[N]
T_r	Rise time	[s]
T_s	Settling time	[s]
T_{up}	Upward thrust	[N]
V	Velocity	[m/s]
v	Vertical distance	[m]
V_e	Exit velocity	[m/s]
V_i	Initial velocity	[m/s]
w_f	Width of the computer frame	[m]
X'	Normalised X-location inside the computer vision frame	
Z'	Normalised Z-location inside the computer vision frame	
$^\circ$	Degree	

Coordinate Frame

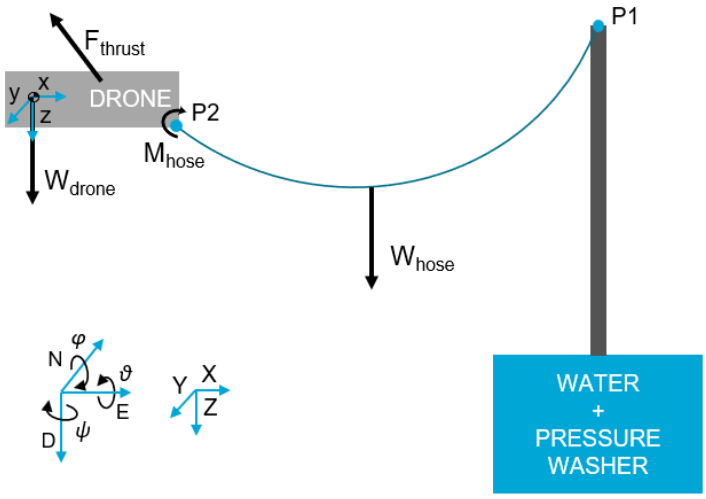


Figure 1: Schematic side view of the tethered hydro-propelled aerial system showing the coordinate frame definitions.

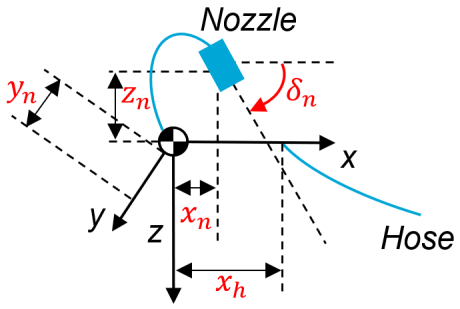


Figure 2: Schematic overview of the drone including the sign conventions.

List of Figures

List of Tables

Part I

Introduction & Literature Review [graded as AE4020]

1

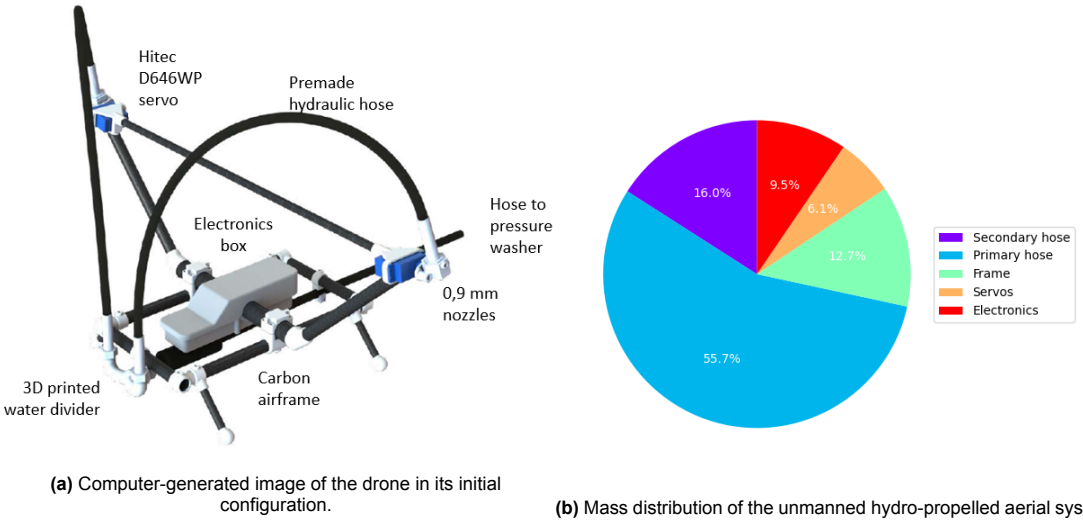
Introduction

Drones are becoming an increasingly integrated part in our lives while drone technology evolves rapidly. A trend can be seen where drones are becoming smaller, lighter, more efficient, and cheaper while being used for a wide range of purposes. Many applications exist where drones are used such as to detect gas leaks and provide rapid medical aid (e.g. Zipline[1]), or gathering intelligence. Depending on the size, propulsion, flight configuration, level of autonomy, and payload, drones are used for different applications (Vergouw et al., 2016). A client has reached out to Actiflow with the objective to design, build, and test a drone that is able to spray water at a predefined location that meets the following requirements:

1. Drone uses water jets as the propulsion mechanism
2. Drone must be able to operate at a minimum distance of 50m from the launch site
3. Drone must effectively deliver 15L of water per minute
4. Drone must be able to deliver water systematically in an area of 20m x 20m
5. On-board materials and systems must be able to withstand heat coming from fire and electronics may not become hotter than 80°C when flying close to a fire
6. Drone must be able to come down safely in case of damage or actuator failure
7. No injuries or harm may be caused by the water jets

While the above specifications are still under debate and may be modified, removed, or complemented, this is the starting point for the drone design. The system, shown in Figure 1.1, is provided by Actiflow, together with a limited simulation model. The initial design is showcased in Figure 1.1a and its mass distribution along with the primary hose is given in Figure 1.1b. Although the use case for the drone is still under consideration, the aim of the research is to build an autonomously operating drone that adheres to the specifications mentioned in this section by building the flight software for the current platform designed by Actiflow. It is expected that together with the flight controller software, the design of the drone must also be updated in order to have an improved flight performance.

[1] Zipline is a worldwide fixed-wing drone delivery company that started in Africa to provide rapid medical aid. https://www.flyzipline.com/



(a) Computer-generated image of the drone in its initial configuration.

(b) Mass distribution of the unmanned hydro-propelled aerial system.

Figure 1.1: Illustration of the drone and the mass distribution of the system.

1.1. Thesis Objective and Research Questions

The main research objective is ”to design, build, and test a tethered and hydro-propelled UAV that is able to safely fly throughout its entire flight envelope in the presence of disturbances”. As the mission is yet to be defined, the research is split up into several sub-objectives: (i) modify the flight code and attitude controller to have the most stable flight possible in the current flight configuration, (ii) come up with a design iteration in hard- and/ or software to improve flight performance, and (iii) the attitude controller can withstand disturbances and fly safely throughout its entire flight envelope.

Reaching the research objectives will lead to new insights into tethered hydro-propelled unmanned aerial vehicles (UHAVs) in terms of dynamics and (robust) control. This will aid future research in the design, safety, controllability, cost reduction, and autonomy of hydro-propelled drones that can be used for e.g. leisure activities, fire detection and suppression, and washing of large surfaces.

The following research question is linked to this objective:
”Is a hydro-propelled UAV capable of steady, controlled flight and reject external and tether disturbances, while being cost-efficient and safe?”

This research question can be split up into multiple sub-questions to make it specific, measurable, achievable, relevant, and timed, which together answer the main research question.

Q1: What is the dynamic behaviour of the drone due to bank and yaw coupling in lateral attitude control?

(a) Can the design of the drone be adjusted to limit the (dominant) yawing motion during lateral stability control?
(b) How can the controller be optimised to deal with bank and yaw coupling?

Q2: What is the influence of the three flexible water hoses on the flight dynamics and can it be limited or dealt with?

(a) Does the primary hose act as a natural damper?
(b) Can the attachment point of the primary hose be adjusted for improved performance?
(c) How can the controller deal with oscillations and eigenfrequencies of the primary hose?
(d) How large is the impact of the non-minimum phase effect of the two secondary flexible hoses on the attitude control?
(e) How can this non-minimum phase effect be reduced through a mechanical solution?
(f) How can this non-minimum phase effect be reduced by flight software?

Q3: In terms of cost efficiency, what is the minimum number of actuators and moving parts to operate the drone?

(a) How many controlled degrees of freedom (DoF) does the drone in its current configuration have?
(b) What is the flight envelope of the current system?
(c) How can the flight envelope and DoF be increased by adjusting the design of the drone?

Q4: Which safety measures have to be undertaken to safely operate the drone in case a malfunction appears?

(a) What is the effect of a nozzle breaking down?
(b) What is the effect of an air bubble insight the hose?
(c) In what conditions is the drone able to safely fly without tipping over?
(d) What damage can the water jets cause to humans, animals, or the environment?

1.2. Research Approach

As a starting point of the research, a drone was given with a set flight configuration. Parameters such as the number of controllable nozzles, weight (distribution), dimensions, center of mass, and flight control platform were given. The first phase focused on stable flight with the current system by adjusting the flight code and the proportional-integral-derivative (PID) controller provided by the Pixhwak 4 (PX4) open-source platform[2]. By investigating studies about the dynamics of tethered quadrotors (Goodarzi et al., 2014; Kotaru & Sreenath, 2020; T. Lee et al., 2007; T. Lee, 2015; T. Lee et al., 2018), it was concluded that modelling the dynamics of the entire system in simulation was too expensive for this research. As the drone is constructed from small carbon rods and 3D printed joints, it was expected that time spent on a validated simulation model is more expensive than extensive flight testing, while also eliminating the problem of the reality gap in research (de Croon & de Wagter, 2021). Therefore, the correct PID model was implemented into the flight controller, the gains were tuned empirically, and the log files were studied. Several main adjustments were made concerning the flight code. Firstly, because the drone's pitch angle is defined by its 3D location due to the tether weight, this angle is not controlled by the controller. In addition, when hovering at high pitch angles, the roll angle must be redefined for lateral stability. The angle between the axis connecting the two nozzles and the horizon is defined as the bank angle and the drone controls the bank angle rather than the roll angle. Secondly, the nozzle effectiveness was taken into account based on its angle relative to the Earth's horizon. Through initial investigation of the dynamics of the system, it was expected that the current flight configuration is difficult to control. The bank and yaw coupling reduces the controlled degrees of freedom of the system. Disturbances due to wind gusts or an oscillating water hose have to be dealt with by the controller.

Additionally, the weight of the two secondary flexible hoses that distributes the water to both nozzles, is mainly above the rotation axis of the servos. When deflecting the nozzles forward to increase the pitch angle and prevent the drone from tipping over, the weight above the axis of rotation will cause the system to initially rotate pitch down, increasing the chance of tipping over. As the two flexible hoses contribute to 22% of the total weight of the drone, it was expected that the controller must take this non-minimum phase effect into account.

Combining the two hypotheses will limit the freedom of the drone and reduce flight performance. It was expected that after testing these effects, the design of the drone had to be modified. After proposing several design options to the stakeholders a final design was chosen, built and tested. In addition, based on the literature review in Chapter 2, the area of robust control for coupling problems was exploited when the hypothesis would be confirmed that the coupling between yaw and bank imposes attitude stabilisation problems.

1.3. Experimental Setup

As mentioned in Section 1.2, the method of testing was by flight tests rather than running simulations. A drone arena has been built by Actiflow that has 360° water screens and a floor drain. In addition, the materials and electronics were provided by the client and are given in the following list:

[2] https://github.com/PX4/PX4-Autopilot | Accessed: 31-07-2022

- Adjustable airframe composed of carbon rods and 3D printed joints
- Kranzle 1150 T high pressure washer
- Pixhawk 4 Mini
- FrSky Taranis remote controller and its receiver
- Lechler nozzles T25
- D646WP Hitec servos
- ESP8266 Wi-Fi telemetry
- Garmin LIDAR-Lite v3
- 2 Cell Lipo batteries
- Necessary power converters and electricity cables
- A ground station

It was measured that the high-pressure washer in combination with the nozzles provided 20N of thrust which means that the current system is able to lift a platform of ± 2 kg when the nozzles are in the vertical position. This reduces to 1.41 kg when sitting at a 45° angle, limiting the weight of the drone to <1.4kg because part of the hose also had to be carried. The water pump may be replaced by a pump with more power at a later stage of the project when the performance of the drone is sufficient. However, the focus on the test phase remains on the current system setup.

1.4. Hard- and Software

The hardware used for this research is the Pixhawk 4 Mini, developed by Dronecode, a Linux Foundation non-profit. This is an open-source flight control platform for drone developers which provides a flexible set of tools for drone developers to create tailored applications for drones[3]. The motivation to use this platform is because the hardware is relatively cheap, the software is free, there is a large and active community of developers, it supports a large variety of compatible hard- and software, and there is a well-documented developer's guide.

1.5. Contribution

In the long run, this research will contribute to the field of autonomous water-jet propelled UAVs that can have various functionalities such as fire suppressing, cleaning, or the attitude control mechanism can contribute to water propelled underwater vehicles for surveillance, detection, or lifting a drowning person out of the water.

1.6. Structure

The report is divided into three parts. Part I covers the introduction, followed by the literature review in Chapter 2 where the research opportunities are researched and relevant research is assessed. During the first phase, new insights were gained such that the scope of the research had to be adjusted. The adjustments are covered in Chapter 3. In Part II the scientific paper is presented which discusses the main findings and conclusions of the research. Finally, in Part III, additional work is presented. Chapter 4 gives a quick update on how the data was analysed from the log files. The general controller architecture of the flight stack and the custom controller validation is discussed in Chapter 5. The software implementation is touched upon in Chapter 6 while the hardware implementation is discussed in Chapter 7. Position control and computer-vision is discussed in detail in Chapter 8. A deepdive into the dynamics of the system is discussed in Chapter 9, which was a setup towards a simulation model of the system. Finally, this work is concluded in Chapter 10.

[3] https://px4.io/ | Accessed: 08-03-2022

2

Literature Review

The aim of the literature review is to orient in which field of drone research this research fits and to search for a research opportunity within this field. In addition, all relevant previous literature may be used to the advantage of this research. First, the different types of drones are discussed in Section 2.1, followed by an overview and the characteristics of commercially available jet packs in Section 2.2. Fire-extinguishing UAVs are covered in Section 2.3. Next, a deepdive into the dynamics of the tether, thrust-vectoring mechanism, and flight control is given in Section 2.4, Section 2.5, and Section 2.6, respectively.

2.1. Drone Types

Drones are flying robots that are able to fly thousands of kilometres or navigate in confined spaces that are remotely controlled or rely on autonomous flight software, also known as unmanned aerial vehicles (UAVs) (Cavoukian, 2012; Krijnen & Dekker, 2014). UAVs have seen rapid popularity due to their wide applications in both military and civil settings which stimulates research and development (Elijah et al., 2021). According to Vergouw et al., 2016, drones can be differentiated into four categories: (i) the type of drone, (ii) the degree of autonomy, (iii) the size and weight, and (iv) the power source. When designing a drone, these specifications are important to consider as the application of the drone drives the characteristics of the platform. To narrow the scope of research, it is important to study the type of drone subject to this research.

2.1.1. Main Drone Types

Types of drones can be differentiated upon their flight configuration. Fixed-wing systems have an airplane-like configuration where fixed wings generate lift through forward flight. Fixed-wing type drones can efficiently fly long ranges at high speeds, but cannot hover and need a start and landing strip. Another type of widely used drones are multirotor systems. In contrast to the traditional helicopter, drones are usually equipped with multiple smaller rotors necessary for their stability and simplicity in control. This case, the rotors do not need variable pitch angle which is complex and more prone to failures (H. Yang et al., 2017). In contrast to fixed-wings, multirotors can vertically take-off and land, and hover. However, their flight range and airspeed is limited, and due to rotational and translational coupling it is an underactuated system. Fully actuated multirotors are currently researched, but their rotor configuration is application-dependent (Rashad et al., 2020). In addition, VTOL (vertical take-off and landing) or STOL (shortrun take-off and landing) are aircraft that are able take-off and land without or with a very short runway and transition in mid-air to a fixed-wing type aircraft (Intwala & Parikh, 2015). Other systems like ornithopters fly using flapping wings such as the DelFly (De Croon et al., 2016). More interesting for this research are drones using (hydro-)jets and thrust-vectoring mechanisms such

as Zapata's UHAV[1] or Honeywell's T-Hawk (Agbeyangi et al., 2016). Finally, lighter than air systems exist that are lifted by light gasses or heating of air. Many drone types are underactuated systems and Section 2.6 will elaborate on the classification and control of underactuated systems.

2.1.2. Level of Autonomy

During the design of a drone, it is important to study the level of autonomy that needs to be designed. Depending on the use-case of the drone, the level of autonomy and the human-drone interaction are important factors that influence the physical and control system design (Christ et al., 2016). An increasing level of autonomy influences e.g. the number of sensors onboard, robustness of the flight software, control interface, energy consumption, and cost. Research has shown that based on NASA's Task Load Index (TLX), an increasing level of autonomy decreases the perceived workload of the operator (Hart & Staveland, 1988; Steinfeld et al., 2006). These are all factors that need to be considered in the design of a UAV. According to the United States Department of Defense, there are four levels of autonomy for unmanned systems Huang et al., 2008. At the first level of autonomy, the human operator makes all the decisions and the system does not have control over its environment. The second level is a human delegated system where the system can perform functions independent of the human operator. Engine and automatic controls that are activated by a human controller are examples. The third level of autonomy is a human supervised system. Both the system itself and the human supervisor can initiate actions based on sensed data. The fourth and final level of autonomy is a fully autonomous system where commands are translated into tasks without human interaction. A more technical and descriptive approach is taken by (Johnson et al., 2020) in categorising the level of autonomy. The first category is defined as ”Manual control with stability augmentation”. That is, feedback of the aircraft motion through its sensors is added to the pilot input to enhance the flying quality of the aircraft. The following level of autonomy is defined as ”Navigation and trajectory following” which uses satellite navigation to follow a prescribed course. ”Path planning” is the third category that generates its trajectory based on information derived onboard to avoid e.g. obstacles using radar, lasers, or cameras. The final category according to the authors is ”collision avoidance and traffic management” which is an active area of research where UAVs may encounter or collaborate with other vehicles.

Finally, Floreano and Wood, 2015 identifies three levels of autonomy. The first is ”sensory-motor autonomy” where high-level human commands are translated into combinations of control signals. An example is to follow a pre-programmed trajectory using Global Positioning System (GPS) waypoints. The second level is ”reactive autonomy” and requires the system to deal with perturbations such as wind gusts or electro-mechanical failure and avoid obstacles and other vehicles. For the final level of autonomy, ”cognitive autonomy”, the drone can learn, recognise patterns, areas of people, resolve conflicting information, and plan for e.g. battery recharge.

2.1.3. Size and Weight

The size and weight is another important parameter to categorise drones. Research and development focuses on the creating smaller and lighter drones, capable of the same tasks whereas larger drones are used for military or industrial purposes (Vergouw et al., 2016). Large fixed-wing drones are in the range of 20-150kg and 25-100kg for multirotors. Small drones are below the threshold of 20 and 25 kg, respectively. Mini drones are in the range of grams up to several kilograms. While Australia categorises aerial vehicles in micro (<250g), very small (250g-2kg), small (2-25kg), medium (25-150kg), and large (>150kg) drones[2]. As the size of the drone decreases, the moment of inertia is also scaled down considerably leading to higher manoeuvrability (Kumar & Michael, 2012). However, this enhanced agility comes at the expense of open-loop stability and requires increased control. This, in turn leads to higher on-board sensing and computation which affects weight, size, and flight time (Floreano & Wood, 2015).

[1] https://www.zapata.com | Accessed: 25-07-2022

[2] https://www.casa.gov.au/drones/drone-rules/drone-safety-rules/types-drones | Accessed: 29-07-2022

2.1.4. Energy Source

The final categorisation is based on the energy source. According to Vergouw et al., 2016, the energy source can be battery cells, solar cells, fuel cells, and (airplane) fuel. Airplane fuel (kerosene) is mainly used in large drones, while battery cells are common in micro to small drones due to the many advantages such as efficiency, precise control, reliability, and reduced noise (Donateo et al., 2017). Fuel cells are rarely used in drones because of the low energy density and also solar cells are rarely used in the drone industry because of their low energy conversion efficiency and high weight. Battery-powered drones are limited in terms of flight time and therefore, multiple solutions are being researched to increase the flight time. Boukoberine et al., 2019 reviews different alternatives to increase flight time which include hybrid powered drones combining the different energy sources, laser-beam recharging, and tethered drones. The latter provides an interesting field of study as the UHAV will be a tethered drone.

The drone can be classified as a tethered, hydro-jet propelled UAV with a ”manual control with stability augmentation” level of autonomy. The size and weight is very small as the drone initially is designed to be no heavier than 1.5kg, and the energy source is an electric water pump and onboard battery cells for flight control and actuation. When the prototype is tested, the system will be upgraded to a small device in the order of several kilograms.

2.2. Overview of Jet Pack Development

A jet pack is a device that is worn on the back which provides thrust through a jet to lift the operator. According to the Oxford English Dictionary, ”a jet is a rapid stream of liquid or gas forced out of a small opening”. Although jets are widely used for multiple purposes, the technique behind a jet propelled system is hundreds of years old. The first known application of a jet propelled system dates back to the year 62 where Hero of Alexandria invented the Aeolipile, where a jet of steam is used to spin a sphere (Geatrix, 2012). This principle was first mathematically described by Sir Isaac Newton as Newton’s third law of motion, stating ”that for every action, there is an equal and opposite reaction” (Geatrix, 2012). The first jet propulsion flight systems were developed around the 1930s during the time the jet engine was invented (Henry et al., 1995). A mass is accelerated such that a reaction force is created, known as thrust. Thrust acts in a linear direction and thrust vectoring is the act of adjusting the direction of thrust (Geatrix, 2012). This section will outline the commercially available jet packs that use water or air as the propellant in Section 2.2.1 and Section 2.2.2, respectively.

2.2.1. Water

Hydro-propelled flight systems is a subcategory of jet propelled flight systems. Water is accelerated to provide thrust, which has gained attention in the last decade, especially in the recreational sector. Manufacturers of hydro-propelled flight systems are, amongst others, Jetlev-Flyer, Zapata Racing, and Stratospheric Industries. The idea originated from Raymond Li who came up with the idea to exchange hydrogen or nitrogen for water as the propellant. By pushing the water up through a hose, two nozzles adjust the direction of the water downwards to push the weight up. The Jetlev-Flyer is controlled via arm movements up and down and body movements left and right (Jetpack America, 2016). Since Jetlev launched the first commercial hydro-propelled jet pack, many systems took flight. The physics behind the water jet packs is explained by Vonk and Bohacek, 2013, however the control system of the water jet packs remains relatively undiscussed in the literature.

Control of the hydro-propelled jet packs is mainly done by human operators but the flight configuration varies between manufacturers as well as within their products. As mentioned before, with the human operators, weight distribution of the pilot is used for attitude control and in addition to weight, thrust vectoring mechanism can be added. The altitude is controlled using a thrust regulator that sits either on the hydro-pump or jet ski or can be handheld by the pilot. Two examples of such flight systems from two different companies with varying flight configurations are shown in Figure 2.1.

(a) Zapata Flyboard [3]. **(b)** Jetlev-Flyer [4].

Figure 2.1: Two hydro-propelled human flying systems produced by different companies.

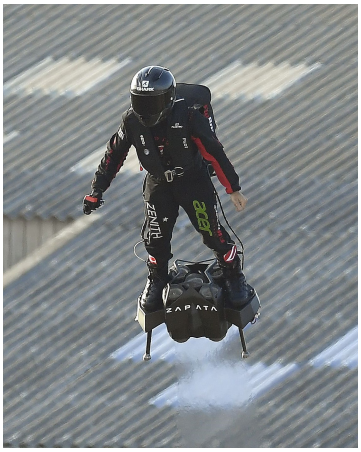
The Zapata Flyboard and the Jetlev-Flyer differ in flight configuration. In Figure 2.1a, it can be seen that the centre of mass is above the propulsion system, there is no thrust vectoring so the attitude is controlled solely by weight distribution of the pilot and the thrust. Figure 2.1b shows the Jetlev-Flyer which has the centre of mass located below the nozzles and the direction of thrust can be twisted around the horizontal axis using the hand grips. This adds an extra attitude controller, besides weight distribution of the pilot and the thrust. Although very little data exist and literature is limited due to the commercial interest to keep it undisclosed, it is expected that the Jetlev-Flyer is a more controllable flight system than the Zapata because of the thrust vectoring mechanism and its weight distribution. As human flight on platforms requires a lot of training to avoid injuries or dangerous situations, Gerken and Bright, 2017 designed a semi-autonomous hydro-propelled jet pack to reduce the skills needed for flight via an onboard thrust controller. It was concluded that the problem with the significant system lag between the water pump and the nozzles can be reduced by using model predictive control (MPC) in contrast to linear quadratic optimal control or a proportional-integral-derivative (PID) controller with very limited predictive power.

Furthermore, Zapata has launched, besides several different human-piloted hydro-propelled drones, the Aquadrone that can carry 100kg of payload and is able to operate autonomously. The drone has 4 nozzles in a '+' configuration that can rotate around a single axis, giving attitude control[5]. Again, limited information is available about the flight performance of the Aquadrone but a Youtube video [6] shows that the drone is manoeuvrable and stable during flight. Although the drone is tethered via a water hose, autonomous flight is limited to one hour and it is assumed that this limitation is due the limited power for the onboard flight controller.

2.2.2. Air

Besides hydro-propelled flight systems, jet packs flying on air have recently also taken flight. While Zapata produced the Flyboard Air and the EZ-FLY, Gravity Industries Ltd introduced the Jet Suit, and Yves Rossy founded the Jetman project. All projects have in common that flight control is handled with thrust vectoring and weight control while all three designs have a different design philosophy and hence the various systems and flight configurations as can be seen in Figure 2.2. FusionFlight designed a remote controlled jet engine drone with four thrust vectoring nozzles (FusionFlight, 2022). FusionFlight is currently working on manned applications of the JetQuad [7]. One of the main differences between the air- and hydro-propelled flight systems is the tether. The tether limits the system by the length of the hose while it highly increases its flight time as the propellant is provided by the tether.

[3] www.eurocfd.fr | Accessed: 27-06-2022
[4] www.jetlev-flyer.com | Accessed: 27-06-2022
[5] https://www.zapata.com | Accessed: 25-07-2022
[6] https://www.youtube.com/watch?v=TrJ9ugDSIlk - Time: 1.45 | Accessed: 27-06-2022
[7] https://fusionflight.com/jetquad/ | Accessed: 31-07-2022



(a) Zapata's Flyboard Air [8].

(b) Gravity Industries' Jet Suit [9].

(c) Yves Rossy's Jetman [10].

Figure 2.2: Three air propelled jet packs/ suits with different flight configurations.

2.3. Fire-Extinguishing UAVs

Water-propelled vehicles can be used for many different applications besides recreational use. One of the popular fields of research which accelerates the development of hydro-propelled UAVs, is the use of hydro-propelled flight systems against fire hazards. Dubai has showcased with the Dolphin that a Flyboard can aid firefighters with extinguishing fires near shorelines or boats[11]. While humans can be carried towards a fire, often fires are inaccessible or dangerous to humans resulting in an interesting field of research towards remote controlled or autonomous firefighting. Liljeback et al., 2006 propose the water hydraulic snake robot, SnakeFighter, that moves on the ground inspired by snake movement for fire intervention tasks. Spurny et al., 2021 present a novel approach for a quadcopter to extinguish fires by controlling and estimating the MAV state, autonomously finding building entrances, localising the fire, and discusses how to extinguish it. Ando et al., 2018 demonstrate a novel design of a hose-type robot which can fly directly into the fire while being propelled by water jets. The nozzle outlet direction of two of the four nozzles can be controlled towards the fire. In Ando et al., 2019; Yamaguchi et al., 2019 this design is iterated where mechanical suppression of tether vibrations and fire-fighting strategies are analysed. In D.-H. Lee et al., 2021 another firefighting aerial vehicle is designed and modelled to prevent firefighters from approaching dangerous fires. The aerial vehicle is attached to a flexible water hose, actuated by four water jets, while the fire can be suppressed by an additional 1-DoF controllable nozzle on top. While above aerial vehicles are specifically designed for the use-case of extinguishing or suppressing fires, X. Liu and Zhou, 2019 focus on the theory behind attitude control of water-propelled drones and proposes two actuating mechanisms which are applied to different unmanned water aerial vehicles (UWAVs) which is discussed in more detail in Section 2.4, Section 2.5, and Section 2.6.

2.4. Tether

As mentioned in Section 2.2.2, one of the main limitations of untethered UAVs is the limited flight time. In recent years, tethered UAV's have been designed to cope with this issue where energy can be supplied (Fagiano, 2017). While tethers can have a useful purpose, it also imposes complexities. A distinction in literature is made between systems with taut and loose tethers. Dynamics of loose cables are described by Winget and Huston, 1976 and flow-structure interaction in a flexible hose is studied by Xie et al., 2016 and Hyvärinen et al., 2020. Fechner et al., 2015 and Dunker, 2018 discuss the dynamics and the physics of tether line drag on kite power systems.

[8] www.theverge.com | Accessed: 27-06-2022
[9] www.gravity.co | Accessed: 27-06-2022
[10] www.yvesrossy.com | Accessed: 27-06-2022
[11] https://www.bbc.com/news/av/world-middle-east-38711494 | Accessed: 25-07-2022

Ioppo, 2017 compares stability and positional dynamics of a quadcopter connected to a taut and loose tether, to conclude that the system was in both cases notably robust to disturbances due to the well actuated nature of the quadcopter while a vertical taut tether imposed the most stable flight. Klinker, 2016 studies a tethered quadrotor with a spherical position PID controller as opposed to the conventional Cartesian position controller. The spherical position controller uses tether dynamics as feedforward control model for which the controller consumed less energy.

Modelling the dynamics and control of a tethered quadrotor or hanging payload has been done in (T. Lee, 2015) and (Goodarzi et al., 2014), respectively. The tether is divided into n number of links of length l with a point mass and ball-and-socket joints at both ends and modelled using Lagrangian mechanics (T. Lee et al., 2007; T. Lee et al., 2018). A drone carrying a water hose with water flow and pressure forces is unprecedented and stated as a recommendation for future work by Kotaru and Sreenath, 2020. In Ando et al., 2018, the hose is also modelled as a chain of n number of links connected through elastic and joints and dampers that are able to bend along three axes. D.-H. Lee et al., 2021 is limited in its simulation results by not taking dynamics into account.

In X. Liu and Zhou, 2019, a tether rising from underwater is analysed which is attached to the drone's center of mass. The force of the hose acting on the drone is defined as an interference force. Furthermore, there is an interference torque imposed by the hose and the Coriolis effect of the water inside the airframe that causes an additional torque. As these forces and torques cannot be expressed explicitly by simple formulas, physics concerning the tether are simplified and disturbances such as vibrations are recommended to be handled by the controller as disturbances. With an increasing altitude, the thrust of the system decreases and the weight carried by the drone increases. This creates a balance between throttle of the pump and altitude. The article shows furthermore that the underwater part of the hose acts as a natural damper for the vertical vibration of the system. Taking these two phenomena into account, an open-loop altitude controller suffices for said system. As altitude is regulated by modifying the water pump's pressure, the benefit of open-loop altitude control is that the controller does not need to deal with the delay between an increase in throttle and an increase in thrust due to the length of the hose. A closed-loop system should therefore be able to handle this system lag. The article assumes that the hose, attached to the center of mass of the drone, always points downwards and therefore does not contribute to the lateral forces and torques acting on the drone.

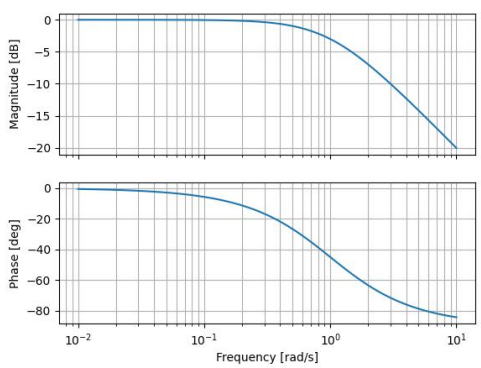
Vibrations Vibrations and dynamic chaos is undesired in mechanical systems as they cause disturbance, discomfort, damage, and possible destruction of the system. Therefore, dealing with vibrations is an important part of the design of mechanical systems (Eissa & Sayed, 2006). It is expected that vibrations will occur in the loose tether that the controller has to deal with. One end of the tether is fixed (P1) while the other end is connected to the drone (P2). As the mass distribution of the system, as displayed in Figure 1.1b, shows that the weight of the hose is a significant part of the total mass of the system, one can imagine that vibrations in the hose will play a vital role in the dynamics of the system. The importance of vibrations are once more stressed when the drone has to act on the end of a tether 2-5 times its current length.

Traditionally, there are two ways to minimise the effect of residual vibrations: (i) redesign the dynamic characteristics of the system to obtain acceptable dynamics (passive vibration suppression) or (ii) using a closed-loop control method (active vibration suppression) (Economou et al., 2001). The first method is preferred and should always be exploited thoroughly as a mechanical solution to vibrations is usually more robust. The design could be altered such that natural damping of the hose vibrations occurs. Yamaguchi et al., 2019 proposes a renewed design of a flying robot with water jets with a mechanical solution to suppress oscillation of the hose. By using wires along the sides of the hose, passive damping of the hose occurred. However, when a vibration in the mechanical system remains, filters can be used to remove some frequencies to suppress interfering frequencies and reduce noise, reduce aliasing and calm resonances. This research will focus on the methods to calm resonances as this might be caused when the controller reacts on the vibration of the tether. The filters are often used in combination with a PID controller to obtain a robust controller. Four common filters are the low- and high-pass, notch, and bandpass filters. As low-pass and notch filters are most common in control theory, this literature review will focus on those two filters (Ellis, 2012).

