**MINISTRY OF EDUCATION AND TRAINING**
**HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION**

---

**FACULTY OF MECHANICAL ENGINEERING**

# GRADUATION THESIS

## TOPIC: RESEARCH, DESIGN AND FABRICATION OF FIREFIGTING QUADCOPTERS

**Supervisor:** **PhD HA LE NHU NGOC THANH**

**Students :** **VU DUC BINH……..ID: 20146478**

**DOAN THANH NAM……...ID: 20146506**

**TRAN LE NHAT HUY…….ID: 20146494**

**Class:** **201461**

**Academic year :** **2020-2024**

**Ho Chi Minh City, July – 2024**

**HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY OF MECHANICAL ENGINEERING**

---

**MECHATRONICS ENGINEERING**

# GRADUATION THESIS

## TOPIC: RESEARCH, DESIGN AND FABRICATION OF FIREFIGTING QUADCOPTERS

**Supervisor:** **PhD HA LE NHU NGOC THANH**

**Students :** **VU DUC BINH……..ID: 20146478**

**DOAN THANH NAM……...ID: 20146506**

**TRAN LE NHAT HUY…….ID: 20146494**

**Class:** **201461**

**Academic year :** **2020-2024**

**Ho Chi Minh City, July - 2024**

## COMMITMENT

- Topic: Research, Design and Fabrication of firefighting quadcopters
- Supervisor: PhD Ha Le Nhu Ngoc Thanh
- Students: Vu Duc Binh.....ID: 20146478

  Doan Thanh Nam..... ID: 20146506

  Tran Le Nhat Huy.....ID: 20146494

- Thesis Deadline: 4/7/2024
- Class: 201461
- Phone number: 0888360856
- Email: 20146478@student.hcmute.edu.vn
- Thesis submission date:
- Commitment: “We hereby declare that this dissertation: **Research, Design and Fabrication of FireFighting Quadcopters** is our work and has been conducted by ourself. We have not copied from any published article without citing the source. If there is any violation, we will take full responsibility.”

**Group of students**

- Vu Duc Binh

  Doan Thanh Nam

  Tran Le Nhat Huy

## ACKNOWLEDGMENT

On behalf of the research team, we would like to extend our deepest gratitude to PhD Ha Le Nhu Ngoc Thanh for his unwavering guidance and expertise throughout our academic journey. His mentorship and inspiration have been instrumental in the successful completion of this project.

We also extend our sincere appreciation to the esteemed faculty members of the Department of Mechatronics, particularly, and to the broader Faculty of the Mechanical Engineering and the Ho Chi Minh City University of Technology and Education. Their dedication to teaching and unwavering support have been invaluable throughout our academic endeavors.

We are immensely grateful to our families, friends, and loved ones for their encouragement, care, and assistance throughout our studies and the completion of this project. Their support has been a source of strength and motivation.

While we have strived to produce a high-quality project, we acknowledge that our limitations in time and experience may have resulted in shortcomings. We humbly welcome any constructive criticism and valuable feedback from our esteemed lectures to further refine our knowledge and skills. We are eager to learn and grow from your insights.

Group of students

Vu Duc Binh

Doan Thanh Nam

Tran Le Nhat Huy

## ABSTRACT

## RESEARCH, DESIGN AND FABRICATION OF FIREFIGHTING QUADCOPTERS

UAVs are becoming increasingly popular in society. Their biggest strength is that they can fly to difficult-to-reach locations, hazardous areas for humans, or rough terrain. With their great potential, UAVs are being utilized to revolutionize many industries, being used in various fields like agriculture, military, and surveillance equipment, significantly increasing productivity and work efficiency.

However, there are still limitations and challenges in the current firefighting field. Based on this reality, our team has researched and implemented a UAV capable of carrying heavy objects, specifically a fire extinguisher ball. The fire extinguisher ball is held and released through a clipper controlled automatically by companion computers like Jetson Nano. Additionally, flight modes are established for the UAV to enable it to automatically search for fire sources using a camera with an autofocus function. AI models are built to optimize fire detection quickly and with high reliability. The choice of propulsion engine, UAV frame, etc, is made in accordance with pre-calculated aerodynamic equations.

As a result, the team has researched and developed a fire-fighting quadcopter UAV capable of carrying a load of approximately 1.5 kg, with the most notable advantage being its ability to automatically search for fire sources. Hardware-in-the-Loop (HILT) simulation tools are used to ensure it runs according to the pre-programmed code, making its operation smoother and more stable outdoors.

# TABLE OF CONTENT

## TABLE OF TABLES

## TABLE OF IMAGES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| UAV | **U**nmanned **A**erial **V**ehicle |
| FPV | **F**irst **P**erson **V**iew |
| LiDAR | **L**ight **D**etection and **R**anging |
| VTOL | **V**ertical **T**ake-**O**ff and **L**anding |
| FC | **Fl**ight **c**ontroller |
| IMU | **I**nertial **M**easurement **U**nit |
| GPS | **G**lobal **P**ositioning **S**ystem |
| PID | **P**roportional-**I**ntegral-**D**erivative |
| ROS | **R**obot **O**perating **S**ystem |
| QGC | **QG**round**C**ontrol |
| ESC | **E**lectronic **S**peed **C**ontroller |
| PWM | **P**ulse **W**idth **M**odulation |
| HITL | **H**ardware-**i**n-**t**he-**l**oop |
| GPIO | **G**eneral-**P**urpose **I**nput/**O**utput |
| BEC | **B**attery **E**liminator **C**ircui |

# INTRODUCTION

## 1. Reason for choosing a topic

- The realm of public safety is undergoing a transformative shift with the emergence of firefighting quadcopters. These unmanned aerial vehicles (UAVs) represent a significant advancement in robotics and artificial intelligence, offering innovative solutions to longstanding challenges faced by fire departments globally.

- *[3]*Firefighting quadcopters possess the distinct advantage of traversing hazardous or inaccessible areas with agility, surpassing the limitations of human firefighters. Equipped with advanced sensor technology and autonomous navigation systems, these drones gather real-time data on fire behavior and provide crucial situational awareness. This enhances informed decision-making by fire crews, leading to increased efficiency in response efforts.

- Furthermore, the capabilities of these quadcopters extend beyond data collection.*[2]*By leveraging advanced sensors, they facilitate early fire detection, allowing for prompt intervention and potentially mitigating the spread of blazes. Similarly, the ability to deliver firefighting agents with precision to specific locations significantly reduces response times and minimizes the risk to human life.

- Therefore, our team has proposed the design and development of a firefighting drone integrated with multiple functions, including image processing and highly accurate navigation algorithms. The topic is: Research, Design, and Fabrication of a Firefighting Quadcopter.

## 2. Research objectives of the topic

- Calculate and select suitable components to assemble the Drone.
- Build dynamic equations suitable for Drones.
- Build fire identification using cameras and JetsonNano.
- Design and fabricate a pick-up handle with the function of picking up and dropping fire-extinguishing ball.
- Build mathematical models and algorithms for drone movement to perform fire extinguishing tasks.

## 3. Research subjects and scope

### 3.1. Research subject

- Motors; ESC; Divided Source
- JetsonNano
- Camera IMX519
- Pixhawk 6C
- MAVROS
- HILT (Hardware in the loop)
- Yolov9 (Recognition Flame)

### 3.2. Research scope

- Calculate and assemble Drones using engines Sunsky x2814-7 1100 KV
- Use control modes for Drones depending on the specific case
- Identify and follow when a fire is detected, then drop a fire extinguisher onto that fire

### 3.3. Research method

- Calculate parameters for components and assemble them.
- Set parameters for Pixhawk 6C.
- Design and fabricate the mechanism to hold and release the ball.
- Hardware simulation (HITL) for Drones.
- Fly for real, test the Drone's ability to detect fire, how to move when it detects fire and release fire balls to put out the fire.

### 3.4. Structure of project

**INTRODUCTION**

CHAPTER 1: RESEARCH TOPIC OVERVIEW

CHAPTER 2: STRUCTURE AND PRINCIPLES OF OPERATION
CHAPTER 3: OVERVIEW OF QUADCOPTER KINEMATIC

CHAPTER 4: CONTROLLER DESIGN

CHAPTER 5: APPLICATION FOR FIREFIGHTING QUADCOPTER

CHAPTER 6: RESULT OF SIMULATION AND EXPERIMENT

# CHAPTER 1: RESEARCH TOPIC OVERVIEW

## 1.1. What is Drone/UAV

A drone, also called an unmanned aerial vehicle (UAV), is a broad term that refers to an aircraft that operates autonomously or by remote control, with no pilot on board. Drones are usually small-or-medium-sized, and these 'flying robots' can carry out a wide range of tasks, from stealth military operations to package delivery to aerial photography.

*Figure 1. 1 Drone/UAV*

## 1.2. History of evolution and Development

### 1.2.1. History of Aviation

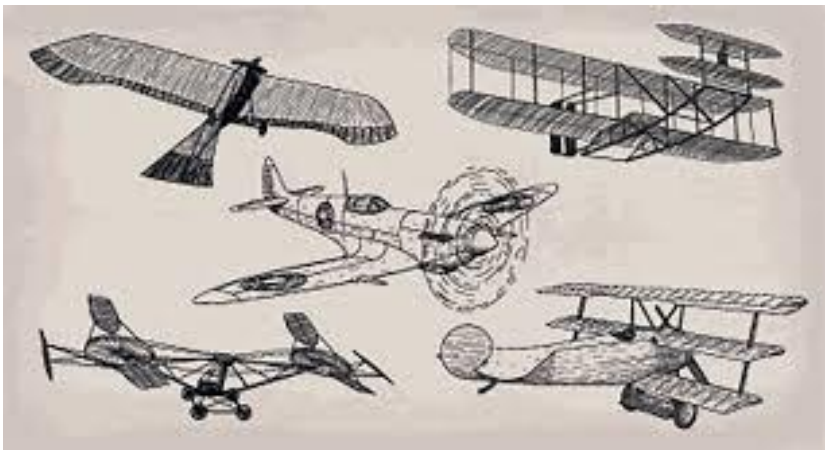


*Figure 1. 2 Some types of Plane*

The desire to fly has been a long-standing human dream, predating the knowledge and tools for experimentation. Today, flight has transformed global travel. There are explore key historical moments that have significantly influenced modern aviation:

- **1490 – Leonardo da Vinci's 'flying machine':** Leonardo created sketches of a bird-inspired flying machine and conceptualized designs that anticipated the modern helicopter, parachute, and hang-glider.
- **1783 – The hot air balloon:** Their original hot-air balloon design was later improved and used to make larger balloons that could go even higher.
- **1903 – The first (official) aeroplane:** the Wright Flyer, successfully flew in front of five people for around 12 seconds in 1903. By 1905, after improving the designs, their aircraft could stay in the air for 39 minutes!
- **1939 – The first helicopter flight in the USA: The** design was later improved and developed into the helicopters that we know today.

### 1.2.2. History of Drone

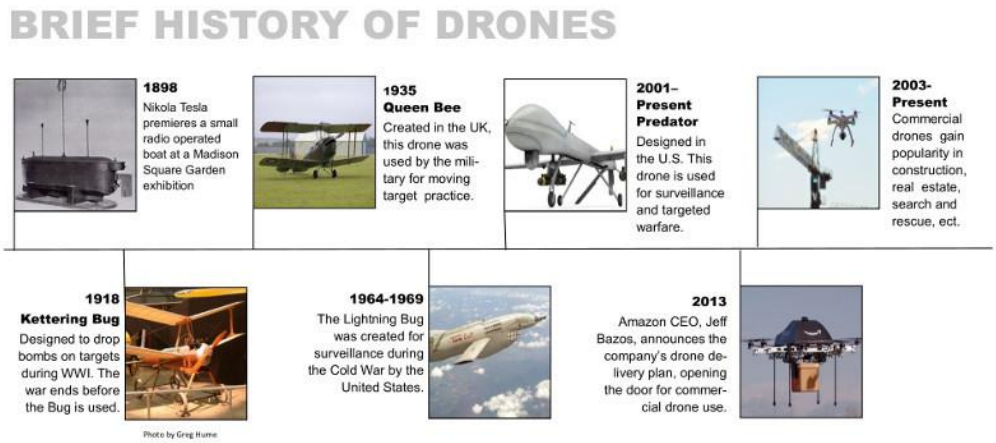



*Figure 1. 3 History of Drones*

Pilotless vehicles were first developed during WWI by Britain and the USA. These early drones evolved through the inter-war period and saw significant use in the Vietnam War for reconnaissance and combat. Advancements continued globally, with new models featuring improved endurance and capabilities. Modern drones serve diverse roles, from environmental monitoring to military operations:

- **1917**: The first pilotless vehicles, such as Britain's Aerial Target, a small radio-controlled aircraft, were tested.
- **1918**: The American aerial torpedo known as the Kettering Bug made its first flight
- **1935**: Britain developed radio-controlled aircraft like the DH.82B Queen Bee for target practice, leading to the term 'drone.'
- **1960s – 1970s**: During the Vietnam War, drones were extensively used for reconnaissance and other combat roles.
- **Recent Years**: Modern drones have diversified functions, including environmental monitoring, disaster search operations, photography, and commercial deliveries

## 1.3. Application of Drone into practice

- **Agriculture:** .*[9]* Drones significantly enhance agricultural productivity by providing regular land observation data to stakeholders. They are increasingly used for spraying, fertilization, and crop damage detection, optimizing production and reducing physical labor. Drones also save time in research, seed planting, livestock monitoring, and crop yield prediction. Their integration with smart farming techniques allows producers to efficiently monitor and plan agricultural activities. As drone and satellite data usage grows, these technologies will further streamline agricultural management

*Figure 1. 4 Drone spraying Pesticide*

- **Environment:** Drones play a crucial role in environmental control and emergency response in urban areas. They aid in preventing pollution,

combating poaching, and tracking endangered animals using thermal cameras. Additionally, drones help monitor animal behavior and health, and are used by oil companies to inspect for oil and gas leaks, quickly detecting and mitigating potential risks.

*Figure 1. 5 Drone flying in the city*

- **Mapping:** Drones, becoming increasingly widespread in mapping, can map almost all terrains quickly and in three dimensions. For this purpose, LiDAR Drones with sensors provide highly successful and accurate data. LiDAR technology offers important solutions in the evaluation of agricultural products as well as the mapping of landforms.

*Figure 1. 6 Drone-based land surveying*

- **Logistic:** Drones revolutionize the logistics industry by transporting food, packages, and goods, especially in urgent or remote areas. However, their limited carrying capacity remains a significant challenge. In warehouse management, drones scan various materials, while heavy-duty models could streamline material transportation, potentially reducing road traffic.