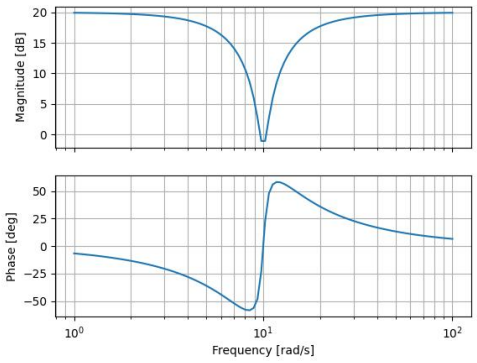
Filters can be described by a transfer function. ”A transfer function is the Laplace transform of the impulse response of a linear, time-invariant system with a single input and single output when you set the initial conditions to zero” (Douglas, 2019a). It describes the relation between the input and the output of the system. Writing it in the Laplace domain as compared to a differential equation in the time-domain, has the advantage that convolution in the time domain becomes a multiplication in the Laplace domain. In control theory, transfer functions are often described by a Bode plot which is the frequency response of a system (Olsder & van der Woude, 2005).

Low-pass filters are used to pass frequency signals below a cut-off frequency by attenuating frequencies higher than the cut-off frequency as depicted in the Bode plot in Figure 2.3a. This filter is ideal for reducing noise in a measurement signal as noise tends to be of higher frequency than process changes. However, low-pass filters can also be used to remove resonance. Resonance occurs when the system has a high gain around the resonant frequency and when this frequency is around the 180° loop phase, a low-pass frequency might be used to remove the resonance (Ellis, 2012).

A notch filter can also be used in controllers against noise and resonance. A notch filter is a type of band-stop filter that blocks a certain frequency interval and passes the other frequencies relatively unaltered. This is particularly convenient to filter undesired resonance peaks. A notch filter is a band-stop filter with a very narrow stopband as is depicted in the Bode plot in Figure 2.3b. By carefully designing the notch filter, the bandwidth of the removed frequencies is adjusted to target the troublesome frequencies. You don’t want a resonant frequency coupling through the controller which commands the actuators at that frequency which might further excite the flexible mode causing an unstable feedback loop or reduce the performance. Notching out that frequency, might help alleviate this problem. By combining a low-pass and a high-pass filter, a notch filter is created (Douglas, 2019a). Low-pass and notch filters can both cure resonance while notch filters create less phase lag in the control loop (Ellis, 2012).



(a) Bode plot of a low-pass frequency filter: $H(s) = \frac{1}{s+1}$.



(b) Bode plot of a notch filter: $H(s) = \frac{100s^2+100s+10000}{10s+125s+1000}$.

Figure 2.3: Bode plots of two types of filters; (i) the low-pass filter and (ii) the notch filter.

Additionally, there are more ways to suppress the residual vibrations with the use of e.g. input shapers. Input shapers create a reference signal that cancels its own vibration. Singhose and Vaughan, 2011 argues that input shapers are always superior to low-pass and notch filters. However, the research comprises of a feasibility study to design, build, and test a hydro-propelled drone and does not emphasise on the advancement of superior filters. Therefore, potential implementation of a relatively simple low-pass or notch filter is preferred.

2.5. Thrust-Vectoring

Thrust-vector control is widely used in fighter jet and spacecraft applications. Generally, three methods of thrust-vectoring are used: (i) reactive fluid injection, (ii) exhaust flow deflection, and (iii) engine mechanical manipulation (Bernacchia, 2019). This research focuses on the third method as the nozzles are mechanically actuated by design. As mentioned in Section 2.4, X. Liu and Zhou, 2019 discusses

the performance of thrust-vectoring and flow-regulation which was tested on two different platforms. Both aerial systems demonstrated sufficient attitude control for stable flight. It is shown that the thrust-vectoring system, having two nozzles is a non-minimum phase system as the nozzles are located under the drone's center of mass. The flow-regulated UAV is a minimum phase system. The control commands come from a remote controller and it is proposed for future research to study the guidance law so that the UWAV can perform a specific flight mission autonomously. In addition, flexible hose modelling and robust control for coupling and uncertainties must be studied.

Vinayagam and Sinha, 2014 assesses the concepts of thrust-vectoring for a twin-engine airplane by showing the equations of motion and simulating Nonlinear Dynamic Inversion (NDI) control as it is an ideal tool to evaluate the different thrust vector concepts. Full six degrees of freedom actuation for a quadrotor is researched by Ryll et al., 2012 through tilted propellers as a form of thrust vectoring.

2.6. Flight Control

In this section, various flight control techniques are discussed. Because the open-source Pixhawk Autopilot software uses PID Control for attitude and position control, some background knowledge on PID control is given in Section 2.6.1. Disturbance rejection is touched upon in Section 2.6.2 while feedforward control is discussed in Section 2.6.3. Finally, as the first analysis of the UHAV system showed non-minimum phase characteristics, control of these systems is discussed in Section 2.6.4.

An underactuated mechanical system (UMS) has fewer control actuators than degrees of freedom (Y. Liu & Yu, 2013). A quadcopter is an underactuated system as there are four rotors (control actuators) while it has six degrees of freedom (DoF). Rotational and translational movements are coupled, resulting in four control degrees of freedom. Mathematical modelling of UMS provides the base for control.

The reasons that cause underactuation are summarised below which classify UMS (Y. Liu & Yu, 2013; Olfati-Saber, 2000):

- Dynamics of the system by nature
- By design for cost reduction or practical purposes
- Actuator failure
- Artificially imposed for research purposes to gain insights in control of high-order UMS

The drone is underactuated by design to save weight and reduce the number of moving parts. It will be investigated whether the system can be sufficiently controlled with 2 actuators rotating around a single axis, or to conclude that coupling effects restrict attitude control.

Y. Liu and Yu, 2013 describes the aim of UMS control as "to find a feedback control law that stabilises the system in the presence of various uncertainties and external disturbances". Partial feedback linearisation (PFL), energy-based methods such as backstepping, sliding mode control (SMC), fuzzy control, and optimal control, are control techniques that are discussed. D.-H. Lee et al., 2021 uses SMC since the control theory is well-known for its robustness in facing disturbances and uncertainties. This is important as potentially the ground-effect, wind gusts and the water hose can impose disturbances. In addition, explicit knowledge of the system is not necessary. For the highly underactuated system in Kotaru and Sreenath, 2020, the system is controlled by linearising the dynamics about a given desired time-varying trajectory, to finally implement a finite-horizon linear quadratic regulator (LQR). Bernacchia, 2019 discusses LQR and SMC as suitable attitude controller for a thrust-vectored lunar landing vehicle and the SMC showed the best results. In D.-H. Lee et al., 2021, a PID controller is compared with an SMC in a simulation to conclude that in both absence and presence of disturbances, the SMC showed to be superior.

Initially, the fixed-wing attitude controller from PX4 is used as a starting point for this research which is described in Dronecode, 2022c. This controller needs to be altered to meet the requirements of the attitude of the UHAV and therefore, a brief summary of PID control, feedforward control, and gain tuning is given.

2.6.1. PID Control

PID controllers are widely used feedback control loops in various industries due to its strong robustness, simple principle, and convenient use. It consists of three units: (i) the proportional unit (P), (ii) the integral unit (I), and (iii) the derivative unit (D). The proportional unit is the basis of the controller and sets the performance boundaries of the controller (Ellis, 2012). The integral unit eliminates the steady-state error but may increase overshoot. Finally, the derivative unit can accelerate the response of the system and reduce the overshoot (L. Wang et al., 2021). In addition to Pixhawk's tuning guide in Dronecode, 2022b, Ellis, 2012 describes the general idea behind tuning a PID controller. The P- and D-gains act in the high-frequency zone while the I-gain acts in the lower-frequency zone. Adding a D-gain results in a higher possible P-gain which improves the response of the system. The tuning guide is enumerated below:

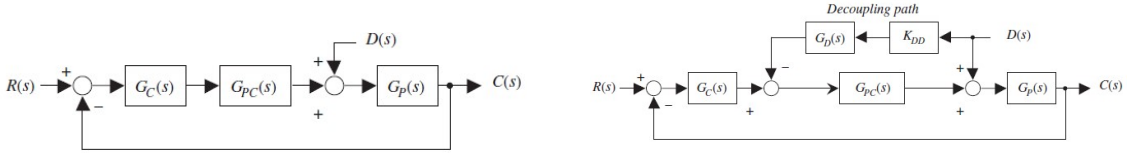
1. Tune the controller as if it was a P controller (I- & D-gains set to zero), but allow more overshoot ($\pm 10\%$).
2. Add D-gain to cure the overshoot of the P-gain. Practically, this means that the D-gain must be increased until the system starts to oscillate after which is must be reduced by 50%.
3. Add I-gain such that the steady-state error is reduced to zero. In reality, it is required to look at the log files for integral-gain tuning.

2.6.2. Disturbance Response

A controller must be able to follow the commands that it receives through an input signal, as well as being able to reject disturbances. Ellis, 2012 gives an overview of disturbance response methods. Here, disturbance response is defined as the response of the system output to the disturbance force, taken as input. The ideal disturbance response is $-\infty$dB. One way of improving the disturbance response is a slow-moving system with a large moment of inertia such that the system has a low gain. Secondly, a system that has high mechanical stiffness responds little to disturbances. One can imagine that a small and lightweight drone, attached to a flexible hose, is highly influenced by disturbances.

Finally, command and disturbance responses profit from high proportional gains as it provides a higher bandwidth and better ability to reject high-frequency disturbances. The I-gain helps in rejecting low-frequency disturbances. However, another type of gain is the disturbance decoupling gain which measures disturbances to improve the system's disturbance response. Conversely, feedforward gains, which are discussed in Section 2.6.3, improve command response.

When low stability margins prohibit raising the loop gains, disturbance decoupling can aid the disturbance response of the system. However, disturbance decoupling requires to measure the disturbance which can be complex. Figure 2.4 illustrates disturbance decoupling in a control loop.



(a) Feedback control loop with disturbance (Ellis, 2012).

(b) Feedback control loop with disturbance decoupling (Ellis, 2012).

Figure 2.4: Disturbance decoupling

Zheng et al., 2009 discusses a novel active disturbance rejection framework (ADRF) for multivariable systems using a unique dynamic disturbance decoupling control (DDC) strategy. Using an extended state observer, little knowledge about the system dynamics is required. This is favourable as in reality, little about the exact dynamics of physical systems and their disturbances are known as it uses real-time input and output data. Hao et al., 2019 discusses disturbance rejection control for flexible air-breathing

10

11

hypersonic vehicles. While robust control is achieved over the years through (higher-order) SMC, LQR, fuzzy control, and backstepping control, robustness to light disturbances are handled through inertial ability of the vehicle. Chen et al., 2015; T. Wang et al., 2016; J. Yang et al., 2016 describe several disturbance estimation techniques.

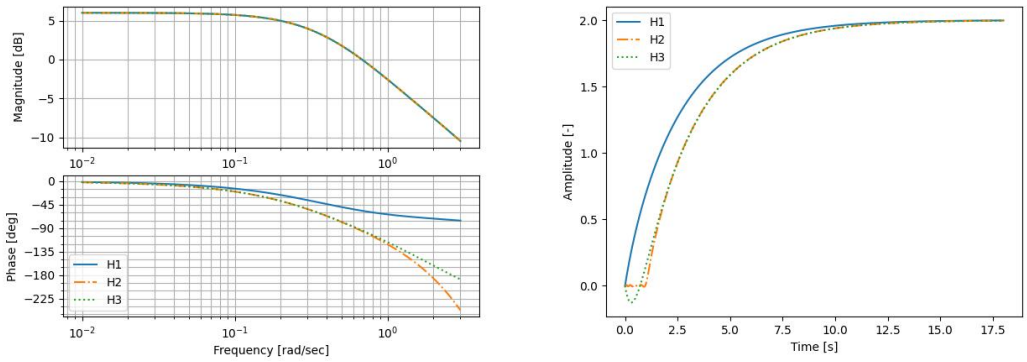
2.6.3. Feedforward Control

As mentioned in Section 2.6.2, feedforward control impacts the command response of the system using knowledge of the system (Ellis, 2012). Knowing the operation of the system by $\pm 20\%$ can already improve the controller dramatically. An additional benefit of feedforward control is the elimination of the steady-state error during a constant command.

A feedback controller regularly acts on the error between the reference signal and the measured state estimate. This error is effected by setpoint changes, external disturbances, and sensor noise. Looking at this in terms of frequencies, usually setpoint changes have a low frequency, disturbances tend to have higher frequencies, while noise usually has a high frequency. A low-pass filter might solve for this problem as discussed in paragraph 2.4. However, when the disturbances and noise act in the same frequency zone, this can no longer be filtered by the low-pass filter. In this situation, a feedforward loop can be used (Douglas, 2018). In the case the controller needs to ignore the disturbance, feedforward control might also work. Feedforward control works when the system dynamics are known such that it can generate the right output directly based on the reference signal. However, there is always a difference between the model and the physical process. Therefore, the feedforward controller takes care of the bulk behaviour while the feedback loop takes care of the error.

2.6.4. Non-Minimum Phase Systems

System behaviour can be described by transfer functions and are often categorised. One of the characterisations is minimum or non-minimum phase systems. A non-minimum phase system is a system that shows either inverse response (undershoot) or time-delays which is shown in Figure 2.5 (Uren & van Schoor, 2011). Figure 2.5a, shows that there are multiple transfer functions possible with the same magnitude response, but with an affected phase response. Adding time delay or pushing the zero into the right half plane. The additional phase response that is added to the system by time delay or a positive zero, influences the step response function, which is shown in Figure 2.5b. Moving in the wrong direction first slows down the system response which makes it harder to control such a system. If you increase the gain of the controller, it may become unstable because the phase margin is smaller compared to the minimum phase system (Douglas, 2019b).



(a) Bode plot to illustrate that multiple transfer functions can have equal magnitude responses but varying phase responses.

(b) Step response to illustrate direct response (H1), time delay (H2), and inverse response (H3).

Figure 2.5: Bode plot and step response of three transfer functions: $H1(S) = \frac{s+2}{s^2+3+1}$, $H2(S) = \exp -s\frac{s+2}{s^2+3+1}$, and $H3(S) = \frac{-s+2}{s^2+3+1}$.

Using predictive or nonlinear controllers is a way to overcome the problems of non-minimum phase systems. However, the easiest solution is to slow the system down by lowering the gain to maintain stability. If the system cannot act fast enough to the error that is generated by the non-minimum phase components, fed into the system through the feedback loop, the system is going to behave as if there is no error (Douglas, 2019b). Time delay is a by-product of non-minimum phase systems which is undesired in a control system. One way to control systems with time-delays is using a Smith predictor structure or internal model control (IMC) (Uren & van Schoor, 2011). A plant model is used to predict the future output such that the control law can act immediately on the reference input before becoming unstable. Additionally, MPC can be used that uses an optimisation routine to minimise a cost function to derive optimal values of the PID gains.

2.7. Next Steps in the Research Field

Having discussed above literature in the field of control of manned and unmanned aerial platforms, propelled by air or water, whether it is attached to a tether, and thrust-vectored or weight-distributed attitude control, the literature showed several research opportunities. One can summarise the relevant research opportunities through robust control for coupling effects of yaw and roll and automated flight of predefined missions for an unmanned hydro-propelled aerial platform (X. Liu & Zhou, 2019), precise aiming of the water spray into the direction of e.g. the fire (Spurny et al., 2021), use of robust parameter identification methods to increase stability in flight (Ando et al., 2018), control the flying length of the water hose and increase its robustness to external disturbances (Ando et al., 2018), fluid-structure interaction (Ando et al., 2018; Kotaru & Sreenath, 2020), and comparison of a PID or linear quadratic optimal controller with an SMC in an experimental setup (D.-H. Lee et al., 2021). SMC showed to be superior in most cases with a tethered drone due to its robustness against disturbances and the limited necessary knowledge about the system. NDI control has, to the knowledge of the author, not yet been implemented in the research field of hydro-propelled drones but may be of interest due to its great performance with thrust-vectoring control (Vinayagam & Sinha, 2014). Yamaguchi et al., 2019 concludes that weight and water flow are limiting factors and must be reduced. Therefore, suppression of tether vibrations with limited actuators is desired such that an analysis on whether the system can be controlled by using less actuators is interesting.

3

Redefined Scope

After the literature review and the preliminary analysis of the dynamics of the system, the research scope was adjusted. Multiple designs were presented to different stakeholders at Actiflow to conclude that the most optimal design was to use pressure differentiation. However, when discussing this with the senior design engineer at Actiflow, the conclusion was drawn that constructing a flow regulator would be unfeasible, given the limited resources of budget and time. Continuation of the trade-off, taking into account cost and manufacturability, led to the introduction of the canting keel. The high pace at which the canting keel could be implemented and its exceptional performance led to an adjustment of the research scope. In addition to attitude control, position control could be added as an extra deliverable. Because the GPS signal was blocked inside the drone arena and autonomous missions of the UHAV require vision, it was decided that vision-based position control would be implemented. To the best of the author's knowledge, no UHAV has been built with a position controller that can track waypoints.

Therefore, the research question was adjusted:
"Is a novel tethered unmanned hydro-propelled aerial vehicle capable of setpoint tracking and position control in the presence of disturbances, using minimum actuation?"

Sub-question **Q4** was replaced by the following sub-question:
Q4: How can camera feedback be integrated into the system to provide real-time position control?

(a) Which computer vision algorithm can be used for real-time motion tracking of the drone and/ or its water jets?
(b) What is a simple, yet effective user interface that interacts with a human operator to control the position of the drone?
(c) What is the necessary software, hardware and system architecture?

Part II

Scientific Paper

Stability control and positional water jet placement for a novel tethered unmanned hydro-propelled aerial vehicle using real-time water jet detection

X. F. van Beurden
Supervisors: Dr. ir. C. de Wagter and ir. O.L. van Dijk
Section Control & Simulation, Department Control and Operations, Faculty of Aerospace Engineering
Delft University of Technology, Delft, The Netherlands

Abstract

Aerial platforms designed for water jet placement are gaining interest in the areas of fire-fighting, washing, and irrigation. A novel, lightweight, and simplistic design is proposed that reduces the number of actuators and limits ineffective water discharge. External camera feedback was used for position control as a first step towards autonomous flight. An initial prototype of an unmanned hydro-propelled aerial vehicle (UHAV) connected to a water hose was designed and fabricated. Flight tests were conducted to show that attitude control with uniaxial thrust-vectoring of two nozzles was impossible due to undamped vibrations and coupling effects. By redesigning the PID controller, pitch rate damping was accomplished. Furthermore, a design trade-off led to the introduction of a canting keel to reduce bank-yaw coupling effects due to asymmetric nozzle deflections. Flight tests proved that the iterated design with a hose length of 3m was capable of disturbance rejection and setpoint tracking. An external camera was used to show that the Lucas-Kanade optical flow algorithm and the implementation of the YOLOv5 segmentation model can be used for positional water jet placement. By increasing the pitch rate damping, improving the water jet detection algorithm and implementing a cost function for water discharge at the area of interest, autonomous missions can be flown in the future.

1 Introduction

Drones are flying robots, including unmanned aerial vehicles (UAVs), that can fly thousands of kilometres or navigate in confined spaces [1, 2]. Depending on the application of the drone, the flight configuration is determined. While fixed-wing systems are used for long-range flights, various ways exist to achieve hovering flight. Rotors [3, 4], thrusters [5], ornithopters such as the Delfly [6], kites [7], and lighter-than-air systems [8, 9] are designed for hovered flight. This research is concerned with hovered flight using water thrusters to combine lift and water supply at a predetermined location.

Thrust is generated through the acceleration of mass, of which various types of gasses or working fluids can be used for propulsion[1]. Recent research has shown that manned flight using jetpacks is possible for military[2] and recreational[3] purposes. Additionally, it can support fire suppression techniques[4]. The human pilot often controls the attitude through weight displacement or manual thrust vectoring. For unmanned systems, attitude is controlled using thrust vectoring [10], pressure differentiation [11], rudders or flaps [12, 13, 14], weight displacement [15], or control moment gyros [16, 17]. For the initial design, thrust vectoring is used due to its simplicity in design and control.

Generally, three methods of thrust vectoring are used: (i) reactive fluid injection, (ii) exhaust flow deflection, and (iii) engine mechanical manipulation [18]. This research is concerned with the mechanical manipulation of the thrusters. The thrust vector is adjusted by actuating a servo to create control torques that modify the vehicle's attitude. In manned flight, the thrusters can be attached to the arms, and the thrust is vectored intuitively by the human operator, such as the military jetpack. For unmanned or automated systems, one or multiple thrusters are automatically vectored from underneath the body, like Yves Rossy's Jetman project[5].

Using water as the working fluid has a range of potential use-cases. With its unique propulsion system, one can combine thrust vectoring for control and manoeuvrability with a useful application such as irrigation, fire suppression, and washing.

One of the key differences between using air or water as accelerating fluid is the collection of it. While air can be extracted from the air, water needs to be supplied via a hose, creating a tethered UAV system. Fagiano [19] studied the advantages of power supply via a tether. While this can increase

[1] https://www.grc.nasa.gov/www/k-12/airplane/thrsteq.html — Accessed: 22-08-2022
[2] https://www.gravity.co — Accessed: 25-07-2022
[3] https://www.zapata.com — Accessed: 25-07-2022
[4] https://www.bbc.com/news/av/world-middle-east-38711494 — Accessed: 25-07-2022
[5] https://yvesrossy.com/ — Accessed: 25-07-2022

flight time, tethered UAVs also impose complexities. Ioppo [20] compared the stability and positional dynamics between a quadcopter connected to a taut tether and a quadcopter connected to a loose tether. It was concluded that taut tethers impose the most stable flight. Winget and Huston [21] described the dynamics of loose cables while flow-structure interaction was studied by Xie et al. [22] and Hyvärinen et al. [23]. The dynamics of tethered quadrotors have been modelled in [24, 25, 26, 27]. However, the modelling of a drone carrying a hose with water flow and pressure forces is unprecedented, according to the study of Kotaru and Sreenath [28]. These forces will lead to vibrations in the tether propagating towards the drone. Passive vibration suppression for a tethered unmanned hydro-propelled aerial vehicle (UHAV) was exploited in [29], while closed-loop control methods to actively suppress the vibrations of a swinging payload were studied in [30, 31, 32].

For the UHAV, tip of the tether is controlled by the drone. Many studies have been conducted about the control of flexible beams, as reviewed in [33]. Continuum robots, whose tip position is controlled by concentric tube designs, pneumatic actuators, and tendons, are proposed in [34]. A water hydraulic snake-type robot, suitable for unmanned firefighting that can control its movement on the ground plane, was demonstrated in [35]. This robot is limited to operations on the ground as it cannot cover large distances through the air, limiting its operational effectiveness at higher altitudes for fire suppression, washing, or irrigation. Liu and Zhou [11] studied two control methods of a UHAV system. The nozzle-gimbaled UHAV consists of two nozzles, while the flow-regulated UHAV has four nozzles. Both designs are shown in Figure 1a and Figure 1b, respectively. As the hose is attached to the center of mass, designed to hang freely into a water basin, the drone is designed for vertical thrust and hovering at levelled flight. While this is beneficial for carrying payloads, it limits use-cases that require the water to be sprayed at an angle, such as washing or fire-fighting where straight water discharge wastes most of the water due to evaporation [36]. The nozzle-gimbaled UHAV has non-minimum phase in lateral acceleration, limiting its movement close to an object. The flow-regulated UHAV requires four nozzles and a complex mechanism to distribute the flow between its nozzles. Both problems must be addressed to increase operational effectiveness and decrease development time and cost.

Ando et al. [37, 36] studied the control of a flexible hose by applying translational forces to the tip of the hose. The Dragon Firefighter (DFF) robot can fly to the source directly and uses multiple nozzle modules that emit water to lift the hose for the scalability of the length of the water hose. This system is depicted in Figure 1c. Controlling the tip requires four nozzles, two of which are biaxial nozzles and two are uniaxial nozzles. While this reduces the underactuation, it adds complexity and weight and increases failure modes. Additionally, much water is wasted through the uniaxial nozzles that do not contribute to the water discharge at the area of interest. Yamaguchi et al. [29] worked on improving the DFF by designing a passive vibration-damping mechanism. Vibrations in long, flying continuum robots limit fast and accurate control, thus, vibration suppression is essential for the operational output of the system. However, active vibration suppression through control of the nozzles removes weight and complexity of the design while being flexible and scalable to different designs of a flying continuum robot. Finally, to the best of the author's knowledge, no previous work focused on position control of UHAV systems. UHAV systems that can execute a mission autonomously, such as fire suppression or irrigation, can reduce the number of operators engaging in dangerous or time-consuming tasks.

(a) Nozzle-gimbaled UHAV [11]. (b) Flow-regulated UHAV [11]. (c) Dragon Firefighter [36]

Figure 1: Related work on unmanned hydro-propelled aerial vehicles.

In this research, actuation was limited to two uniaxial nozzles to reduce the design's weight and complexity while increasing the water discharge's operational output. A third actuator was installed, which used weight displacement for attitude control. Two control modes associated with different levels of automation were considered. The first was related to stabilisation, attitude control, and linear accelerations. The latter related to position control using real-time water jet detection through an external camera. In addition, active vibration suppression were studied to eliminate the additional mass of the mechanical hose suppression mechanism. The following two points can summarise the contribution of this research. (i) a novel, simplistic, and lightweight design that limits the waste of water and (ii) position control of the UHAV system using an external camera.

The paper is structured as follows. Section 2 gives an overview of the UHAV system. In section 3, the setup of the experiments and evaluation methods are discussed. Thereafter, section 4 presents the results of the initial design and evaluates proposed solutions that improve flight performance. section 5 discusses the iterated design and section 6 presents the results of the iterated design. Finally, section 7 summarises the results and shares recommendations for future research.

2 DESIGN

The final design of the drone was subject to an iteration based on flight test analyses. This section elaborates on the initial design of the drone. subsection 2.1 discusses the physical design, while the thrust and actuating mechanisms are covered in subsection 2.2 and subsection 2.3, respectively.

2.1 Physical design

Consider a UHAV that is connected via a primary hose to a water supply and pressure washer, as illustrated in Figure 2. Let P1 be the attachment of the stationary pole with the hose and P2 the connection point between the drone and the hose. The North-East-Down (NED) frame is defined by unit vectors $N = [1, 0, 0]$, $E = [0, 1, 0]$, and $D = [0, 0, 1]$. The body-fixed reference frame $\{x, y, z\}$ is located at the center of mass of the UHAV of which x is aligned along the symmetry axis towards P1, y points to the right and z points downward. The rotation matrix $\mathbf{R}$ represents the linear translation from the NED-frame to the body frame where φ, θ, ψ are the roll, pitch, and yaw angles. A heading rotated tangent frame, denoted as pressure frame, is defined by unit vectors $X = [1, 0, 0]$, $Y = [0, 1, 0]$, and $Z = [0, 0, 1]$, where $Z = D$, and X and Y are rotated by the heading such that they always align with the horizon. α, defined as the azimuth angle, denotes the angle between N and the projection of the hose onto the NE-plane while ψ is the heading of the airframe. A newly defined yaw angle, ν, is defined as the difference between the heading and the azimuth angle. Expressing this in mathematical terms leads to Equation 1.

$$\nu = -(\alpha - \psi) \tag{1}$$

For the longitudinal and 2D analyses, $\nu = 0$ is assumed. δ_{n_i} is the deflection of the i-th nozzle around the Y-axis, where $0°$ corresponds to horizontal deflection along the X-axis and $90°$ corresponds to vertical deflection along the Z-axis. $i = 1$ for the nozzle in positive y and $i = 2$ for negative y location. Note that thrust acts in the opposite direction of the nozzle deflection.

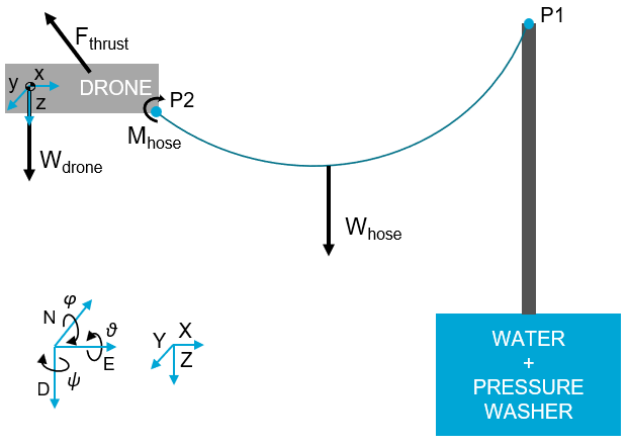


Figure 2: Schematic side view of the UHAV system.

Equation 2 shows the static equilibrium equations for the 2D representation of the UHAV system. It is important for the equilibrium to hold that the nozzles, delivering the thrust, are located between the airframe's center of mass and P2. The thrust needs both an X- (F_{T_X}) and a Z-component (F_{T_Z}) to lift the drone, including (part of) the hose.

$$\begin{aligned} \sum F_X &= F_{h_X} - F_{T_X} \\ \sum F_Z &= F_{h_Z} + W_d - F_{T_Z} \\ \sum M_{cg} &= -M_h - F_{h_Z} x_h + F_{T_X} z_n + F_{T_Z} x_n + F_{T_X} z_n \end{aligned} \tag{2}$$

F_{T_X} and F_{T_Z} are positively defined in the direction as denoted in Figure 2. x_h, z_h, x_n, and z_n are the body-frame locations of, respectively, the x- and z-location of the hose and the x-, and z-location of the nozzle, measured from the center of gravity (cg). The specifications of the system are shown in Table 1 and a technical drawing of the airframe is depicted in Figure 3. The flight computer, along with all the other electronics, are nested inside the 3D printed waterproof electronics box. For this research, the Pixhawk 4 (PX4) Mini[6] was used as the flight computer because the code is open-source, its renowned performance as a versatile and modular flight computer, and the large and active development community. The ground station was QGroundControl (QGC)[7] because of its integration with PX4 and an additional User Diagram Protocol (UDP) connection between a laptop and the PX4 was established through the `pymavlink` [38] package integrated into Python.

[6] `https://px4.io/` — Accessed: 03-08-2022
[7] `https://qgroundcontrol.com` — Accessed: 03-08-2022

Table 1: Specifications and dimensions of the drone

Specifications	**Value**	
Mass drone (wet), m_d	1.25	[kg]
Mass primary hose (wet), $m_{h,1}$	0.5	[$\mathrm{kg\,m^{-1}}$]
Length primary hose, $l_{h,1}$	3.0	[m]
Length secondary hose, $l_{h,2}$	0.4	[m]
Width airframe, w_d	0.73	[m]
Height airframe, h_d	0.28	[m]
Length airframe, l_d	0.77	[m]
x location of the nozzles, x_n	0.21	[m]
y location of the nozzles, y_n	0.37	[m]
z location of the nozzles, z_n	0.33	[m]
Pressure water pump, P_{pump}	120	[bar]
Mass flow water, $\dot{m}$	0.13	[$\mathrm{kg\,s^{-1}}$]
Thrust, T	19.0	[N]
Diameter nozzle, d_{nozzle}	0.9×10^{-3}	[m]

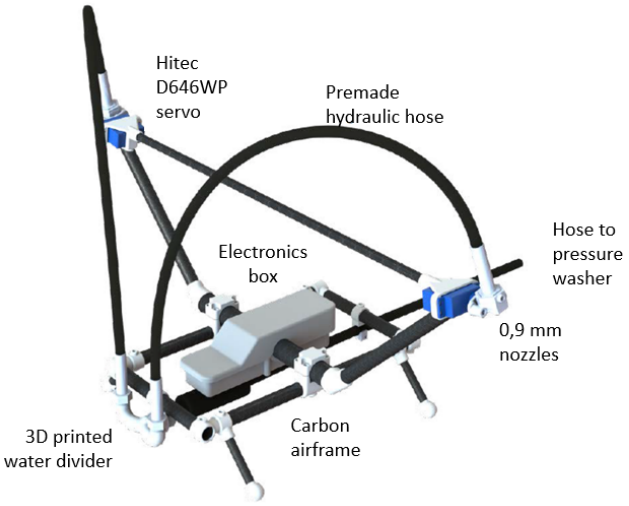


Figure 3: CAD drawing of the initial design of the drone. The frame is composed of carbon fibre rods and by 3D-printed joints.

2.2 *Thrust*

The lift of the drone is generated through thrust. Thrust is generated through the reaction of accelerating the water. Thrust is equal to the change in momentum over time, denoted by Equation 3, where the exit and atmospheric pressure are assumed to be equal[8].

$$T = \frac{dp}{dt} = \dot{m}(V_e - V_i) \tag{3}$$

[8] https://www.grc.nasa.gov/www/k-12/airplane/thrsteq.html — Accessed: 22-08-2022

Here, $\dot{m}$ is the mass flow rate of the water, and V_e and V_i are the exit and initial velocities of the water, respectively. $\dot{m} = \rho V A$ shows that for incompressible flow and a constant cross-sectional area of the nozzle, the mass flow can be increased by increasing the fluid's velocity. This can be achieved by raising the pressure of the pressure washer or by decreasing the cross-sectional area of the nozzle [39]. An optimum exists, where decreasing the diameter of the nozzle reduces the mass flow and thrust decreases.