*Figure 1. 7 Drone-based warehouse inventory*

- **Military:** Drones were first used for military purposes, serving roles from surveillance to transportation of supplies and weapons. Modern military drones, like the MQ-9 Reaper, boast advanced features for reconnaissance and air strikes, with long flight ranges and high altitudes. They play a crucial role in contemporary warfare, evolving into sophisticated tools for diverse missions.

*Figure 1. 8 Drone is used by military*

## 1.4. All of usage notes and instruction

❖ **Before you fly:**

- **Pre-flight checklist:** Checking the battery levels, propellers for damage, and ensuring the flight area is clear of obstacles and people.
- **Understand the Controls:** Familiarize yourself with your drone's specific controls and practice in a safe, open space before venturing outdoors.
- **Check the geological:** Check the regulations in your area

❖ **While flying:**

- **Location:** Be aware of your surroundings and fly in open areas away from crowds, power lines, and buildings. Respect privacy laws and avoid flying over people or property without permission
- **Weather Matters:** Don't fly your drone in high winds, rain, snow, or other adverse weather conditions that could affect its stability and control.
- **Start Low and Slow:** Especially for beginners, take off slowly and gradually increase altitude as you get comfortable with the controls.

❖ **Addition usages notes:**

- **Battery Life:** Be mindful of battery life and plan your flights accordingly. Most drones will give you a warning when the battery is low, and it's important to land before it dies to avoid losing control.
- **Harassment of privacy:** Don't fly your drone over private property or areas where people have a reasonable expectation of privacy.

# CHAPTER 2: STRUCTURE AND PRINCIPLES OF OPERATION

## 2.1. The common of configuration and operating principles of some Drone's genre

### 2.1.1. Multicopter

**Multi-rotor unmanned aerial vehicles (UAVs) offer a cost-effective solution for aerial observation tasks.** These UAVs, characterized by their multiple rotors (commonly configured as tricopters, quadcopters, hexacopters, or octocopters), provide superior maneuverability and precise. Their ease of operation and relative affordability further contribute to their popularity in various fields.

*Figure 2. 1 Multicopter*

### 2.1.2. Fix-wing

In contrast to multi-rotor UAVs, **fixed-wing utilize a fixed, airfoil-shaped wing to generate lift, akin to a conventional airplane**. This design eliminates the need for rotors solely dedicated to maintaining altitude. Consequently, fixed-wing achieve greater energy efficiency as their propulsion system focuses solely on forward motion.

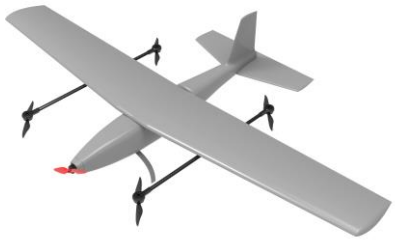


*Figure 2. 2 Fix-wing*

### 2.1.3. Single-Drone

**Single-rotor unmanned aerial vehicles (UAVs) exhibit robust construction and durability.** Their design closely resembles that of traditional helicopters, employing a single, large rotor that functions similarly to a rotating wing. To maintain directional control and stability during flight

*Figure 2. 3 Single-Drone*

### 2.1.4 Fixed-wing hybrid VTOL

**Hybrid Vertical Take-Off and Landing (VTOL) represent an emerging class of aerial vehicles that combine the advantages of fixed-wing and rotor-based designs.** These UAVs integrate rotors into their fixed wings, enabling vertical takeoff, landing, and hovering capabilities. While currently limited in commercial availability, advancements in technology are expected to propel hybrid VTOL UAVs towards greater popularity in the near future.

*Figure 2. 4 Fixed-wing hybrid VTOL*

### 2.1.5 Compare advantages and disadvantages in each drones

| Drone types | Pros | Cons | Technical used |
|---|---|---|---|
| **Multi-rotor** | • Accessibility<br>• Ease of use<br>• Good camera control<br>• VTOL and hover flight<br>• Can operate in a confined area | • Short flight times<br>• Small payload capacity | • Visual inspections<br>• Thermal reports<br>• Photography & Videography<br>• 3D scans |
| **Fixed-wing** | • Long endurance<br>• Large area coverage<br>• Fast flight speed | • Launch and recovery needs a lot of space<br>• No VTOL/hover<br>• Harder to fly, more training needed<br>• Expensive | • Aerial Mapping<br>• Agriculture<br>• Inspection<br>• Construction<br>• Security |
| **Single -rotor** | • VTOL and hover flight<br>• Long endurance (with gas power) | • More dangerous | • Aerial LIDAR laser scan<br>• Drone surveying |

| | | | |
|---|---|---|---|
| | • Heavier payload capability | • Harder to fly, more training needed<br>• Expensive | • Carrying heavy payloads |
| **Fixed-Wing Hybrid** | • VTOL<br>• Long-endurance flight | • Not perfect at either hovering or forward flight<br>• Still in development | • Drone Delivery |

*Table 2. 1 Compare all types of Drones*

## 2.2. Drone Structure

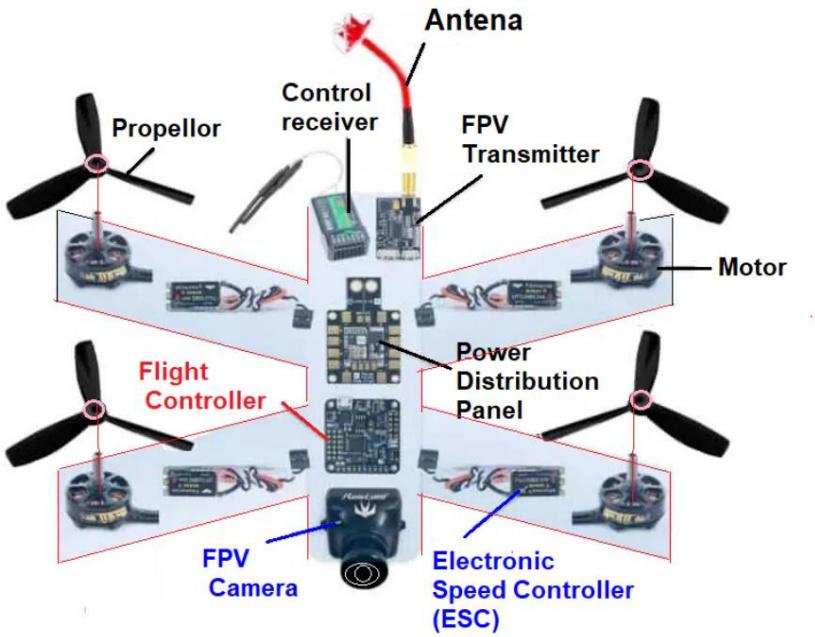



*Figure 2. 5 All components in Drone*

A drone's structure can be broken down into these key parts:

- **Frame:** Lightweight and strong skeleton, holding everything together.
- **Motors:** Electric motors that spin the propellers, providing thrust.
- **Propellers:** Rotating blades that generate lift for flight.
- **Battery:** Powers the entire drone, with high capacity for flight time.
- **Flight Controller:** The "brain" that interprets pilot input, processes sensor data, and controls motors for flight stability.

### 2.2.1. Frame

**H-frame quadcopter**: *[18]*As the name suggests, the H frame has an extended arm form from the base forming an H shape. The H frame was the first and most popular non-pilot airframe. Sturdy frame, can carry many FPV devices and LiPo batteries. The frame is wide and stable when rolled. The frame is good for FPV racing beginners because the battery lasts longer and there is more space for building and modifications**.**

*Figure 2. 6 H-Frame*

**X-frame Quadcopter**: The X-frame has a shape similar to the letter X. The strength of the Versatility and predictability in handling make the X-frame a top choice for FPV racing.

*Figure 2. 7 X-Frame*

**Folding Quadcopter**: A folding drone is a type of drone that can be folded for easy carrying and storage. They are often made of lightweight and durable materials, such as carbon fiber or synthetic resin. They are compact and easy to carry, making them ideal for travel or other activities where you need to carry a drone.

*Figure 2. 8 Folding Quadcopter*

Since the frame is the backbone of a drone, supporting all its parts and facing the most stress during flight, it needs to be light, strong, and easy to work with. Prioritizing these requirements, we opted for the Tarot Iron Man 650 Foldable Quadcopter Frame. It meets our needs by being:

- The material is made from carbon fiber, it weighs an impressive 476 grams, minimizing weight for optimal flight performance.
- The frame is CNC for high precision and a high level of accomplishment.
- Allowing for the easy installation of additional payloads, making it a versatile choice for the needs.

*Figure 2. 9 Iron Tarot 650*

### 2.2.2. Dynamic systems

#### *2.2.2.1 Motors*

Brushless DC motors have become a mainstay in applications demanding precise control, exceptional longevity, and high efficiency. Unlike brushed DC motors that use physical brushes for commutation, brushless motors rely on electronic controls and strategically place in magnetic fields for rotor rotation. This translates to several advantages:

- Enhanced Efficiency: The absence of physical brushes eliminates friction, a major source of energy loss in brushed motors. This translates to better efficiency and cooler operation in brushless motors.

- Superior Control**:** Electronic controls enable precise adjustments to the speed and direction of the rotor. This level of control allows for smooth operation, optimized performance, and dynamic braking capabilities.

- Increased Durability**:** Brushless motors eliminate the wear and tear associated with physical brushes. This results in a longer lifespan and reduced maintenance needs compared to brushed counterparts.

- Spark-Free Operation**:** Without brushes, brushless motors eliminate sparking, making them ideal for applications in flammable or explosive environments.

**Selecting the optimal drone motor hinges on several factors:**

- Drone Weight: This is the foundation. Motors need enough thrust to lift the entire drone's weight.

- Motor Thrust: This is the most critical factor. The combined thrust of all motors should ideally be at least double the drone's weight (2:1 ratio) for stable flight. More aggressive maneuvers might require higher ratios (3:1 or 4:1).

- Motor Size and KV Rating: These influence the motor's ability to handle propellers and generate thrust. Generally, a higher KV rating indicates potentially higher speeds but might require more current.

- Current and Voltage Levels: Compatibility with your battery and Electronic Speed Controllers (ESCs) is crucial, as these determine the power delivered to the motors.

  Each propeller will produce a different thrust performance, it is necessary to ensure that the motor is strong enough to make the propeller rotate, at this point we need to pay attention to torque.

  Thus, with the requirement to lift a mass of 4 kg, we decided to choose the Sunsky x2814-7 1100 KV engine, with the thrust of each engine about 2360 g, so the ratio is about 2.36 :1 suitable with requirements ranging from 2:1 to 4:1.


*Figure 2. 10 Motor SunSky x2814-7 1100KV*

*[32]***Specifications of motors**

| | |
|---|---|
| Rotor Diameter | 35 mm |
| Shaft Thickness | 4 mm |
| No. of Stator Arms | 12 |
| No. of Rotor Poles | 14 |
| Motor KV | 1100 |

| | |
|---|---|
| No-Load Current (A/10V) | 1.5 A |
| Motor Resistance | 32 mQ |
| Max Continuous Current | 40A/30S |
| Max Continuous Power | 600 W |
| Weight | 108 g |
| Overall Shaft Length | 54 mm |
| Body Length | 36 mm |
| Max Lipo Cell | 3-4S |
| Recommend ESC | 50A |

*Table 2. 2 Specifications of motor*

### *2.2.2.2. Propeller*

| Propeller Type | Voltage | Current | Thrust(G) | Power (W) |
|---|---|---|---|---|
| EMP9X6 | 10V | 22.9A | 1.25kg | 229 |
| | 11V | 26.3A | 1.4kg | 289.3 |
| | 12V | 29.4A | 1.62kg | 352.8 |
| | 13V | 32.6A | 1.8kg | 423.8 |
| | 14V | 36.4A | 2.0kg | 509.6 |
| | 14.8V | 38.9A | 2.1kg | 575.72 |

*Table 2. 3 Parameters for EMP9X6 Propeller*

Based on parameters of motors, EMP9X6 propeller as the most suitable for the drone's specifications. However, this type is quite difficult to find in Vietnam, so we have decided to replace it with the APC 1045."

*Figure 2. 11 Propellers APC 1045*

| Specifications | Values |
|---|---|
| Pitch | 114.3 mm |
| Propeller Diameter | 254 mm |
| Hub Diameter | 16.51 mm |
| Hub Thickness | 8.9 mm |
| Shaft diameter | 6.35 mm |
| Weight | 15 g |
| Color | Gray |

*Table 2. 4 Specifications of Propellers*

### 2.2.3 Flight control system

*Figure 2. 12 Pixhawk 6C*

The flight controller (FC) is a crucial component of a UAV, responsible for controlling the motors and propellers to maintain stability and achieve desired flight maneuvers. The FC receives data from various sensors, including inertial sensors, barometric pressure sensors, and magnetometers, to determine the UAV's position, speed, and orientation. Based on this information, the FC employs control algorithms to precisely adjust the speed and direction of the motors, ensuring the UAV remains stable and follows the intended flight path.

**Sensors**

The flight controller is connected to an array of sensors that provide it with essential information about the UAV's state, such as altitude, orientation, and speed. Common sensors include:

- **Inertial Measurement Unit (IMU)**: Measures angular rates and accelerations to determine the UAV's orientation and motion.
- **Barometer**: Measures atmospheric pressure to estimate altitude.
- **Magenetometer**: Determine the drone’s orientation by the compasss.

**Control**

UAVs achieve maneuverability by generating differences in speed between their four motors. The flight controller utilizes sensor data to calculate the desired speed for each motor. This information is then transmitted to the Electronic Speed Controllers (ESCs), which convert the desired speeds into signals understood by the motors.

The computation of motion, sensor data fusion and filtering, as well as flight safety and stability estimation are all handled by algorithms. The most prevalent flight control algorithm is called PID control, which stands for Proportional-Integral-Derivative control.

**Communication**

Effective communication is an integral aspect of the flight controller's functionality. Part of its role is to present sensor data in a clear and understandable format for the pilot. One crucial piece of information is battery level, which can influence the pilot's decision to continue flying or return for charging. Currently, communication is primarily achieved through Wi-Fi and radio frequencies, with cellular solutions also gaining traction.

The Pixhawk 6C *[26]* is the latest open-source autopilot designed specifically for drones, building upon prior Pixhawk versions with improved performance and features. It's a collaborative effort between Holybro's hardware expertise and the PX4 open-source software development team. This fusion ensures both powerful hardware and software compatibility for your next drone project.

**Core Specifications:**

- **Processor:** High-performance STM32H7 microprocessor for efficient handling of complex flight control algorithms.
- **Form Factor:** Available in two options:
    - Standard 6C: Full-featured with various connectivity options.

- 6C Mini: Compact version with built-in motor/servo header, ideal for space-constrained builds (with slightly less connectivity).

- **Software Compatibility:** PX4 Autopilot and ArduPilot (open-source flight control software platforms).

**Some outstanding features:**

- **Redundant Inertial Measurement Unit (IMU):** This critical safety feature provides a backup IMU in case the primary unit malfunctions. This redundancy mitigates the risk of flight control issues and potential crashes caused by sensor failure.
- **IMU Temperature Control:** The Pixhawk 6C actively regulates the temperature of the IMU sensors which contributing to smoother and more precise flight performance.
- **High-Performance Processor:** The 6C is equipped with a powerful STM32H7 microprocessor, enabling it to handle complex flight control algorithms with greater efficiency.
- **Open-Source Hardware Design:** The Pixhawk 6C adheres to the open-source hardware philosophy, providing a level of transparency that fosters innovation within the drone development community.
- **Open-Source Software Compatibility:** The Pixhawk 6C seamlessly integrates with popular open-source flight control software platforms like PX4 and ArduPilot. This compatibility grants users access to a vast array of software functionalities that provide support and ongoing development resources.

### 2.2.4. Flight control software

#### *2.2.4.1. PX4 Firmware*


*Figure 2. 13 PX4 Firmware*

PX4 Firmware *[23]* is an open-source flight control software for drones and other unmanned vehicles. It provides a flexible toolkit for UAV developers to share technology to create solutions for drone applications. PX4 provides a standard for supporting hardware and related software for drones, enabling the ecosystem to build and maintain hardware and software in a scalable way. Beside that:

- **Open and Flexible Architecture:** PX4 is designed with an open architecture, allowing users to customize and extend functionality to meet their needs. It uses the C/C++ programming language and can be integrated with a variety of sensors and controllers.
- **Support for Multiple Types of Aircraft:** PX4 supports not only quadcopters but also a variety of other aircraft such as fixed-wing aircraft, VTOL (vehicles capable of switching between vertical and horizontal movement), and other multi-rotor models.

- **PX4 Community:** The PX4 community is large and diverse, including developers, aerospace engineers, researchers, and end users. Community support and contributions play a vital role in maintaining and developing the system.

### *2.2.4.2. ROS*


*Figure 2. 14 Robot Operating System*

ROS*[31]*which stands for Robot Operating System, is actually not a true operating system like Windows or macOS. It's more like a **framework** that provides a bunch of tools and libraries specifically designed for robot software development.

- **Open-source:** Anyone can use and contribute to ROS, making it a collaborative and constantly evolving platform.
- **Modular design:** ROS breaks down complex robot tasks into smaller, manageable pieces of code. These pieces are called nodes, and they communicate with each other to achieve the overall goal.
- **Code reusability:** Because ROS is modular, developers can easily reuse existing code for different robot applications. This saves time and effort.
- **Tools and libraries:** ROS provides a variety of tools and libraries that handle common robotics tasks like device drivers, sensor data processing, message-passing, and more.

### *2.2.4.3. MavROS*

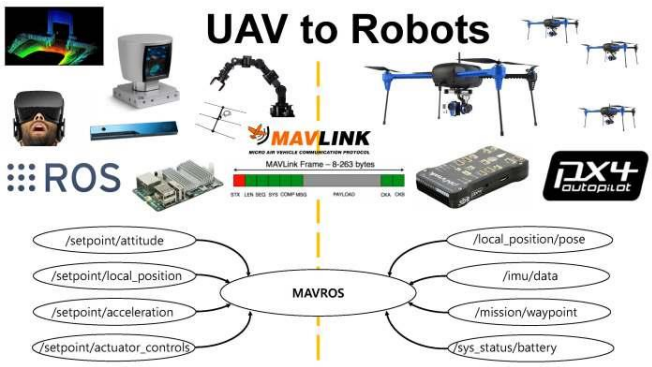



*Figure 2. 15 MavROS*

MavROS is a software package that bridges the gap between ROS and autopilots using the MAVLink communication protocol. It essentially allows you to control unmanned aerial vehicles (UAVs) through ROS.

- **MAVLink communication:** MavROS translates ROS messages into MAVLink messages, which is a widely used communication protocol for drones and autopilots. This enables ROS applications to send commands and receive data from the UAV.

- **ROS integration:** MavROS provides a set of ROS nodes that handle tasks like sending control commands to the autopilot,

  receiving sensor data from the UAV, and managing the overall communication between ROS and the autopilot.

#### *2.2.4.4. Gazebo*


*Figure 2. 16 Gazebo*

Gazebo*[17]*is a free and open-source 2D/3D robot simulator widely used in robotics research, development, and education. It provides a powerful platform to create realistic simulations of robots interacting with their environments.

- **Realistic Physics and Dynamics:** Gazebo simulates physics accurately, including factors like gravity, friction, and inertia. This allows developers to test robot movement, stability, and performance in a controlled virtual environment before deploying them in the real world.
- **Sensor Simulation:** Gazebo can simulate various types of sensors commonly used in robots, such as cameras, LiDAR, and IMUs (Inertial Measurement Units). This allows developers to test robot perception and how the robot interacts with its surroundings based on sensor data.
- **Extensive Robot Libraries:** Gazebo comes with a rich library of pre-built robot models, encompassing everything from simple mobile robots to complex industrial manipulators. Users can also create custom robot models using plugins or import models from CAD software.
- **Multiple Environments:** Gazebo allows creation of diverse simulated environments, indoors, outdoors, on uneven terrain, or even underwater. This versatility helps developers test robots in various scenarios they might encounter.

- **ROS Integration:** Gazebo integrates seamlessly with the Robot Operating System (ROS), a popular framework for robot software

development. This allows developers to leverage existing ROS tools and libraries within their simulations.

#### *2.2.4.5. QGroundControl*

*Figure 2. 17 QGroundControl*

**QGroundControl***[28](***QGC):** An open-source ground control station (GCS) for drones and UAVs. It offers a user-friendly interface for hobbyists and professionals alike.

- **Flight Control at Your Fingertips:** Take full command of your drone, including initiating takeoff, landing, and waypoint navigation. QGC allows you to control your camera and other onboard sensors for real-time adjustments.
- **Mission Planning Made Easy:** Unleash the power of QGC's mission planner. Create intricate missions with designated waypoints, specific altitudes, desired speeds, and more. Schedule camera captures or sensor data collection at designated points for a truly automated experience.
- **Real-Time Knowledge is Power:** Gain valuable insights into your drone's health through real-time telemetry data. Monitor critical aspects like altitude, speed, battery level, and GPS position to ensure safe and efficient flight operations.
- **Unlock Sensor Potential:** QGC doesn't stop at basic telemetry. It displays real-time sensor data, allowing you to view camera images,

accelerometer readings, and other sensor information crucial for understanding your drone's environment and performance.

- **Customization is Key:** QGC caters to individual needs. A wide range of plugins exist to extend its functionality. Explore plugins for tasks like creating detailed maps, detecting objects autonomously, or enabling fully autonomous flight for your drone.
- **Embrace the Open-Source Community:** As a free and open-source project, QGC is constantly evolving. The community actively contributes to its development, ensuring a steady stream of new features and improvements.

### 2.2.5. Power System

#### *2.2.5.1. Battery Lipo*


*Figure 2. 18 Battery Lipo GN3 4 cells*

When choosing a LiPo battery for radio-controlled (RC) applications, particularly drones and multirotors, two crucial factors must be considered: discharge current and capacity.

The battery's continuous discharge rate indicates the maximum current it can deliver consistently without sustaining damage. In this scenario, with four motors each requiring 60 Amps, the total current draw is 4 motors * 60 Amps/motor = 240 Amps. Therefore, the LiPo battery's continuous discharge rating must exceed 240 Amps to ensure safe and reliable operation.

The GNB 5500mAh 4S 70C LiPo*[29]*battery emerges as a strong candidate based on the established criteria. With a 4S configuration

(indicating four cells), it delivers a capacity of 5500mAh, providing ample flight time for your drone.

**Discharge Current:** The battery boasts a remarkable 70C discharge rate. This translates to a maximum continuous discharge current of 70C * 5500mAh = 385,000mAh, or 385 Amps. This value comfortably surpasses the combined current requirement of the four motors (240 Amps), ensuring sufficient power for demanding maneuvers.

The GNB 5500mAh 4S 70C LiPo battery offers an ideal combination of high capacity and exceptional discharge current, making it a well-suited choice for powering RC applications with multiple motors, particularly when extended flight times and robust performance are desired

| Specifications | Values |
|---|---|
| Nominal Capacity | 5500mAh |
| Dimensions | 13.9x4.4x3.6cm (DxWxH) |
| Nominal Voltage | 14.8V |
| Discharge Rate | 70C/140C |
| Connector Type | XT60 |
| Weight | 450 grams |

\

*Table 2. 5 Specifications of Battery Lipo*

### *2.2.5.2. Power divider*

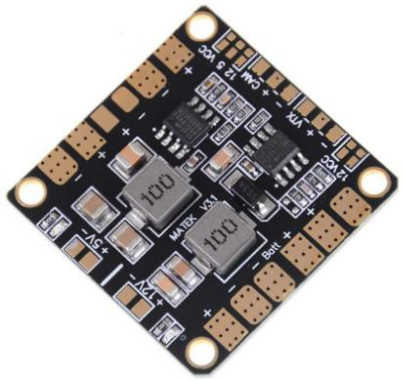


*Figure 2. 19 Power Distribution Board*

Power Distribution Board (PDB) with BEC (Battery Eliminator Circuit) is an essential component for multirotor drones, particularly FPV (First Person View) racing drones like the QAV250 and ZMR250. It serves as a central hub for distributing power from the battery to various electronic components, including the flight controller, motors, servos, and FPV equipment. Additionally, the BEC provides regulated voltage options (5V and 12V) for powering these components.

- **Power Distribution:** Efficiently distributes power from the battery to various components, reducing clutter and improving wiring organization.

- **Voltage Regulation:** Provides regulated 5V and 12V outputs, ensuring stable power supply for sensitive electronics.

- **Overcurrent Protection:** Safeguards against electrical overload by protecting components from excessive current draw.

- **Compact Design:** Integrates multiple functions into a compact PCB, saving space and weight.

### *2.2.5.3. ESC*


*Figure 2. 20 ESC*

The Hobbywing Skywalker 60A ESC (Electronic Speed Controller) *[20]* is brushless ESC designed for use in electric radio control (RC) airplanes and other electric RC vehicles.

- **Continuous Current:** 60A - This is the maximum amount of current the ESC can handle continuously without overheating.

- **Burst Current:** 80A (for up to 10 seconds) - The ESC can handle short bursts of higher current for things like quick acceleration.

- **Input Voltage:** 2-6S LiPo or 5-18S NiMH/NiCd - This is the range of battery voltages the ESC can handle. LiPo batteries are Lithium Polymer batteries, while NiMH/NiCd are Nickel Metal Hydride and Nickel Cadmium batteries, respectively. "S" refers to the number of cells in the battery pack.

- **BEC:** 5V @ 5A - The ESC has a built-in BEC (Battery Eliminator Circuit) that can provide power to your receiver and servos. The 5V refers to the voltage output, and the 5A refers to the maximum current output.

Parameters for ESC:

| Specifications | Values |
|---|---|
| Continuous Current | 60A |
| Burst Current | 80A |
| Input Voltage | 2-6S LiPo or 5-18S NiMH/NiCd |
| Size | 73*30*12 mm |
| Weight | 68 gram |
| Input Wires | Red-14AWG-100mm*1; Black-14AWG-100mm*1 |
| Output Wires | Black-14AWG-100mm*3 |
| Output connectors | 4.0 gold connectors (female) |

*Table 2. 6 Specifications of ESC*

### 2.2.6. GPS (M10)


*Figure 2. 21 GPS M10 Holybro*

The Holybro M10 GPS*[20]*is a compact and versatile positioning module designed to elevate the performance of your drone. Leveraging the advanced u-blox M10 series chip, it delivers superior accuracy and reliable signal acquisition even in challenging environments.

The M10 GPS uses multi-constellation GNSS powered by u-blox M10 series, a concurrent GNSS receiver which can receive and track multiple GNSS systems. Owing to the multi-band RF front-end architecture all four major GNSS constellations, GPS, Galileo, GLONASS and BeiDou can be received concurrently.

### 2.2.7. JetsonNano

` *Figure 2. 22 JetsonNano Developer Kit*

The NVIDIA Jetson Nano*[22]* is a compact, high-performance embedded system specifically designed for developers integrating artificial intelligence (AI) at the edge. This single-board computer offers exceptional processing capabilities within its small form factor, enabling the execution of complex machine learning models for real-time applications. Its suitability extends to a variety of industrial settings, including robotics, intelligent camera systems, and automated manufacturing processes.

## 2.3. Transmitting data and images on Drone

### 2.3.1. Camera Adrucam IMX519


*Figure 2. 23 Camera Adrucam IMX519*

The Sony IMX519*[14]* is a popular 16-megapixel image sensor widely used in various camera modules for applications like drones (UAVs), robotics, security systems, and machine vision. Some specifications:

- **Resolution:** 16 megapixels (4656 x 3496)
- **Pixel Size:** 1.22 µm x 1.22 µm
- **Sensor Format:** 1/2.5 inchs optical format
- **Image Sensor Type:** Stacked CMOS (Complementary Metal-Oxide-Semiconductor) sensor

  With many advancements:

  - Interface: It connects and communicates with a computer or microcontroller through digital interfaces such as MIPI CSI-2, USB, or UART.
  - Applications: It is suitable for applications like cameras, security cameras, robot cameras, surveillance cameras, etc.
  - Integration: It is pre-integrated on evaluation boards like ArduCAM, Raspberry Pi, etc. to enable easy integration with embedded systems.

### 2.3.2. Video Signal

*Figure 2. 24 GStreamer*

GStreamer*[19]* is a powerful multimedia framework that facilitates the creation of applications involving media processing, streaming, editing, and playback. It offers a modular and flexible approach, enabling developers to build complex media pipelines by combining various processing elements called plugins.

- **Pipeline Architecture:** GStreamer's pipeline architecture allows for a structured and flexible approach to media processing. Each element in the pipeline performs a specific task, and the elements are connected to form a dataflow graph which creation of complex workflows by combining and chaining elements.
- **Plugin Ecosystem:** GStreamer boasts a vast ecosystem of plugins, providing a wide range of functionalities for media processing. These plugins are categorized into various groups based on their purpose, such as video codecs, audio codecs, filters, and sinks
- **Cross-Platform Support:** GStreamer is a cross-platform framework including Linux, Windows, macOS, and Android. This portability allows developers to build applications that can run on various platforms without significant modifications.
- **Performance Optimization:** GStreamer is designed for performance and efficiency, making it suitable for real-time media processing applications. It utilizes techniques like multithreading, hardware acceleration, and efficient memory management to ensure smooth and responsive media playback and processing.