2.3 *Actuating mechanism*

Control torques are generated by adjusting the direction of thrust around a single axis. Both nozzles are connected to the primary water hose through two flexible, secondary water hoses. By rotating the nozzles around the y-axis, thrust in the X-, Y-, and Z-directions is regulated. The nozzles can rotate independently of each other in order to generate symmetric and asymmetric control torques onto the airframe. By rotating the nozzles around an axis, the change in forces applied to the airframe are nonlinear which increases complexity in the controller design. Considering $\varphi = 0$, Figure 4, Equation 4 and Equation 5 illustrate that the effect of a 10° nozzle deflection on the force in X- and Z-direction is determined by δ_n. When the nozzle points downward in the direction of gravity, a 10° deflection results in a 2% decrease of thrust along the Z-axis and a 17% increase in thrust along the X-axis. Nozzle efficiency and coupling effects must be taken into account when designing the controller. The grey marked area in Figure 4 shows the deflection limits of the actuators in order to operate safely. This corresponds to the stability criteria denoted by equilibrium Equation 2.

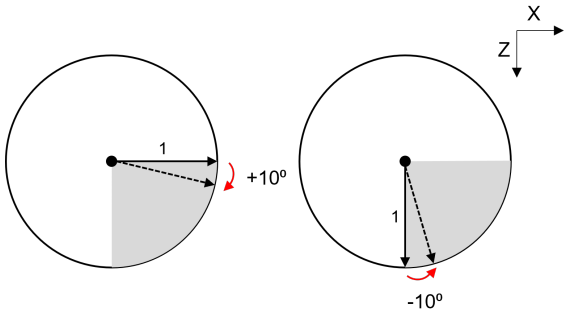


Figure 4: Visualisation of the nozzle deflection limits and their efficiencies for a 10° deflection at the two deflection limits.

$$F_X(\delta) = \begin{cases} \cos(0) = 1 \\ \cos(10) = 0.98 \end{cases} \tag{4}$$

$$F_Z(\delta) = \begin{cases} \sin(0) = 0 \\ \sin(10) = 0.17 \end{cases} \tag{5}$$

Equation 6 and Equation 7 show that by changing the deflection angle (δ_n), the forces and moments that are exerted onto the airframe by the nozzles are controlled. Note that the torsional and bending stiffness of the hose and the inertia of the system influence the dynamics when forces and moments are applied. M_{n_X} and M_{n_Z} are the moments generated by the nozzles around the X- and Z-axis.

$$M_{n_X} = -F_{T_1}\sin(\delta_{n_1}) \cdot y_n + F_{T_2}\sin(\delta_{n_2}) \cdot y_n \quad (6)$$
$$M_{n_Z} = F_{T_1}\cos(\delta_{n_1}) \cdot y_n - F_{T_2}\cos(\delta_{n_2}) \cdot y_n \quad (7)$$

3 METHODOLOGY

This section elaborates on the methodology used to build a stable and controlled UHAV system. By analysing the design characteristics of the drone and creating a framework on how to evaluate its performance, a strategy on how to enhance the drone was built. First, subsection 3.1 analyses the dynamics of the initial design of the drone and discusses its expected shortcomings and complexities. Furthermore, it elaborates on the architecture of the attitude controller. Secondly, subsection 3.2 discusses the design evaluation method. Finally, the trade-off method based on the performance criteria is touched upon in subsection 3.3.

3.1 Current design

The control is split into longitudinal and lateral motion characteristics. Similar to UAVs and rockets, the dynamics can be obtained using Newton's equations. The forces applied to the airframe include the active control forces from the nozzles, the interference torques from the three hoses, and dynamic forces within the water hose such as the Coriolis effect [11]. The latter forces are regarded as disturbances which are handled by the controller.

3.1.1 Longitudinal stability

Longitudinal motion is determined by the total force applied by the nozzles in the Z-direction. Typically caused by power limitations of a pump with constant pressure, the flow rate decreases with an increasing height ($\frac{\partial \dot{m}}{\partial h} < 0$) [11]. During flight tests, the sound of the pump or the water discharge did not significantly change. The effect of nozzle deflections is much larger compared to the decrease in flow rate and therefore, this effect is neglected. However, when increasing the size of the system, it is recommended that this effect is taken into account. Subsection 2.1 showcases that the flying equilibrium is determined by the interaction between the thrust and moment forces of the nozzles and the hose. By deflecting the nozzles symmetrically, the drone will move in the XZ-plane until a new equilibrium is reached where the hose exerts different forces and moments onto the drone. This position is directly related to the length and the weight of the primary hose. The hose will hang onto its own weight, while being supported at both ends, P1 and P2. The supporting forces are calculated using the physics of a catenary [40]. The shape of the curve of the hose is independent of its linear mass density which makes it applicable to all chains that are assumed to be non-elastic [41]. The catenary equation is given by Equation 8. In order to plot the catenary, one must calculate the parameter a, known as the catenary constant. a is found through iterations according to Equation 9, using Equation 10 and Equation 11 [42].

$$y = a\cosh\left(\frac{x}{a}\right) \quad (8)$$

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)} \quad (9)$$
$$f(a) = 2a\sinh\left(\frac{h}{2a}\right) - \sqrt{L^2 - v^2} \quad (10)$$
$$f'(a) = 2\left[\sinh\left(\frac{h}{2a}\right) - \left(\frac{h}{2a}\right)\cosh\left(\frac{h}{2a}\right)\right] \quad (11)$$

L is the length of the cable, v and h are the vertical and horizontal distance between P1 and P2, respectively. The tension T in the cable at distance s from the low-point is given by Equation 12. Taking the $\sin$ and $\cos$ of the angle of the catenary at that point will give the corresponding forces in X- and Z-direction.

$$T = \lambda g\sqrt{s^2 + a^2} \quad (12)$$

Where λ is the mass per meter of the cable. With the static equilibrium equations, the 2D location of the drone relative to the pole can be determined with the use of Equation 12 for various drone layouts.

To determine the X-Z position where the drone will hover, the tension, weight of the drone, and thrust must cancel out. The thrust in X- and Z-direction are given by Equation 13 and Equation 14, respectively, assuming $\delta_{n_1} = \delta_{n_2}$. The small, constant lateral deflection angle of the nozzles is neglected due to the small angle approximation. The upward and forward motion of the UHAV can be described by Equation 15 and Equation 16, respectively.

$$F_{T_X} = F_T \cdot \cos(\delta_n) \quad (13)$$
$$F_{T_Z} = \big(F_T \cdot \sin(\delta_n)\big)\cos(\mu) \quad (14)$$

$$ma_Z = -F_{T_Z} + F_{h_Z} + m_d g \quad (15)$$
$$ma_X = -F_{T_X} + F_{h_X} \quad (16)$$

Figure 5 shows the flight envelope of the drone with a 3m hose. The moment at P2, due to deflection of the hose was assumed to be zero.

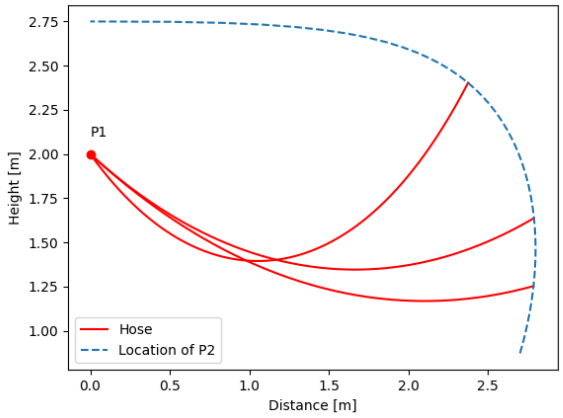


Figure 5: 2D space of drone flight envelope and corresponding catenary curves of the 3m hose with the specifications from Table 1.

The pitch angle of the drone is determined by the moment around the y-axis. As the shape of the catenary changes, so do the forces and moment at P2. An increase in height results in more weight of the hose being carried by the drone and the pitch angle will decrease. The nozzle deflection must therefore act independently of the pitch angle of the drone. Thus the first goal of the controller is not to control the pitch angle, but to control the nozzle deflection in the pressure frame. Figure 6 shows that the nozzle deflection in the pressure frame determined the height of the drone (for a fixed hose length) and that the pitch angle varied with its vertical position.

Figure 6: Relationship between pitch angle and nozzle deflection.

Pitch rate control While the pitch angle is not directly controlled, the second goal of the controller is to dampen the pitch rate. The changing pitch angles, nozzle deflections, and the vibrations of the secondary hose that translate to pitch angle disturbances onto the airframe, cause oscillations in pitch rate. Two factors are considered that affect the pitch rate. The first factor is that a clockwise (CW) deflection of the nozzle creates a counterclockwise (CCW) reaction torque onto the airframe. When $\dot{\theta} < 0$, a CCW deflection of the hose is necessary to reach the first goal. However, the CW reaction torque contributes to an additional negative moment around the y-axis, decreasing the pitch angle even more. Therefore, the system is non-minimum phase. The flexibility of the hose reduced the non-minimum phase effect as the movement of the weight of the secondary hose is delayed. The second factor is the moment of the nozzles around the y-axis. Considering the layout of the airframe, the nozzles are located forward and above the center of gravity. Looking at Equation 17, the deflection and the pitch angle influence the moment of the nozzle around the cg. It is assumed that the deflection of the secondary hose does not influence the location of the cg. This relationship is plotted in Figure 7, which shows that for an angle of $\delta_n + \theta > 30°$ the moment around the y-axis decreases. For $\dot{\theta} < 0$ while $\delta_n = cst$ (goal 1), the moment around the y-axis decreases which results in a smaller pitch angle.

$$M_{n_Y} = -z_n \cdot \cos(\delta_n + \theta) + x_n \cdot \sin(\delta_n + \theta) \qquad (17)$$

Figure 7 illustrates the relationship of Equation 17. For $z_n > 0$, the nozzles are placed under the cg and for $z_n < 0$, the nozzles are placed above the cg. x_n was held constant which is the reason all three lines cross at $\delta_n + \theta = 90°$. The red cross is one of the possible hovering points and demonstrates that the primary hose exerts a moment of $M_{h_Y} \approx -2.75$ Nm. A negative slope at the red cross means that the design is statically stable. When the pitch angle increases and the nozzle remains fixed in the pressure frame, the moment of the nozzle around the y-axis decreased. Adding a positive D-gain results in a further decrease of the moment. The green line ($z_n = -0.35$) shows that by moving the cg down with respect to the nozzles, the effect is larger. The solid blue line ($z_n = 0.25$) shows that when the nozzles are located below the cg, the design is statically unstable. Note that throughout the entire flight envelope, $\delta_n + \theta > 35°$ holds.

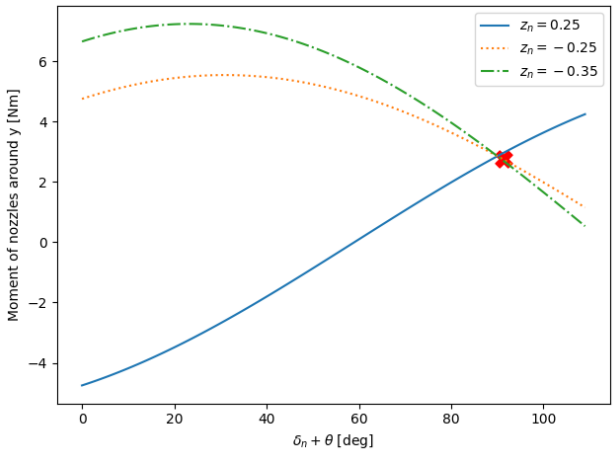


Figure 7: Moment around the y-axis of the nozzles due to the angle of the nozzles relative to the airframe. z_n refers to the z-location of the nozzle and the red cross denotes an operational configuration during flight where the primary hose creates a counteracting moment of $M_{h_Y} \approx -2.75$ Nm.

3.1.2 Lateral stability

For stability and position control, the roll angle and roll rate must also be controlled. As the orientation of the nozzles is decoupled from the drone's pitch angle, the roll angle of the nozzles rather than the roll angle of the drone have to be controlled. This angle is defined as the bank angle, μ, which is the angle between the body's y-axis and its projection onto the XY-plane. In the extreme case that the drone has a pitch-up angle of $90°$, roll and yaw (in the body reference frame) are inverted. This is visualised in Figure 8. To clarify this in mathematical terms, the bank angle is defined in Equation 18.

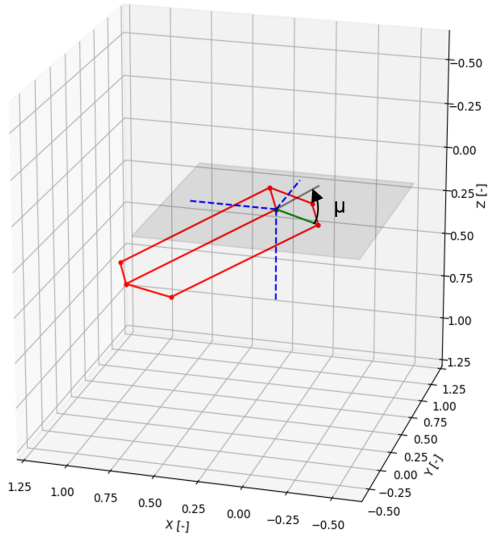


Figure 8: Visualisation of bank angle μ.

$$\mu = \sin^{-1}\left(\frac{Z_{n_1} - Z_{n_2}}{w_d}\right) \tag{18}$$

A more efficient way to compute the bank angle is to compute the cross-product of the body's y-axis with the Y-axis. In order to get the y-axis, the rotation matrix that rotates the roll, pitch and yaw angle from the NED frame to the body frame is multiplied by the unit y-axis in order to get the y-term of the rotation matrix ($\mathbf{R}_y$). The cross-product multiplication of $\mathbf{R}_y \times [0, 1, 0]^T$ is shown in Equation 19.

$$\mathbf{R}_y \times \mathbf{y} = \begin{bmatrix} -\sin(\varphi)\cos(\theta) \\ 0 \\ \begin{pmatrix} \sin(\varphi)\sin(\theta)\cos(\psi) \\ -\cos(\varphi)\sin(\psi) \end{pmatrix} \end{bmatrix} \tag{19}$$

According to Equation 20, the magnitude of the resulting vector from Equation 19 is equal to the dot product of both vectors times the sine of the angle between the two vectors. However, the bank angle is defined as the angle of the rotated y-axis with respect to its projection onto the xy-plane. Therefore, one must take the first term of the resultant matrix in Equation 19 which corresponds to the magnitude of the body in the YZ-plane. Hence, the formula of the bank angle, μ, is given in Equation 21.

$$|\mathbf{A} \times \mathbf{B}| = |\mathbf{A}||\mathbf{B}| \sin\theta \tag{20}$$

$$\mu = -\sin^{-1}\left(-\sin(\varphi)\cos(\theta)\right) \tag{21}$$

The bank rate is computed through Equation 22. Although this is a simple computation, for high pitch angles, this is not valid as the yaw angles start to translate into high roll angles. However, during flight, such pitch angles are not reached and therefore the computation remains valid.

$$\dot{\mu} = \dot{\varphi} \cdot \cos(\theta) + \dot{\psi} \cdot \sin(\theta) \tag{22}$$

The yaw rate, $\dot{\nu}$, is also defined differently compared to the classical definition. The definition is given in mathematical terms by Equation 23.

$$\dot{\nu} = \dot{\psi} \cdot \cos(\theta) + \dot{\varphi} \cdot \sin(\theta) \tag{23}$$

The bank acceleration is controlled by applying a torque around the X-axis. The bank and yaw acceleration are defined by Equation 24 and Equation 25.

$$\ddot{\mu} = \frac{M_X}{I_{XX}} \tag{24}$$

$$\ddot{\nu} = \frac{M_Z}{I_{ZZ}} \tag{25}$$

Where I_{XX} and I_{ZZ} are the moments of inertia around the X- and Z-axis, respectively, which is a function of attitude. M_X and M_Y are given by Equation 26 and Equation 27.

$$M_X = y_n \left(F_{T_{Z_2}} - F_{T_{Z_1}} \right) \tag{26}$$

$$M_Z = y_n \left(F_{T_{X_1}} - F_{T_{X_2}} \right) \tag{27}$$

Here, the subscripts 1 and 2 correspond to the i-th nozzle. Similar to multicopters, the bank and translational motion are coupled. Additionally, the yaw angle is also coupled to translational motion, which is shown in Equation 28.

$$ma_{lat} = \left(F_{T_{Y_1}} + F_{T_{Y_2}} - F_{h_Y} \right) \cos(\nu) + \left(F_{T_{X_1}} + F_{T_{X_2}} \right) \sin(\nu) \tag{28}$$

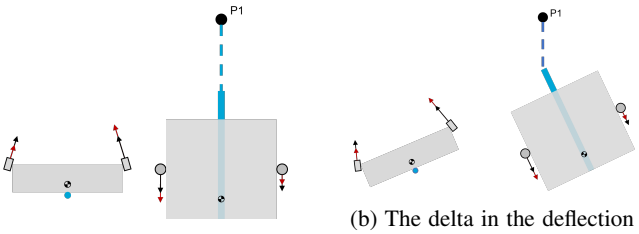
Where ma_{lat} is the lateral acceleration perpendicular to the azimuth angle and F_{T_Y} is given by Equation 29.

$$F_{T_{Y_i}} = \left(F_T \cdot \sin(\delta_{n_i}) \right) \sin(\mu) + F_T \cdot \sin(\delta_{n_{x_i}} - \mu) \tag{29}$$

Where $\delta_{n_{x_i}}$ is the deflection of the i-th nozzle around the x-axis. This angle was fixed throughout the research. Equation 30 shows that the ratio between thrust in X- and Z-direction of each nozzle is coupled. Therefore, Equation 28 proves that the bank and yaw accelerations are coupled. When $F_{T_{Z_1}} > F_{T_{Z_2}}$, then $F_{T_{X_1}} < F_{T_{X_2}}$ must be true. Note that $F_{T_{Y_i}}$ is constant, irrespective of the defection of the nozzle.

$$F_{T_i} = \sqrt{F_{T_{X_i}}^2 + F_{T_{Z_i}}^2 + F_{T_{Y_i}}^2} \tag{30}$$

Bank-yaw coupling A 10° nozzle deflection is illustrated by Figure 4, which shows that a deflection generates torque around the bank and yaw axes. When the nozzle deflection is 0°, a deflection creates a torque almost purely around the X-axis while a nozzle deflection of 90° generates a torque almost purely around the Z-axis. In the latter case, a reaction to a bank disturbance will initially result in a large yawing moment. Figure 9 shows the coupling between the bank and yaw angle. The negative bank and negative yaw angle are counteracting the movements around P1.



(a) During levelled flight of the drone, the nozzles are deflected such that $\delta_{n_1} > \delta_{n_2}$.

(b) The delta in the deflection results in a negative bank and a negative yaw angle, rotating the drone around P1.

Figure 9: Illustration of bank-yaw coupling due to a difference in nozzle deflection angle between the left and the right nozzle, resulting in negative bank and negative yawing moments. *Angles are illustrative.*

The effect of the bank-yaw coupling is depicted in Figure 10 where the degree of bank-yaw coupling is plotted for various inertia ratios. The coupling is defined by Equation 31.

$$\eta = \frac{\dot{\mu}}{\dot{\nu}} \left| \frac{M_X}{M_Z} \right| \cdot \frac{I_{ZZ}}{I_{XX}} \tag{31}$$

For $\eta > 1$, the bank rate is dominant and for $\eta < 1$, yaw rate is dominant. Similarly, when $\frac{I_{ZZ}}{I_{XX}} > 1$, rotation around the X-axis takes less effort than rotation around the Z-axis and vice versa. The plot shows that by an increasing inertia ratio, the coupling value moves up for the same deflections. In other words, when it becomes harder to accelerate around the yaw axis compared to the bank axis, an asymmetric nozzle deflection becomes more effective in bank than in yaw, along the entire scope of deflections.

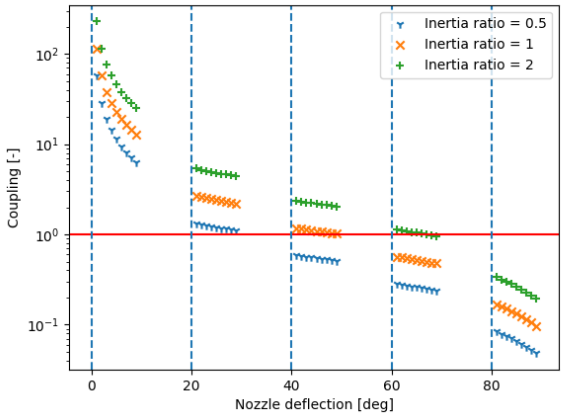


Figure 10: Bank-yaw coupling effects for various inertia ratios. The dotted vertical lines denote the right nozzle deflection that remains constant during the following 9 left nozzle deflection. When left and right nozzle deflection are equal, there is no coupling and therefore this point is not shown in the graph.

3.1.3 Controller design

An attitude controller was designed to control the attitude of the drone and deal with internal and external disturbances, based on the goals of the controller and the flight characteristics discussed in subsubsection 3.1.1 and subsubsection 3.1.2. Internal disturbances come from within the system while external disturbances relate to e.g. wind. The controller must correct for the nozzle efficiency and thus computes the commands in thrust percentages which are converted to deflection angles in the pressure frame, see Equation 32. Finally, the nozzle deflection was converted into the body reference frame followed by a servo command $[-1, 1]$ which is sent directly to the servos.

$$\delta_n = \sin^{-1}(T) \tag{32}$$

$T\epsilon[0, 1]$ is the percentage of upward thrust and δ_n acts in the pressure frame. The controller assumes $\nu = 0$. Remember that in contrast to fixed-wing or multicopter UAVs, the pitch angle is not controlled but the symmetric nozzle deflections are controlled instead. For this research, this is denoted as the thrust command, expressed in vertical thrust. A PID controller is implemented to compute the thrust and bank angle commands. Figure 11 shows the control diagram of the attitude controller.

The thrust command is directly fed forward to the mixer as no feedback for its position in the pressure frame is necessary. The first goal of longitudinal control is reached by subtracting the pitch angle from the servo command such that the nozzle deflection remains constant, independent of the pitch angle. In order to reach the second goal, a gain on the pitch rate (D-gain) was introduced. However, due to non-minimum phase, a $\Delta\delta_n$ creates two opposing moments around the y-axis. It is expected that the effect of the changing moment is greater than the non-minimum phase due to the flexibility of the hose.

The bank angle controller works using a cascaded loop method where the outer loop computes the error between the setpoint and the estimated bank angle that, multiplied by a P-gain, generates a rate setpoint. The rate setpoint is fed forward to improve the response of the system. The inner loop calculates the error in rates and multiplies this by a gain (D-gain) and adds an integrator (I). For each nozzle, the deflection is calculated to execute half of the bank command to reach the desired angular rate. While hovering, the azimuth angle α is determined by the relative position to P1. Therefore, α is not controlled and is a side-effect of the rotation around P1 due to lateral accelerations. The thrust and bank angle commands are added in the mixer before being converted into the servo command, see Equation 34.

$$P(\mu) = \dot{\mu}_{sp} \cdot P + \dot{\mu}_e \cdot D + I \tag{33}$$

$P(\mu)\epsilon[-1, 1]$ is the positive or negative bank angle command in percentages fed to the mixer. The mixer adds the bank and thrust commands (in percentages) before converting the commands into body frame deflections of the nozzles, where nozzle efficiencies apply. Equation 34 shows the mixer.

$$mix_s = \pm\frac{P(\mu)}{2} + T \tag{34}$$

Here, $mix_s\epsilon[0, 1]$ where 0 corresponds to horizontal deflection along X and 1 corresponds to vertical deflection along Z. In Equation 35, the pitch and bank commands are converted into deflections in the pressure frame. Next, the pitch rate, multiplied by gain P3, and the pitch angle are subtracted. Additionally, the bias (β) of the servo in its initial state (deflection in the pressure frame at $PWM = 1500$) must be subtracted after which the command is converted into the servo command $C\epsilon[-1, 1]$ by dividing it by σ, the angle between $PWM = 800$ and $PWM = 2200$ of the servo. This is shown in Equation 35.

$$C = \frac{\cos^{-1}(mix_s) - \dot{\theta} \cdot P_3 - \theta - \beta}{\sigma} \tag{35}$$

3.2 Performance analysis

Once the gains of the controller are tuned, the performance of the UHAV system was analysed based on its longitudinal and lateral flight performance. Stable flight is paramount before implementing the position controller. Based on the following metrics, the ability of steady, stable flight was analysed:

- Bank angle setpoint tracking

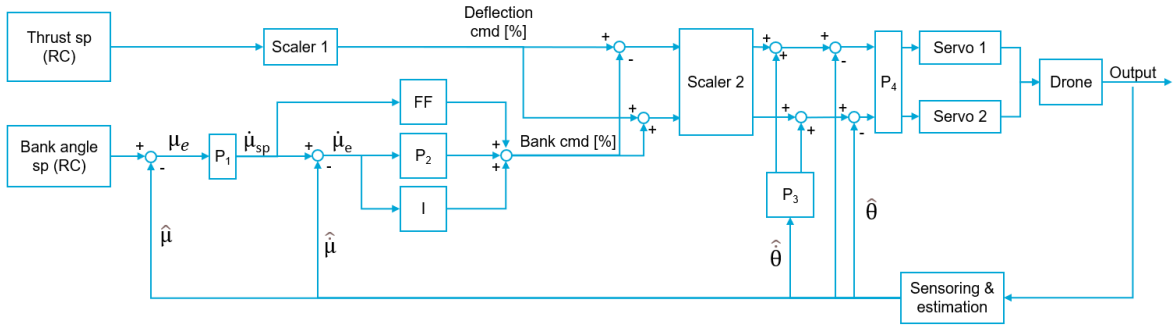


Figure 11: Control diagram of the inner loop attitude controller.

- Pitch angle disturbance rejection
- Bank angle disturbance rejection

Based on the performance of the above metrics, design solutions were proposed to improve the UHAV system on all three metrics. Design choices were tested and analysed and based upon their improvement to the system, a trade-off was made. The most promising design solution was developed and compared with the current design.

3.3 Design trade-off

Design choices are evaluated on four criteria that are weighted on their significance for this research. Based on the performance on each of the criteria and its weight factor, the iterated design was chosen. The four trade-off criteria are:

- **Flight performance:** a qualitative analysis on the stability and controllability of the drone.
- **Manufacturability:** the degree to which the iterated design can be effectively manufactured and repaired.
- **Cost:** the material and labour cost of producing a single version of the iterated design.
- **Mass:** the total mass of the resulting design.

Table 2 shows the different trade-off criteria. Flight performance is the most important metric as it is paramount for the success of the drone to have stable and controlled flight in the presence of disturbances. For longitudinal motion, 3 points are awarded when the solution contributes to more stable flight performance. 2 points are given when no significant improvements are made and 1 point is awarded when the solution has a negative effect. For lateral motion, 5 points are scored when bank and yaw motion are decoupled. If the new design results in positive coupling, i.e. coupled bank and yaw moments both cause a rotation in clockwise/ counter-clockwise direction around P1, 4 points are scored. 3 points are awarded for a design with positive bank-yaw coupling but with complex or unknown side-effects. When the bank-yaw coupling is not affected but the design is improved through another way, this results in 2 points. 1 point is awarded if no improvement is made.

Manufacturability has a weight of 3 because it has an impact on the timeline of the product development, both in terms of building, testing, repairing, and iterating the design. If the design can be 3D printed, it scores 3 points. When the parts can be bought off the shelf, it scores 2 points and when the parts must be custom made by external parties it scores 1 point.

Cost is split into two parts: (i) labour cost and (ii) material cost. Looking at labour costs, the design will score 3 points if it only requires software updates with the option of minor hardware adjustments. When both the software and hardware needs significant updates, it will score 2 points. When additionally, mechanically complex moving parts or major software software improvements in need of extensive research need to be designed, built and tested, the iteration scores 1 point. Looking at material costs, 2 points are given to a design of which only needs 3D printed or off-the-shelf products and 1 point is scored when expensive custom made hardware is needed. The scores of both parts are added for the final score.

An estimation of the mass of the iterated design is compared with the mass of the initial design. Based on the percentage in- or decrease, points are scored. Define m_1 and m_2 as the masses of the initial and the proposed design solution, respectively. 5 points are awarded if $m_2 < m_1$. The design scores 4 points if $m_1 \leq m_2 < 1.05m_1$. 3 points are awarded when $1.05m_1 \leq m_2 < 1.10m_1$, 2 points for $1.10m_1 \leq m_2 < 1.20m_1$, and finally, 1 is awarded if $m_2 > 1.20m_1$.

Once all the scores are determined, they are multiplied by the weight factor and added per design. The design with the highest score is analysed, designed, built and tested and its results are presented in section 6.

Table 2: Weight distribution of the trade-off criteria

Category	Weight
Performance	5
Manufacturability	3
Cost	3
Mass	1

4 TRADE-OFF

In subsection 4.1, the results of the initial design are discussed, based on the performance metrics from subsection 3.2. The performance led to possible solutions which are touched upon in subsection 4.2. Finally, the trade-off is made in subsection 4.3.

4.1 Results initial design

The results of the initial design are shared based on the four categories discussed in subsection 3.2. First, the longitudinal stability is covered, followed by the lateral stability.

4.1.1 Longitudinal stability

The thrust setpoint deflects the nozzles based on its RC input and σ was tuned such that the nozzle remained fixed in the pressure frame in the presence of pitch angle perturbations. To determine the sign and magnitude of the D-gain, a step input was simulated by cutting away a mass attached to the drone to measure its response in pitch angle. For this experiment, D-gains in the range $D = [-0.2; 0.2]$ were set. The airframe was hung freely from the ceiling, unattached to the hose and hence, no thrust applied. For $D \leq 0$, the pitch angle oscillation did not converge and the experiment needed to be stopped after several seconds to avoid damage to the airframe. For $D > 0$, the pitch angle converged, the results are shown in Table 3. T_r is the rise time, η is the percentage overshoot, T_s 15 is the settling time of the system for which the pitch angle is within 15% of the setpoint, and ζ is the damping ratio. $D = 0.15$, $D = 0.2$, and $D = 0.3$ were promising results for the flight test.

Table 3: Step response characteristics of the converging oscillations.

D [-]	T_r [s]	η [%]	T_s 15 [s]	ζ[%]
0.1	0.13	77	16.1	1.2
0.15	0.12	72	4.2	5.0
0.2	0.12	65	3.0	14.4
0.3	0.11	115	7.6	10.8

The experiment demonstrated that a positive D-gain was necessary for converging pitch oscillations. Figure 12 shows that the airframe can be broken down into two separate systems. The first is the rigid carbon fibre airframe corresponding to the slower oscillation but larger in amplitude. The second consists of the two flexible secondary hoses which correspond to the faster but low amplitude vibration which can be seen at $t \approx 10$. Airframe and secondary hose oscillations can become in phase, amplifying each other. This happened for negative D-gains resulting in a diverging pitch angle of the airframe. Because the experiment showed that two oscillations are separately visible, a third goal of the controller with respect to longitudinal motion is introduced. The goal is to limit the effect of the disruptive interaction between the rigid airframe and the flexible secondary hoses.

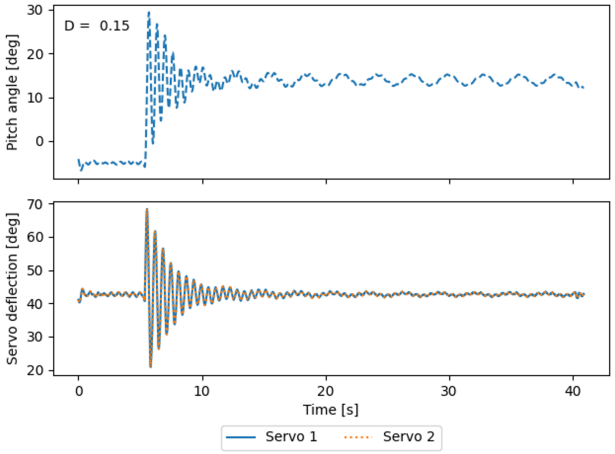


Figure 12: Servo and pitch angle response after a step input of cutting away a mass. Servo deflections are presented in the pressure frame.

The flight test confirmed that a positive D-gain on the pitch angle was necessary for stable flight. Figure 13 illustrates disturbance rejection with a positive D-gain.

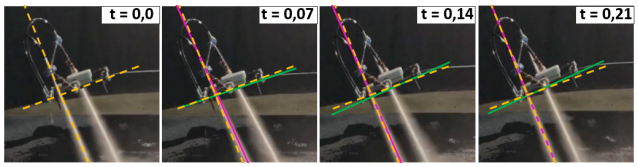
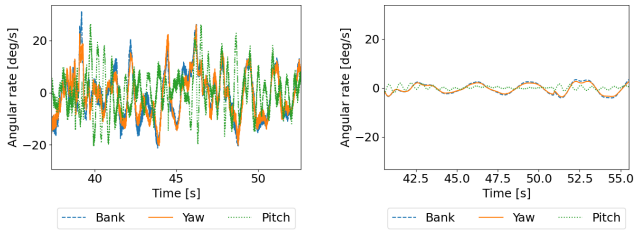


Figure 13: Time domain illustration of a pitch rate oscillation. The orange dotted line is the reference line that remains unchanged while the solid lines are aligned along the frame and nozzle deflection. At $t = 0.07$, the green line indicates that the pitch angle was decreased and the purple line shows that the nozzle reacted through a forward deflection which caused the drone to pitch up. This is seen at $t = 0.14$ while the nozzle is still deflected forward. The nozzle returned to its original position at $t = 0.21$ but the pitch angle had overshoot. The system was not damped and remained in oscillation.