## 2.4 The Flight Controller

### 2.4.1 Transmitter (Controller)

*Figure 2. 25 RadioMaster Pocket*

**The RadioMaster Pocket***[30]* **is a compact and powerful radio controller.** It comes in two flavors: ExpressLRS and MPM CC2500. Both versions run EdgeTX software for easy customization. The detachable sticks and folding antenna make it highly portable. Plus, the long-lasting battery keeps you flying for hours. The reliable 2.4 GHz operating frequency ensures accurate control. Status LEDs and a backlit LCD display provide clear information without sacrificing size. It's the perfect balance of portability and functionality for remote control enthusiasts

- **Basic Control:**
  - **Flight Control:** The primary function is to provide precise and reliable control of your drone or RC model through its sticks and switches.
  - **Channel Outputs:** It transmits control signals (typically PWM or SBUS) to your model's flight controller, allowing you to maneuver it as desired.
- **Programming and Customization:**
  - **EdgeTX Software:** Pre-installed EdgeTX software offers extensive customization options. You can program flight modes, adjust control

throws and expo, configure mixes, and fine-tune various settings for your specific model.

- **Model Profiles:** Create and store multiple profiles for different models, each with its own configuration settings for easy switching between them.

- **Communication and Connectivity:**

  - **Internal RF Module Options:** Choose between the ExpressLRS or MPM CC2500 multi-protocol internal radio module depending on your needs and receiver compatibility.

  - **Supported Protocols (Dependent on Module):** Depending on the chosen module, it can support various communication protocols like FrSky, Spektrum, Crossfire, and of course, ExpressLRS for long-range and low-latency control.

| Specifications | Value |
|---|---|
| Dimensions | 156.665.1125.3mm(Folded), 156.673.1154.8mm (Unfolded) |
| Weights | 288 grams |
| Operating frequency | 2.4GHz - 2.48GHz |
| Internal RF options | Multi-protocol CC2500/ELRS 2.4GHz |
| Operating voltage | 6.6V - 8.4V DC |
| Operating system | EdgeTX |
| Control Channels | Maximum 16 |

*Table 2. 7 Specifications of RadioMaster Pocket*

### 2.4.2 Receiver


*Figure 2. 26 BETAFPV ELRS 2.4G*

The BETAFPV ELRS 2.4G Nano Receiver*[15]* is a lightweight and miniaturized receiver designed for use with drones and other radio-controlled models. It is based on the ExpressLRS open-source project, which aims to achieve the best possible link performance in terms of speed, latency, and range.

- Small and lightweight: The receiver weighs only 0.7 grams, making it ideal for use in small drones where weight is a critical factor.
- Long range: The receiver can provide a long range connection, even in challenging environments.
- Low latency: The receiver offers low latency for responsive control of your drone.
- Compatible with EdgeTX and OpenTX: The receiver is compatible with popular radio control systems such as EdgeTX and OpenTX.

# CHAPTER 3: MATHEMATIC MODEL OF QUADCOPTER

## 3.1. Diagram of quadcopter kinematic

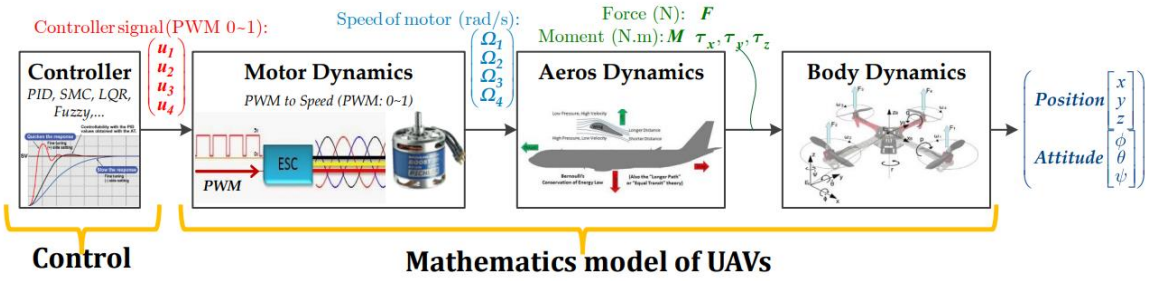



*Figure 3. 1 Diagram of quadcopter dynamic*

In short, the diagram explains the basic functions of a quadcopter. The crucial part is its dynamics, which allows the quadcopter to maintain its desired position and altitude during flight. To achieve this, there are three key components involved in the dynamic stage:

- **Motor dynamics:** This is the study and application of physical principles to understand and control how motors generate force to ensure that the drone can perform the desired movements and tasks. This includes studying thrust through motors, response time equivalent to the amount of input given, increasing motor efficiency, adjusting PID, ... Understanding these is a key factor in controlling the speed and altitude of the UAV.

- **UAV aerodynamics:** UAV aerodynamics is the study of how airflow interacts with a UAV. Aerodynamics affects many aspects of UAV performance, including lift, drag, and thrust. There are many factors that can affect UAV aerodynamics: material, size, flight speed, weather conditions, ... Modeling them can understand the airflow around the body, avoid unwanted force and moment effects on the UAV.

- **Body dynamics:** Body dynamics focuses on studying the motion and behavior of the UAV. This includes the translational and rotational motion of the UAV. Factors affecting these parameters include mass, moment of inertia, load distribution (attached components) on the UAV. Researching body dynamics helps to increase the stability, controllability and flight performance of the UAV.

### 3.1.1. Coordinate systems in Quadcopter

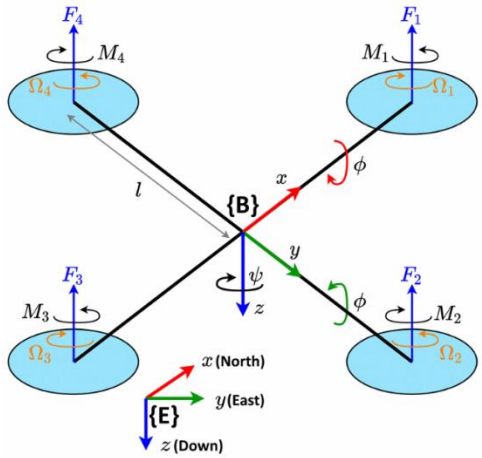



*Figure 3. 2 Coordinates on Drone*

There are 3 coordinates on UAV:

- **Earth Frame (E) or Inertial Frame (I):** This is a globally fixed reference frame, often denoted as NED (North, East, Down). It serves as a constant reference point, with the X-axis aligned with geographic North, the Y-axis aligned with geographic East, and the Z-axis pointing downwards, completing a right-handed coordinate system. The Earth frame remains stationary throughout the UAV's flight, providing a universal reference for absolute position.
- **Vehicle Frame (V):** This reference frame is body-fixed, meaning it is rigidly attached to the center of mass of the UAV. Initially, during system initialization, the Vehicle frame perfectly aligns with the Earth frame. Both frames share the same origin and utilize the same right-handed NED convention. The Vehicle frame serves as a convenient intermediate reference for relating measurements obtained on-board the UAV to the global Earth frame.
- **Body Frame (B):** The Body frame is a concept derived from the Vehicle frame. It is obtained by applying a rotational transformation to the Vehicle frame. This transformation accounts for the UAV's orientation in space relative to its initial alignment. The Body frame axes remain parallel to the Vehicle frame axes, but their orientation in the Earth frame

changes dynamically as the UAV maneuvers. The Body frame is crucial for representing the UAV's attitude (orientation) and for controlling its flight dynamics through motor inputs.

### 3.1.2. Euler Angles

Euler angles are a parametric representation used to describe the orientation of a rigid body in 3D space relative to a fixed reference frame. They achieve this by specifying a sequence of three rotations around designated axes.

**Rotation matrix:**

Considering a reference frame {V} with origin O, the initial rotation of {V} around an axis with angle ψ can be determined as follows:

$$\{V\} \to \{V_1\} : R_V^{V_1} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

With the third rotation of $\{V_2\}$ around x axis with angle $\phi$ can be calculated as:

$$\{V_2\} \to \{B\} : R_{V_2}^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

The rotation matrix from {V} to {B}, from this coordinate to other coordinate:

$$R_V^B = R_{V_2}^B R_{V_1}^{V_2} R_V^{V_1} = Rot(x,\phi).Rot(y,\theta).Rot(z,\phi)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \sin\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta \end{bmatrix}$$

Switching the coordinate *P(x,y,z)* {V}, position *(P)* in {B} is calculated:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}_B = R_V^B \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}_V$$

Or point P(x,y,z) in {V} is computed:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}_V = R_B^V \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}_B$$

Describing the transformation matrix from {B} to {V}:

$$R_V^B = (R_B^V)^{-1}$$

$$= \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta cc\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi & c\phi s\theta cc\psi - s\phi s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

Summarize:

$$R_V^B = R_{V_2}^B R_{V_1}^{V_2} R_V^{V_1} = Rot(x,\phi).Rot(y,\theta).Rot(z,\phi)$$

$$= \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta \end{bmatrix}$$

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

*The formulas in sections 3.1.2 and 3.1.1 are referenced from source [8]*

## .3.2. Body Dynamics

### 3.2.1. High altitude dynamic equation of UAV

We obtain that the angular velocity of the {B} coordinate system is equal to the sum of the speeds of all roll, pitch and yaw axes observed from the fixed coordinate system at the {E} axis.

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = w_{b/e} = w_b^{\ roll} + w_b^{\ pitch} + w_b^{\ yaw}$$

With the following parameters:

- The angular speed motioning on Roll axis:

$$w_b^{\ roll} = R_B^B \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix}$$

- The angular speed motioning on Pitch axis:

$$w_b^{\ pitch} = R_{v_2}^b \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} = R_{v_2}^b (x, \phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix}$$

- The angular speed motioning on Yaw axis:

$$\omega_b^{yaw} = \boldsymbol{R}_{v_1}^b \omega_{v_1} = \boldsymbol{R}_{v_1}^b \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}_{v_1} = \boldsymbol{R}_{v_2}^b \left(\phi\right) \boldsymbol{R}_{v_1}^{v_2} \left(\theta\right) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}_{v_1}$$

Thus:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = w_{b/e} = w_b^{\ roll} + w_b^{\ pitch} + w_b^{\ yaw} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_{v_2}^b (x, \phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_{v_2}^b (x, \phi) R_{v_1}^{v_2} (y, \theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

$$= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad [6]$$

### 3.2.2. Position dynamic equation of UAV

- In coordinate {E} of quadcopter:

$$\{E\} : \xi = (x_e, y_e, -h)$$

- Linear velocity in coordinate {B}: $(u, v, w)^T$
- Angular speed of each axis on coordinate {B}: $(p, q, r)^T$
- The relation between linear velocity in {E} and linear velocity {B}:

$$\frac{d}{dt}\begin{pmatrix} x_e \\ y_e \\ -h \end{pmatrix} = R_B^E \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\theta s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi & c\phi s\theta c\psi - s\phi s\psi \\ -s\theta & s\phi c\theta & c\theta c\phi \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

### 3.2.3. Equation of Coriolis's force

The efficiency of quadcopter is based on thrust (F) and drag moment (M)

- Thrust: $F_i$ is external force exerted by the engine. This thrust is proportional to the thrust coefficient.

$$F_i = b * \Omega_i^2 (i = 1, 2, 3, 4)$$

- Drag moment: $M_i$ is drag moment for engines. This drag moment is also proportional to the drag coefficient.

$$M_i = d * \Omega_i^2 (i = 1, 2, 3, 4)$$

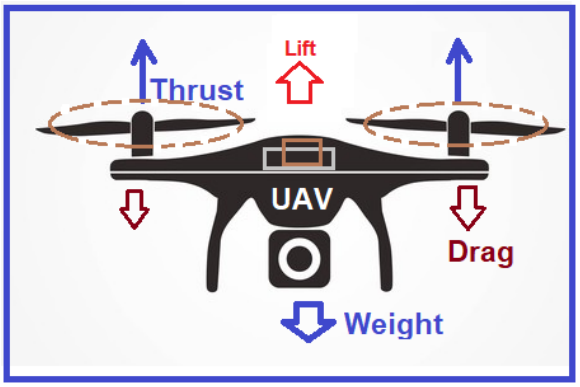



*Figure 3. 3 The Graph for lifting UAV*

- Based on Newton's second law in the inertial coordinate system, the equation for translational motion for the body frame will be:

$$\sum \vec{f}_b = m\vec{a}_b \Leftrightarrow f_b = m\left(\frac{\vec{dv}}{dt}\right)_b$$

- Because the quadcopter flying in the atmosphere will be subjected to a Coriolis force, the full equation is:

$$m\left(\frac{dv}{dt}\right)_b = m\left(\left(\frac{dv}{dt}\right)_b + w_b \times v\right)$$

- Suppose **m** is total mass of drone, $f_b$ is total force interact to body frame of drone:

$$f_b = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}_b = (f_b)_{Gravity} + (f_b)_{Aeros} + \\ = R_e^b \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}_e + \begin{pmatrix} 0 \\ 0 \\ -\sum F_i \end{pmatrix}_b$$

With: $\sum F_i = \sum b * \Omega_i^2 (i = 1,2,3,4)$

- Based on all of these equations, the final equation as follows:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = R_e^b \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ -\sum F_i \end{pmatrix} - w_b \times v$$

Or:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} -g \sin\theta \\ g \sin\phi \cos\theta \\ g \cos\theta \cos\phi \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ -\sum F_i \end{pmatrix} - \left( w_{b/e}(t) \times v \right)$$

### 3.2.4. Euler's equation

With Euler's equation of motion, consider the angular momentum for any rigid body (body frame) with the coordinate system {B}:

$$L_b = J w_b \ , \ w_b = [w_x, w_y, w_z]^T \ , \ J = diag(J_x, J_y, J_z)$$

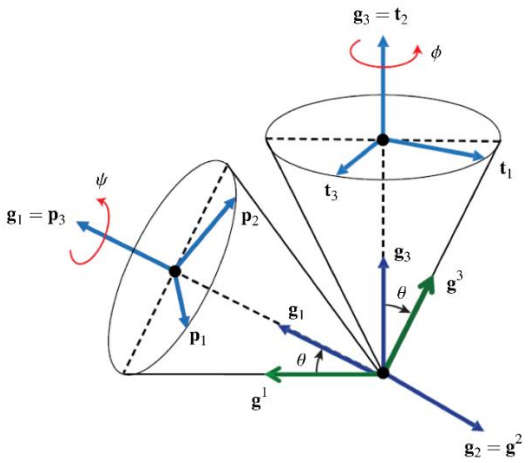



*Figure 3. 4 Coordinate in Euler's equation*

Suppose $\Gamma$ is the external moments affecting the body frame that converge at a central point, Newton's equation of torque and moment:

$$\left(\frac{dL}{dt}\right) = \Gamma\left(\Gamma = [\Gamma_x, \Gamma_y, \Gamma_z]^T : external\_moment\right)$$