Figure 14 compares controlled with uncontrolled hovered flight. Figure 14b shows a more stable attitude compared to Figure 14a. This demonstrated that in order to realise the third goal, the controller should not actuate the nozzles because there was no actuator that could dampen the disruptive vibrations of the secondary hoses. However, this is in contradiction to the first goal of the controller where actuation is necessary to comply with the thrust setting and not to flip over in the presence of disturbances.



(a) Bank, pitch, and yaw angular rates during hovering with pitch and without bank control.

(b) Bank, pitch, and yaw angular rates during hovering without flight control.

Figure 14: Comparison of flight performance between uncontrolled and controlled hovered flight.

4.1.2 Lateral stability

Figure 10 shows that the inertia ratio was a determining factor for the performance of the lateral attitude stability. The degree of bank-yaw coupling was tested during a flight test where a 10° bank angle step input was given. Figure 15 shows the result of the flight test. In Figure 15a the initial nozzle deflection setpoint was set at 40° (64% upward thrust) and it can be seen that the bank angular rate was equal or slightly higher than the yaw angular rate. Figure 15b shows that the yaw angular rate was higher than the bank angular rate when the nozzles were deflected 65° (91% upward thrust). Although the yawing motion was damped by the hose due to its stiffness (rotational spring characteristic), it resulted in a satisfactory stabilising result. Additionally, the rotational inertia of the drone around the z-axis was not high enough to limit yawing motion. During the experiment, the drone was flying close to the ground which means that with less than 64% thrust, the drone could not fly. Hence, dominant bank motion did not exist in its flight envelope with the current amount of thrust.

In Appendix A:, the results of an experiment with a short tether are showcased. The reduced mass of the hose resulted in the ability of hovered flight with lower upward thrust. A manual bank input was given with the RC which resulted in dominant bank motion for a nozzle setting of 35°.

Due to the coupling effects, the drone was unable to reject disturbances throughout its entire flight envelope without causing unwanted yawing motions. A selected angular setpoint could not be reached before the yawing angle interfered with the attitude of the drone.

4.2 Proposed solutions

With the current design, it was proven that stable, controlled flight in the presence of internal and external disturbances was not possible. In this subsection, solutions to the problems are discussed to improve the stability of the drone. subsubsection 4.2.1 discusses solutions to improve the longitudinal stability while subsubsection 4.2.2 covers solutions to reduce the coupling effects of bank commands.

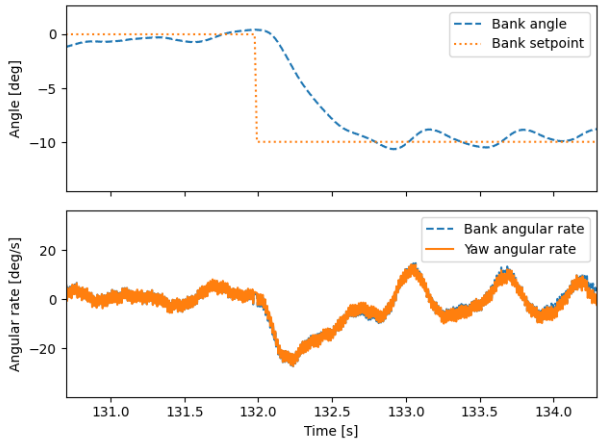
4.2.1 Longitudinal stability

Subsubsection 4.1.1 shows the shortcomings of the system in pitch behaviour. The three goals of the controller have opposing interests which made the design of a controller complicated. The following three solutions are discussed and evaluated in this subsection:

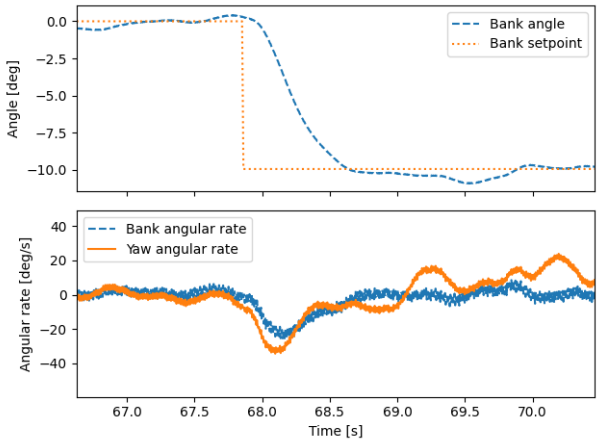
1. Reduction of weight above axis of rotation
2. Low-pass filter on pitch angle
3. Lower P-gain on the pitch angle

The first solution deals with reducing the effect of the secondary hoses. By decreasing or lowering the weight, it is expected that the non-minimum phase characteristic and the interaction of the flexible secondary hose onto the rigid airframe is reduced. The latter two solutions also try to reduce the interaction between the two oscillating systems. Figure 14 indicates that delay in the nozzle deflections created a more stable system in the short term, while complying with the primary goal of the controller (constant nozzle deflection in the pressure frame) in the long term. Firstly, this can be done by applying a first- or second-order low-pass filter to the pitch angle. Secondly, by introducing a P-gain smaller than 1 to the pitch angle and adding an integrator to remove the steady state error between the nozzle deflection and its setpoint. The results of all three solutions are presented below.

Weight reduction As discussed in subsubsection 4.1.1, the mass of the flexible hose affected the behaviour of the pitch angle. While the weight adds an additional vibration into the system, Table 3 shows that it could also be used to dampen the oscillation. Ideally, the hose would be incorporated into the rod connecting the two nozzles, removing the (flexible) weight above the axis of rotation. A control law would have to be designed that reacts to the vibrations of the rigid airframe only. However, weight reduction would reduce the impact of a rotation of the nozzles. The use of a different material of the secondary hose resulted in a weight reduction of 47%, without losing its pressure rating. Additionally, this more flexible hose could be attached to the airframe which reduced the weight acting above the axis of rotation and reduced its flexibility. The step responses of the converging motions for different secondary hose configurations and D-gains on pitch angle are shown in Table 4. The same experimental setup as described in subsubsection 4.1.1 was used.



(a) Bank yaw coupling effect during a 10° bank angle step input. The thrust setpoint was 64% which corresponded to a nozzle deflection of 40° while the pitch angle was 45°. $\eta = 1$.



(b) Bank yaw coupling effect during a 10° bank angle step input. The thrust setpoint was 91% which corresponded to a nozzle deflection of 65° while the pitch angle was 26°. $\eta \approx 0.55$.

Figure 15: Validation of the bank-yaw coupling for two different initial nozzle settings that illustrate the change in coupling.

Table 4: Step response characteristics of the converging oscillations for different secondary hose configurations and D-gains.

	D [-]	**T_r [s]**	**η [%]**	**T_s 15 [s]**	**ζ[%]**
Heavy	0.1	0.13	7	16.1	0.6
	0.15	0.12	72	4.2	5.0
	0.2	0.12	65	3.0	6.9
Light	0.1	0.12	111	-	1.4
	0.15	0.12	11	7.1	4.2
	0.2	0.13	102	3.2	5.6
Light + tie-wrap	0.0	0.12	163	-	0.3
	0.1	0.12	117	-	1.4
	0.15	0.11	148	-	3.5
	0.2	0.11	145	30.6	3.6

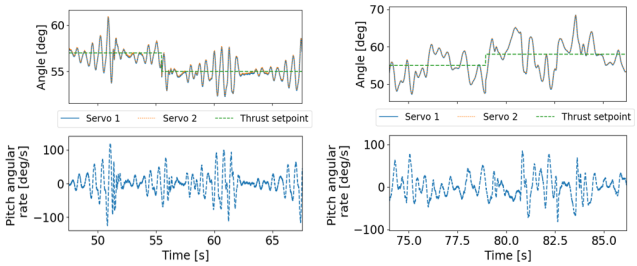
The results show that the heavy secondary hose performed best. As the deflection was more effective with more weight, the oscillation was damped more effectively. Again, the non-minimum phase characteristic was not sufficient to have an impact on the lighter hose. Using the lightweight hose, the settling time marginally increased. Tie-wrapping the secondary hose onto the airframe to further decrease its weight above the axis of rotation showed to be effective, however, resulted in a lower damping ratio. The reason for this must be further analysed, however, a possible explanation is the loss of energy. Vibrations are damped when energy from the system dissipates. Increasing the flexibility of the hose, losses due to mechanical and aerodynamic forces increase when the hoses are bent and moved through the air. The heavy configuration scored 2 points (per definition) on flight performance and as the performance of the light secondary hose was comparable with the initial hose, this configuration also scored 2 points. Tie-wrapping the hose onto the airframe showed not to be beneficial and therefore received 1 point. By integrating the secondary hose into the rigid airframe, the pitch rate of the airframe was not disturbed by the secondary hoses. Therefore, three points were awarded to this configuration.

In terms of manufacturability, the heavy nozzles can be ordered off-the-shelf, resulting in 2 points. The light hose including the nozzles were manufactured in the work place which added the flexibility of repairing and adjusting the nozzles for optimal thrust. Therefore, a score of 3 points were awarded for the latter two designs. The integrated hose scored 1 point as it will require custom made parts to enable this configuration.

The heavy, lightweight, and tie-wrapped hoses scored 3 points on labour cost as minor adjustments were necessary. The integrated nozzle scored 1 point as the water supply for the nozzles has to be redesigned. For material cost, the heavy nozzle and integrated hoses scored 1 point as the cost is more than 10 times higher compared to the light hose. The lightweight hose scored 2 points. Therefore, the four designs scored 4, 5, 5, and 2 points in order of appearance for the cost. The light and tie-wrapped hose scored 5 points on mass as $m_2 < m_1$, whereas the heavy hoses scored 4 points as the total weight of the airframe remained unchanged. Because solid and custom made parts need to be designed that can withstand the high pressure of the water, it is expected that the weight increase and thus 4 points were awarded.

Table 5: Trade-off table for the secondary tether configuration.

Category	Heavy	Light	Light + tie-wrap	Integrated
Performance, 5	2	2	1	3
Manufacturability, 3	2	3	3	1
Cost, 3	4	5	5	2
Mass, 1	2	5	5	4
Total	**30**	**39**	**34**	**31**



(a) Step input response of a thrust step input for a first-order low-pass filter with cutoff frequency of 0.7 Hz.

(b) Step input response of a thrust step input for a second-order low-pass filter with cutoff frequency of 1.0 Hz.

Figure 16: Comparison of pitch angle and nozzle deflection behaviour between a first- and second-order low-pass filter on the pitch angle.

Low-pass filter on pitch angle For filter design, it is important to choose a filter that eliminates the disturbing frequencies. Flight data was analysed using the Fast Fourier Transform to choose the cutoff frequency of the low-pass filter. First- and second-order Butterworth low-pass filter were applied to flight data to compare the results in order to build a suitable filter. Equation 36 and Equation 37 denote the transfer functions in the Laplace domain of a first- and second-order low-pass filter respectively [43].

$$H(S) = K\frac{\omega_c}{s+\omega_c} \tag{36}$$

$$H(S) = K\frac{\omega_c^2}{s^2+2\zeta\omega_c s+\omega_c^2} \tag{37}$$

Where K is the gain, ω_c is the cutoff frequency, and ζ is the damping ratio. The second-order filter has a steeper cutoff frequency but also results in more phase lag which is undesirable. The results of two flight tests with the implemented low-pass filters are presented in Figure 16. Filtering out the desired frequencies also added phase lag and its combination with the unresponsiveness to the higher frequencies created an unstable motion. For a high cutoff frequency, the system was amplified. A low cutoff frequency resulted in a slow system that was unable to respond to the high-frequency disturbances. Note that the sudden damping in Figure 16a was due to a manual interference to prevent a crash from happening.

Lower P-gain on pitch angle The initial controller used a P-gain on the pitch angle of 1. Lowering the P-gain reduces the deflection due to a change in pitch angle. As a result, the deflection of the nozzle will no longer remain fixed in the XZ-plane. Adding an integrator to the nozzle deflection angle removes this steady state error. This way, the controller reduces the relative deflections between the airframe and the secondary hoses in the short term, while maintaining a constant nozzle angle in the long term. The I-gain determines the speed at which the nozzle converges to its desired angle. Figure 17 shows the results of the introduced time delay. In contrast to the low-pass filter, the nozzles reacted to the high-frequency vibrations, however, a slow response to the thrust step input realised calmer behaviour as compared to Figure 14a.

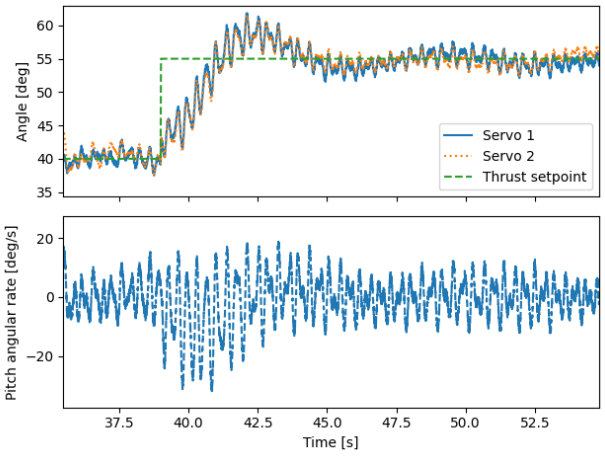
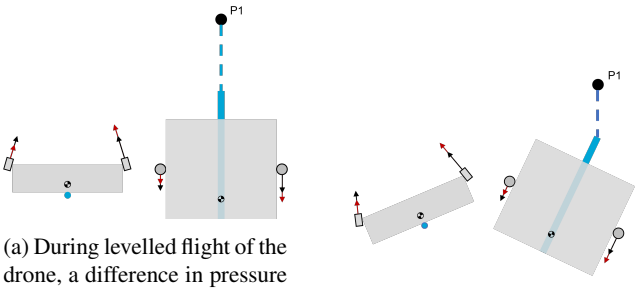


Figure 17: Step response of servo deflections and pitch angular rate for a thrust increase of 15°, or 27%. The added time delay resulted in calmer pitch rates compared to Figure 14a.

Manufacturability, cost, and mass are irrelevant for the trade-off between the implementation of the low-pass filter and the artificial delay on the thrust response. The flight performance was reduced due to the implementation of the filters while the artificial time delay improved the longitudinal stability of the drone. Therefore, the latter software iteration was implemented.



(a) During levelled flight of the drone, a difference in pressure is generated resulting in an increased thrust of the right nozzle and a decreased thrust in the left nozzle.

(b) The pressure difference results in a negative bank and a positive yaw angle, rotating the drone around P1.

Figure 18: Illustration of bank-yaw coupling due to a difference in pressure between the left and the right nozzle, resulting in negative bank and positive yawing moments.
Angles are illustrative.

4.2.2 Lateral stability

The other problem that caused an uncontrollable and unstable system is the bank-yaw coupling. By creating a positive relationship between the bank and yaw movements such that both bank and yaw tend to rotate around P1 in the same direction, or by decoupling the motions, controlled flight is possible. Four solutions are covered in this subsection, however, many more solutions exist to address this problem. The proposed solutions are:

- Pressure differentiation
- Decreased inertia ratio
- Additional degree of freedom for both nozzles
- Canting keel

Pressure differentiation With pressure differentiation, one can regulate the mass flow between the right and the left nozzle. Looking at Equation 3, increasing the mass flow increases the thrust and vice versa. It was shown by Equation 6 and Equation 7 that deflections exert a moment onto the airframe. With pressure differentiation, the deflection remain the same but the magnitude of F_{T_1} and F_{T_2} changes. When $F_{T_2} > F_{T_1}$ and $\delta_{n_1} = \delta_{n_2} = 90°$, $M_X > 0$ and $M_Z = 0$ which means that the motion is purely around the X-axis. For $F_{T_2} > F_{T_1}$ and $\delta_{n_1} = \delta_{n_2} = 0°$, $M_X = 0$ and $M_Z < 0$ which means that the motion is purely around the Z-axis. The coupling effect is illustrated in Figure 18. A negative bank acceleration corresponds to a positive yawing acceleration of which both rotate around P1 in clockwise direction. The Coriolis effect of the water inside the secondary hoses is neglected [11] . Based on this analysis, this solution scored 4 points on flight performance. Partial decoupling of bank and yaw motion is achieved and pressure differentiation results in positive coupling.

As custom parts need to be assembled for this design, manufacturability scored 1 point. In terms of cost, labour cost scored 1 point as both software and hardware updates are necessary. The hardware consists of complex mechanical parts that need to be reliable and controllable under high pressure in a confined space. Material cost scored 1 point as the materials need to be custom made from high-grade materials. This led to a total score of 2. The mass of the iterated design consists of an extra servo and the mechanical parts that can regulate the mass flow under high amounts of pressure. The additional mass was estimated to be 6%. This led to a score of 3 points.

Decreased inertia ratio Increasing the rotational inertia of the drone around the body's z-axis while reducing or keeping the rotational inertia around the body's x-axis constant, reduces the dominance of the yawing motion due to a difference in nozzle deflections. One way of doing this, is by increasing the length of the drone. Figure 19 shows the result of a flight test with an extended frame in x-direction. The increased inertia around the body's z-axis showed an improved effect on the bank-yaw coupling. When comparing Figure 19 with Figure 15, one can see that the extended drone showed a dominant bank rate while the initial design showed equal rates for less thrust. The moment of the hose on the drone was also increased, which increased the dominance of oscillations in the hose. Yamaguchi et al. [29] described a lightweight method to mechanically suppress oscillations in the hose of a UWAV. However, this adds unwanted complexity and weight to the design of the drone. Additionally, the rotational inertia is dependent on the pitch angle of the drone creating complexities in controller design. Based on the flight performance, the design scored 1 point because the advantage of the increased inertia ratio affected the moment of the hose such that distur-

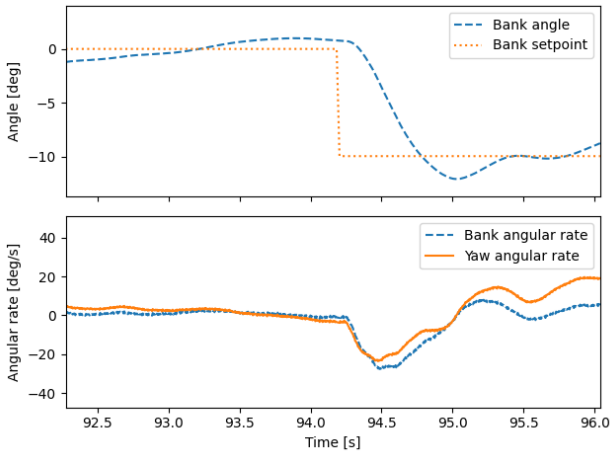
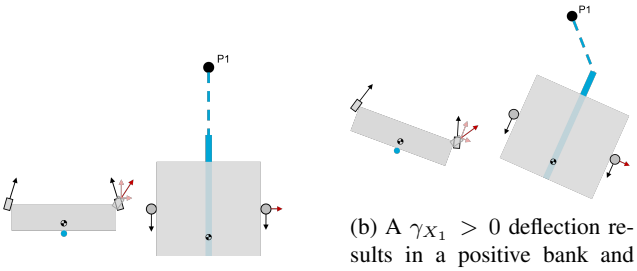


Figure 19: Bank angle step input response for the extended drone in x-direction, a thrust setpoint of 43°. The pitch angle during this bank angle was 26°. $\eta \approx 1.2$.



(a) During levelled flight, nozzle 1 is deflected around the X-axis affecting the forces in Z- and Y-direction.

(b) A $\gamma_{X_1} > 0$ deflection results in a positive bank and yawing motion. The additional force in the Y-direction creates an acceleration in the positive Y-direction.

Figure 20: Illustration of bank-yaw coupling due to a $\gamma_{X_1} > 0$ deflection. The bank angle and lateral acceleration create a CCW rotation around P1, while the yaw angle creates a CW rotation around P1.
Angles are illustrative.

bances interfered increasingly with the attitude of the drone.

The frame could be extended by using larger carbon rods that connected the front with the back of the airframe. Therefore, 3 points were awarded to manufacturability. For labour and material cost, this design scored 3 and 2 points, respectively. Software did not need to be adjusted and minor hardware changes were made. The extra material for the base frame was negligible. Extending the base frame to twice its length, increased the mass of the drone by approximately 6%. This led to a score of 3.

Additional DOF for both nozzles When the nozzles can rotate in- and outward, forces in the Y-direction can be directly controlled. Liu and Zhou [11] investigated a UHAV with thrust-vectoring around 2 axes where the hose was attached to the center of mass and pointed downward. In contrast to the UHAV, the nozzles were located below the center of gravity.

Let γ_{X_i} be defined as the i-th nozzle deflection angle around the X-axis. A free body diagram can be drawn to evaluate the moments around the center of mass of the drone. Figure 20 illustrates a $\gamma_{X_1} > 0$ deflection and its affect on the attitude of the drone. Similar to the initial design, in case $\delta_n = 90°$ a deflection of γ_{X_i} is in terms of thrust more effective in the Y-direction than in the Z-direction. This is illustrated by rotating Figure 4 to the YZ-frame. M_Z and a_y are dominant compared to M_X, restricting effective bank torques. As such, bank disturbance rejection will result in large lateral accelerations which complicates navigation in confined spaces and sensitive position control.

A rotation of the nozzle around the Z-axis does not affect the force in Z-direction and hence a bank torque. However, as the force in X-direction decreases, the yaw angle will be decreased. This is beneficial for the rotation around P1. As the bank-yaw coupling is not decoupled nor converted into positive coupling, this design scored 2 points.

As the construction of the new design consists of the integration of two additional servos and several moving mechanical parts that have to be 3D printed, the manufacturability scored 2 points. In terms of labour cost however, the design scored 1 point as a mechanical design needs to be designed, built, and tested to withstand the forces that are related to drone flight under high pressure. As mentioned, the material cost consists of two servos and 3D printed parts and as such, 2 points were scored which adds up to 3 points for cost. It was estimated that the mass of the airframe will be increased by 14% which corresponded to a score of 2.

Canting keel Weight displacement is a simple, yet effective method for flight control. By shifting the cg while the center of lift remains unaltered, a moment is created. To manage the weight displacement, a solution was inspired by offshore sailing vessels which are often equipped with a canting keel. By deflecting the keel to windward, the righting moment of the keel is increased, decreasing the roll angle [44]. In Figure 21, a schematic drawing of the canting keel is shown. The torque ($\boldsymbol{\tau}_k \epsilon \mathbb{R}^3$) generated by the keel can be split into two parts. The moment of the counterweight hung at the end of the keel and the reaction torque of the servo deflecting the keel, given by Equation 38.

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_w + \boldsymbol{\tau}_s \tag{38}$$

Where $\boldsymbol{\tau}_w(t)$ is the torque created by the counterweight acting around the pressure frame X-axis and $\boldsymbol{\tau}_s(t)$ is the torque created due to an angular acceleration of the keel (α_k) around the body x-axis. As the weight is below the axis of rotation, when the servo rotates in CCW direction, $\boldsymbol{\tau}_s < 0$ and at this position $\boldsymbol{\tau}_w < 0$, which means this is a minimum phase system. Both the deflection itself and the counterweight cause torques into the same direction which is desired for disturbance rejection and waypoint tracking.

$\boldsymbol{\tau}_w$ acts around the X- and Y-axes. When placing the canting keel in front of the center of mass of the airframe, this will have a pitch down moment on the drone. However, the center of mass of the airframe will move down which means that the stabilising moment of the nozzles will increase (see Figure 7) in combination with increased non-minimum phase characteristics. Additionally, the moment of inertia around the y-axis (I_{yy}) is increased by adding weight underneath the drone. It is expected that this decreases the pitch vibrations.

As $\boldsymbol{\tau}_s$ acts around the x-axis, when $\theta \neq 0$, $\boldsymbol{\tau}_s$ will act around the bank and yaw axes. Like shown in Figure 18, this bank-yaw coupling will work complementary to each other, creating positive coupling for disturbance rejection and waypoint tracking.

The weight of the keel will move the center of gravity of the drone forward and down. Two effects are expected on pitch rate damping. The first is the increase in moment of inertia around the y-axis and secondly, the distance between the nozzles and the center of gravity is increased. This increases the damping moment of the D-gain on pitch.

Based on the first analysis where partial decoupling of bank and yaw motion is achieved (for $\theta = 0$), positive coupling moments are introduced ($\theta \neq 0$), and the pitch rate damping is expected to increase, the canting keel scored 4 points for flight performance. The increased inertia around the y-axis is a positive side-effect.

In terms of manufacturability, this design scored 3 points as a canting keel mechanism can be 3D printed and assembled onto the current airframe. The labour cost scored 2 points as structural software and hardware adjustments are necessary, including the design of a canting keel mechanism. The material costs involve a servo and some 3D printed plastics which resulted in a score of 2 points. Hence, the total score for the cost was 4 points.

Additional mass of the drone includes a third servo and the materials for assembling the keel. The battery pack can be used as the counterweight which means that limited extra weight is necessary. It was estimated that the weight increase is 14% which corresponded to a score of 2.

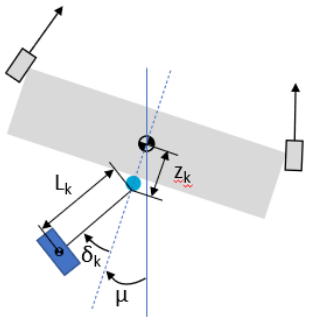


Figure 21: Schematic drawing of the rear view of the canting keel design

4.3 Trade-off

Based on the analysis of the proposed solutions to the longitudinal and the lateral stability in subsubsection 4.2.1 and subsubsection 4.2.2, respectively, the designs were traded off. Table 6 shows an overview of the total scores per proposed design iteration. It can be concluded that the canting keel design has the highest score.

Table 6: Trade-off table for the lateral stability proposes solutions.

Category	**Pressure**	**Inertia**	**DOF**	**Keel**
Performance, 5	4	1	2	4
Manufacturability, 3	1	2	2	3
Cost, 3	2	5	3	4
Mass, 1	3	3	2	2
Total	**32**	**29**	**27**	**43**

5 Iterated Design Analysis

This subsection shares the analysis of the iterated design. The adjustments to the physical design are summarised in subsection 5.1, followed by a description of the dynamics of the actuating mechanism of the canting keel in subsection 5.2. Finally, the flight controller architecture is discussed in subsection 5.3.

5.1 Physical design

In Figure 21, the rear view of the canting keel mechanism attached to the drone is given. Table 7 shows the updated parameters of the iterated design. In Figure 22, the CAD drawing of the new design is given.

Table 7: Updated specifications and dimensions of the drone. Unaltered parameters are left out.

Specifications	Value	
Mass drone without keel (wet), m_d	1.04	[kg]
Mass keel w/ weight, m_k	0.28	[kg]
x location of the keel, x_k	0.33	[m]
y location of the keel, y_k	0.0	[m]
z location of the keel, z_k	0.21	[m]

Figure 22: CAD drawing of the iterated design of the drone.

5.2 *Actuating mechanism*

Longitudinal motion is controlled by symmetric deflections of the nozzles, creating no control torques around the X- or Z-axis, as discussed in subsubsection 4.2.1. The P-gain on the pitch angle was lowered and an integrator was added to the nozzle deflection to create artificial delay. The canting keel is actuated for lateral control. Equation 26 and Equation 27 still hold, however, $F_{T_{X_1}} = F_{T_{X_2}}$ which means that $F_{T_{Z_1}} = F_{T_{Z_2}}$ and no bank-yaw coupling moment was introduced by the nozzles. Moreover, for $\theta \neq 0$, a keel deflection generates a torque around the x-axis, thus creating a moment around the X- and Z-axes. Equation 40 shows that for $\delta_k, \dot{\delta}_k > 0$, $M_X < 0$ and $M_Z > 0$, both contributing to clockwise motion around P1.

τ_w and τ_s are computed according to Equation 39 and Equation 40.

$$\tau_w = m_k g \cdot -\left(L_k \sin(\mu + \delta_k) + z_k \sin(\mu)\right) \tag{39}$$

$$\tau_s = -I_k \dot{\delta}_k \tag{40}$$

Where I_k is moment of inertia of the keel around the axis parallel to the x-axis, passing through the hinge of the keel.

5.3 *Controller design*

The attitude controller of the canting keel is displayed in Figure 23. Equation 41 shows the computation of the keel command. The thrust command was calculated conform Equation 34 and Equation 35 with the bank command set to zero.

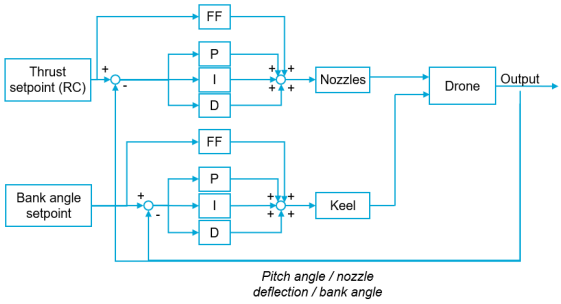


Figure 23: Attitude PID controller for the canting keel design.

$$C_k = \Big[\Big(\sin^{-1}(P(\mu_{sp})) \cdot FF + \sin^{-1}(P(\mu_e)) \cdot P - \sin^{-1}(P(\dot{\mu}_e)) \cdot D + \sin^{-1}(I)\Big) - \beta_k\Big] \cdot (\sigma_k)^{-1} \tag{41}$$

The keel command C_k is constrained such that $|\delta_k| < 45°$ in the body frame. Due to nozzle efficiencies, the inputs for the controller are in setpoint or error percentages, denoted by $P(x)$ with x the corresponding variable. This is converted to radians by the sine function. The integrator, I, is the integrated percentage error over time. β_k is the bias term of the canting keel deflection and σ_k is the total deflection flexibility of the servo in radians which converts the keel command to a servo command.

As the iterated design showed that stable flight, disturbance rejection, and setpoint tracking was possible, position control was introduced. Figure 24 shows the architecture of the position controller. The cascaded outer loop consists of a PID controller that generates thrust and bank angle setpoints that serve as the input for the attitude controller.

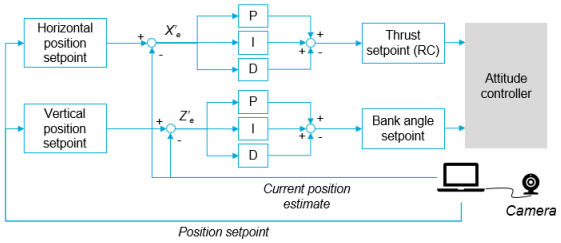


Figure 24: Outer loop PID controller.

Lidar A Lidar sensor was mounted onto the drone to estimate the current height of the drone. Trigonometric relations were used to correct for the pitch and bank angle of the drone. The remote controller sent a feedforward command for the thrust setting after which the PID controller computed its thrust commands using the error between the current altitude and its setpoint. Equation 42 shows the thrust command that was sent to the attitude controller.

$$T_{sp} = RC \cdot FF + h_e \cdot P + \dot{h}_e \cdot D + I \tag{42}$$

T_{sp} is the thrust setpoint in percentage of upward thrust, RC is the remote control setpoint and h_e is the height error in meters. Because the Lidar does not work for $h < 0.4$m, a thrust command to hover above 1m was fed forward by the remote controller for safety reasons. For simplicity, the difference in changing tension force of the hose was not taken into account during climbing or descending but the integrator removed this steady state error. The bank command was generated by the remote controller.

Visual motion capture Computer vision is widely used in the UAV industry to track vehicles, navigate in confined spaces, fire suppression, washing, or visual inspection. Onboard vision can be used for indoor navigation with the advantage of being able to accurately determine its velocity using optical flow [45]. However, during this research, an external camera was used to track the drone and set waypoints for path planning as it does not add complexity and weight to the system, is cheap, reliable, and useful for various flight operations where a broad view of the surroundings is necessary. When a camera with a third person view is used to determine the drone's position or attitude, this is called visual motion capture (MoCap) [46]. Once the drone is tracked, a waypoint inside the camera's view can be set. The difference between both screen coordinates is sent to the flight computer using a secondary MAVLink connection. The ground station converts the pixel frame of the video into a normalised X'Z'-coordinate frame where the top-left corner corresponds to point $[X' = 0; Z' = 0]$ and the right-bottom point corresponds to point $[X' = 1; Z' = 1]$. The thrust command is computed conform Equation 42, however, the feedback is coming from the ground station rather than from the Lidar.

For both the lateral and the longitudinal position, a PID controller was implemented to send attitude setpoints to the attitude controller. The lateral forces that the nozzles exerted onto the airframe were computed according to Equation 43 and their efficiency was used to scale the attitude setpoint that was generated, see Equation 44. It is assumed that X points into the direction of the camera such that Y is always perpendicular to the camera's point of view.

$$F_Y = 2F_T \cdot \sin(\mu) \tag{43}$$

$$\mu_{sp} = \sin^{-1}(X'_e) \cdot P - \sin^{-1}(\dot{X}'_e) \cdot D + \sin^{-1}(I) \tag{44}$$

Where X'_e is the normalised horizontal error between its current X'-location (X') and the X'-waypoint (X'_{wp}). Note that the entire term inside the brackets must be smaller than 1 and $X'_e > \dot{X}'_e$ must hold to not fly away from the waypoint.