In addition, due to the influence of the Coriolis force, we have the following equation:

$$\left(\frac{dL}{dt}\right) = \left(\frac{dL}{dt}\right)_b + w_b \times L_b = \dot{L}_b + w_b \times L_b$$

Finally, the Euler's equation as follow:

$$\Gamma = \dot{L}_b + w_b \times L_b = J\dot{w}_b + w_b \times L_b$$

Summarize:

$$\begin{cases} \dot{L}_B = J\dot{w}_b = \left[ J_x\dot{w}_x, J_y\dot{w}_y, J_z\dot{w}_z \right]^T \\ w_b \times L_b = w_b \times J_b w_b = \begin{vmatrix} e_1 & e_2 & e_3 \\ w_x & w_y & w_z \\ J_x w_x & J_y w_y & J_z w_z \end{vmatrix} \\ \Gamma = \left[ \Gamma_\phi, \Gamma_\theta, \Gamma_\psi \right]^T \\ J = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \\ w_b = \left[ p, q, r \right]^T \end{cases}$$

From these equations, the acceleration angular in {B} coordinate:

$$\Rightarrow \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = J\dot{w}_b + w_b \times Jw_b$$

$$\dot{w}_b = \frac{1}{J}\left( \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} - w_b \times Jw_b \right)$$

### 3.2.5. Deploy the motion kinematic model in {E} coordinate

Equation of lifting force in {B} coordinate:

$$f_b = R_e^b \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}_e + \begin{pmatrix} 0 \\ 0 \\ -\sum F_i \end{pmatrix}$$

Transformation to {E} coordinate:

$$\begin{cases} f_e = R_b^e f_b \\ \Rightarrow f_e = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}_e + R_b^e \begin{pmatrix} 0 \\ 0 \\ -\sum F_i \end{pmatrix}_b \end{cases}$$

Equation of the second laws Newton:

$$f_e = m\ddot{\xi}$$

$$\ddot{\xi} = (\ddot{x}, \ddot{y}, \ddot{z})^T$$

With $(\ddot{x}, \ddot{y}, \ddot{z})$ is the acceleration of quadcopter in Earth coordinate {E}:

From (1) and (2), deduce:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} -\dfrac{U_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)}{m} \\ -\dfrac{U_1(\cos\phi\sin\theta\cos\psi - \sin\phi\cos\psi)}{m} \\ g - \dfrac{U_1\cos\phi\cos\theta}{m} \end{pmatrix}$$

The equation of acceleration angular in {B} coordinate:

$$\begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = I\dot{w}_{b/e} + (w_b \times Iw_b) \qquad (3)$$

Additionally:

$$I = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_Z \end{pmatrix},\ w_b = \begin{pmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{pmatrix}^T \quad (4)$$

Replace (4) into (3):

$$\begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_Z \end{pmatrix} \begin{pmatrix} \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{pmatrix}^T + \left( \begin{pmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{pmatrix}^T \times^T \left( \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_Z \end{pmatrix} \begin{pmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{pmatrix}^T \right) \right)$$

$$\Rightarrow \begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \dfrac{\tau_\phi}{I_x} \\ \dfrac{\tau_\theta}{I_y} \\ \dfrac{\tau_\psi}{I_Z} \end{pmatrix} + \begin{pmatrix} \dfrac{\dot{\theta}\dot{\psi}(I_y - I_Z)}{I_x} \\ \dfrac{\dot{\phi}\dot{\psi}(I_z - I_x)}{I_y} \\ \dfrac{\dot{\phi}\dot{\theta}(I_x - I_y)}{I_z} \end{pmatrix}$$

To sum up, the model of equation mathematics for designing the controller:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} -\frac{U_1(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)}{m} \\ -\frac{U_1(\cos\phi\sin\theta\cos\psi-\sin\phi\cos\psi)}{m} \\ g-\frac{U_1\cos\phi\cos\theta}{m} \\ \frac{\dot{\theta}\dot{\psi}(I_y-I_Z)}{I_x} \\ \frac{\dot{\phi}\dot{\psi}(I_z-I_x)}{I_y} \\ \frac{\dot{\phi}\dot{\theta}(I_x-I_y)}{I_z} \end{pmatrix}$$

*The formulas in sections 3.2.4 and 3.2.5 are taken from source [7].*

## 3.3. Aerodynamics

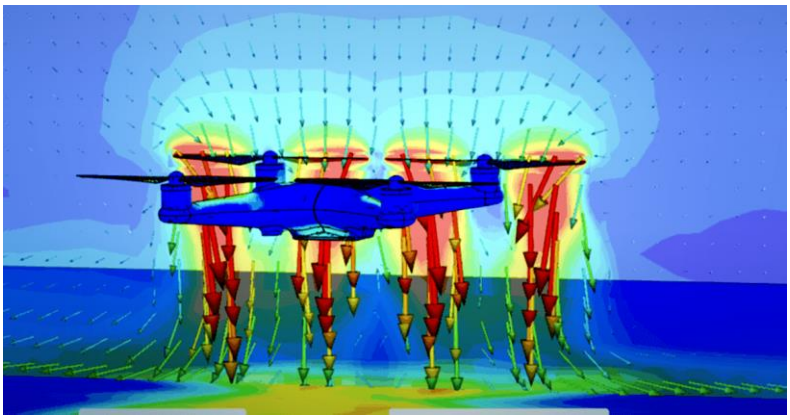


*Figure 3. 5 Computational Fluid Dynamics*

*[6]*Consider two factors related to aerodynamics: force and moment:

- For force (N):

$$f_{b_{aeero}} = \begin{bmatrix} 0 \\ 0 \\ -\sum_{i=1}^{4} F_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\sum_{i=1}^{4} b\Omega_i^{\,2} \end{bmatrix}$$

- For moments (N):

$$\Gamma = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} lF_4 - lF_2 \\ lF_1 - lF_3 \\ -M_1 + M_2 - M_3 + M_4 \end{pmatrix} = \begin{pmatrix} lb(\Omega_4^{\,2} - \Omega_2^{\,2}) \\ lb(\Omega_1^{\,2} - \Omega_3^{\,2}) \\ ld(-\Omega_1^{\,2} + \Omega_2^{\,2} - \Omega_3^{\,2} + \Omega_4^{\,2}) \end{pmatrix}$$

$\sum_{i=1}^{4} F_i = \sum_{i=1}^{4} b\Omega_i^{\,2}$ is lifting force of motors with thrust coefficient ***b***;

$\sum_{i=1}^{4} M_i = \sum_{i=1}^{4} (-1)^i d\Omega_i^{\,2}$ drag moments of motors with drag coefficient ***d***;

## 3.4. Dynamics of motors

The linear relation among speed, $K_v$ and voltage is illustrated to this equation:

$$w = K_v * V$$

With:

- W(rad/s): angular speed of n=motors
- $K_v$(rad/ (s.V) ): coefficient acceleration of motor per 1 Volt
- V: voltage of motor

### 3.4.1. Electrical operating model:

Using Kirchoff's law:

$$v_a - v_R - v_L - v_{emf} = 0$$

$$\Leftrightarrow v_a - i_a R - L_a \frac{di_a}{dt} - v_{emf} = 0 \qquad (1)$$

The electrodynamic force behind the emf is generated when the motor rotates:

$$e = v_{emf} = K_e \dot{\theta}_m$$

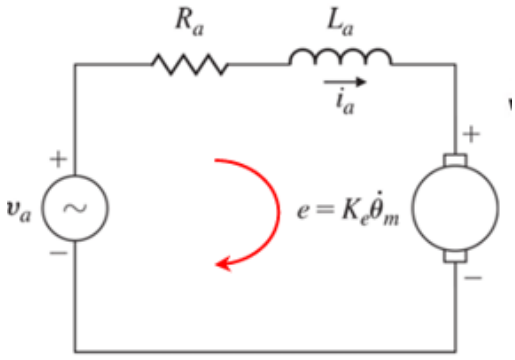



*Figure 3. 6 Circuit using Kirchoff's law*

### 3.4.2. The mechanical operating model

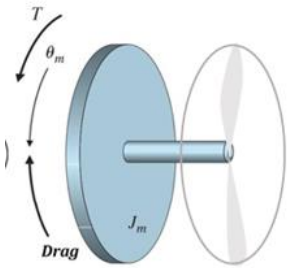



*Figure 3. 7 Mechanical operation model*

Using Newton's law:

$$\begin{cases} T_m - Drag = J_m\ddot{\theta} \\ Drag = d\Omega^2 \end{cases}$$

With:

- $J_m$: moment of inertia of motor
- $T_m$: Torque when rotating through an angle
- d: Damping coefficient
- $R_a$: motor resistance
- $L_a$: Inductance of the coil
- $V_{emf}$: Back electromotive force (emf)
- $I_a$: Current (A)

### 3.4.3. The relation between $K_t$ and $K_e$

*[1]*Considering the power electrical put into the motor:

$$W_{in} = v_a i_a = K_e w_m + i_a{}^2 R_a$$

For the power of mechanism:

$$W_{me} = T_m w_m = K_t i_a w_m$$

Ignoring the waste power, thus: $K_e = K_t$

From the equation of electrical model:

$$\begin{aligned} & v_a - v_R - v_L - v_{emf} = 0 \\ & \Leftrightarrow v_a - i_a R - L_a \frac{di_a}{dt} - K_e w = 0 \qquad (2) \\ & \Leftrightarrow i_a = \frac{(v_a - K_e w)}{R_a} \left( L_a : is\_ignored \right) \end{aligned}$$

Again:

$$\begin{cases} T_m = K_t i_a \\ Drag = dw^2 \end{cases} \qquad (3)$$

Replace (3), (2) into (1), the last equation is:

$$K_t \left( \frac{(v_a - K_e w)}{R_a} \right) - dw^2 = J_m \ddot{\theta} = J_m \dot{w} \quad (4)$$

Assume that speed on motors include stable state and instantaneous state:

$$\begin{aligned} w &= w_0 + \Delta w \\ v_a &= v_{a0} + \Delta v \end{aligned} \quad (5)$$

Put (5) into (4):

$$\frac{K_t (v_{a0} + \Delta v)}{R_a} - \frac{K_t K_e (w_0 + \Delta w)}{R_a} - d(w_0 + \Delta w)^2 = J_m (\dot{w}_0 + \Delta \dot{w})$$

Expand above equation:

$$\frac{K_t}{R_a}v_{a0} + \frac{K_t}{R_a}\Delta v_a - \frac{K_t K_e}{R_a}w_0 - \frac{K_t K_e}{R_a}\Delta w - dw_0{}^2 - 2d\Delta w w_0 - d\Delta w^2 = J_m\dot{w}_0 + J_m\Delta\dot{w}$$

At stable time with $v_{a0}$, $w_0$, thus: $K_t\left(\frac{v_{a0}}{R_a}\right) - \frac{K_t K_e w_0}{R_a} - dw_0{}^2 = 0$

Cause $\Delta w^2 \approx 0$ so the equation is encapsulated:

$$J_m\Delta\dot{w} - \frac{K_t}{R_a}\Delta v_a + \left(\frac{K_t K_e}{R_a} + 2dw_0\right)\Delta w = 0$$

Divided two sides for $\frac{K_t K_e}{R_a} + 2dw_0$:

$$\frac{J_m R_a}{K_t K_e + 2dw_0 R_a}\Delta\dot{w} + \Delta w = \frac{K_t}{K_t K_e + 2dw_0 R_a}\Delta v$$

Set $\tau = \frac{J_m R_a}{K_t K_e + 2dw_0 R_a}; K_S = \frac{K_t}{K_t K_e + 2dw_0 R_a}$ and Laplace two sides:

$$\tau s.\Delta W(s) + \Delta W(s) = K_s\Delta V(s)$$

$$\Leftrightarrow G(s) = \frac{\Delta V(s)}{\Delta W(s)} = \frac{K_s}{(\tau s + 1)}$$

# CHAPTER 4: CONTROLLER DESIGN

## 4.1. The relation between PWM and thrust

The force, F[N], generated by the motor and fan is related to the input PWM signal, u[PWM], through a first-order transfer function. This relationship can be mathematically expressed using the following equation *[11]:*

$$F(s)[N] = K_{th}\left(\frac{1}{\tau s+1}\right)u(s)$$

Inverese Laplace transform with $u(s)=\frac{1}{s}$:

$$F(t)[N] = L^{-1}\left\{\frac{K_{th}}{s} * \left(\frac{1}{\tau s+1}\right)u(t)\right\}$$

$$\Rightarrow F(t)[N] = K_{th}\left(1-e^{\frac{-t}{T}}\right) * u(t)$$

$$\Rightarrow F(t)[N] = K_{th}u(t), t \to \infty$$

With:

- $K_{th}$: positive gain
- $\tau[s]$: time constant

When: $\lambda = \frac{\Omega_0^{\,2}}{u_0}$, $u_0$ and $\Omega_0$ at stable position => $K_{th} = b\lambda$

$$\Rightarrow F(t)[N] = b\frac{\Omega_0^{\,2}}{\Omega^2}u(t)$$

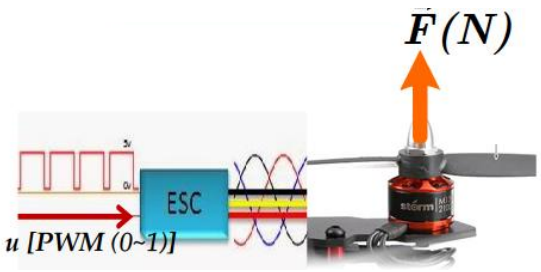



*Figure 4. 1 Relation between PWM and F(N)*

## 4.2. The relation between PWM and moment

The moment, M[Nm], generated by the motor and fan is related to the input PWM signal, u[PWM], through a first-order transfer function. This relationship can be mathematically expressed using the following equation:

$$M(s)[N] = K_d\left(\frac{1}{\tau s+1}\right)u(s)$$

Inverse Laplace transform with $u(s)=\frac{1}{s}$:

$$M[Nm] = L^{-1}\left\{\frac{K_d}{s}*\left(\frac{1}{\tau s+1}\right)u(t)\right\}$$
$$\Rightarrow M[Nm] = K_d\left(1-e^{\frac{-t}{T}}\right)*u(t)$$
$$\Rightarrow M[Nm] = K_d u(t), t\to\infty$$

With:

- $K_{th}$: positive gain
- $\tau[s]$: time constant

When: $\lambda=\frac{{\Omega_0}^2}{u_0}$, $u_0$ and $\Omega_0$ at stable position => $K_{th}=b\lambda$

$$\Rightarrow M[N] = d\frac{{\Omega_0}^2}{\Omega^2}u(t)$$

The formulas in sections [4.1] and [4.2] are taken from source [10]

## 4.3. Distribution control

Set $u_1$, $u_2$, $u_3$ và $u_4$ are the control signals for motor i, where i = 1, 2, 3, 4 *[12]*