Two vision algorithms were tested to track the drone. The Lucas-Kanade optical flow [47] and the YOLOv5n instance segmentation algorithm [48] were tested. The optical flow method was chosen because of its computational efficiency and the drone accelerated within the Lucas-Kanade thresholds to keep tracking the right pixel. The drawback is that a human operator was needed to be present during the operation because the algorithm had to be reset in case of losing the object due to visual obstruction flying out of sight.

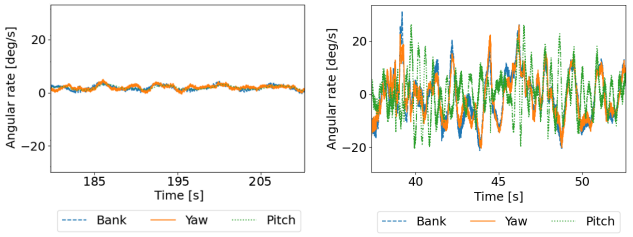
The YOLOv5 segmentation model was trained to detect the water jets using a dataset of 348 images from various test flights. The centroid of the detected shape was used as input for the position controller. Because the shape continuously changed, a second order low-pass filter was used to smoothen the detections of the centroids. This resulted in phase lag of the detection, however, as the drone moved slowly and its goal was not to reach the waypoint as fast as possible, this was acceptable. Once the water was detected, one of the two centroids of the detected segments was used to determine its X'Y'- location which was sent, together with the waypoint, to the flight computer. No human operator was needed for this operation as the location was automatically detected.

6 Results

In this section, the results of the iterated design are displayed. This is done by looking at the performance of the longitudinal and lateral stability in subsection 6.1 and subsection 6.2, respectively. Position control is discussed in subsection 6.3.

6.1 Longitudinal stability

In Figure 25, hovered flight is compared between the initial and the iterated design. Figure 25b shows that the initial design was not able to withstand disturbances coming from the hose, which translated into large bank, pitch, and yaw rates. The current design was able to partially decouple the bank-yaw motion such that internal and external disturbances were damped without creating unwanted instabilities. Figure 25a shows that the iterated design behaved calm while hovering. The yaw angular rate was not controlled but only constrained by the hose and hence, it had the highest amplitude.



(a) Bank, pitch, and yaw angular rates during hovering with the improved attitude controller.

(b) Bank, pitch, and yaw angular rates during hovering with pitch control and without roll control.

Figure 25: Comparison of flight performance between the initial and iterated design. Stable and calm flight was realised by the design iterations.

6.2 *Lateral stability*

Performance of lateral stability was evaluated based on the capability of disturbance rejection and bank angle setpoint tracking. Figure 26 shows the step responses to a bank angle step input for two different pitch angles. Note that the peaks of the bank and yaw angular rates at the moment of the step input have opposite signs. This illustrates the positive coupling (i.e. both rotating in clockwise direction around P1). Compared to Figure 15, both peaks are in equal direction causing negative coupling in opposite directions around P1. As expected, Figure 26a shows that the bank motion for low pitch angles had a very limited effect on the yaw angle because $\tau_s \approx 0$. While in Figure 26b, the effects of the bank-yaw coupling due to keel deflections can be seen. Keel deflections created a torque around the Z-axis, which translated into oscillations into the primary hose. Therefore, banking showed a more jerky behaviour compared to Figure 26a. However, it shows that lateral disturbances could be rejected due to the dominant bank rate.

Comparing the experiments leading to Figure 15 and Figure 26, the pitch angle decreased due to the pitch down motion of the canting keel. Figure 26 shows that this was favourable because the flight performance improved for lower pitch angles.

6.3 *Position control*

This subsection is split into height control using the Lidar sensor and visual control using the external camera.

Height Figure 27 shows the step response of the system for a height step input. In addition, a disturbance was introduced by pushing the drone up and down and the results show that the drone quickly recovered to its setpoint.

The oscillations around the height setpoint was the result of a trade-off between high or low gains on the height error. High gains resulted in fast nozzle deflections and high pitch rates while lower gains increased the delay and height fluctuations but led to slower, steadier, and more robust behaviour of the drone.

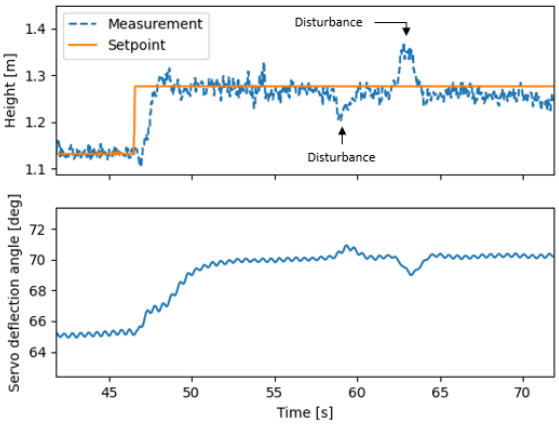
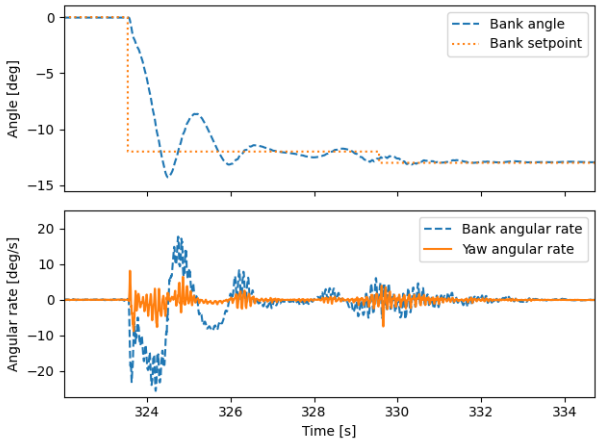


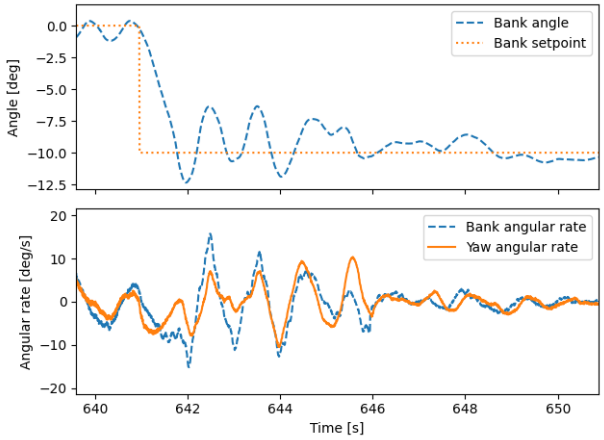
Figure 27: Step input response of a 0.15m altitude increment and external disturbances, using the Lidar.

Visual motion capture In Figure 28, the results of the position controller are shown, using the Lucas-Kanade optical flow tracking algorithm. The horizontal and vertical distances were estimated based on the webcam's factsheet and horizontal distance to the UHAV[9]. With optical flow, it was not possible to track the center of mass of the drone. In order to move into the X'-direction, the drone banked into that direction. However, as the tracker was placed underneath the center of mass, a bank angle into the direction setpoints initially increased the distance between the current position and the setpoint. This was unaccounted for by the controller. In addition, pitch oscillations with constant height $\left(\frac{\partial Z}{\partial t} = 0\right)$, resulted in fluctuations in the Z'-plane $\left(\frac{|\partial Z'|}{\partial t} > 0\right)$. This explains the behaviour of the vertical oscillation in Figure 28. With low gains, it was possible to create a position controller that moved the drone slowly to its position setpoint.

[9] https://www.trust.com/nl/product/18679-trino-hd-video-webcam#downloads — Accessed: 16-12-2022



(a) Step response for a bank angle step input of 13°. The bank rate was dominant as compared to the yaw rate. During this motion, the pitch angle was 5° and a thrust setpoint of 95% which corresponded to a nozzle deflection of 72°. $\eta \approx 4.25$.



(b) Step response for a bank angle step input of 10°. The bank rate was dominant as compared to the yaw rate. During this motion, the pitch angle was 20° and a thrust setpoint of 70% which corresponded to a nozzle deflection of 44°. $\eta \approx 1.9$.

Figure 26: Comparison of the step responses to a bank angle step input for 2 different thrust setpoints, hence 2 different pitch angles.

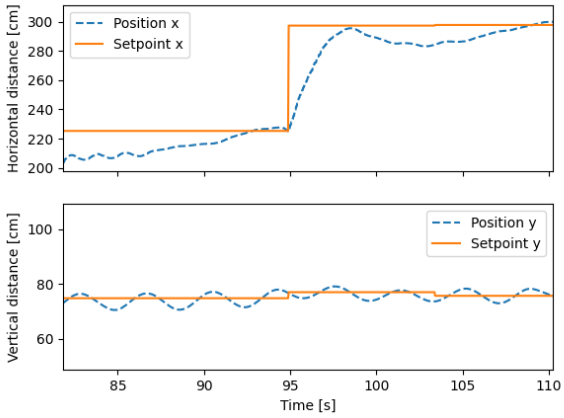


Figure 28: Step input response of a horizontal and vertical distance step, using the Lucas-Kanade optical flow tracking algorithm.

The trained YOLOv5 segmentation model resulted in an mAP50-95 score of 0.745. Running the trained algorithm on a test video, the algorithm detected 891/892 instances with a detection threshold of 0.25. The custom YOLOv5n Segmentation algorithm ran at approximately 7.9 frames per second on an 11th Gen Intel Core i7 CPU.

Figure 29 shows a compilation of the algorithm during a test flight. In Figure 30, the results are displayed. At the location of the centroid, the water jet was approximately 10 centimetres wide and 20 centimetres in vertical direction. With the goal of spraying water onto a target, the margins are included in Figure 30. During the snapshot of 24 seconds, the setpoint was wetted for 71% of the time. This included moving time.

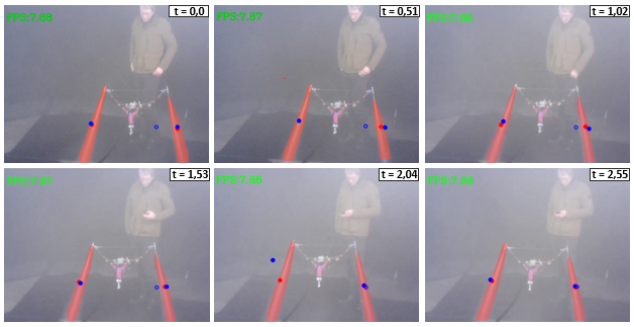


Figure 29: Image series of waypoint tracking using the YOLOv5 Nano segmentation model to detect the water jets. The solid red dots are located at the centroid of each segment, the solid blue dots correspond to the computed X'Z'-location of the water jet as its detected position is passed through a second order low-pass filter. The setpoint is denoted by a blue hollow circle.

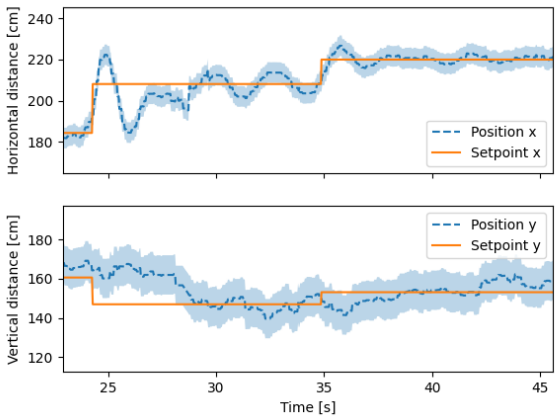


Figure 30: Step input response of a horizontal and vertical distance step, using the YOLOv5-Nano Segmentation model to track the water streams. The marked blue area represents the wetted area around the detected centroid.

Similar to the Lucas-Kanade method, the tracking point was not located at the center of mass of the drone. As can be seen in Figure 30, the centroid of the detected water jets had an offset in the X'- and Z'-plane. A rotation of the drone resulted in a deviation between the distance from the center of mass of the drone and the detected point. This was not taken into account by the flight controller and resulted in fluctuations around the setpoint. Because the offset was much larger compared the offset of the optical flow tracking point, the oscillations were larger.

7 Conclusions

The final goal of the project aims at developing a lightweight and simplistic unmanned hydro-propelled aerial vehicle capable of autonomously detecting a target location to maximise its wetted time. In this research, a prototype of a UHAV flying with a hose length of 3m was tested and evaluated. This led to new insights about the physical design and software architecture. Stable and controlled flight in the presence of disturbances was not possible with the initial hard- and software architecture.

For longitudinal stability, the first goal was to fly with controlled nozzle deflections in the pressure frame for position control along the constraints of Figure 5. Because the pitch angle was not controlled due to the geometry of the system, the second goal of the controller was to dampen the pitch rate that was excited due to nozzle deflections and primary hose oscillations. The third goal was to limit the disruptive interaction between the rigid airframe and the flexible secondary hoses. Integrating the hoses into the airframe would reduce the interaction of the moving parts, however, this was unfeasible in terms of manufacturability and cost. The controller had no means of actively reducing the interaction, other than to stop deflecting the nozzles. A trade-off between the first and third goal had to be made by lowering the P-gain on the nozzle deflection, for short term compliance of the third goal, and adding an integrator to comply with the first goal in the long term. By adding a positive D-gain on the pitch, the pitch rate was damped. The D-gain created an additional stabilising moment around the y-axis. Furthermore, it is expected that the flexibility of the secondary hose increased the mechanical and aerodynamic damping of the pitch rate. Because of the flexibility of the hose, the non-minimum phase effect of nozzle deflections was reduced.

For lateral stability, disturbance rejection and setpoint tracking was impossible throughout the entire flight envelope due to bank-yaw coupling effects of asymmetric nozzle deflections. Proposed solutions to the lateral and longitudinal problems were traded-off, resulting in an iterated physical design and updated attitude controller. The bank-yaw coupling was addressed by the implementation of a canting keel, often used by offshore sailing vessels to create an additional righting moment. It was demonstrated that for $\theta = 0$, the bank-yaw motion was decoupled and for $\theta \neq 0$, actuation of the keel resulted in positive coupling. A deflection of the keel was minimum phase such that both the weight and the reaction torque contribute to the same moment. Because the keel created an additional pitch-down moment, the effect of the bank-yaw coupling was reduced to dominant bank rates throughout its flight envelope as compared to yaw rates. Furthermore, increasing the distance between the nozzles and the center of gravity, increased the additional stabilising moment of the nozzles due to the D-gain on pitch. Simultaneously, the increased moment of inertia around the y-axis reduced pitch rate oscillations.

Finally, it was shown that real-time position control based on visual motion capture, using a webcam, was possible. The Lucas-Kanade (LK) optical flow method, known for its lightweight computational requirements, demonstrated great performance in tracking pixels at the maximum frame rate of 30. The YOLOv5 model was effective in detecting the water jets but the maximum frame rate of 7.9 was much lower than LK. In addition, the LK algorithm provided a more robust signal and the offset of the location tracker to the center of mass of the drone was smaller compared to the detected water jets. Due to the offset, the detected element did not represent the movement of the center of mass. This was not taken into account by the controller.

Future research should focus on the development of a more robust flight controller combined with a strategy to maximise water discharge onto a target area.

To improve the longitudinal stability, pitch rate damping should be improved. The interaction between the secondary hoses and the rigid airframe due to nozzle deflections was disruptive to the pitch rate. Therefore, it is recommended to integrate the the secondary hoses with the airframe. Removing the weight above the axis of rotation eliminates the

non-minimum phase characteristic of the nozzle deflection. Furthermore, the flexible hose does not create disruptive moments around the y-axis. This way, the controller deals with active vibration suppression of the primary hose and external disturbances only. In addition, it is expected that the artificial delay in the thrust setpoint tracking can be removed.

When integrating the secondary hoses with the airframe remains unfeasible, pitch rate damping due to flexible hose movements should be investigated further. Modelling the interaction between the secondary hoses and the airframe could lead to interesting solutions on how to increase pitch rate damping. Furthermore, the implementation of a notch filter is proposed. It was shown that the cutoff frequency of a low-pass filter on the pitch angle could not be set correctly as the pitch oscillations were either amplified or the system was too slow to reject disturbances. However, a notch filter could filter out the vibrations due to the interaction between the secondary hoses and the airframe without a significant phase shift.

Next, it was shown that the flight performance was improved due to the canting keel characteristics, while very limited weight was added. For future designs, optimising the length of the drone and the location and dimensions of the keel could increase natural damping of the pitch rate.

To enhance position control for maximising the water discharge at a specified target, the YOLOv5 detection model should be improved. By generating a larger dataset under various indoor flight conditions (e.g. mist and sunlight), the detection rate and detection threshold can be raised. Combining YOLOv5 segmentation with LK could increase the detection rate of the water jets. Furthermore, it is important that the controller uses the center of mass of the drone for its control strategy. Through the detection of the water jets, the center of mass of the drone can be estimated. By minimising a cost function, a model predictive controller can compute the most efficient flight path to maximise the amount of water delivered at the area of interest. Future research will need to determine whether external vision remains acceptable for its missions or that onboard vision needs to be integrated into the system.

This research has laid the foundation towards autonomous flight of a lightweight and simplistic UHAV. Enhancing the pitch rate damping, refining the water jet detection algorithm and implementing a cost function for water discharge at the area of interest, will enable conducting autonomous missions in the future.

REFERENCES

[1] Ann Cavoukian. *Privacy and drones: Unmanned aerial vehicles*. Information and Privacy Commissioner of Ontario, Canada Ontario, 2012.

[2] Daan Krijnen and Coen Dekker. Ar drone 2.0 with subsumption architecture. In *Artificial intelligence research seminar*, 2014.

[3] Ramy Rashad, Jelmer Goerres, Ronald Aarts, Johan B. C. Engelen, and Stefano Stramigioli. Fully actuated multirotor uavs: A literature review. *IEEE Robotics & Automation Magazine*, 27(3):97–107, 2020. doi: 10.1109/MRA.2019.2955964.

[4] Aditya Intwala and Yash Parikh. A review on vertical take off and landing (vtol) vehicles. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 2(2):187–191, 2015.

[5] Abayomi O Agbeyangi, Joseph O Odiete, and Adam B Olorunlomerue. Review on uavs used for aerial surveillance. *Journal of Multidisciplinary Engineering Science and Technology*, 3(10):5713–5719, 2016.

[6] GCHE De Croon, M Perçin, B Remes, R Ruijsink, and C De Wagter. The delfly. *Dordrecht: Springer Netherlands. doi*, 10:978–94, 2016.

[7] Miles L Loyd. Crosswind kite power (for large-scale wind power production). *Journal of energy*, 4(3):106–111, 1980.

[8] Chunhua Zhang and John M Kovacs. The application of small unmanned aerial systems for precision agriculture: a review. *Precision agriculture*, 13(6):693–712, 2012.

[9] Y Inoue, S Morinaga, and A Tomita. A blimp-based remote sensing system for low-altitude monitoring of plant variables: A preliminary experiment for agricultural and ecological applications. *International Journal of Remote Sensing*, 21(2):379–385, 2000.

[10] Soheil Jafari, Seyed Alireza Miran Fashandi, and Theoklis Nikolaidis. Control requirements for future gas turbine-powered unmanned drones: Jetquads. *Applied Sciences*, 8(12):2675, 2018.

[11] Xiaoming Liu and Hao Zhou. Unmanned water-powered aerial vehicles: Theory and experiments. *IEEE Access*, 7:15349–15356, 2019. doi: 10.1109/ACCESS.2019.2895335.

[12] Jean Michel Pflimlin, Paolo Binetti, Daniel Trouchet, Philippe Soueres, and Tarek Hamel. Aerodynamic modeling and practical attitude stabilization of a ducted fan uav. In *2007 European Control Conference (ECC)*, pages 4023–4029. IEEE, 2007.

[13] Jean Michel Pflimlin, Philippe Souères, and Tarek Hamel. Position control of a ducted fan vtol uav in crosswind. *International Journal of Control*, 80(5):666–683, 2007.

[14] Florent Le Bras, Robert Mahony, Tarek Hamel, and Paolo Binetti. Adaptive filtering and image based visual servo control of a ducted fan flying robot. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1751–1757. IEEE, 2006.

[15] J H Gerken and Glen Bright. Design of a semi-autonomous water-driven flight platform. In *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pages 1–6, 2017. doi: 10.1109/M2VIP.2017.8211500.

[16] Hyungjoo Yoon and Panagiotis Tsiotras. Singularity analysis of variable speed control moment gyros. *Journal of Guidance, Control, and Dynamics*, 27(3):374–386, 2004.

[17] Larry Lipera, Jason D Colbourne, Mark B Tischler, M Hossein Mansur, Michael C Rotkowitz, and Paul Patangui. The micro craft istar micro air vehicle: Control system design and testing. In *Annual Forum Proceedings-American Helicopter Society*, volume 57, pages 1998–2008. AMERICAN HELICOPTER SOCIETY, INC, 2001.

[18] David Bernacchia. Design of thrust vectoring attitude control system for lunar lander flying testbed. Master's thesis, University of Bologna, 2019.

[19] L. Fagiano. Systems of tethered multicopters: Modeling and control design. *IFAC-PapersOnLine*, 50 (1):4610–4615, 2017. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2017.08.653. URL https://www.sciencedirect.com/science/article/pii/S240589631731039X. 20th IFAC World Congress.

[20] Piero Graziano Ioppo. The design, modelling and control of an autonomous tethered multirotor uav. Master's thesis, Stellenbosch: Stellenbosch University, 2017.

[21] JM Winget and RL Huston. Cable dynamics—a finite segment approach. *Computers & Structures*, 6(6):475–480, 1976.

[22] Fangfang Xie, Xiaoning Zheng, Michael S. Triantafyllou, Yiannis Constantinides, and George Em Karniadakis. The flow dynamics of the garden-hose instability. *Journal of Fluid Mechanics*, 800:595–612, 2016. doi: 10.1017/jfm.2016.364.

[23] Jari Hyvärinen, Matts Karlsson, and Lin Zhou. Study of concept for hydraulic hose dynamics investigations to enable understanding of the hose fluid–structure interaction behavior. *Advances in Mechanical Engineering*, 12(4):1687814020916110, 2020. doi: 10.1177/1687814020916110.

[24] Taeyoung Lee. Geometric controls for a tethered quadrotor uav. In *2015 54th IEEE conference on decision and control (CDC)*, pages 2749–2754. IEEE, 2015.

[25] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. *Global Formulations of Lagrangian and Hamiltonian Dynamics on Manifolds*. Springer, 2018.

[26] Dong-Hun Lee, Thinh Huynh, Young-Bok Kim, and Chakir Soumayya. Motion control system design for a flying-type firefighting system with water jet actuators. *Actuators*, 10(10), 2021. ISSN 2076-0825. URL https://www.mdpi.com/2076-0825/10/10/275.

[27] Farhad A Goodarzi, Daewon Lee, and Taeyoung Lee. Geometric stabilization of a quadrotor uav with a payload connected by flexible cable. In *2014 American Control Conference*, pages 4925–4930. IEEE, 2014.

[28] Prasanth Kotaru and Koushil Sreenath. Multiple quadrotors carrying a flexible hose: dynamics, differential flatness and control. *IFAC-PapersOnLine*, 53(2): 8832–8839, 2020.

[29] Tomoka Yamaguchi, Yuichi Ambe, Hisato Ando, Masashi Konyo, Kenjiro Tadakuma, Shigenao Maruyama, and Satoshi Tadokoro. A mechanical approach to suppress the oscillation of a long continuum robot flying with water jets. *IEEE Robotics and Automation Letters*, 4(4):4346–4353, 2019. doi: 10.1109/LRA.2019.2932582.

[30] Tom Kusznir and Jaroslaw Smoczek. Sliding mode-based control of a uav quadrotor for suppressing the cable-suspended payload vibration. *Journal of Control Science and Engineering*, 2020, 2020.

[31] Xiao Liang, Yongchun Fang, Ning Sun, and He Lin. Nonlinear hierarchical control for unmanned quadrotor transportation systems. *IEEE Transactions on Industrial Electronics*, 65(4):3395–3405, 2017.

[32] Hassan Alkomy and Jinjun Shan. Vibration reduction of a quadrotor with a cable-suspended payload using polynomial trajectories. *Nonlinear Dynamics*, 104(4):3713–3735, 2021.

[33] Chang Tai Kiang, Andrew Spowage, and Chan Kuan Yoong. Review of control and sensor system of flexible manipulator. *Journal of Intelligent & Robotic Systems*, 77(1):187–213, 2015.

[34] Ian D Walker. Continuous backbone "continuum" robot manipulators. *International Scholarly Research Notices*, 2013, 2013.

[35] Pal Liljeback, Oyvind Stavdahl, and Anders Beitnes. Snakefighter - development of a water hydraulic fire fighting snake robot. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6, 2006. doi: 10.1109/ICARCV.2006.345311.

[36] Hisato Ando, Yuichi Ambe, Tomoka Yamaguchi, Masashi Konyo, Kenjiro Tadakuma, Shigenao Maruyama, and Satoshi Tadokoro. Fire fighting tactics with aerial hose-type robot "dragon firefighter". In *2019 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*, pages 291–297. IEEE, 2019.

[37] Hisato Ando, Yuichi Ambe, Akihiro Ishii, Masashi Konyo, Kenjiro Tadakuma, Shigenao Maruyama, and Satoshi Tadokoro. Aerial hose type robot by water jet for fire fighting. *IEEE Robotics and Automation Letters*, 3(2):1128–1135, 2018.

[38] ArduPilot. Pymavlink. `https://github.com/ArduPilot/pymavlink`, 2022. Accessed: 2022-12-12.

[39] NASA. Mass flow rate. `https://www.grc.nasa.gov/www/k-12/rocket/mflchk.html`, 2021. Accessed: 2022-08-22.

[40] Rod Deakin. Catenary curve. 2019.

[41] F Behroozi. A fresh look at the catenary. *European Journal of Physics*, 35(5):055007, 7 2014. doi: 10.1088/0143-0807/35/5/055007. URL `https://doi.org/10.1088/0143-0807/35/5/055007`.

[42] Paul Cella. Methodology for exact solution of catenary. *Journal of Structural Engineering*, 125(12):1451–1453, 1999.

[43] Jonathan Dodge. *From Continuous-Time Domain to Microcontroller Code*. UnitedSiC, 10 2018. URL `https://unitedsic.com/appnotes/UnitedSiC_AN0019_a2d.pdf`. Application Note AN0019.

[44] Mark Hobbs and P Manganelli. Measurement of accelerations and keel loads on canting keel race yachts. In *RINA Modern Yacht Conference*, 2007.

[45] Yuncheng Lu, Zhucun Xue, Gui-Song Xia, and Liangpei Zhang. A survey on vision-based uav navigation. *Geo-spatial information science*, 21(1):21–32, 2018.

[46] A Masiero, F Fissore, R Antonello, A Cenedese, and A Vettore. A comparison of uwb and motion capture uav indoor positioning. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:1695–1699, 2019.

[47] Bruce D Lucas, Takeo Kanade, et al. *An iterative image registration technique with an application to stereo vision*, volume 81. Vancouver, 1981.

[48] Ultralytics. Yolov5 in pytorch. `https://github.com/ultralytics/yolov5`, 2022. Accessed: 2022-11-21.

APPENDIX A: BANK-YAW COUPLING

Figure 31 and Figure 32 show the results of a flight test for a manual bank angle input with a tether of 1m. Note that this change in tether length changed the dynamics of the system. The experiment was performed to demonstrate the varying bank-yaw coupling. However, the results cannot be directly compared with the results presented in this paper.

In Figure 31 the initial nozzle deflection setpoint was set at 35° (57% thrust) and it can be seen that the bank angular rate is higher than the yaw angular rate. Figure 32 shows that the yaw angular rate was higher than the bank angular rate when the nozzles were deflected 60° (87% thrust). Although the yawing motion was damped by the primary hose due to its stiffness (rotational spring characteristic), it did not result in a satisfactory stabilising moment. Additionally, the rotational inertia of the drone around the z-axis was not high enough to limit yawing motion.

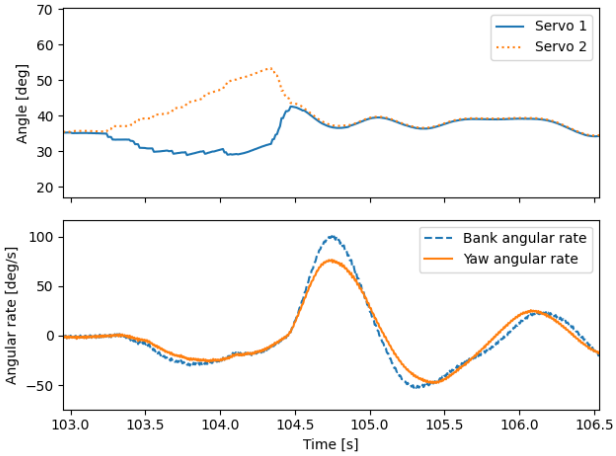


Figure 31: Bank-yaw coupling effect for an initial nozzle setting of 35°.

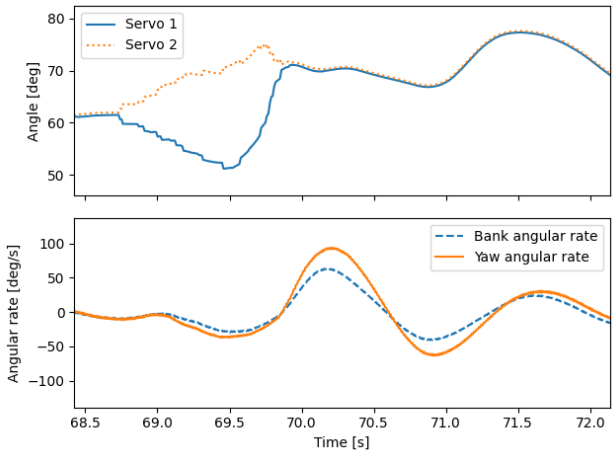


Figure 32: Bank-yaw coupling effect for an initial nozzle setting of 60°.

Part III

Additional Work

4

Data Analysis

Data analysis was at the core of the research. Flight data was collected in order to gain insights into the behaviour of the drone, but also for example for debugging the code. For this, it was important to collect the right data from the PX4 to be able to analyse it. This required the right setup of the SD card inside the flight computer, a converter from ULog files to interpretable csv files, and finally the scripts that visualise the data.

The SD card was used for storing flight logs, firmware version, mission information, and other information. The SD card should be FAT32 formatted in order to be compatible with PX4. Furthermore, a file was added in the `etc/logging/logger_topics.txt` file to replace the automated logged files for custom topics. Chapter 6 discusses several additional topics that were included for analysis. The topics were logged at full rate.

The PX4 community has put effort into making an online analysis tool for end-users that automatically creates interactive plots that can be used for flight analysis. Users can upload their log files onto a web-server which generates these plots. However, the plots were unsuitable for analysis of a custom made vehicle that uses unconventional definitions and actuators, such like the UHAV. In order to create custom files, the `pyulog` package [1] was used to parse ULog files to extract information and convert them into csv files. For this, the `ulog2csv` command was used. Once the excel files were generated onto the computer, multiple Python scripts were built to interpret the information for analysis. Examples are Fast Fourier Transforms (FFT), time-series analyses, first- and second-order low-pass filters, and attitude and actuator plots.

[1] https://github.com/PX4/pyulog | Accessed: 07-03-2022

5

Controller

This chapter focuses on the general architecture of the control system used in the PX4 Autopilot software. PX4 uses PID position and attitude controllers throughout all of its airframes due to their simplicity and robustness. As a foundation, the controller of a fixed-wing aircraft was taken from PX4's Autopilot software (Dronecode, 2022a). The flight software was adjusted such that stable and controlled flight was possible in the presence of external and internal disturbances. This chapter discusses the flight software platform in Section 5.1 and covers the verification of the customised control software in Section 5.2.

5.1. Overview of PX4 Control

This section provides a high-level overview of the PX4 soft-and hardware stack[1]. Figure 5.1 shows a schematic drawing of the system's layout. The flight controller, a PX4 Mini, contains the flight software with all the necessary apps and modules, middleware used for internal and external communication, and drivers. A Wi-Fi module, RC receiver, power unit, Lidar, and 2 servos are plugged into the flight controller. The Wi-Fi module communicates with QGroundControl which runs on the ground station to monitor the flight performance, track the vehicle in the presence of a GPS module, and adjust the settings in real-time when necessary.

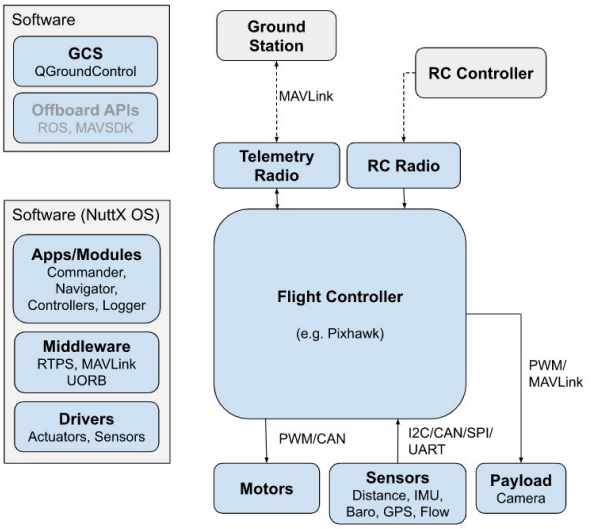


Figure 5.1: High-level overview of the PX4 system architecture [2].

[1] http://docs.px4.io/main/en/concept/px4_systems_architecture.html | Accessed: 02-08-2022

5.2. Controller Verification

The purpose of controller verification is to test whether the control system operates according to the control architecture. This section describes the tests that were performed to verify that the correct code was flashed onto the PX4 Autopilot.

Bank angle To verify whether the bank angle and bank rate are measured correctly, the attitude of the PX4 was changed carefully according to Figure 5.2 after which the log files of the bank angle were evaluated.

Figure 5.2: Bank angle verification scheme; the PX4 Autopilot was (i) rolled by 90°, (ii) rolled back, (iii) pitched up by 90°, (iv) yawed by 90°, and finally (v) yawed and (vi) pitched back into its original position

Figure 5.3 shows the log file of the demonstrated motions in Figure 5.2. At 0° pitch, roll motion is equal to bank motion. At 90° pitch, yawing motion is then defined as a change in bank angle. The figure clearly shows that the rolling and yawing motions are defined as bank motion depending on the pitch angle.

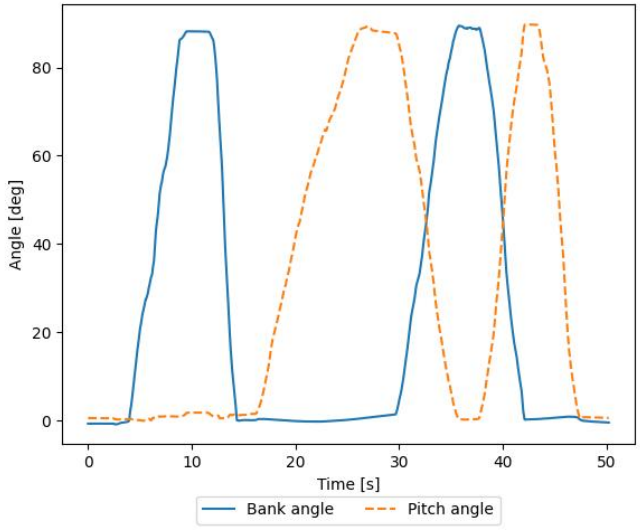


Figure 5.3: Bank and pitch angle during the bank angle verification scheme

Verification of the bank rate is done by comparing the numerical derivative of the bank angle with the formula of the bank rate given by Equation 5.1 as computed by the autopilot. This results in the following plot as shown in Figure 5.4. During the cross-over period, both bank rates are slightly different which is due to the fact that the pitch angle decreases during the yawing motion at 90° pitch angle. Bank angle is defined as pure yaw in this state, which means that $\cos(\theta) \cdot \dot{\psi}$ results in less accurate calculations for a decreasing pitch angle. However, during flight, this cross-over region cannot be reached.

$$\dot{\mu} = \dot{\varphi} \cdot \cos(\theta) + \dot{\psi} \cdot \sin(\theta) \tag{5.1}$$

[2]https://docs.px4.io/main/en/concept/px4_systems_architecture.html | Accessed: 13-06-2022

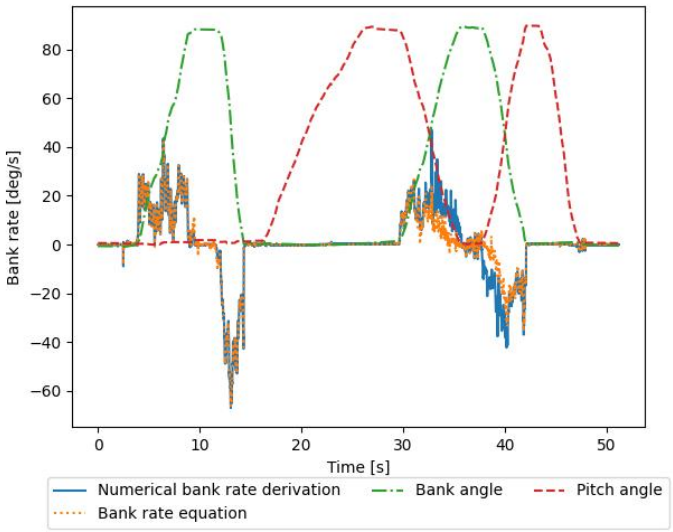


Figure 5.4: Comparison of the numerical bank rate with the real time calculated bank rate done by the autopilot. When the drone starts to bank at high pitch angles, the bank rate is incorrect as the yaw angles start to translate into high roll angles. However, during flight, such pitch angles are not reached and therefore the computation remains valid.

Thrust control The measurements for the servo deflections in combination with the remote control inputs must also be verified. Independent of the attitude of the drone, the nozzles have to be in the safety region ($0° < \delta_n < 90°$) with respect to the pressure frame. When the left stick of the RC is pushed all the way up, this means that maximum upward thrust is given and both nozzles are in a vertical position. When the stick is down, this means that the nozzles are in their horizontal position. In addition, it has to be verified that the nozzles remain in the same position with respect to Earth while changing the pitch angle. This is shown in Figure 5.5 with a corresponding photo series in Figure 5.6 that verifies the results in Figure 5.5. The fact that the deflection of the nozzles remains approximately the same, except for the first reaction of the pitch rate due to the D-gain (P_3), means that the P-gain (P_4) is tuned correctly.

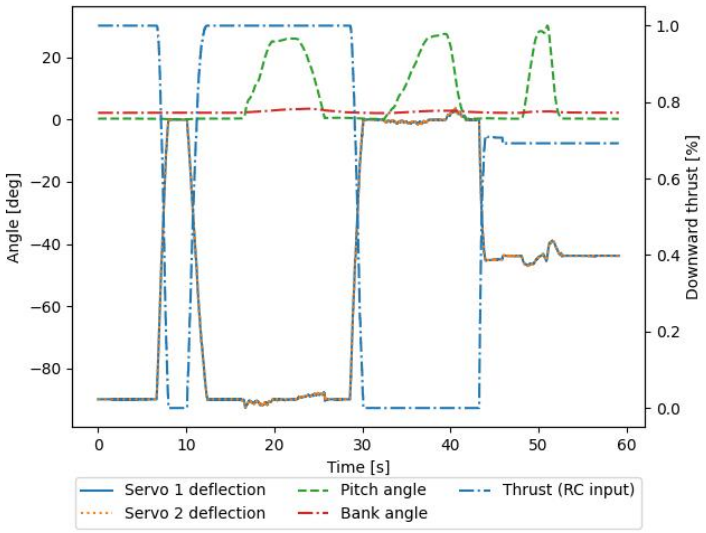


Figure 5.5: Servo outputs in Earth reference frame to verify that both the control stick input, as well as the pitch angle gains, are tuned correctly.

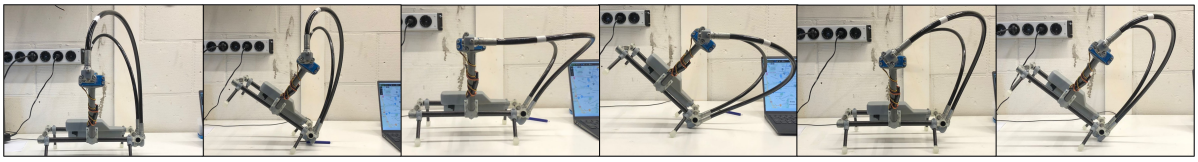
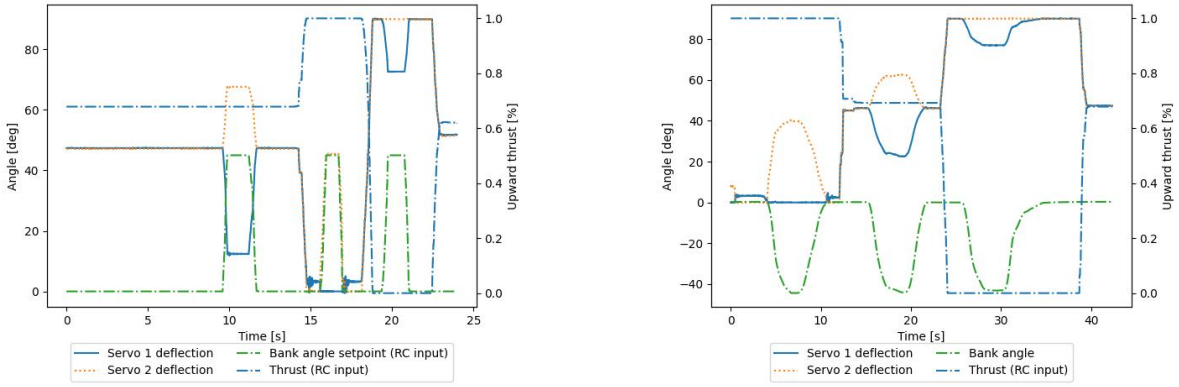


Figure 5.6: Servo deflection scheme to verify the deflection envelope, servo trim values, and p-gain on the pitch angle.

Lateral control To verify whether the controller acts according to the control architecture and obeys the rule that the nozzles must remain in their 90° deflection window, bank angle setpoints are generated through the RC and physical rotation of the drone. The results are shown in Figure 5.7.



(a) Bank angle setpoints are given to the drone through RC stick inputs to verify the nozzle deflections ($\varphi \approx \dot{\varphi} \approx \theta \approx \dot{\theta} \approx 0$)

(b) The drone is rotated on a flat surface to verify the nozzle deflections ($\theta \approx \dot{\theta} \approx 0$)

Figure 5.7: Bank inputs to verify the nozzle efficiencies of the controller

As can be seen in Figure 5.7, the lower the upward thrust, i.e. a more horizontal nozzle deflection, the more effective deflections are with respect to forces in the negative z-direction. This can be clearly seen in Figure 5.7a, where the bank angle setpoint is adjusted with an RC stick input, through the asymmetric deflection at $t = 10$s and the more significant deflection at $t = 16$s. In addition, the boundaries of the 90° deflection window can be seen at $t = 16$s and $t = 20$s. All three phenomena are also seen in Figure 5.7b where the bank angle is adjusted instead of the bank angle setpoint. Note that the deflections are in the same direction in both figures for opposite bank angle setpoint and bank angle as the bank angle setpoint of Figure 5.7b is zero, so it tries to counteract the bank motion. A bank angle command results in a ΔF_x on the drone which causes an undesired yawing motion. It was expected that this leads to stability problems during flight at high upward thrust flight modes.

Pitch response In order to identify whether the D-gain on the pitch must be positive, negative, or zero, the following test was executed. Three values were tested ($D = 0.1$, $D = 0.0$, $D = -0.15$) during which a step input was given. Figure 5.8 shows the experimental setup. In Figure 5.9, the results are shown. During this test, the drone was unattached to the hose and the secondary hose was not filled with water. Water increases the weight of the secondary hose creating an increased reaction torque of the servo and increased mass that interacts with the rigid airframe.

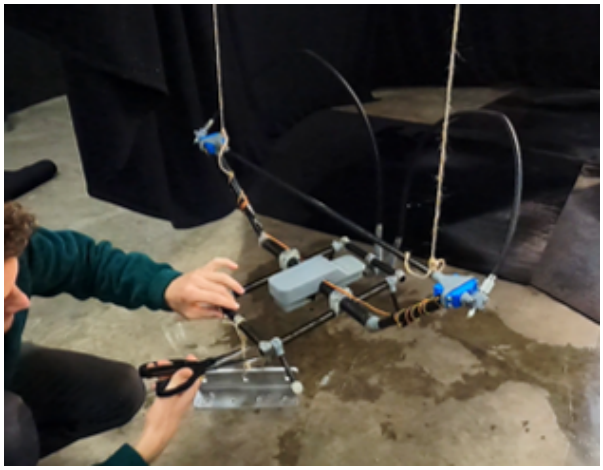
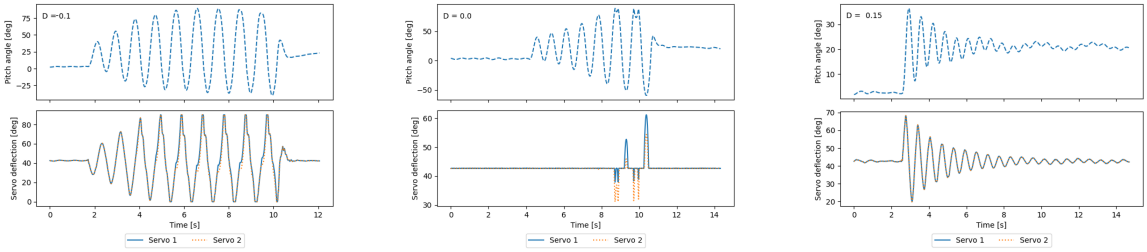


Figure 5.8: Experimental setup to simulate a step input. A weight is attached to the drone through a line which is cut away. During the tests, the weight remained constant.



(a) Step input response of the pitch angle and servo deflections for a D-gain on the pitch angle of $D = -0.1$

(b) Step input response of the pitch angle and servo deflections for a D-gain on the pitch angle of $D = 0.0$

(c) Step input response of the pitch angle and servo deflections for a D-gain on the pitch angle of $D = 0.1$

Figure 5.9: Comparison of different D-gains on the pitch angle to support the hypothesis of a non-minimum phase system

Figure 5.9a shows that with a negative gain, the pitch angle is amplified and the motions becomes out of control. The weight of the secondary hose becomes in phase with the oscillation of the rigid airframe. In Figure 5.9b, the D-gain is set to zero. This means that when σ is tuned correctly, the servo deflections remain constant in the pressure frame. This is shown by the plot, however, when the pitch angle becomes too small or too large, the limitation of the servo deflections cause the four peaks. Although the amplification is larger is smaller compared to Figure 5.9a, the pitch rate is undamped. Finally, Figure 5.9c shows that a positive D-gain dampens the pitch rate. Therefore, it was expected that positive D-gains were necessary during flight to prevent unstable pitch motion. Additionally, this shows that the effect of the non-minimum phase is reduced to the flexibility of the hose.

6

Software Implementation

This chapter provides an overview of some of the main software adjustments that were performed. Section 6.1 gives a general description of the architecture of the flight software. In Section 6.2, the internal messaging protocol is discussed, followed by an explanation of the mixer files in Section 6.3. The ground station is covered in Section 6.4. Finally, the implementation of the low-pass filters is discussed in Section 6.5. The hard- and software that was used for the external camera feedback is discussed in Chapter 8.

6.1. Software Overview

PX4 is an open source flight control software for unmanned vehicles. It is specialised in drone software and with an active community of 600+ developers, it provides a flexible set of tools to create tailored drone software for various applications. PX4 provides a software stack through Github, allowing an open source community to build and maintain software in a scalable way. The code is mainly written in C/C++. The following four properties make PX4 a popular toolkit for drone developers[1]:

- Modular architecture
- Open source
- Configurability
- Autonomy stack

The modular architecture utilises a port-based architecture which means that the extended system does not lose its performance robustness when developers add components. The code is divided into thousands of scripts that exchange variables with each other through the publication and subscription of variables. More information is covered in Section 6.2.

The software is supported by hundreds of developers which means that it is not just built for a single company, but is intended to serve the needs of thousands of users for recreational of commercial use. The optimised application programming interface (API) and software development kit (SDK) makes sure that features are easily deployed and reconfigured. Finally, the flightstack is deeply coupled with embedded computer vision for autonomous flight. This lowers the barrier to connect cameras or other sensors to develop autonomous flight algorithms.

Another reason PX4 is widely adopted by companies and organisations is the following[1]:

- Validated by real world
- Permissive license

[1] https://px4.io/software/software-overview/ | Accessed: 29-12-2022

- Interoperability
- Powerful safety features

Thousands of real-world applications have been developed with PX4 based systems. Examples are Avy[2] and Aerotain[3]. In parallel, a dedicated flight test team is testing thousands of flight hours each month on hardware and in simulations to ensure the safety and reliability of the codebase. Under the BSD 3-clause license, the software is free to use and can be commercialised. PX4 also aims to lead the standardisation of the advancement of communications, power management solutions, and peripherals communication. In addition, powerful safety features such as return and failsafe modes and automatic parachute opening. This can be easily modified for a tailored solution.

The above summarises but is not limited to the value of the PX4 flightstack. However, the downside of the highly modular and interoperable codebase is the complexity of the code. This means that the code can be easily broken and care must be taken in pulling or fetching the newest updates from the Github repository when developing the code for a novel flight mode or airframe locally.

When using a Windows computer, it is advised to setup the PX4 development environment running on Ubuntu Linux within Windows Subsytem for Linux 2 (WSL2) for building the code. The default ground station is QGroundControl (QGC).

6.2. uORB Messaging

Part of the middleware is the internal messaging protocol, named uORB. This messaging API is used for inter-thread and inter-process communication through the `publish()` and `subscribe()` commands. Individual message channels between applications are called topics[4]. A tutorial on how to add a new topic is described on one of the development pages on the PX4 website[5]. Figure 6.1 shows an overview of the messaging traffic between modules and topics.

[2] https://avy.eu/ | Accessed: 29-12-2022
[3] https://www.aerotain.com/ | Accessed: 29-12-2022
[4] https://dev.px4.io/v1.9.0_noredirect/en/apps/hello_sky.html | Accessed: 11-04-2022
[5] https://dev.px4.io/v1.9.0_noredirect/en/middleware/uorb.html | Accessed: 11-04-2022

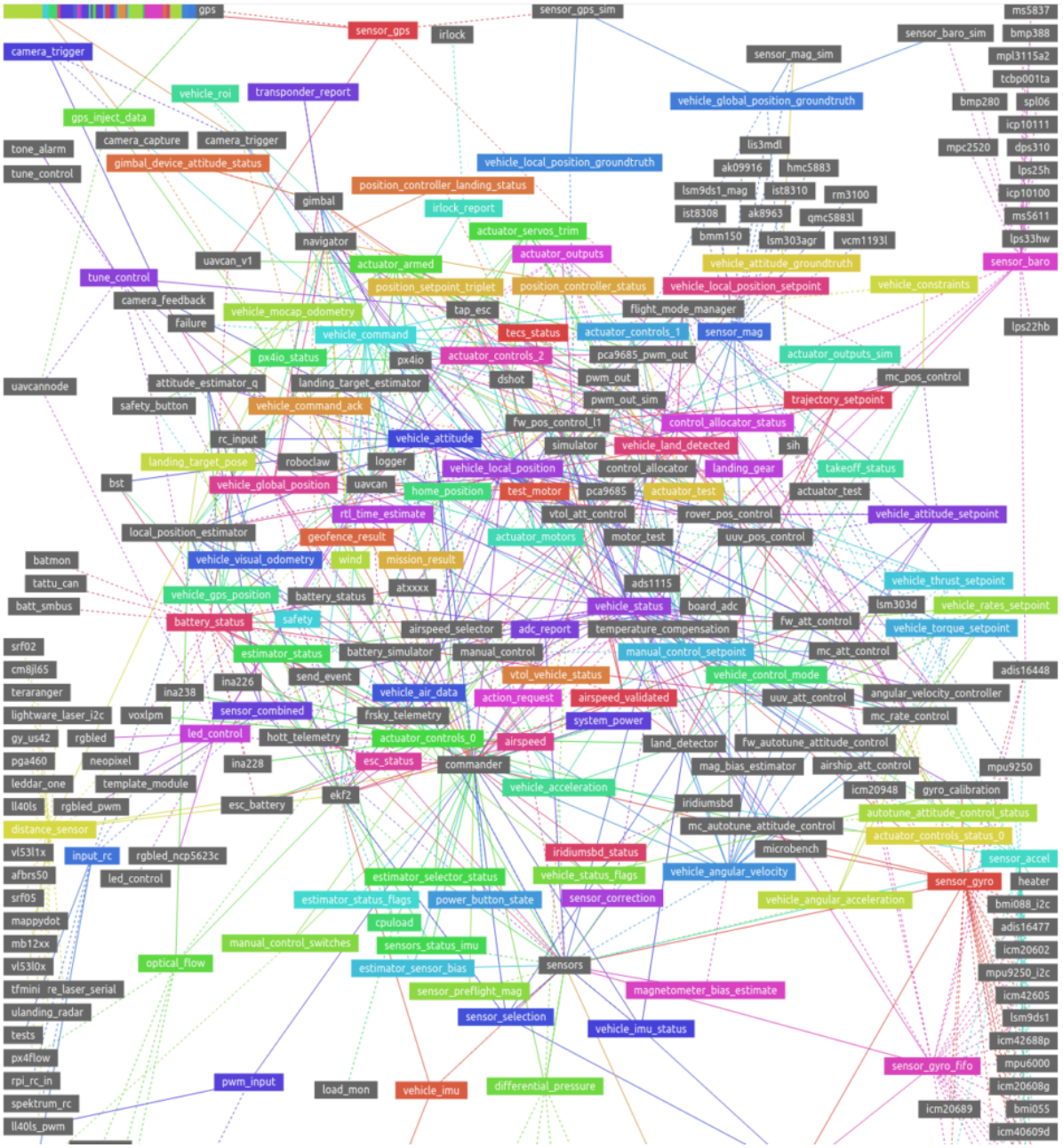


Figure 6.1: uORB publication/ subscription graph showing the communication between the modules and topics [6]. Modules are displayed in the grey boxes with rounded corners and topics are shown as coloured rectangles. The lines indicate the connections between the topics and modules.

However, to analyse sensor data that is not yet published by the PX4 Autopilot by default, one needs to create new messages containing the right data to save in the flight log file. As mentioned before there are two options for the uORB messaging API. The first is to publish the information and the second option is to subscribe to the information. Publishing means that the data is sent to the messaging API and subscribing means that the data is retrieved such that the stored message (information) can be used by the module. An example of an additional parameter that is added is the bank angle. Besides the roll angle, the bank angle needs to be recorded to be able to analyse this angle. This is done by adding a new topic. First, a `.msg` file is created that can register the timestamp of the publication of the message and the parameters that are recorded. When compiling, this script will add an automatically generated header file to the `build` directory. This is imported by the header file that

[6] https://docs.px4.io/main/en/middleware/uorb_graph.html | Accessed: 02-05-2022

supports the C-script where the message will be used. This is the `FixedwingAttitudeControl.hpp` file. To the `FixedwingAttitudeControl.cpp` file, one needs to advertise and publish the message, containing the calculated parameters for the log file to record the data. Finally, the `.msg` needs to be added to the MicroSD card of the flight computer such that it will log the extra topic in the folder `/etc/logging/logger_topics.txt`.

6.3. Mixer files

Mixer files translate the force commands from the controller to actuator commands that control the servos. The controller sends a command in the range $[-1, 1]$ which corresponds to minimum and maximum deflection of the actuator. By default, the mixer file mixes roll, pitch, and yaw commands together, which are sent to the corresponding actuators. The UHAV mixes the bank and thrust commands inside the attitude control script before publishing the command. This way, the servo efficiencies are dealt with by the controller instead of the mixer file. The mixer file was manually adjusted to eliminate mixing of different commands after these were published. Additionally, the canting keel was added to a custom mixer file such that the MAIN3 pin of the FMU would actuate the canting keel. Figure 6.2 shows the enhanced mixer file used during the final configuration of the UHAV.

```
MAIN 1 (servo 1) is attached to the roll input of the controller
M: 1
S: 0 0  10000  10000      0 -10000  10000

MAIN 2 (servo 2) is attached to the pitch input of the controller.
M: 1
S: 0 1 -10000 -10000      0 -10000  10000

MAIN 3 (canting keel) is attached to the yaw input of the controller.
M: 1
S: 0 2  10000  10000      0 -10000  10000
```

Figure 6.2: Custom mixer file that serves the three FMU main pinouts separately.

6.4. QGroundcontrol

To monitor the flight status, battery level, and adjust settings during flight a ground station is needed. QGroundControl (QGC) is one of the default ground stations compatible with PX4. There are two ways to communicate with QGC. Using the serial port directly via a USB cable and through the Wi-Fi module. Note that the baudrate of the FMU and the Wi-Fi module must match in order to transmit data. Once the custom firmware is compiled, it is flashed onto the FMU through PX4. The sensors and remote controller must be carefully calibrated before flight.

Real-time parameter tuning is done through QGC. When the firmware is flashed, QGC reads adjustable parameters and lists them in the parameter section. For the UHAV, custom parameters were created to adjust the different gains of the PID controllers. When a new parameter is needed, this is defined inside the header file. Said parameter must also be defined in the `params.c` file that exists for each module. Correctly setting the minimum and maximum values, units, and description are important steps for safety and readability of the code and instructions of use. Now that the parameter is defined, it can be used with the `.get()` extension inside the source file. When the firmware is flashed, all parameters automatically appear in QGC.

6.5. Low-pass Filter Implementation

The continuous time first-order low-pass filter must be discretised when implementing this into computer code. One way of doing this is transferring the continuous-time transfer function in the s-domain to a discrete time-domain transfer function in the z-domain. For discretisation, the bi-linear method, or Tustin's method, is used. The Tustin method is similar to a second-order trapezoidal integration. The inverse z-transform is used to convert the discrete transfer function to the time domain. By replacing

each z-inverse with a unit delay, one could solve for y. For first- and higher-order continuous-time transfer functions, the discretisation method is described in Dodge, 2018. In Equation 6.1, the coefficients of the time-domain equation, given by Equation 6.6, are computed. Similarly, Equation 6.7 and Equation 6.14 give the equations to implement a second-order low-pass filter into the microcontroller using Tustin's discretisation method.

$$H(S) = \frac{B_1 s + B_0}{A_1 s + A_0} \tag{6.1}$$

$$a_0 = 1 \tag{6.2}$$

$$a_1 = \frac{A_0 - A_1 K}{A_0 + A_1 K} \tag{6.3}$$

$$b_0 = \frac{B_0 + B_1 K}{A_0 + A_1 K} \tag{6.4}$$

$$b_1 = \frac{B_0 - B_1 K}{A_0 + A_1 K} \tag{6.5}$$

$$y(i) = b_0 x(i) + b_1 x(i-1) - a_1 y(i-1) \tag{6.6}$$

$$H(S) = \frac{B_2 s^2 + B_1 s + B_0}{A_2 s^2 + A_1 s + A_0} \tag{6.7}$$

$$a_0 = 1 \tag{6.8}$$

$$a_1 = \frac{2A_0 - 2A_2 K^2}{A_0 + A_1 K + A_2 K^2} \tag{6.9}$$

$$a_2 = \frac{A_0 - A_1 K + A_2 K^2}{A_0 + A_1 K + A_2 K^2} \tag{6.10}$$

$$b_0 = \frac{B_0 + B_1 K + B_2 K^2}{A_0 + A_1 K + A_2 K^2} \tag{6.11}$$

$$b_1 = \frac{2B_0 - 2B_2 K^2}{A_0 + A_1 K + A_2 K^2} \tag{6.12}$$

$$b_2 = \frac{B_0 - B_1 K + B_2 K^2}{A_0 + A_1 K + A_2 K^2} \tag{6.13}$$

$$y(i) = b_0 x(i) + b_1 x(i-1) + b_2 x(i-2) - a_1 y(i-1) - a_2 y(i-2) \tag{6.14}$$

$x(i)$ and $y(i)$ denote the in- and output values at time-step i. The results are verified with Python's imported package `scipy.signal.cont2discrete(system, dt, method = 'gbt', alpha=0.5)` where `system` is an array of the numerator and the denominator of the continuous-time transfer function.

7

Hardware Implementation

7.1. Physical Design and Building

The drone was built using carbon rods, 3D printed parts, and bolts. Figure 7.1 shows the technical drawing of the final design of the drone. Actiflow was of great help by providing the drawings, printing files, and parts to build the drone. In order to select the best material and dimensions that could withstand the forces related to flight, such as high pressure, impact, and wear and tear of moving parts, several plastics in different shapes were tested. Additionally, the electronics box had to be carefully designed to stay dry inside during flight and remain in tact during a potential crash to keep the electronics dry. Using 3D printed parts increases the flexibility of design choices and removes the dependence on third parties such as long delivery times and expensive parts. For that reason, high-grade off-the-shelf secondary hoses and nozzles were exchanged for cheap hydraulic hoses and custom made nozzles. This removed weight and added the flexibility to test different nozzle cross-sectional areas to optimise thrust. Finally, a diameter of 0.9mm was selected based on performance tests.

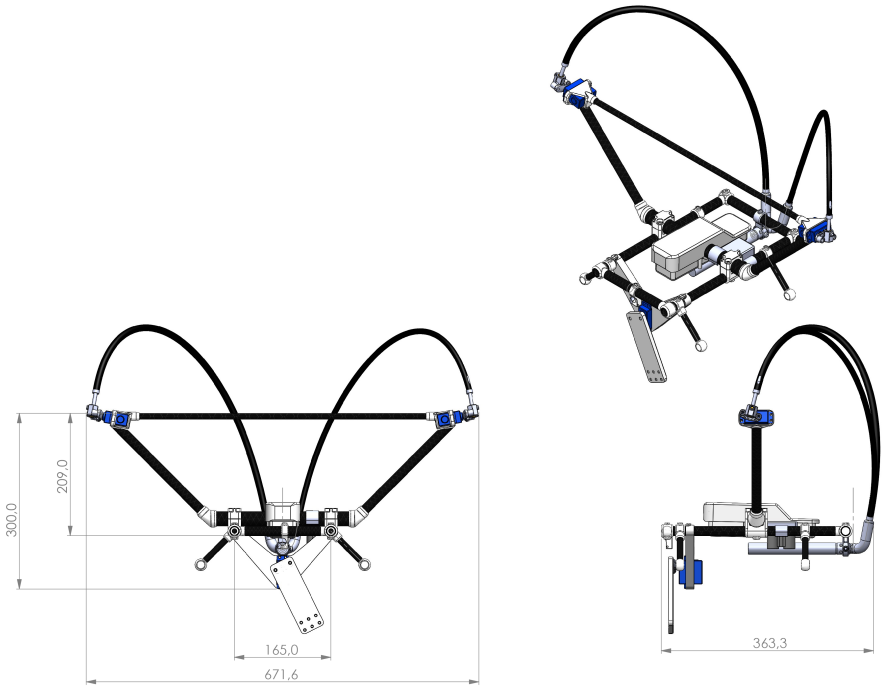


Figure 7.1: Technical drawing of the drone.

The two main limitations of the design are (i) the flexible secondary hoses with their center of mass above the axis of rotation and (ii) the missing landing mechanism. Analysis showed that the flexible hose caused a delayed reaction once actuated, leading to complex vibrations and oscillations that the controller must deal with. By incorporating the transportation of water along the carbon rod connecting both nozzles, the weight is removed. In future work, it is recommended to investigate the integration of the secondary hose and look into creating a non-flexible 'hose'. Secondly, it is recommended that a landing gear should be designed and tested such that the drone can automatically take-off and land without the help of a human operator. Test flights have shown that the most fragile parts during landing are the T-junction that divides the water as it is under high pressure and the canting keel.

7.2. Flight Computer

The flight computer that is being used, is the Pixhawk 4 mini (PX4 mini), which is shown in Figure 7.2. The reason for the mini is that the shape of the flight computer is more compact than the Pixhawk 4 which is more convenient for the drone, while the specifications are sufficient. As can be seen in Figure 7.2, the PX4 mini has multiple in- and outputs that are described in more detail in Section 7.2.1, Section 7.2.2, Section 7.2.3, and Section 7.2.4. The PX4 mini itself is equipped with several sensors and power boards of which the most important are listed below [1]:

- Main Flight Management Unit (FMU) processor
- On-board processors
  - Accelerometer
  - Gyrometer
  - Magnetometer
  - Barometer
- GPS

[1] https://docs.px4.io/master/en/flight_controller/pixhawk4_mini.html | Accessed: 02-05-2022

- Power system

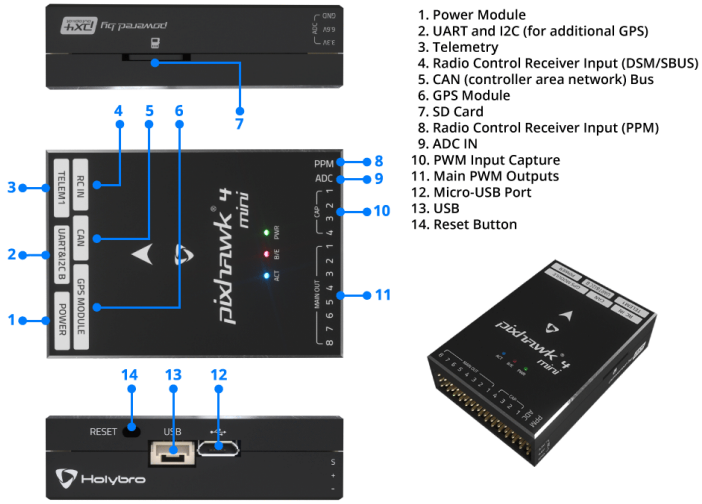


Figure 7.2: Overview of the Pixhawk 4 mini [2].

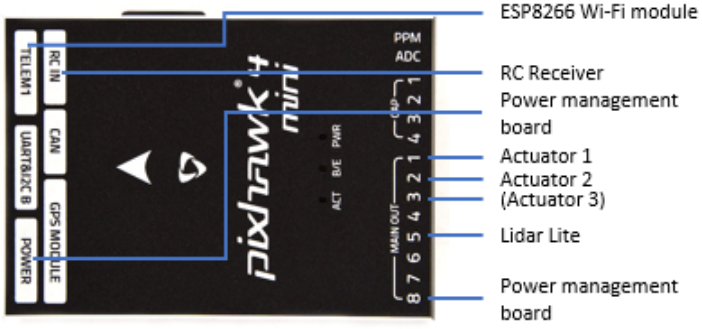


Figure 7.3: Wiring overview of the Pixhawk 4 Mini.