Assume $\delta_{roll}$ is control signal for Roll axis:

$$\begin{cases} u_4 = u_0 + \delta_{roll} \\ u_2 = u_0 - \delta_{roll} \end{cases} \Rightarrow u_4 - u_2 = 2\delta_{roll}$$

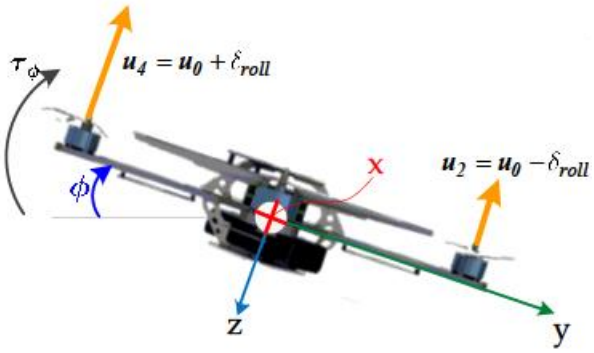



*Figure 4. 2 Drone rotates around X-axis*

Assume $\delta_{pitch}$ is control signal for Pitch axis:

$$\begin{cases} u_1 = u_0 + \delta_{pitch} \\ u_3 = u_0 - \delta_{pitch} \end{cases} \Rightarrow u_1 - u_3 = 2\delta_{pitch}$$

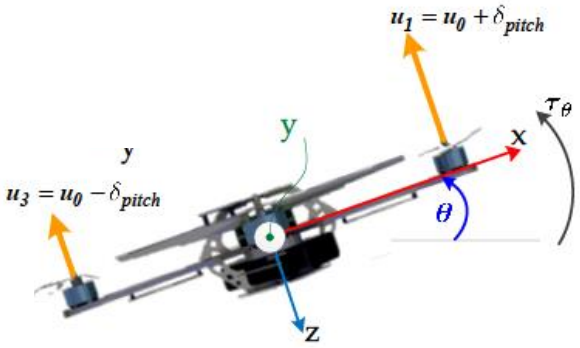



*Figure 4. 3 Drone rotates the Y-axis*

Assume $\delta_{yaw}$ is control signal for Yaw axis:

$$\begin{cases} u_1 = u_0 - \delta_{yaw} \\ u_2 = u_0 + \delta_{yaw} \\ u_3 = u_0 - \delta_{yaw} \\ u_4 = u_0 + \delta_{yaw} \end{cases} \Rightarrow -u_1 + u_2 - u_3 + u_4 = 4\delta_{yaw}$$

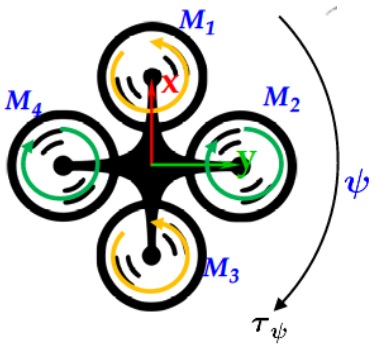



*Figure 4. 4 Drone rotates the Z-axis*

Assume $\delta_{th}$ is control signal for thrusting:

$$\begin{cases} u_1 = u_0 + \delta_{th} \\ u_2 = u_0 + \delta_{th} \\ u_3 = u_0 + \delta_{th} \\ u_4 = u_0 + \delta_{th} \end{cases} \Rightarrow u_1 + u_2 + u_3 + u_4 = 4\delta_o + 4u_0$$

Summarizing the system equations:

$$\begin{cases} u_4 - u_2 = 2\delta_{roll} \\ u_1 - u_3 = 2\delta_{pitch} \\ -u_1 + u_2 - u_3 + u_4 = 4\delta_{yaw} \\ u_1 + u_2 + u_3 + u_4 = 4\delta_o + 4u_0 \end{cases}$$

Solving each of $u_i$ values:

$$\begin{cases} u_1 = u_0 + \delta_{th} - \delta_{yaw} + \delta_{pitch} \\ u_2 = u_0 + \delta_{th} + \delta_{yaw} - \delta_{roll} \\ u_3 = u_0 + \delta_{th} - \delta_{yaw} - \delta_{pitch} \\ u_4 = u_0 + \delta_{th} + \delta_{yaw} + \delta_{roll} \end{cases}$$

Turning into the matrix:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} u_0 \\ u_0 \\ u_0 \\ u_0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta_{th} \\ \delta_{roll} \\ \delta_{pitch} \\ \delta_{yaw} \end{pmatrix}$$

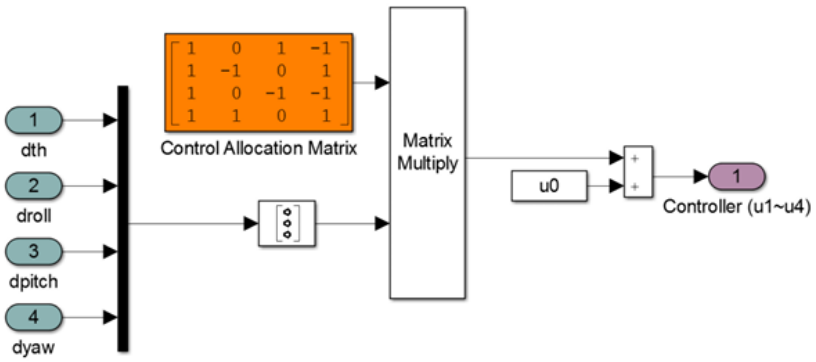



*Figure 4. 5 Simulation by Matlab for control motor signal*

## 4.4. Attitude control

### 4.4.1. Roll axis

In order to design a roll axis controller, a linear model of the quadcopter in roll motion with $\phi = 0, \dot{\phi} = 0, \dot{\theta} = 0$ *[11]*

The mathematics motion model with Roll axis:

$$\ddot{\phi} = \frac{1}{I_x} U_2, U_2[Nm]$$
$$= \frac{1}{I_x}(F_4 - F_2)l = \frac{1}{I_x} K_{th} l(u_4 - u_2)$$
$$\Rightarrow \ddot{\phi} = \frac{2K_{th}l}{I_x} \delta_{roll}$$

To achieve the preset value $\phi$, find the relationship between it and $\delta_{roll}$:

From mathematics motion model with Roll axis:

$$\begin{cases} \dot{\phi} = w_{\phi}(t) \\ \dot{w}_{\phi}(t) = \dfrac{2K_{th}l}{I_x}\delta_{roll} \end{cases}$$

Apply Laplace to both sides of the first equation:

$$L\{\dot{\phi}(t)\} = L\{w_{\phi}(t)\}$$
$$\Rightarrow s\phi(t) = w_{\phi}(s)$$
$$\Rightarrow G_2(s) = \frac{\phi(t)}{w_{\phi}(s)} = \frac{1}{s}$$

Apply Laplace to both sides of the second equation:

$$L\{\dot{w}_{\phi}(t)\} = L\left\{\frac{2K_{th}l}{I_x}\delta_{roll}\right\}$$
$$\Rightarrow sw_{\phi}(s) = \frac{2K_{th}l}{I_x}\delta_{roll}(s)$$
$$\Rightarrow G_1(s) = \frac{w_{\phi}(s)}{\delta_{roll}(s)} = \frac{2K_{th}l}{I_x s}$$

Finally, the transfer function is:

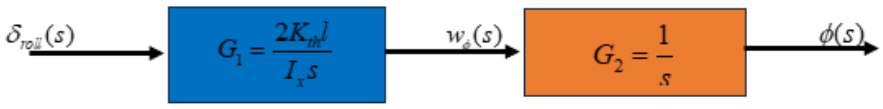



*Figure 4. 6 Transfer function for Roll axis control*

### 4.4.2. Pitch axis

In order to design a Pitch axis controller, a linear model of the quadcopter in roll motion with $\phi = 0, \dot{\phi} = 0, \dot{\theta} = 0$ *[11]*

The mathematics motion model with Pitch axis:

$$\ddot{\theta} = \frac{1}{I_y} U_3, U_3[Nm]$$
$$= \frac{1}{I_y}(F_1 - F_3)l = \frac{1}{I_y} K_{th} l(u_1 - u_3)$$
$$\Rightarrow \ddot{\theta} = \frac{2K_{th}l}{I_y}\delta_{pitch}$$

To achieve the preset value $\theta$, find the relationship between it and $\delta_{pitch}$:

From mathematics motion model with Roll axis:

$$\begin{cases} \dot{\theta} = w_\theta(t) \\ \dot{w}_\theta(t) = \dfrac{2K_{th}l}{I_y}\delta_{pitch} \end{cases}$$

Apply Laplace to both sides of the first equation:

$$L\{\dot{\theta}(t)\} = L\{w_\theta(t)\}$$
$$\Rightarrow s\theta(s) = w_\theta(s)$$
$$\Rightarrow G_2(s) = \frac{\theta(s)}{w_\theta(s)} = \frac{1}{s}$$

Apply Laplace to both sides of the second equation:

$$L\{\dot{w}_\theta(t)\} = L\left\{\frac{2K_{th}l}{I_y}\delta_{pitch}\right\}$$
$$\Rightarrow sw_\theta(s) = \frac{2K_{th}l}{I_y}\delta_{pitch}(s)$$
$$\Rightarrow G_1(s) = \frac{w_\theta(s)}{\delta_{pitch}(s)} = \frac{2K_{th}l}{I_y s}$$

Finally, the transfer function is:

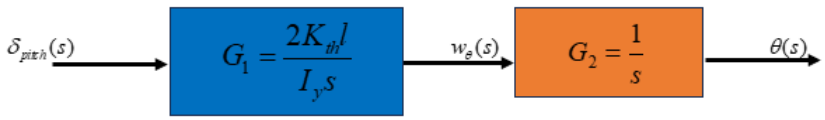



*Figure 4. 7 Transfer function for Pitch Axis control*

### 4.4.3. Yaw axis

In order to design a Yaw axis controller, a linear model of the quadcopter in roll motion with $\phi = 0, \dot{\phi} = 0, \dot{\theta} = 0$ *[11]*

The mathematics motion model with Yaw axis:

$$\ddot{\psi} = \frac{1}{I_z} U_4, U_4[Nm]$$
$$= \frac{1}{I_z}(-M_1 + M_2 - M_3 + M_4) = \frac{1}{I_y} K_{th} l(-u_1 + u_2 - u_3 + u_4)$$
$$\Rightarrow \ddot{\theta} = \frac{4K_{th} l}{I_z} \delta_{yaw}$$

To achieve the preset value $\psi$ , find the relationship between it and $\delta_{yaw}$:

From mathematics motion model with Roll axis:

$$\begin{cases} \dot{\psi} = w_{\psi}(t) \\ \dot{w}_{\psi}(t) = \dfrac{4K_{th} l}{I_z} \delta_{yaw} \end{cases}$$

Apply Laplace to both sides of the first equation:

$$L\{\dot{\psi}(t)\} = L\{w_{\psi}(t)\}$$
$$\Rightarrow s\psi(s) = w_{\psi}(s)$$
$$\Rightarrow G_2(s) = \frac{\psi(s)}{w_{\psi}(s)} = \frac{1}{s}$$

Apply Laplace to both sides of the second equation:

$$L\{\dot{w}_{\psi}(t)\} = L\left\{\frac{4K_{th} l}{I_z} \delta_{yaw}\right\}$$
$$\Rightarrow s w_{\psi}(s) = \frac{4K_{th} l}{I_z} \delta_{yaw}(s)$$
$$\Rightarrow G_1(s) = \frac{w_{\psi}(s)}{\delta_{yaw}(s)} = \frac{4K_{th} l}{I_z s}$$

Finally, the transfer function is:

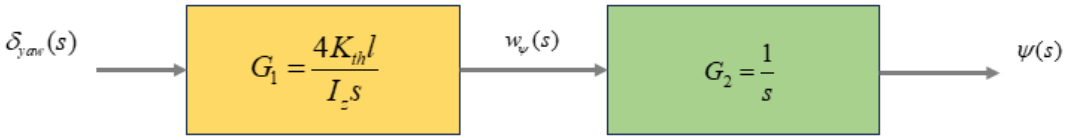



*Figure 4. 8 Transfer fucntion for Yaw axis control*

## 4.5. Position control

### 4.5.1. Transfer function for controlling position

*[10]* The mathematics motion model:

$$\begin{cases} \ddot{x} = -\dfrac{U_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)}{m} \\ \ddot{y} = -\dfrac{U_1(\cos\phi\sin\theta\cos\psi - \sin\phi\cos\psi)}{m} \end{cases}$$

Set: $u_x = \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \quad (4.1)$

$u_y = \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \quad (4.2)$

Thus:

$$\begin{cases} \ddot{x} = -\dfrac{U_1 u_x}{m} \\ \ddot{y} = -\dfrac{U_1 u_y}{m} \end{cases}$$

When UAV at stable altitude so $U_1$ is constant. So the signal control are $u_x$ *and* $u_y$

From equation (4.1): $\sin\theta = \dfrac{u_x - \sin\phi\sin\psi}{\cos\phi\cos\psi}$

From equation (4.2):

$$\sin\theta = \frac{u_y + \sin\phi\cos\psi}{\cos\phi\sin\psi}$$

$$\Rightarrow \frac{u_x - \sin\phi\sin\psi}{\cos\phi\cos\psi} = \frac{u_y + \sin\phi\cos\psi}{\cos\phi\sin\psi}$$

$$\Rightarrow \frac{u_x - \sin\phi\sin\psi}{\cos\psi} = \frac{u_y + \sin\phi\cos\psi}{\sin\psi}$$

$$\Rightarrow u_x \sin\psi - \sin\phi\sin\psi^2 = u_y\cos\psi + \sin\phi\cos\psi^2$$

$$\Rightarrow \sin\phi = u_x\sin\psi - u_y\cos\psi$$

$$\Rightarrow \phi = \arcsin(u_x\sin\psi - u_y\cos\psi)$$

Replace this equation to (4.2):

$$u_x = \cos\phi\sin\theta\cos\psi + \sin\psi(u_x\sin\psi - u_y\cos\psi)$$

$$\sin\theta = \frac{u_x - u_x\sin^2\psi + u_y\sin\psi\cos\psi}{\cos\psi\cos\phi}$$

$$\sin\theta = \frac{u_x\cos^2\psi + u_y\sin\psi\cos\psi}{\cos\psi\cos\phi} = \frac{u_x\cos\psi + u_y\sin\psi}{\cos\phi}$$

$$\Rightarrow \theta = \arcsin(\frac{u_x\cos\psi + u_y\sin\psi}{\cos\phi})$$

Summarize:

$$\begin{pmatrix}\phi_d\\ \theta_d\end{pmatrix} = \begin{pmatrix}\arcsin(u_x\sin\psi - u_y\cos\psi)\\ \arcsin(\frac{u_x\cos\psi + u_y\sin\psi}{\cos\phi}\end{pmatrix}$$

To simplify the position control, set the yaw angle to 0 ($\psi = 0$)

Replace $\psi = 0$ into calculated equations about Roll and Pitch angles:

$$\begin{pmatrix}\phi_d\\ \theta_d\end{pmatrix} = \begin{pmatrix}\arcsin(u_y)\\ \arcsin(\frac{u_x}{\cos\phi}\end{pmatrix}$$

All off above mathematic models:

$$\begin{cases} \dot{x}(t) = v_x(t) \\ \dot{v}_x(t) = \ddot{x}(t) = -\dfrac{1}{m}U_1 u_x(t) \\ \dot{y}(t) = v_y(t) \\ \dot{v}_y = \ddot{y}(t) = -\dfrac{1}{m}U_1 u_y(t) \end{cases} \quad (1)$$

Laplace transformation (1) at the second and fourth equation both sides:

$$\begin{cases} L\{\dot{v}_x(t)\} = L\left\{-\dfrac{1}{m}U_1 u_x(t)\right\} \\ L\{\dot{v}_x(t)\} = L\{-\dfrac{1}{m}U_1 u_x(t)\} \end{cases}$$

$$\Rightarrow \begin{cases} sv_x(s) = -\dfrac{1}{m}U_1 u_x(s) \\ sv_y(s) = -\dfrac{1}{m}U_1 u_x(s) \end{cases}$$

$$\Rightarrow \begin{cases} G_{x1} = \dfrac{v_x(s)}{u_x(s)} = -\dfrac{1}{ms}U_1 \\ G_{y1} = \dfrac{v_y(s)}{u_y(s)} = -\dfrac{1}{ms}U_1 \end{cases}$$

Laplace transformation (1) at the first and third equation both sides:

$$\begin{cases} L\{\dot{x}(t)\} = L\{v_x(t)\} \\ L\{\dot{y}(t)\} = L\{v_y(t)\} \end{cases}$$

$$\Rightarrow \begin{cases} sx(s) = v_x(s) \\ sy(s) = v_y(s) \end{cases}$$

$$\Rightarrow \begin{cases} G_{x2} = \dfrac{x(s)}{v_x(s)} = \dfrac{1}{s} \\ G_{y2} = \dfrac{y(s)}{v_y(s)} = \dfrac{1}{s} \end{cases}$$

Block Diagram for Position Control:

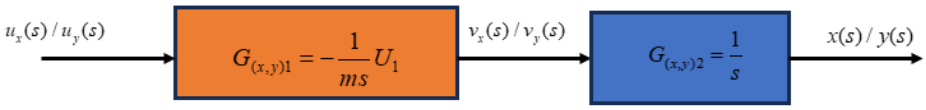



*Figure 4. 9 Block Diagram for Position Control*

### 4.5.2. P-P Controller Design for Position Control

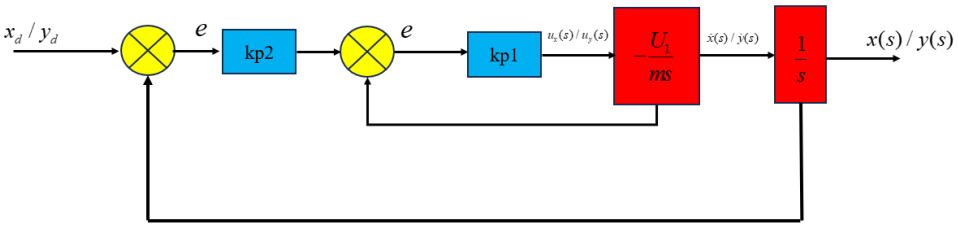



*Figure 4. 10 Block Diagram for P-P Control Position*

*[13]* Based on Block Diagram: $u_x = k_{p1}(k_{p2}(x_d - x) - \dot{x})$

Replace to equation:

$$\ddot{x} = -\frac{U_1 u_x}{m}$$

$$\Leftrightarrow \ddot{x} = -\frac{U_1(k_{p1}(k_{p2}(x_d - x) - \dot{x}))}{m}$$

$$\Leftrightarrow \ddot{x} - \frac{1}{m}U_1 k_{p1}\dot{x} - \frac{1}{m}U_1 k_{p1}k_{p2}x = -\frac{1}{m}U_1 k_{p1}k_{p2}x_d$$

Laplace transformation in both sides:

$$\left[s^2 - \frac{1}{m}U_1 k_{p1}s - \frac{1}{m}U_1 k_{p1}k_{p2}\right]x(s) = -\frac{1}{m}U_1 k_{p1}k_{p2}x_d(s)$$

$$G_h(s) = \frac{x(s)}{x_d(s)} = \frac{-\frac{1}{m}U_1 k_{p1}k_{p2}}{s^2 - \frac{1}{m}U_1 k_{p1}s - \frac{1}{m}U_1 k_{p1}k_{p2}}$$

**$G_h(s)$** convert to a second-order transfer function:

$$G_h(s) = \frac{w^2{}_n}{s^2 + 2\xi w_n s + w^2{}_n}$$

With: $w^2{}_n = \frac{1}{m} U_1 k_{p1} k_{p2}; 2w_n\xi = \frac{1}{m} U_1 k_{p1}; U_1 = mg$

So: $k_{p1} = -\frac{2\xi w_n}{g}; k_{p2} = -\frac{w_n}{2\xi}$

Set: $w_n = 12; \xi = 0.707 \Rightarrow k_{p1} = -1.723; k_{p2} = -8.48$

## 4.6. Altitude control

### 4.6.1. The transfer function for controlling altitude

*[12]* The mathematic model of altitude:

$$\ddot{z} = g - \frac{U_1(\cos\phi\cos\theta)}{m}$$

Considering the z-axis in the Earth coordinate system with the positive direction pointing downwards, and considering the altitude ***h*** with the positive direction pointing upwards, we can express the relationship as: ***z = -h***:

$$=> \ddot{h} = -g + \frac{U_1(\cos\phi\cos\theta)}{m}$$

With: $U_1 = \sum_1^4 F_i = 4K_{th}(u_0 + \delta_{th})$

$$\ddot{h} = -g + \frac{4K_{th}(u_0 + \delta_{th})_1 \cos\phi\cos\theta}{m} \quad (*)$$

Simplify the signal, set $(u_0 + \delta_{th})\cos\phi\cos\theta = u_0 + \delta^{'}{}_{th}$ (**)

When quadcopter at stable position:

$$4K_{th}u_0 = mg \Rightarrow g = \frac{4K_{th}u_0}{m}$$

Replace (**) to (*):

$$\ddot{h} = -g + \frac{4K_{th}u_0}{m} + \frac{4K_{th}\delta'_{th}}{m}$$

$$\Rightarrow \ddot{h} = \frac{4K_{th}\delta'_{th}}{m}$$

Finding the relation between $\delta'_{th}$ and $\delta_{th}$:

$$\Rightarrow \delta_{th} = \frac{u_0 + \delta'_{th}}{\cos\phi\cos\theta} - u_0$$

With mathematic model:

$$\begin{cases} \dot{h}(t) = v_h(t) \\ \dot{v}_h(t) = \dfrac{4K_{th}\delta'_{th}}{m} \end{cases}$$

Laplace at the first equation:

$$L\{\dot{h}(t)\} = L\{v_h(t)\}$$
$$\Rightarrow sh(s) = v_h(s)$$
$$\Rightarrow G_2(s) = \frac{h(s)}{v_h(s)} = \frac{1}{s}$$

Laplace at the second equation:

$$L\{\dot{v}_h(t)\} = L\left\{\frac{4K_{th}\delta'_{th}}{m}\right\}$$
$$\Rightarrow sv_h(s) = \frac{4K_{th}\delta'_{th}(s)}{m}$$
$$\Rightarrow G_1(s) = \frac{v_h(s)}{\delta'_{th}(s)} = \frac{4K_{th}}{ms}$$

The transfer function for controlling altitude as follow:

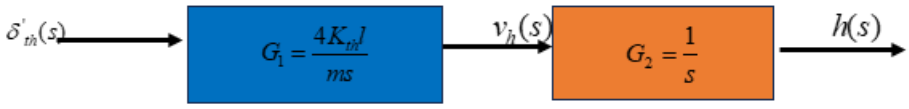



*Figure 4. 11 The block Diagram for Altitude control*

### 4.6.2. P-P Controller Design for Altitude Control

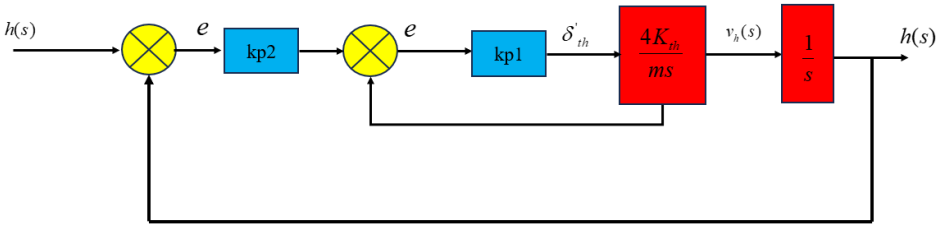



*Figure 4. 12 The Block Diagram P-P Controller for Altitude Control*

*[13]* Based on Block Diagram:

$$\delta'_{th} = k_{p1}(k_{p2}(h_d - h) - \dot{h})$$

Replace to equation which describe altitude based on mathematic model:

$$\ddot{h} = \frac{4K_{th}}{m} k_{p1}(k_{p2}(h_d - h) - \dot{h})$$

$$\Leftrightarrow \ddot{h} + \frac{4K_{th}}{m} k_{p1}k_{p2}h + \frac{4K_{th}}{m} k_{p1}\dot{h} = \frac{4K_{th}}{m} k_{p1}k_{p2}h_d$$

Laplace transformation both sides:

$$L\left\{\ddot{h}+\frac{4K_{th}}{m}k_{p1}k_{p2}h+\frac{4K_{th}}{m}k_{p1}\dot{h}\right\}=L\left\{\frac{4K_{th}}{m}k_{p1}k_{p2}h_d\right\}$$

$$\Rightarrow\left[s^2+\frac{4K_{th}}{m}k_{p1}s+\frac{4K_{th}}{m}k_{p1}k_{p2}\right]h(s)=\frac{4K_{th}}{m}k_{p1}k_{p2}h_d(s)$$

$$\Rightarrow G_h(s)=\frac{h(s)}{h_d(s)}=\frac{\frac{4K_{th}}{m}k_{p1}k_{p2}}{s^2+\frac{4K_{th}}{m}k_{p1}s+\frac{4K_{th}}{m}k_{p1}k_{p2}}$$

Convert to a transfer function second order:

$$G_h(s)=\frac{w^2{}_n}{s^2+2\xi w_n s+w^2{}_n}$$

With:

$$\begin{cases}w^2{}_n=\dfrac{4K_{th}}{m}k_{p1}k_{p2}\\ 2\xi w_n=\dfrac{4K_{th}}{m}k_{p1}\end{cases}$$

Thus:

$$k_{p1}=\frac{\xi w_n m}{2K_{th}};\ k_{p2}=\frac{w_n}{2\xi}$$

The final Block Diagram for Controlling Dynamic:

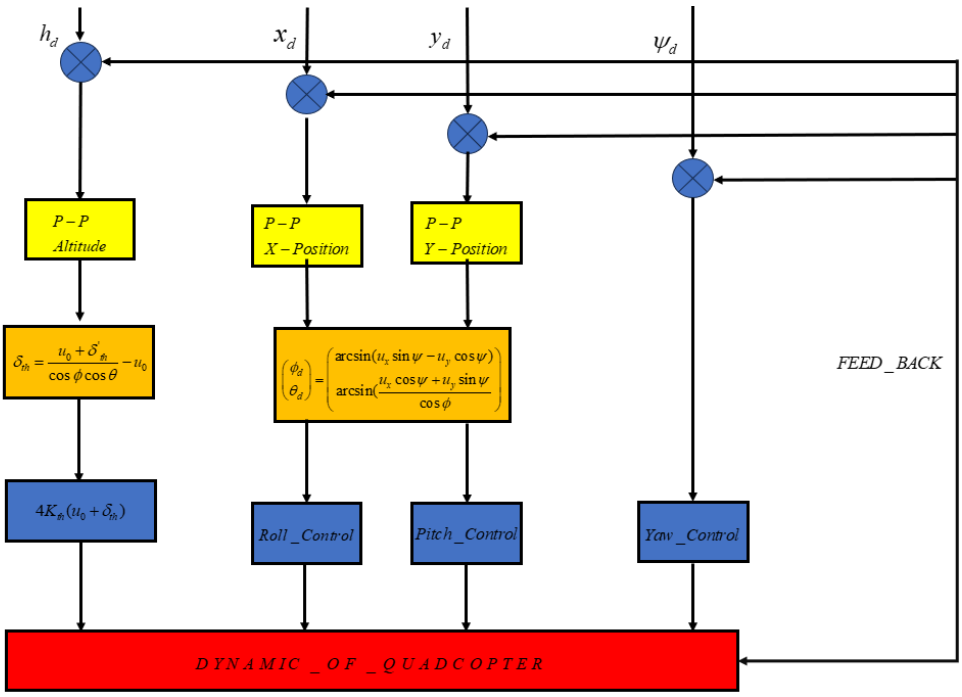



*Figure 4. 13 Block Diagram for Dynamic Control*

# CHAPTER 5: APPLICATION FOR FIREFIGHTING QUADCOPTER

## 5.1. Gripper

### 5.1.1. Components on gripper

#### *5.1.1.1. Gear Reducer*

The DC GA25-370 130rpm is a direct current (DC) gear motor designed for applications demanding control and torque at low speeds. Its integrated gearbox efficiently reduces the motor's rotational speed to a precise 130 revolutions per minute (rpm). This geared configuration makes the DC GA25-370 particularly suitable for scenarios where precise control and sustained turning force is more essential than velocity control.

*Figure 5. 1 The gear reducer*

| Specifications | Values |
|---|---|
| **Operating Voltage** | 12V |
| **Gear Ratio** | 46.8:1 |
| **No-load Speed** | 130 rpm |
| **Load Speed** | 100 rpm (at 10% load) |
| **Rated Torque** | 0.9 kgf.cm (0.83 lb.ft) |

| | |
|---|---|
| **Maximum Torque** | 4.4 kg (9.7 lb) |
| **Gearbox Length (L):** | 21 mm (0.83 in) |

*Table 5. 1 Specifications of Motor GA25-370*

### *5.1.1.2. H-Bridge L-298N*

It uses four switches to control the direction of current flow to a load, typically a DC motor. This allows the H-bridge to control the motor's direction (forward or backward) and even braking. The versatility of H-bridges lies in their ability to manipulate the flow of current, making them a indispensable electrical equipment of DC motor control in various fields.