During the assembly phase, it was important to read the instructions carefully in order to comply with all the compatible hardware and not to break any hardware when using the wrong voltage.

7.2.1. Power Supply

The power is supplied via an onboard battery pack. A dual cell pack delivers 7.4V and 1500mAh. A power management board (PMB) is bolted into the electronics box which serves the purpose of a power distribution board as well as a power module. Multiple devices are powered by the PMB. The schematic drawing of the architecture is shown in Figure 7.4. As shown, the FMU and the servos both need to be separately powered. By connecting one of the unused pins of the FMU, the power is supplied to all pins.

Furthermore, the Lidar needed to be supplied with power through a BEC. As the batteries provide 7.4 V and the lidar has an operating voltage of 4.75 - 5.0V, a BEC is needed. This is done through soldering the BEC in between the power unit and the Lidar. An overview of the pinouts can be found in the PX4 user manual [3]

[2] https://docs.px4.io/main/en/flight_controller/pixhawk4_mini.html | Accessed: 02-05-2022
[3] https://docs.px4.io/main/en/sensor/lidar_lite.html | Accessed: 02-05-2022

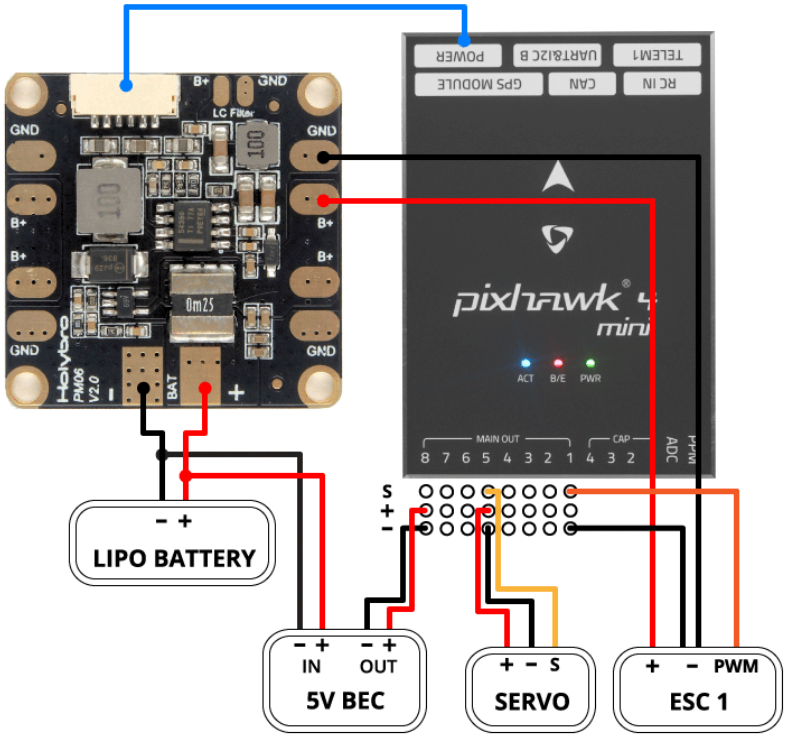


Figure 7.4: Schematic drawing of the pinouts of the power management board [4]. Note that not all connections are shown in the overview.

It is important that the all the conductors are covered with silicone or another anti-conducting material to make sure that no shortage can occur. Likewise, the FMU is covered with tape.

7.2.2. Remote Control

The remote controller (RC) had to be setup and calibrated correctly before use. In QGC, a protocol for arming/ disarming and changing the flight modes was setup. Next, the RC was calibrated. The nozzles had to be trimmed before flight such that they aligned with the horizon for zero thrust. By adjusting the subtrim, it was possible to change this angle. Current setting on the Taranis is 20.7 in subtrim. The nozzle deflections were maximised by setting it to -100 min and 100 max.

The remote control provides the manual control setpoints when the flight mode permits this. The way the remote controller is setup is as follows:

- Left poke up and down is setpoint x with range $[-1, 1]$
- Left poke left and right is setpoint r with range $[-1, 1]$
- Right poke up and down is setpoint z with range $[0, 1]$
- Right poke left and right is setpoint y with range $[-1, 1]$

7.2.3. Telemetry

The PX4 is equipped with multiple serial ports where telemetry units can be plugged in. The ESP8266 Wi-Fi module, which is the default module of PX4, converts a Wi-Fi signal to serial. For this, the baudrate of the serial port must be set to 921600. For the Wi-Fi module to be compatible with the PX4, the appropriate software was flashed onto the module. To flash the software, a ESP32-Prog board

[4] https://docs.px4.io/main/en/assembly/quick_start_pixhawk4_mini.html#power | Accessed: 02-05-2022

was used. The following connections were soldered between the Wi-Fi module and the programming board:

- GND to GND
- RX to TX
- TX to RX
- VPROG to 3V3
- GND to IO0 (this is a loop onto the Wi-Fi module itself)

The `esptool.py` was downloaded via Pip and the latest firmware was installed. Once the programming board is connected to the computer via a USB port, the command line can be used to run the `esptool` with the correct settings. `cd` into the right folder and the following line was typed into the command prompt:
`esptool.py --baud 921600 --port COM4 write_flash 0x00000 firmware-1.2.2.bin`.

7.2.4. Servos

The servos are actuated by sending a pulse-width modulation (PWM) signal between 800 and 2200 microseconds (μs). When the servo is deflected completely to the left, it receives a signal of $800\mu s$, in the middle a signal of $1500\mu s$ and completely to the right is $2200\mu s$. The deflection envelope of the servo is approximately 128°. The PWM value of the servo that is plugged into the MAIN OUT 1 port is logged in the output[0] command in the `actuator_outputs` uORB message and so on.

8

Position Control

8.1. System Architecture

Accurate determination of the local position of a drone is fundamental for navigation and executing missions. A popular method is using Global Positioning System (GPS) data to track the position of the vehicle. Although navigation using the GPS is available for small drones, the information is not sufficiently precise for altitude regulation nor is it always available inside buildings or other confined spaces (Floreano & Wood, 2015). Even it were available, the execution of specific tasks requires the use of additional information. The use of onboard or external camera vision that is processed in order to provide relevant information is a popular method of gaining this additional information. Onboard vision, called visual inertial odometry (VIO), often uses optical flow methods to determine its velocity and light-weight computer vision algorithms are developed for obstacle avoidance and navigation. As the camera is carried onboard, the drone is not restricted by boundaries where the area is covered by sensors. However, a more complex user interface (UI) is required to guide the navigation from a ground station. The UHAV is constrained in its position due to the hose and there is a relationship between the X- and Z-location. This makes the use of VIO more complex.

Motion capture (MoCap) uses external vision to locate and guide the drone. MoCap systems generally require markers on the body and multiple cameras to accurately detect the attitude and position of the drone. The advantage is that no onboard camera is required and computations are done externally. The disadvantage is that the drone is restricted to a constrained area with camera coverage. As the UHAV is constrained by its tether, this does not limit the freedom of the drone. Additionally, the UI for off-board navigation is simpler to create.

In this research, a single camera was used to capture the drone on a 2D plane. The external camera is fixed at P1 and aims at an azimuth angle of 0°. The camera sends the images to the ground station. With computer vision, the position of the drone was determined. Python was used to design a UI that interacts with the human operator such that the location of the drone onto the 2D surface could be controlled via mouse-clicks onto the video frame. The current location and location setpoint were sent to the FMU that actuates accordingly to move to the waypoint. Figure 8.1 gives a schematic overview of the system.

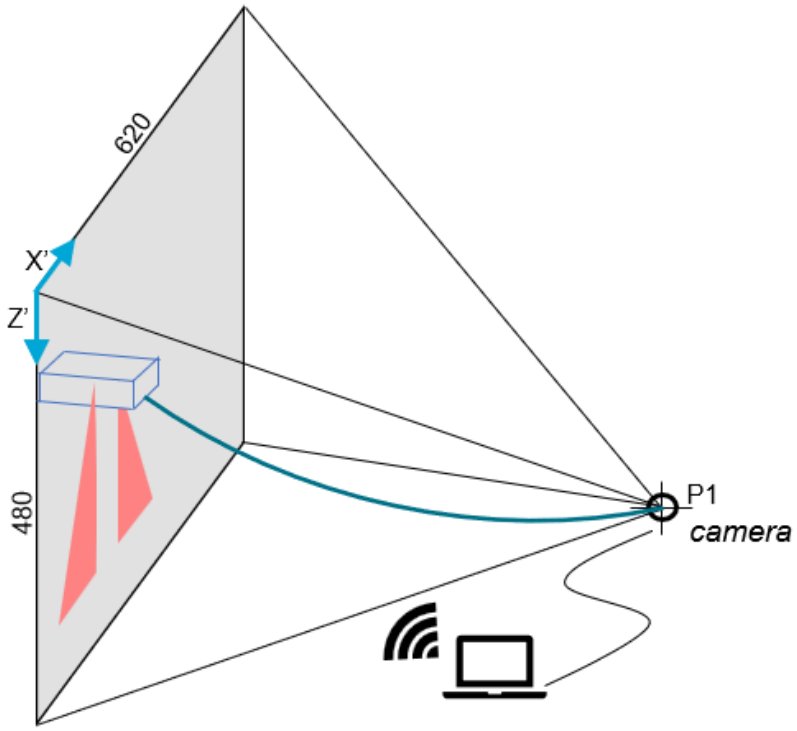


Figure 8.1: Schematic overview of UWAV system, including external camera, viewing frustum, and detection of the water jets. The length of the frame is denoted in pixels and angles and sizes are illustrative.

The webcam that was used, was the Trust Trino HD 720P webcam. The shape of the view is 480 x 620 pixels. Measuring the distance from the camera to the drone was approximately 2.70 meters. Using the specifications of the webcam from Table 8.1, the size of the frame can be determined.

$$\frac{1}{2}w_f = \tan\left(\frac{1}{2}\alpha_v\right) \cdot d_d \tag{8.1}$$

Where w_f is the width of the frame in meters, α_v is the viewing angle of the camera, and d_d is horizontal distance from the camera to the drone. This resulted in $w_f = 2.64$ from which it could be determined that $h_f = 2.04$, assuming square pixels. The size of 1 pixel is $\frac{264}{620} = 0.43x0.43$ centimeter.

Table 8.1: Specifications of the Trust Trino HD 720P webcam[1].

Feature	Value
Width	620 [pix]
Height	480 [pix]
Max. FPS	30 [fps]
Viewing angle	52 [deg]

8.2. Computer Vision

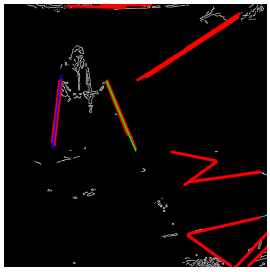
To have a system with vision-based control, it is important that frames are processed (near) real-time. Therefore, not only the accuracy of the detection is important but also the speed of the model. The

[1] https://www.trust.com/nl/product/18679-trino-hd-video-webcam#downloads | Accessed: 16-12-2022

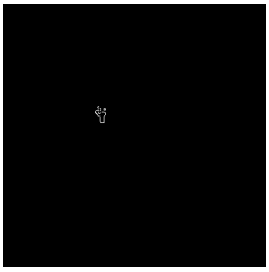
speed depends on multiple factors. Firstly, the processing power and memory of the hardware (ground station) can greatly impact the speed. For example, a GPU is more powerful than a CPU, however, most computers only have a CPU. Secondly, the complexity of the algorithm used for image processing affects the speed. For example, according to Horn and Schunck, 1980, optical flow algorithms typically analyse the the movement of a pixel in a video stream. In contrast, a machine learning algorithm such as YOLO (You Only Look Once), analyses the entire image in order to detect an object Redmon et al., 2016. Although the latter is more powerful, the computational load is heavier. Thirdly, the input size of the image affects the speed of the computer-vision algorithm. A smaller image is less computationally intensive than a large image. However, by downscaling the image, information is lost that can potentially lead to a reduction in accuracy of the algorithm. Finally, it is important to consider the functionality of the system. Depending on the use-case, different strategies should be used. When optimising the computer-vision algorithm for real-time object detection and tracking, the above factors have to be taken into account.

To determine the best algorithm, several object detection methods were tested on a single frame after which the sufficient algorithms were tried out on a pre-recorded video. Using the OpenCV library from Python, several filters are implemented on a series of images taken from a video. The final goal is to be able to detect the water jets. However, to prove that position control is feasible, any other part that can sufficiently be tracked was accepted. Figure 8.2 shows several methods that were used to detect the drone. In addition, two methods were tested on video frames, as the algorithm uses optical flow to remove the background or track an object.

(a) Raw image taken from the webcam without image processing.



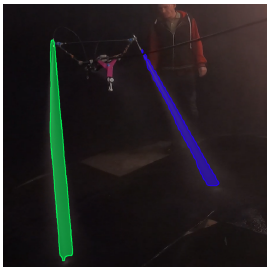
(b) Canny edge detection with a Hough transform to find straight lines in the figure.



(c) HSV or RGB filtering on the canting keel, followed by Canny edge detection.

(d) Background subtraction to find the contours of the jet streams and/ or the drone.

(e) Lucas-Kanade optical flow algorithm to track a specific point in the image.



(f) Mask R-CNN machine learning algorithm, trained to detect the water jets.

(g) YOLOv5-seg algorithm (nano and small), trained to detect the water jets.

Figure 8.2: Various object detection and object tracking algorithms that are tested on their performance to track the drone or water jets in real-time. Note that the OpenCV library was used for the implementation of the algorithms, with the exception of the machine learning algorithms.

The Hough transform was tried in combination with colour filtering and canny edge detection filters to eliminate noise from the picture. As can be seen in Figure 8.2b, multiple lines were detected. Based on a-priori knowledge of the angle that the water jets make, multiple lines could be filtered. By using the least squares method, a single line of the water jet could be determined. However, the method was not robust and trials onto videos led to defections throughout the entire image. It is expected that the reason for this, is the increasing amount of water mist in the air when flying for a long time.

Colour filtering based on the red, green, blue (RGB) and the hue, saturation, value (HSV) colour scheme

was tried. It was not possible to detect the object in a variety of test images and thus Canny edge detection was added. This is shown in Figure 8.2c. Similar to the Hough transform, the method proved to be not robust in a variety of circumstances.

Background subtraction, shown in Figure 8.2d, was a promising optical flow method. When running a video of the drone flying through the air, the body of the drone and the water jets were detected. However, during hovering, the detections were very limited. It was expected that the location of the water hitting the ground, would also be detected. However, the model failed to identify this location.

A solution to the hovering problem was the Lucas-Kanade (LK) optical flow algorithm. The algorithm uses an indicated pixel in the frame (chosen by the human operator) and is based on the assumption that the movement of this pixel is small and its brightness is conserved. The algorithm tracks the movement of this pixel by comparing a small window around the pixel of two consecutive frames. As such the pixel can be tracked without having to move in the frame. This method proved to work well on a series of test videos, while running at 30fps.

The Mask R-CNN (Region-based Convolution Neural Network) (He et al., 2017) and YOLOv5 (Redmon et al., 2016) segmentation models were trained on a custom dataset. These models were chosen because of their claimed ability to process in real-time. Around a 120 snapshots were captured from different videos to have a variety of settings such as lighting and mist. Using Roboflow [2], the water jets at all images were indicated and with e.g. mirroring and adding noise, new training samples were created that the model learn from. This resulted in a dataset of 348 images. The YOLOv5-nano instance segmentation model [3] and the Mask R-CNN model [4] were trained on Google Colab. The YOLOv5-nano model was chosen because of its lightweight computational load and Google Colab was used to train the model on a GPU rather than on the CPU of a laptop. With a small dataset of only 348 images, both models had an outstanding performance in the detection of the water jets. However, after optimising the size of the input image, the Mask R-CNN was running at 0.5 frames per second (fps) while the YOLOv5 algorithm reached 7.9 fps. It was concluded to use the YOLOv5 model. It is expected that the detection rate of both models can be increased by optimising the machine learning models.

There are several limitations using YOLO. The centroid of the segmentation of the water jet was used to indicate its location. Because of the volatility of the shape of the segment, a second order low-pass filter was applied to average out this location. False identifications of other segments could still lead to deviations of the centroid. In the future, the controller needs to deal with this. Additionally, while YOLO runs at 7.9Hz, the position controller of the FMU computes its attitude commands at 100Hz. To prohibit jerky movement by the attitude controller, the previous signal was stored in memory until it was updated by the ground station with a new position. When the frame rate is increased, the position controller can use more accurate information to compute its attitude commands. Finally, when the water jet is not detected, a strategy of estimating its most accurate position should be developed.

For future missions, it is important to not only detect the water jets, but also to determine the position of the center of gravity of the drone. This location must be used by the position controller as it is considered to be the most stable point of the system. Else, errors occur due to the movement of body parts that are not relevant for the movement being tracked. Another solution that should be considered, is to integrate YOLO and LK to enhance the detection speed.

8.3. Data Transmission

The information from the computer vision algorithm and the human input was sent to the FMU. MAVLink is the light-weight communication protocol of PX4 and it is used to communicate with QGroundControl [5]. The service provides a number of standard messages for exchanging data, while it is also possible to create custom messages. An additional connection, next to QGroundControl, was setup from the same groundstation to communicate with the drone. As PX4 has only serial connections, a separate Wi-Fi module was plugged into one of the telemetry ports that transforms a wireless communication signal into

[2] https://www.makesense.ai/ | Accessed: 24-10-2022
[3] https://github.com/ultralytics/yolov5 | Accessed: 24-10-2022
[4] https://github.com/matterport/Mask_RCNN | Accessed: 24-10-2022
[5] https://docs.px4.io/main/en/middleware/mavlink.html Accessed: 14/11/2022

a serial message that can be handled by the Autopilot. The Wi-Fi module (ESP8266) was flashed with the latest firmware as recommended by Dronecode in order to have the right settings to communicate with the PX4. First, the baudrate of the telemetry port needs to match the baudrate of the Wi-Fi module. Secondly, the message protocol needs to be configured. The UDP message protocol was used for data transmission. UDP sends messages to and listens to messages from to a single IP address in combination with a network port. In the firmware of the Wi-Fi module, the baudrate and network ports are defined which needs to be used. QGroundControl establishes the connection automatically. The secondary connection has to be setup separately. Note that the following instructions only work for the default operating system on a laptop (Windows in this case) and not through a virtual machine or a subsystem for Linux (e.g. WSL2).

When the ground station is connected to the Wi-Fi module of the drone, the host and remote IP addresses can be extracted using the `ipconfig` command in the command terminal. The IPv4 Address is the IP address of the ground station that is associated with the Wi-Fi connection and the Default Gateway address is the IP address of the drone. The drone listens to port 14550 and sends to port 14550 as well.

Once the baudrate, IP addresses and network ports are configured, one can test the connection through `pymavlink`, a Python package to communicate through MAVLink. The use of Pymavlink is discussed in Section 8.3.1. For more complex systems like VIO and MoCap, MAVROS is integrated to handle complex computations for robotics and communicate with the FMU. This is shortly touched upon in Section 8.3.2. For sending floats from the ground station to the FMU, setting up a connection with Pymavlink and using standard MAVLink messages was sufficient.

8.3.1. Pymavlink

First a connection has to be established with the FMU through the `mavutil.mavlink_connection` function by setting the udpin to the Default Gateway IP address and port that the PX4 listens to (14550 by default). When a heartbeat is received, the connection is established. Custom and default messages can be sent using the MAVLink protocol. The message that was used to send the data was the `mav.vision_position_estimate_send` message. The PX4 flight stack subscribes to the default messages and the data can be used in the flight code. By adding the messages to the logger topics, the data can be retrieved for analysis.

8.3.2. MAVROS

In order to use offboard communication from a laptop towards the FMU, MAVROS can be used. MAVROS is a robot operating system (ROS) package that enables communication through MAVLink between devices running ROS. The source installation guide is used from the official PX4 website [6]. However, this installation is compatible with the Kinetic or Melodic ROS version which cannot run on Ubuntu 20.04. For ROS Noetic, which runs on Ubuntu 20.04, the catkin workspace can only be built by running the following command in terminal: `sudo pip3 install git+https://github.com/catkin/catkin_tools.git` [7].

Once the catkin workspace is built, including the installations of MAVROS and MAVLink, the communication between the Autopilot and the ground computer can be established. The MAVROS node can then be launched by running the command `roslaunch mavros px4.launch` in the terminal. To confirm that the connection has been established, the command `rostopic echo /mavros/state` can be run in the terminal. This will display information about the state of the autopilot. It is important to ensure that the autopilot is configured correctly and is able to communicate with the computer before attempting to run MAVROS.

Custom uORB message can be sent using the PX4 instructions [8]. Again, one of the steps in the tutorial needed to be adjusted in order to build the PX4 firmware. Using `mavlink-generator` from the MAVLink repository on Github, the correct XML files could be generated inside the PX4 firmware (ChunzPS,

[6] https://docs.px4.io/main/en/ros/mavros_installation.html. Accessed: 14/11/2022
[7] https://stackoverflow.com/questions/71990979/cannot-run-catkin-build-on-ubuntu-20-04 Accessed: 14/11/2022
[8] https://docs.px4.io/main/en/ros/mavros_custom_messages.html

2021). Issues with establishing the MAVROS connection between the devices led to the decision to use Pymavlink. Although MAVROS is a powerful software module that enables the exchange of telemetry data and control commands between the autopilot and a ground control station, MAVROS is unnecessary for sending simple messages through MAVLink.

9

Dynamic Model

An effective approach for the design of a control system, is to create a virtual environment where test flights can be simulated. The dynamic model of a drone is a fundamental building block for creating accurate and realistic simulations of unmanned aerial systems (UAS). The model describes the physical behaviour of the drone and its interaction with the environment.

A first setup of an analytical model was created that could be used for a simulation. However, given the limited time frame of the project and the expectation that the initial design would be subject to modifications, it was determined that creating a reliable model of the system would be ineffective use of the scarce resources. This section summarises the effort that was put into creating a dynamic model of the system. Section 9.1 describes the building blocks of the model while a first setup of the equations of motion are given in Section 9.2. The attitude controller is shortly discussed in Section 9.3. Section 9.4 concludes the chapter by giving the shortcomings of the model and the reason behind the complexity of a representative simulation.

9.1. Preliminary Analysis

This section gives an overview of the system and its variables. Section 9.1.1 states the assumptions that were made, followed by a description of the state variables in Section 9.1.2. The control variables are described by Section 9.1.3.

9.1.1. Assumptions

The following assumptions are defined before proceeding to the dynamics of the system (Kotaru & Sreenath, 2020).

- No water flow in the hose and thus no dynamic pressure forces
- Hose is modelled as a series of n links connected by massless ball-and-socket joints
- The hose mechanical properties such as stiffness and torsional forces are ignored
- Water pump delivers a constant mass flow irrespective of the height of the drone

9.1.2. State Variables

The drone has the following state variables:

$$\mathbf{x} = \left[X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}, \varphi, \theta, \psi, p, q, r\right]^T \tag{9.1}$$

The twelve states in Equation 9.1 represent the position (X, Y, Z), the velocity $(\dot{X}, \dot{Y}, \dot{Z})$, the attitude (φ, θ, ψ), and the angular rates (p, q, r).

9.1.3. Control Variables

Control is based on uniaxial thrust vectoring with two nozzles. A third element of indirect control is the hose attached to the tip of the drone, exerting a force based on the position in space of the drone.

The actuator that was used for design phase 1 can only rotate arond the y-axis which means that the control variables for the current design of the drone are $\delta_{y,1}$ and $\delta_{y,2}$ which is shown in Equation 9.2. The deflection in the yz-plane, hence δ_x is pre-determined by the design which means that it is a design parameter and not a control variable.

$$\mathbf{u} = \begin{bmatrix}\delta_{y,1}\delta_{y,2}\end{bmatrix}^T \tag{9.2}$$

It will be examined whether an additional degree of freedom around δ_x is necessary for (optimal) flight control. Adding this degree of freedom means that there are four control inputs and Equation 9.3 applies.

$$\mathbf{u} = \begin{bmatrix}\delta_{x,1}, \delta_{x,2}, \delta_{y,1}, \delta_{y,2}\end{bmatrix}^T \tag{9.3}$$

9.1.4. Thrust and Aerodynamics

The thrust equation of a nozzle extinguishing water is expressed here. As the entire nozzle is being deflected, one does not have to use the deflection theorem of Erich Wilson where the outlet is deflected, changing the outlet area.

Thrust is generated through the acceleration of mass. Thrust is equal to the change in momentum over time, denoted by Equation 9.4.

$$T = \frac{dp}{dt} = \dot{m}(V_e - V_i) \tag{9.4}$$

Here, $\dot{m}$ is the mass flow rate of the water, and V_e and V_i are the exit and initial velocities of the water, respectively. The mass flow and nozzle outlet area are determining factors to reduce or increase thrust. By increasing the water pump pressure, the mass flow can be increased and by constructing the nozzles manually, its size can be adjusted.

However, thrust can also be measured. When the measuring equipment is trustworthy, a more reliable number can be determined compared to analytical formulas as e.g. friction within the hose is not taken into account. In order to measure the total thrust of the nozzles, weight was added to the drone such that it would not lift off when turning on the pressure washer. The weight of the drone including water inside the secondary hoses without thrust was compared to the weight of the drone with thrust. The difference can be converted into Newtons of thrust.

The aerodynamics of the drone were not taken into account as the airspeed of the drone is low and flight was performed inside a building. However, it is expected that wind will affect the dynamics of the system when flying outside. For future research, this has to be taken into account.

9.1.5. Force and Moment Distributions

Sign conventions The nozzle is undeflected when it is pointing downward in the body reference frame. A forward nozzle deflection rotating around the y-axis (δ_y) is considered to be positive. Similar, a leftward nozzle deflection around the x-axis (δ_x) is considered to be positive. Note that the sign conventions in the scientific paper (see Part II) were defined differently.

Thrust Vector

The thrust that is exerted by one nozzle is expressed by Equation 9.5 (Vinayagam & Sinha, 2014).

$$\mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} -T \cdot \sin(\delta_y)\cos(\delta_x) \\ T \cdot \sin(\delta_x) \\ -T \cdot \cos(\delta_y)\cos(\delta_x) \end{bmatrix} \tag{9.5}$$

Catenary

For the first analysis, the hose was modelled as a point force and moment that is exerted on the nose of the drone. Based on the mass and the length of the hose and the position in the inertial frame of the drone, a force conform the catenary equation can be calculated. The dynamics of the catenary are neglected in the first iteration of the dynamic model and its simulation and they are handled by the controller as disruptions. The hose forces are denoted by Equation 9.6.

$$\mathbf{F}_h = \begin{bmatrix} F_{h,x} \\ F_{h,y} \\ F_{h,z} \end{bmatrix} \tag{9.6}$$

The moments in the x, y, z direction by one nozzle is expressed by Equation 9.7.

$$\mathbf{M} = \begin{bmatrix} 0 & T_{z_1} & -T_{y_1} \\ -T_{z_1} & 0 & T_{x_1} \\ T_{y_1} & -T_{x_1} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} 0 & T_{z_2} & -T_{y_2} \\ -T_{z_2} & 0 & T_{x_2} \\ T_{y_2} & -T_{x_2} & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} -M_{h,x} \\ -M_{h,y} \\ -M_{h,z} \end{bmatrix} \tag{9.7}$$

With $M_h = -k_s \cdot \eta$ where M_h is in $[Nm]$. The angle η is the sum of the angle of the hose with horizon $+$ the pitch angle of the drone. With the catenary equation, the angle of the hose with the horizon can be computed. This is described in Part II.

The thrust and moment equations based on the inputs of δ_x, and δ_y are given by Equation 9.8.

$$\begin{bmatrix} T_x \\ T_y \\ T_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} -T & 0 & 0 & -T & 0 & 0 \\ 0 & -T & 0 & 0 & -T & 0 \\ 0 & 0 & -T & 0 & 0 & -T \\ 0 & T \cdot z_1 & -T \cdot y_1 & 0 & T \cdot z_2 & -T \cdot y_2 \\ -T \cdot z_1 & 0 & T \cdot x_1 & -T \cdot z_2 & 0 & T \cdot x_2 \\ T \cdot y_1 & -T \cdot x_1 & 0 & T \cdot y_1 & -T \cdot x_1 & 0 \end{bmatrix} \begin{bmatrix} \sin(\delta_{y,1})\cos(\delta_{x,1}) \\ \cos(\delta_{y,1})\sin(\delta_{x,1}) \\ \cos(\delta_{y,1})\cos(\delta_{x,1}) \\ \sin(\delta_{y,2})\cos(\delta_{x,2}) \\ \cos(\delta_{y,2})\sin(\delta_{x,2}) \\ \cos(\delta_{y,2})\cos(\delta_{x,2}) \end{bmatrix} + \begin{bmatrix} F_{h,x} \\ F_{h,y} \\ F_{h,z} \\ -M_{h,x} \\ -M_{h,y} \\ -M_{h,z} \end{bmatrix} \tag{9.8}$$

One can get the desired angles of the nozzles with Equation 9.9.

$$\begin{bmatrix} \sin(\delta_{y,1})\cos(\delta_{x,1}) \\ \cos(\delta_{y,1})\sin(\delta_{x,1}) \\ \cos(\delta_{y,1})\cos(\delta_{x,1}) \\ \sin(\delta_{y,2})\cos(\delta_{x,2}) \\ \cos(\delta_{y,2})\sin(\delta_{x,2}) \\ \cos(\delta_{y,2})\cos(\delta_{x,2}) \end{bmatrix} = \begin{bmatrix} -T & 0 & 0 & -T & 0 & 0 \\ 0 & -T & 0 & 0 & -T & 0 \\ 0 & 0 & -T & 0 & 0 & -T \\ 0 & T\cdot z_1 & -T\cdot y_1 & 0 & T\cdot z_2 & -T\cdot y_2 \\ -T\cdot z_1 & 0 & T\cdot x_1 & -T\cdot z_2 & 0 & T\cdot x_2 \\ T\cdot y_1 & -T\cdot x_1 & 0 & T\cdot y_1 & -T\cdot x_1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} T_x \\ T_y \\ T_z \\ M_x \\ M_y \\ M_z \end{bmatrix} + \begin{bmatrix} F_{h,x} \\ F_{h,y} \\ F_{h,z} \\ -M_{h,x} \\ -M_{h,y} \\ -M_{h,z} \end{bmatrix} \tag{9.9}$$

9.2. Equations of Motion

9.2.1. Transformation Matrix

The transformation matrix from the vehicle carried normal Earth reference frame to the body fixed reference frame is described in (Mulder et al., 2013) and given Equation 9.10. Where the positive direction of the angles are in accordance with the standard right-hand-rule. As one wishes to apply orthogonal transformations, i.e. a rotation from one orthogonal frame to another orthogonal frame, the transformation matrix must also be orthogonal. Hence, for a transformation matrix **A** in case of an orthogonal transformation, it follows that $\mathbf{A}^{-1} = \mathbf{A}^{T}$ (Mulder et al., 2013).

$$\mathbf{X}^{b} = \mathbb{T}_{bE}\mathbf{X}^{E} \tag{9.10}$$

$$\mathbf{X}^{E} = (\mathbb{T}_{bE})^{-1}\mathbf{X}^{b} = (\mathbb{T}_{bE})^{T}\mathbf{X}^{b} = \mathbb{T}_{Eb}\mathbf{X}^{b} \tag{9.11}$$

Where $\mathbb{T}_{bE}$ is described by Equation 9.12.

$$\begin{aligned} \mathbb{T}_{bE} &= \mathbb{T}_{bE''}\mathbb{T}_{E''E'}\mathbb{T}_{E'E} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \begin{pmatrix} \sin(\varphi)\sin(\theta)\cos(\psi) \\ -\cos(\varphi)\sin(\psi) \end{pmatrix} & \begin{pmatrix} \sin(\varphi)\sin(\theta)\sin(\psi) \\ +\cos(\varphi)\cos(\psi) \end{pmatrix} & \sin(\varphi)\cos(\theta) \\ \begin{pmatrix} \cos(\varphi)\sin(\theta)\cos(\psi) \\ +\sin(\varphi)\sin(\psi) \end{pmatrix} & \begin{pmatrix} \cos(\varphi)\sin(\theta)\sin(\psi) \\ -\sin(\varphi)\cos(\psi) \end{pmatrix} & \cos(\varphi)\cos(\theta) \end{bmatrix} \end{aligned} \tag{9.12}$$

$$
\begin{aligned}
(\mathbb{T}_{bE})^T = & \\
= & \begin{bmatrix}
\cos(\theta)\cos(\psi) & \begin{pmatrix}\sin(\varphi)\sin(\theta)\cos(\psi)\\ -\cos(\varphi)\sin(\psi)\end{pmatrix} & \begin{pmatrix}\cos(\varphi)\sin(\theta)\cos(\psi)\\ +\sin(\varphi)\sin(\psi)\end{pmatrix} \\
\cos(\theta)\sin(\theta) & \begin{pmatrix}\sin(\varphi)\sin(\theta)\sin(\psi)\\ +\cos(\varphi)\cos(\psi)\end{pmatrix} & \begin{pmatrix}\cos(\varphi)\sin(\theta)\sin(\psi)\\ -\sin(\varphi)\cos(\psi)\end{pmatrix} \\
-\sin(\theta) & \sin(\varphi)\cos(\theta) & \cos(\varphi)\cos(\theta)
\end{bmatrix}
\end{aligned} \tag{9.13}
$$

9.2.2. Translational Motion

For the translational motion, the rate of change of six translation variables ($\dot{X}, \dot{Y}, \dot{Z}, \ddot{X}, \ddot{Y}, \ddot{Z},$) can be obtained according to the following motion equations (Equation 9.16):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{9.14}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \tag{9.15}$$

$$\mathbf{F} = m \cdot \mathbf{a} = m \cdot g \cdot \mathbf{a}_3 + \mathbb{T}_{bE} \cdot \mathbf{T} + \mathbb{T}_{bE} \cdot \mathbf{F}_h \tag{9.16}$$

$$
\begin{aligned}
\mathbf{a} = & \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} \\
= g \cdot & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{1}{m} \begin{bmatrix}
\cos(\theta)\cos(\psi) & \begin{pmatrix}\sin(\varphi)\sin(\theta)\cos(\psi)\\ -\cos(\varphi)\sin(\psi)\end{pmatrix} & \begin{pmatrix}\cos(\varphi)\sin(\theta)\cos(\psi)\\ +\sin(\varphi)\sin(\psi)\end{pmatrix} \\
\cos(\theta)\sin(\theta) & \begin{pmatrix}\sin(\varphi)\sin(\theta)\sin(\psi)\\ +\cos(\varphi)\cos(\psi)\end{pmatrix} & \begin{pmatrix}\cos(\varphi)\sin(\theta)\sin(\psi)\\ -\sin(\varphi)\cos(\psi)\end{pmatrix} \\
-\sin(\theta) & \sin(\varphi)\cos(\theta) & \cos(\varphi)\cos(\theta)
\end{bmatrix} \\
& \left(\begin{bmatrix} -T \cdot \cos(\delta_{z,1})\sin(\delta_{y,1}) \\ -T \cdot \sin(\delta_{z,1})\cos(\delta_{x,1}) \\ -T \cdot \sin(\delta_{x,1})\cos(\delta_{y,1}) \end{bmatrix} + \begin{bmatrix} -T \cdot \cos(\delta_{z,2})\sin(\delta_{y,2}) \\ -T \cdot \sin(\delta_{z,2})\cos(\delta_{x,2}) \\ -T \cdot \sin(\delta_{x,2})\cos(\delta_{y,2}) \end{bmatrix} \right) \\
+ \frac{1}{m} & \begin{bmatrix}
\cos(\theta)\cos(\psi) & \begin{pmatrix}\sin(\varphi)\sin(\theta)\cos(\psi)\\ -\cos(\varphi)\sin(\psi)\end{pmatrix} & \begin{pmatrix}\cos(\varphi)\sin(\theta)\cos(\psi)\\ +\sin(\varphi)\sin(\psi)\end{pmatrix} \\
\cos(\theta)\sin(\theta) & \begin{pmatrix}\sin(\varphi)\sin(\theta)\sin(\psi)\\ +\cos(\varphi)\cos(\psi)\end{pmatrix} & \begin{pmatrix}\cos(\varphi)\sin(\theta)\sin(\psi)\\ -\sin(\varphi)\cos(\psi)\end{pmatrix} \\
-\sin(\theta) & \sin(\varphi)\cos(\theta) & \cos(\varphi)\cos(\theta)
\end{bmatrix} \begin{bmatrix} F_{h,x} \\ F_{h,y} \\ F_{h,z} \end{bmatrix}
\end{aligned} \tag{9.17}
$$

9.2.3. Rotational Motion

For the rotational motion, the rate of change of six rotation variables ($\dot{\theta}, \dot{\varphi}, \dot{\psi}, \dot{p}, \dot{q}, \dot{r}$) can be obtained as described in (Mulder et al., 2013). The equation for angular momentum is given by Equation 9.18.

$$\mathbf{B}_{cm} = \mathbf{I} \cdot \Omega \tag{9.18}$$

$$\Omega = [p, q, r]^T$$

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

Where $\mathbf{I}$ is the moment of inertia of the total system, i.e. the drone including the secondary hoses and part of the primary hose that is directly attached to the airframe. As the shape of the system is constantly changing due to deflections of the secondary hoses and the location of the drone with respect to the stationary pole, its moment of inertia is also constantly changing.

The equation for the moments around the center of mass for a body with constant mass but varying moment of inertia is given by Equation 9.19.

$$\mathbf{M}_{cm} = \dot{\mathbf{I}} \cdot \Omega + \mathbf{I} \cdot \dot{\Omega} + \Omega \times \mathbf{I} \cdot \Omega \tag{9.19}$$

$$\dot{\Omega} = \mathbf{I}^{-1} \left(\mathbf{M}_{cm} - \dot{\mathbf{I}} \cdot \Omega - \Omega \times \mathbf{I} \cdot \Omega \right) \tag{9.20}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}^{-1} \left(\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - \begin{bmatrix} \dot{I}_{xx} & -\dot{I}_{xy} & -\dot{I}_{xz} \\ -\dot{I}_{xy} & \dot{I}_{yy} & -\dot{I}_{yz} \\ -\dot{I}_{xz} & -\dot{I}_{yz} & \dot{I}_{zz} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \tag{9.21}$$

Considering that the secondary hoses do not change shape and that the primary hose attached to the drone delivers a constant force, the moment of inertia does not change over time, hence $\dot{\mathbf{I}} = 0$. Equation 9.21 can be simplified to Equation 9.22.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}^{-1} \left(\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \tag{9.22}$$

9.3. Attitude Control

By generating control torques onto the airframe, the attitude is controlled. As the drone's pitch angle is determined by its relative location to P1, this angle is not controlled. Instead, one goal of the controller is to control the angle of the nozzles in the world reference frame. This angle controls the height of the drone and remains constant in the presence of pitch angle perturbations. Another goal of the controller is to dampen the pitch rate. This can be done by nozzle deflections relative to the airframe.

A PID controller is used for attitude control. Input for the attitude controller are nozzle deflection angles and bank angle setpoints, coming from the remote controller. The efficiency of the nozzles must be taken into account and therefore the commands are provided in percentages of upward thrust. This must be converted to world frame deflections before it is finally transformed to body frame coordinates.

9.4. Conclusion

After an initial analysis of the dynamics of the system, it was decided that a useful representation of the real system would not be an efficient use of resources due to expected design changes and the limited timeline of the research. The above analysis lacks the interaction of the primary hose with the drone, and the interaction between the two subsystems, (i) the rigid airframe and (ii) the flexible secondary hoses. This would consist of reaction forces of the servos onto the airframe, flexibility of the secondary hoses that cause a delay in the transportation of its mass after a nozzle deflection, and the mechanical and aerodynamic losses due to the movement and bending of the secondary hoses through the air. Initial flight analyses have demonstrated that the aforementioned factors play a crucial role in the pitch rate damping or amplification of the airframe.

The current design proved stable and controlled flight in the presence of disturbances. The current design will not be subject to major changes so an optimisation of the control strategy is beneficial towards robust and efficient flight. To optimise the control strategy of the UHAV, detailed modelling of the system is recommended as a next step.

10

Conclusion

This research aimed at developing a first prototype of a novel tethered, unmanned hydro-propelled aerial vehicle that is capable of delivering water at a predetermined location for various purposes. A lightweight, simplistic design is proposed that reduces the number of actuators and limits effective water discharge. Additionally, position control was added by using external camera feedback. After testing various object tracking algorithms, the Lucas-Kanade optical flow algorithm and the YOLOv5 segmentation model proved to be reliable to track the drone.

10.1. Addressing the Research Questions

This research aimed to answer the main research question:
"Is a novel tethered unmanned hydro-propelled aerial vehicle capable of setpoint tracking and position control in the presence of disturbances, using minimum actuation?" This research question was split up into four sub-questions to make it specific, measurable, achievable, relevant, and timed, which together answer the main research question. Through a literature, analyses of physical principles, and flight testing, these could be answered.

Q1: What is the dynamic behaviour of the drone due to bank and yaw coupling in lateral attitude control?
A: Due to asymmetric nozzle deflections, negative bank-yaw coupling occurs. An asymmetric nozzle deflection changes the moment around the X- and Z-axis simultaneously, leading to coupled dynamics. Due to nozzle efficiencies, the degree to which bank-yaw coupling occurs varies for different nozzle deflections. In specific cases it was shown that a banking moment caused a much bigger yawing moment. Therefore, lateral disturbance rejection and setpoint tracking was not possible. The word negative is used because a positive bank motion causes a CW motion around P1, while the corresponding yawing motion leads to CCW motion.

(a) Can the design of the drone be adjusted to limit the (dominant) yawing motion during lateral stability control?
A: In order to limit the bank-yaw coupling such that banking can be achieved without (significant) yawing, the motion needs to be (partially) decoupled. By adding an actuator that creates control torques around the axis of the rotation that needs to be achieved only, one achieves decoupled motion. Adding the canting keel creates a control torque around the x-axis only in case $\theta = 0$, and if $\theta \neq 0$, the motion is positively coupled. This means that bank and yaw angles lead to CW or CCW movement around P1. Throughout its entire flight envelope, the banking moment was larger than the yawing moment, hence dominant bank motion was achieved.

(b) How can the controller be optimised to deal with bank and yaw coupling?

A: The controller was not designed to take yawing motion into account. By generating the desired control torques around the X-axis, torques around the Z-axis were handled by the controller as disturbances. However, through an optimisation of the canting keel deflection in combination with small asymmetric nozzle deflections, it is expected that the coupling can further be reduced. In addition, the head direction could be controlled.

Q2: What is the influence of the three flexible water hoses on the flight dynamics and can it be limited or dealt with?
A: The primary hose delivers the water to the drone and therefore cannot be eliminated. Vibrations occur in the primary hose which the controller must handle. Rotations acting perpendicular to the hose (pitch and yaw) are constrained and result in vibrations in the hose while rotational motions around the hose (bank) do not notably affect the vibrations in the hose. The secondary hoses deliver the water to the nozzles. The flexibility of the hoses create an additional system that vibrates and interacts with the rigid airframe. It is expected that due to mechanical and aerodynamic losses these vibrations are damped. However, by integrating the hoses with the airframe, this interaction can be reduced.

(a) Does the primary hose act as a natural damper?
A: The bending moment of the hose acts as a natural damper in pitch and yaw, however, the effect did not have notable impact on the pitch or yaw rate.

(b) Can the attachment point of the primary hose be adjusted for improved performance?
A: By increasing the length of the drone, the attachment point was moved forward. This resulted in more pitch down flight which was beneficial for the canting keel. However, vibrations in the primary hose had a larger moment which was mainly disadvantageous to the yaw rate.

(c) How can the controller deal with oscillations and eigenfrequencies of the primary hose?
A: As the nozzles are located above the center of gravity, they create a pitch up moment. A positive D-gain on the pitch angle, created a stabilising moment. Furthermore, the deflection of the nozzles created a delayed movement of the weight of secondary hose that could dampen or amplify the pitch rate depending on the gains that were set.

(d) How large is the impact of the non-minimum phase effect of the two secondary flexible hoses on the attitude control?
A: The impact was reduced due to the flexibility of the secondary hose. The more flexible the hose, the smaller the impact of the weight above the axis of rotation as the weight is deflected by a delay. However, the delayed displacement of the weight of the secondary hoses was an important pitch rate damper.

(e) How can this non-minimum phase effect be reduced through a mechanical solution?
A: By removing the weight of the secondary hose above the axis of rotation, the effect is removed. In addition, the interaction between the secondary hose and the airframe is removed such that the weight displacement of the secondary hoses due to deflections of the nozzle do not have an effect on the pitch rate.

(f) How can this non-minimum phase effect be reduced by flight software?
A: By reducing the gains, the nozzle deflections are smaller and the effect of the non-minimum phase and the interaction of the secondary hose with the rigid airframe is reduced. A trade-off was made between height setpoint tracking (high gains) and pitch rate damping (lower gains).

Q3: In terms of cost efficiency, what is the minimum number of actuators and moving parts to operate the drone?
A: This research showed that stable and controlled flight with two uniaxial thrust-vectoring nozzles was not possible. Bank-yaw coupling effects restricted lateral disturbance rejection and setpoint tracking. By adding a pressure differentiator or a canting keel, the drone could be controlled. Therefore, it was concluded that a minimum of three actuators were needed for satisfactory flight performance. However,

when adding more actuators, the controlled degree of freedom can increase, resulting in enhanced manoeuvrability. These are design choices and should be discussed with the stakeholders.

(a) How many controlled degrees of freedom (DoF) does the drone in its current configuration have?
A: Two. Bank, yaw, and lateral motion are coupled. Up- and forward motion are coupled. Pitch angle is not controlled and when $\theta = 0$, the bank and yaw angle are decoupled where only the bank angle is controlled.

(b) What is the flight envelope of the current system?
A: The flight envelope of the system is constrained by the tether. Rotate the flight path of figure 5 of the scientific paper around P1 to get the theoretical 3D flight path.

(c) How can the flight envelope and DoF be increased by adjusting the design of the drone?
A: By increasing the number of actuators, the DoF can be increased. Modifying the geometry of the system such as the size of the static pole, the length of the primary hose, or the weight of the drone, the flight path is adjusted. Reeling out the primary hose will add weight to system so the lift needs to be sufficient to carry the additional weight.

Q4: How can camera feedback be integrated into the system to provide real-time position control?
A: In order to keep the system as lightweight and simplistic as possible, an external camera was used to provide position feedback. By placing the camera at P1 (the static pole) facing towards the drone, the ground station could run the YOLOv5 computer vision algorithm to detect the location of the drone. By creating a user interface, a human operator was able to set waypoints. The information about its current position and coordinates of the waypoint were sent to FMU after which the outer loop position controller computed the attitude commands. When combining the YOLOv5 segmentation algorithm with a lightweight optical flow algorithm such as Lucas-Kanade, it is expected that real-time object tracking at ± 30 frames per second can be reached, using a CPU.

(a) Which computer vision algorithm can be used for real-time motion tracking of the drone and/ or its water jets?
A: Several computer vision algorithms were tested on their performance for real-time motion tracking. The goal is to use the center of gravity of the drone for position control while detecting the water jets for water jet placement. Machine learning algorithms proved to be the most robust detection method. The YOLOv5-nano segmentation model performed best in terms of water jet detection with the highest and fastest segmentation rate. With a detection rate of approximately 7.9 frames per second, this was a first step towards real-time detection.

(b) What is a simple, yet effective user interface that interacts with a human operator to control the position of the drone?
A: Using the OpenCV library, one can interact with the frame using the mouse and keyboard. By clicking with the left mouse button, the waypoint on the 2D frame was set. The current location was compared with the waypoint coordinate and vertical and horizontal differences were sent to the FMU using MAVLink.

(c) What is the necessary software, hardware and system architecture?
A: An external camera, a ground station, a human operator, and flight code adjustments were needed. The operator would provide waypoints by clicking on the interface on the ground station. The computer vision algorithm collaborates with the interface to send position commands to the FMU over a UDP connection. The camera was placed at P1, at $\alpha = 0$. An outer loop position controller was implemented that used information from the computer vision algorithm to send attitude commands to the inner loop attitude controller.

10.2. Concluding Remarks

The final goal of the project aims at developing a lightweight and simplistic UHAV that is capable of stable flight in the presence of disturbances while executing a mission autonomously. While the mission is not yet determined, it aims at spraying water onto a target area of interest. After fabricating the drone and setting up the electronic equipment, the attitude controller of the PX4 was adjusted to be compatible with the UHAV, while adhering to the controller design. Due to limitations in flight stability and controllability of the drone, a trade-off between different designs was made to improve the flight performance. Although a discussion with different stakeholders at Actiflow resulted in favour of using pressure differentiation, it was decided that this would not fit into the timeline of the project due to budget and manufacturability constraints. As such, the trade-off resulted in adjustments in the controller architecture and the introduction of a canting keel. The results were promising and stable and controlled flight was possible in the presence of disturbances. As such, the goals of the research were extended to add a position control loop. A Lidar was used for height feedback and an external camera was used to show that position control was possible using object tracking.

At the beginning of the project, the decision was made to conduct this research without a detailed simulation of the system. Analyses of the flight performance were based on physical principles and flight data. Analyses showed a high probability of necessary design changes due to limited actuation and coupling effects. The current system, however, has satisfactory flight performance and an optimisation in its control strategy is recommended. For future development, it is recommended to invest time and resources into the production of an analytical model that can simulate the behaviour of the drone. By carefully modelling (parts of) the system, detailed insights into the dynamic behaviour of the drone can be gained. Additionally, development and performance evaluation can be done without flying, accelerating the progress.

Furthermore, in terms of practical design, two improvements are recommended. To remove weight of the design, the weight of the keel could be replaced by the batteries. Next, a landing mechanism or docking station should be designed such that the drone is able to take-off and land by itself. Depending on the geometry of the static pole and the primary hose, the docking station is required to handle take-off and landing of the drone at different locations and pitch angles.

Another aspect that this research did not focus on, is the safety of the product. While the initial research scope included safety deliverables, discussions with stakeholders led to a focus on position control. However, before the system can be offered as a commercial product, safety must be taken into account. An air-bubble inside the hose or another malfunction could result in a crash. An emergency system must be created that reduces the impact of a crash to guarantee safe operations.

In conclusion, the project successfully delivered a novel tethered unmanned hydro-propelled system, capable of setpoint tracking and position control, using external camera feedback. The system required three actuators, which is a reduction in comparison to other UHAV systems of this kind. When the budget becomes available to continue the development of the UHAV, it is expected that a safe and reliable autonomous system can be built for commercial use.

References

Agbeyangi, A. O., Odiete, J. O., & Olorunlomerue, A. B. (2016). Review on uavs used for aerial surveillance. *Journal of Multidisciplinary Engineering Science and Technology*, *3*(10), 5713–5719 (cit. on p. 7).

Ando, H., Ambe, Y., Ishii, A., Konyo, M., Tadakuma, K., Maruyama, S., & Tadokoro, S. (2018). Aerial hose type robot by water jet for fire fighting. *IEEE Robotics and Automation Letters*, *3*(2), 1128–1135 (cit. on pp. 10, 11, 16).

Ando, H., Ambe, Y., Yamaguchi, T., Konyo, M., Tadakuma, K., Maruyama, S., & Tadokoro, S. (2019). Fire fighting tactics with aerial hose-type robot "dragon firefighter". *2019 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*, 291–297 (cit. on p. 10).

Bernacchia, D. (2019). *Design of thrust vectoring attitude control system for lunar lander flying testbed* (Master's thesis). University of Bologna. (Cit. on pp. 12, 13).

Boukoberine, M. N., Zhou, Z., & Benbouzid, M. (2019). Power supply architectures for drones - a review. *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, *1*, 5826–5831. https://doi.org/10.1109/IECON.2019.8927702 (cit. on p. 8)

Cavoukian, A. (2012). *Privacy and drones: Unmanned aerial vehicles*. Information; Privacy Commissioner of Ontario, Canada Ontario. (Cit. on p. 6).

Chen, W.-H., Yang, J., Guo, L., & Li, S. (2015). Disturbance-observer-based control and related methods—an overview. *IEEE Transactions on industrial electronics*, *63*(2), 1083–1095 (cit. on p. 15).

Christ, P. F., Lachner, F., Hösl, A., Menze, B., Diepold, K., & Butz, A. (2016). Human-drone-interaction: A case study to investigate the relation between autonomy and user experience. In G. Hua & H. Jégou (Eds.), *Computer vision – eccv 2016 workshops* (pp. 238–253). Springer International Publishing. (Cit. on p. 7).

ChunzPS. (2021). Sending custom message from mavros to px4 [Video]. Retrieved November 14, 2022, from https://www.youtube.com/watch?v=S9TOyIH570U&t=2s. (Cit. on p. 66)

De Croon, G., Perçin, M., Remes, B., Ruijsink, R., & De Wagter, C. (2016). The delfly. *Dordrecht: Springer Netherlands. doi*, *10*, 978–94 (cit. on p. 6).

de Croon, G., & de Wagter, C. (2021). Ae4317 autonomous flight of micro air vehicles. university course. *Delft University of Technology* (cit. on p. 4).

Dodge, J. (2018). *From continuous-time domain to microcontroller code* [Application Note AN0019]. UnitedSiC. https://unitedsic.com/appnotes/UnitedSiC_AN0019_a2d.pdf. (Cit. on p. 56)

Donateo, T., Ficarella, A., Spedicato, L., Arista, A., & Ferraro, M. (2017). A new approach to calculating endurance in electric flight and comparing fuel cells and batteries. *Applied energy*, *187*, 807–819 (cit. on p. 8).

Douglas, B. (2018). What is feedforward control? [Video]. Retrieved September 29, 2022, from https://www.youtube.com/watch?v=FW_ay7K4jPE%5C&t=1s. (Cit. on p. 15)

Douglas, B. (2019a). The fundamentals of control theory. (Cit. on p. 12).

Douglas, B. (2019b). What are non-minimum phase systems? [Video]. Retrieved October 4, 2022, from https://www.youtube.com/watch?v=jGEkmDRsq_M. (Cit. on pp. 15, 16)

Dronecode. (2022a). *Px4 drone autopilot*. Retrieved July 31, 2022, from https://github.com/PX4/PX4-Autopilot. (Cit. on p. 47)

Dronecode. (2022b). *Fixed-wing pid tuning guide*. Retrieved September 27, 2022, from https://docs.px4.io/main/en/config_fw/pid_tuning_guide_fixedwing.html. (Cit. on p. 14)

Dronecode. (2022c). *Controller diagrams*. Retrieved September 27, 2022, from https://docs.px4.io/main/en/flight_stack/controller_diagrams.html. (Cit. on p. 13)

Dunker, S. (2018). Tether and bridle line drag in airborne wind energy applications. In R. Schmehl (Ed.), *Airborne wind energy: Advances in technology development and research* (pp. 29–56). Springer Singapore. https://doi.org/10.1007/978-981-10-1947-0_2. (Cit. on p. 10)

Economou, D., Mavroidis, C., & Antoniadis, I. (2001). Proper filter design procedure for vibration suppression using delay-error-order curves. *ASME International Mechanical Engineering Congress and Exposition*, *35609*, 915–923 (cit. on p. 11).

Eissa, M., & Sayed, M. (2006). A comparison between active and passive vibration control of nonlinear simple pendulum. part ii: Longitudinal tuned absorber and negative gφ and gφn feedback. *Mathematical and Computational Applications*, *11*(2), 151–162 (cit. on p. 11).

Elijah, T., Jamisola, R. S., Tjiparuro, Z., & Namoshe, M. (2021). A review on control and maneuvering of cooperative fixed-wing drones. *International Journal of Dynamics and Control*, *9*(3), 1332–1349 (cit. on p. 6).

Ellis, G. (2012). *Control system design guide: Using your computer to understand and diagnose feedback controllers*. Butterworth-Heinemann. (Cit. on pp. 11, 12, 14, 15).

Fagiano, L. (2017). Systems of tethered multicopters: Modeling and control design [20th IFAC World Congress]. *IFAC-PapersOnLine*, *50*(1), 4610–4615. https://doi.org/https://doi.org/10.1016/j.ifacol.2017.08.653 (cit. on p. 10)

Fechner, U., van der Vlugt, R., Schreuder, E., & Schmehl, R. (2015). Dynamic model of a pumping kite power system. *Renewable Energy*, *83*, 705–716 (cit. on p. 10).

Floreano, D., & Wood, R. J. (2015). Science, technology and the future of small autonomous drones. *nature*, *521*(7553), 460–466 (cit. on pp. 7, 62).

FusionFlight. (2022). *Jetquad: Quad turbine vtol drone*. Retrieved July 31, 2022, from https://fusionflight.com/jetquad/. (Cit. on p. 9)

Geatrix, D. R. (2012). *Powered flight: The engineering of aerospace propulsion* (1st ed.). Springer London. (Cit. on p. 8).

Gerken, J. H., & Bright, G. (2017). Design of a semi-autonomous water-driven flight platform. *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 1–6. https://doi.org/10.1109/M2VIP.2017.8211500 (cit. on p. 9)

Goodarzi, F. A., Lee, D., & Lee, T. (2014). Geometric stabilization of a quadrotor uav with a payload connected by flexible cable. *2014 American Control Conference*, 4925–4930 (cit. on pp. 4, 11).

Hao, F., Zhang, D., Cao, L., & Tang, S. (2019). Disturbance decoupling control for flexible air-breathing hypersonic vehicles with mismatched condition. *Asian Journal of Control*, *21*(3), 1100–1110 (cit. on p. 14).

Hart, S. G., & Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology* (pp. 139–183). Elsevier. (Cit. on p. 7).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961–2969 (cit. on p. 65).

Henry, G., Humble, R., & Larson, W. (1995). *Space propulsion analysis and design*. McGraw-Hill. https://books.google.fr/books?id=7OmMKQAACAAJ. (Cit. on p. 8)

Horn, B. K., & Schunck, B. (1980). Determining optical flow (artificial intelligence laboratory). *Massachusetts Institute of Technology, Cam bridge, MA*, *2139* (cit. on p. 64).

Huang, H.-M., et al. (2008). Autonomy levels for unmanned systems (alfus) framework. *National Institute of Standards and Technology*, *1: Terminology* (cit. on p. 7).

Hyvärinen, J., Karlsson, M., & Zhou, L. (2020). Study of concept for hydraulic hose dynamics investigations to enable understanding of the hose fluid–structure interaction behavior. *Advances in Mechanical Engineering*, *12*(4), 1687814020916110. https://doi.org/10.1177/1687814020916110 (cit. on p. 10)

Intwala, A., & Parikh, Y. (2015). A review on vertical take off and landing (vtol) vehicles. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, *2*(2), 187–191 (cit. on p. 6).

Ioppo, P. G. (2017). *The design, modelling and control of an autonomous tethered multirotor uav* (Master's thesis). Stellenbosch: Stellenbosch University. (Cit. on p. 11).

Jetpack America. (2016). *Jetpack america history*. Retrieved July 22, 2022, from https://www.jetpackamerica.com/jetpack-america-history/. (Cit. on p. 8)

Johnson, E. N., Baillieul, J., & Samad, T. (2020). Unmanned aerial vehicle (uav). In *Encyclopedia of systems and control* (pp. 1–6). Springer London. https://doi.org/10.1007/978-1-4471-5102-9_100039-1. (Cit. on p. 7)

Klinker, M. R. (2016). *Tethered uav flight using a spherical position controller* (Master's thesis). Massachusetts Institute of Technology. (Cit. on p. 11).

Kotaru, P., & Sreenath, K. (2020). Multiple quadrotors carrying a flexible hose: Dynamics, differential flatness and control. *IFAC-PapersOnLine*, *53*(2), 8832–8839 (cit. on pp. 4, 11, 13, 16, 68).

Krijnen, D., & Dekker, C. (2014). Ar drone 2.0 with subsumption architecture. *Artificial intelligence research seminar* (cit. on p. 6).

Kumar, V., & Michael, N. (2012). Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, *31*(11), 1279–1291. https://doi.org/10.1177/0278364912455954 (cit. on p. 7)

Lee, D.-H., Huynh, T., Kim, Y.-B., & Soumayya, C. (2021). Motion control system design for a flying-type firefighting system with water jet actuators. *Actuators*, *10*(10). https://www.mdpi.com/2076-0825/10/10/275 (cit. on pp. 10, 11, 13, 16)

Lee, T., Leok, M., & McClamroch, H. (2007). Lagrangian mechanics and variational integrators on two-spheres. *International Journal for Numerical Methods in Engineering* (cit. on pp. 4, 11).

Lee, T. (2015). Geometric controls for a tethered quadrotor uav. *2015 54th IEEE conference on decision and control (CDC)*, 2749–2754 (cit. on pp. 4, 11).

Lee, T., Leok, M., & McClamroch, N. H. (2018). *Global formulations of lagrangian and hamiltonian dynamics on manifolds*. Springer. (Cit. on pp. 4, 11).

Liljeback, P., Stavdahl, O., & Beitnes, A. (2006). Snakefighter - development of a water hydraulic fire fighting snake robot. *2006 9th International Conference on Control, Automation, Robotics and Vision*, 1–6. https://doi.org/10.1109/ICARCV.2006.345311 (cit. on p. 10)

Liu, X., & Zhou, H. (2019). Unmanned water-powered aerial vehicles: Theory and experiments. *IEEE Access*, *7*, 15349–15356. https://doi.org/10.1109/ACCESS.2019.2895335 (cit. on pp. 10–12, 16)

Liu, Y., & Yu, H. (2013). A survey of underactuated mechanical systems. *IET Control Theory & Applications*, *7*(7), 921–935 (cit. on p. 13).

Mulder, J., van Staveren, W., van der Vaart, J., de Weerdt, E., de Visser, C., in 't Veld, A., & Mooij, E. (2013). *Lecture notes ae3203: Flight dynamics*. TU Delft Open. (Cit. on pp. 71, 73).

Olfati-Saber, R. (2000). *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles* (Doctoral dissertation). Massachusetts Institute of Technology. (Cit. on p. 13).

Olsder, G. J., & van der Woude, J. W. (2005). *Mathematical systems theory* (Vol. 4). VSSD Delft. (Cit. on p. 12).

Rashad, R., Goerres, J., Aarts, R., Engelen, J. B. C., & Stramigioli, S. (2020). Fully actuated multirotor uavs: A literature review. *IEEE Robotics & Automation Magazine*, *27*(3), 97–107. https://doi.org/10.1109/MRA.2019.2955964 (cit. on p. 6)

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 64, 65).

Ryll, M., Bülthoff, H. H., & Giordano, P. R. (2012). Modeling and control of a quadrotor uav with tilting propellers. *2012 IEEE International Conference on Robotics and Automation*, 4606–4613. https://doi.org/10.1109/ICRA.2012.6225129 (cit. on p. 13)

Singhose, W., & Vaughan, J. (2011). Reducing vibration by digital filtering and input shaping. *IEEE Transactions on Control Systems Technology*, *19*(6), 1410–1420 (cit. on p. 12).

Spurny, V., Pritzl, V., Walter, V., Petrlik, M., Baca, T., Stepan, P., Zaitlik, D., & Saska, M. (2021). Autonomous firefighting inside buildings by an unmanned aerial vehicle. *IEEE Access*, *9*, 15872–15890 (cit. on pp. 10, 16).

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., & Goodrich, M. (2006). Common metrics for human-robot interaction. *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, 33–40 (cit. on p. 7).

Uren, K., & van Schoor, G. (2011). Predictive pid control of non-minimum phase systems. In V. D. Yurkevich (Ed.), *Advances in pid control*. InTech. (Cit. on pp. 15, 16).

Vergouw, B., Nagel, H., Bondt, G., & Custers, B. (2016). Drone technology: Types, payloads, applications, frequency spectrum issues and future developments. In *The future of drone use: Opportunities and threats from ethical and legal perspectives* (pp. 21–45). T.M.C. Asser Press. (Cit. on pp. 2, 6–8).

Vinayagam, A., & Sinha, N. K. (2014). An assessment of thrust vector concepts for twin-engine airplane. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *228*(6), 960–979 (cit. on pp. 13, 16).

Vinayagam, A., & Sinha, N. (2014). An assessment of thrust vector concepts for twin-engine airplane. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *228*(6), 960–979 (cit. on p. 69).

Vonk, M., & Bohacek, P. (2013). Carried by impulse: The physics of water jetpacks. *Physics Online*, *66*(1), 54–55. https://doi.org/10.1063/PT.3.1865 (cit. on p. 8)

Wang, L., Liu, J., Yang, C., & Wu, D. (2021). A novel interval dynamic reliability computation approach for the risk evaluation of vibration active control systems based on pid controllers. *Applied Mathematical Modelling*, *92*, 422–446 (cit. on p. 14).

Wang, T., Qiu, J., Fu, S., & Ji, W. (2016). Distributed fuzzy H_∞ filtering for nonlinear multirate networked double-layer industrial processes. *IEEE Transactions on Industrial Electronics*, *64*(6), 5203–5211 (cit. on p. 15).

Winget, J., & Huston, R. (1976). Cable dynamics—a finite segment approach. *Computers & Structures*, *6*(6), 475–480 (cit. on p. 10).

Xie, F., Zheng, X., Triantafyllou, M. S., Constantinides, Y., & Karniadakis, G. E. (2016). The flow dynamics of the garden-hose instability. *Journal of Fluid Mechanics*, *800*, 595–612. https://doi.org/10.1017/jfm.2016.364 (cit. on p. 10)

Yamaguchi, T., Ambe, Y., Ando, H., Konyo, M., Tadakuma, K., Maruyama, S., & Tadokoro, S. (2019). A mechanical approach to suppress the oscillation of a long continuum robot flying with water jets. *IEEE Robotics and Automation Letters*, *4*(4), 4346–4353. https://doi.org/10.1109/LRA.2019.2932582 (cit. on pp. 10, 11, 16)

Yang, H., Lee, Y., Jeon, S.-Y., & Lee, D. (2017). Multi-rotor drone tutorial: Systems, mechanics, control and state estimation. *Intelligent Service Robotics*, *10*(2), 79–93 (cit. on p. 6).

Yang, J., Chen, W.-H., Li, S., Guo, L., & Yan, Y. (2016). Disturbance/uncertainty estimation and attenuation techniques in pmsm drives—a survey. *IEEE Transactions on Industrial Electronics*, *64*(4), 3273–3285 (cit. on p. 15).

Zheng, Q., Chen, Z., & Gao, Z. (2009). A practical approach to disturbance decoupling control. *Control engineering practice*, *17*(9), 1016–1025 (cit. on p. 14).