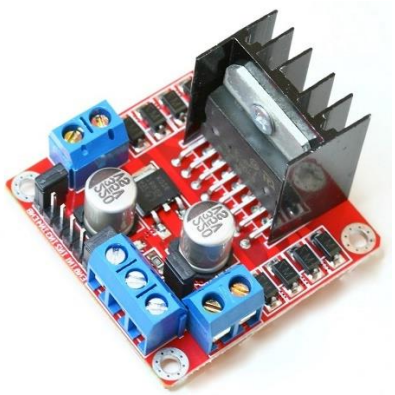


*Figure 5. 2 L-298N*

| Pin name | Description |
|---|---|
| IN1 & IN2 | Motor A input pins. Used to control the spinning direction of Motor A |
| IN3 & IN4 | Motor B input pins. Used to control the spinning direction of Motor B |
| ENA | Enables PWM signal for Motor A |

| | |
|---|---|
| ENB | Enables PWM signal for Motor B |
| OUT1 & OUT2 | Output pins of Motor A |
| OUT3 & OUT4 | Output pins of Motor B |

*Table 5. 2 Description of Pin on L298-N*

#### 5.1.1.3. Gripper Mechanism

The primary material employed in the construction of the gripper mechanism is PLA, a thermoplastic commonly utilized in 3D printing technology. PLA exhibits favorable physical and mechanical properties, including high durability and ease of machining.

The gripper mechanism comprises three primary components:

- **Gripping Unit:** The gripping unit is responsible for grasping or releasing the fire extinguisher ball based on signals transmitted to the motor.

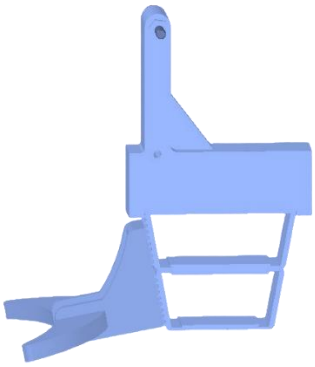


*Figure 5. 3 The Gripping Unit*

- **Support Base:** The support base serves to secure the fire extinguisher ball, preventing it from falling in unintended directions.

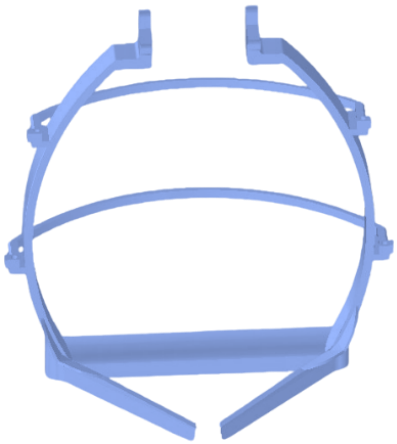


*Figure 5. 4 The Support Base*

- **Motor Housing:** The motor housing encloses the motor and provides a mounting point for the entire gripper mechanism. As the component subjected to the most significant forces, the motor housing must exhibit exceptional structural strength.

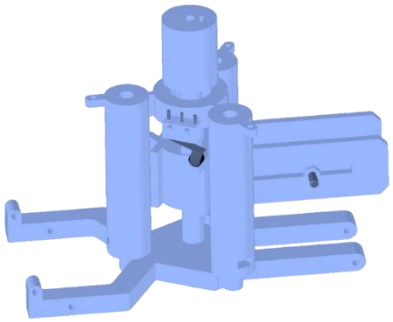


*Figure 5. 5 The Motor Housing*

### 5.1.2. Transmission mechanism of a Gripper

The motor's rotary motion is first converted into linear vertical movement using a screw-nut assembly. The nut, connected to the motor's shaft through the screw, moves up and down along the screw's threads. This linear motion is then converted back into rotary motion for the gripper's arm. A linear guide ensures the nut travels in a straight line, while a rotating shaft connected to the gripper

engages with the nut, transferring its movement into the gripper's rotation. This design allows for controlled raising and lowering of the gripper arm.

This system utilizes a robotic gripper for the deployment of fire extinguishers. The gripper's operation is controlled by a camera-based fire detection and processing unit. Upon identification of a fire surrounding environment, the camera transmits a signal to the motor through a JetsonNano module, initiating its rotation. This action opens the gripper, causing a fire extinguisher to be released and roll towards the fire for extinguishment.

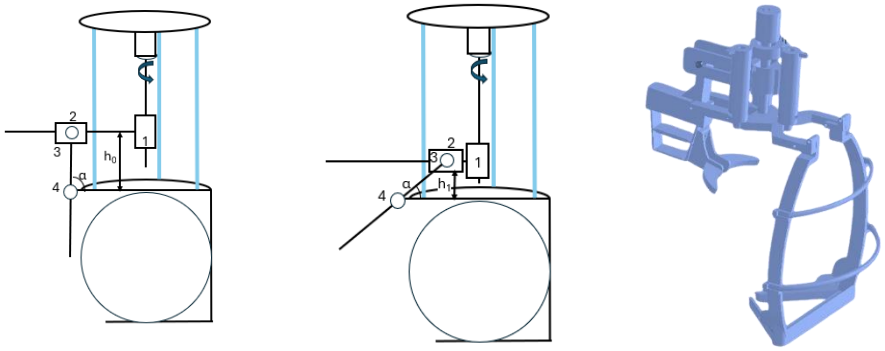



*Figure 5. 6 The transmission mechanism of Gripper*

### 5.1.3. Stress Analysis of Gripper

#### *5.1.3.1. Deformation of Load-Bearing Components in a Gripper*

The primary function of the gripper is to grasp and release fire extinguishing balls. To prevent potential cracks or fractures during operation under load, the team conducted a deformation simulation to identify the components that will bear the most load. Based on the simulation results, the team proposes various measures to enhance the load-bearing capacity of these components, including: Increasing Cross-Section, increasing Ribs,…

Altair Inspire is a comprehensive simulation-driven design software suite that empowers users to generate, optimize, and manufacture

innovative parts within a unified environment. This capability enables users to gain precise insights into the deformation and load-bearing capacities of various structural components. The provided images depict 3D printed components fabricated using PLA (Polylactic Acid) filament. PLA is a thermoplastic polymer known for its favorable deformation characteristics. Based on the visual assessment of the components, it can be concluded that their deformation falls within the acceptable range. There are no apparent signs of fracture or exceeding the components load-bearing limits

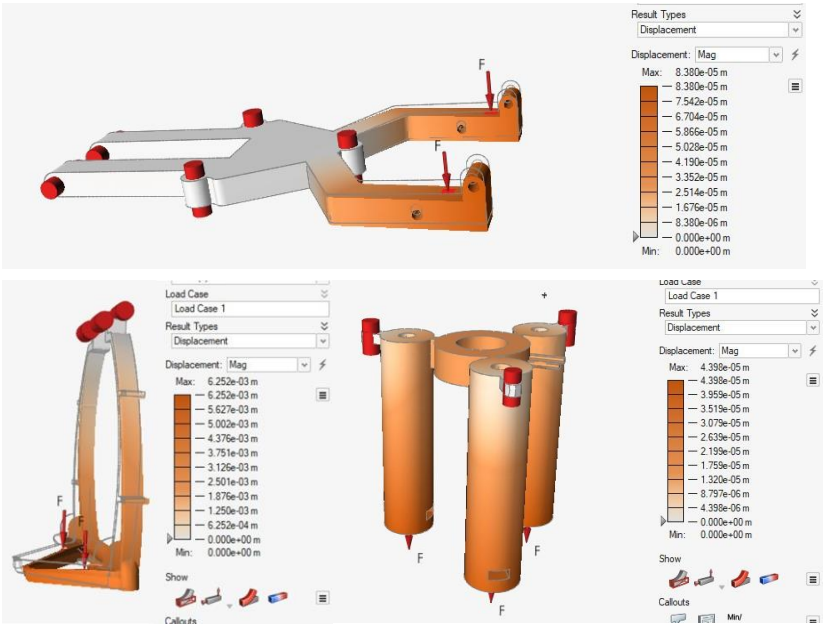
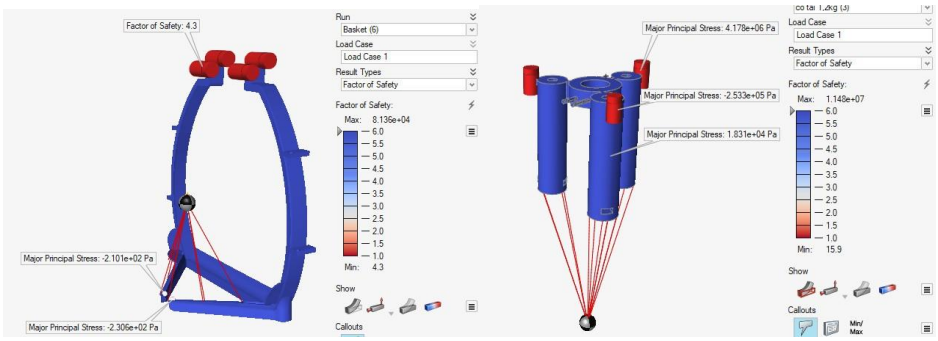



*Figure 5. 7 Deformation Analysis of Load-Bearing Components*

### *5.1.3.2. Simulation and testing the Strength of Components in a Gripper*



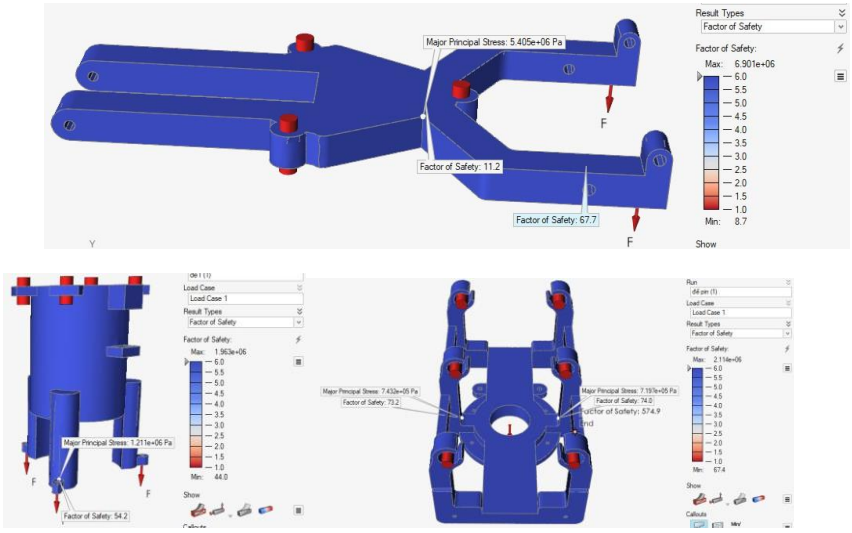



*Figure 5. 8 Strength Analysis of Load-Bearing Components*

Simulation to calculate the stress at all points on the part. We have the yield strength of the material ($[\sigma_{PLA}] = 42$ *(MPa))*. We divide the yield strength of the material by the stress to get the safety factor (FS)

### *5.1.3.3. Analyzing the velocity of Gripper mechanism*

Given the motor speed of 130 rpm and the lead screw pitch of 8 mm, the linear speed of the slider in the vertical direction can be calculated as follows:

$$v_{linear} = \frac{\omega * d}{60}$$

With:

- $v_{linear}$ : the speed of slider (mm/s)
- $d$ : the lead screw pitch (mm)
- $\omega$ : the angular speed of motor (rad/s)

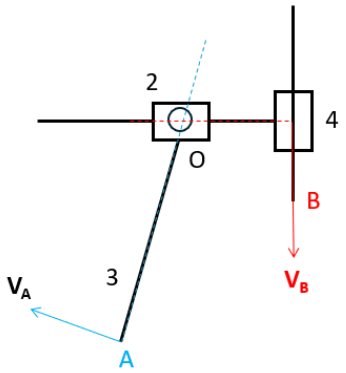



*Figure 5. 9 Illustration for Gripper Mechanism*

The speed of the gripper arm is directly related to the linear speed of the slider. This relationship can be established based on the instantaneous rotation center O. Let $V_A$ represent the speed corresponding to lever arm OA and $V_B$ represent the speed corresponding to lever arm OB. The following expression can be derived:

$$\omega_0 = \frac{V_A}{OA} = \frac{V_B}{OB} = \frac{V_{linear}}{OB}$$

With:

- $V_A$: speed of the end of clipper (mm/s)
- $V_B$: speed of slider on (mm/s)
- OA, OB: distance from O to A and B (mm)

$$V_A = \frac{V_B * OA}{OB}$$

Value OA and VB is constant, value OB consistently changing depend on the $\alpha < 90^0$ (the angle of direction of gripper and horizontal line):

$$OB = 56 - 38 * cos(\alpha)\ (mm)$$

So the value of $V_A$ is:

$$V_A = \frac{\omega * d}{60} * \frac{OA}{(56 - 38 * cos(\alpha))}\ (\text{mm/s})$$

Set some values of $\alpha\ (anpha)$:

| $\alpha\ (anpha)\ (^0)$ | $V_A\ (mm/s)$ |
|---|---|
| 30 | 38,28 |
| 60 | 23,9 |
| 75 | 19,14 |

*Table 5. 3 Values of α (anpha) ($^0$) and $V_A$ (mm/s)*

## 5.2. Images Processing

Image processing plays a crucial role in firefighting. The utilization of AI models like YOLOv9 significantly enhances the accuracy of fire image prediction.

The YOLOv9 model stands as a cutting-edge advancement in the field of machine learning, specifically designed for object detection tasks. YOLOv9 excels in real-time tasks, making it valuable in various fields, especially firefighting where it precisely detects fires in images.

All steps for preparing the datasets:

- *Download Flame Image Data from Kaggle*
- *Upload Data to MakeSense.ai*

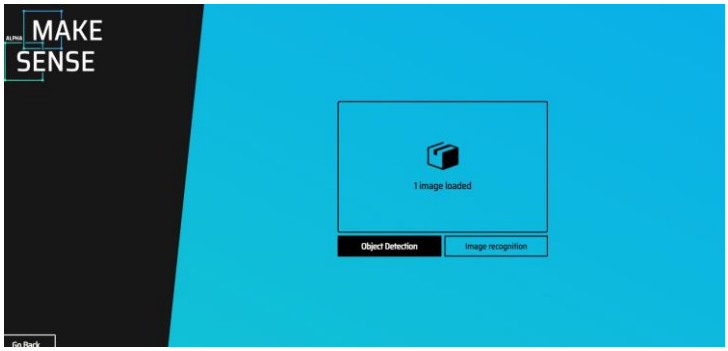



*Figure 5. 10 UI on MakeSense.ai*

- *Crop Flame Images*

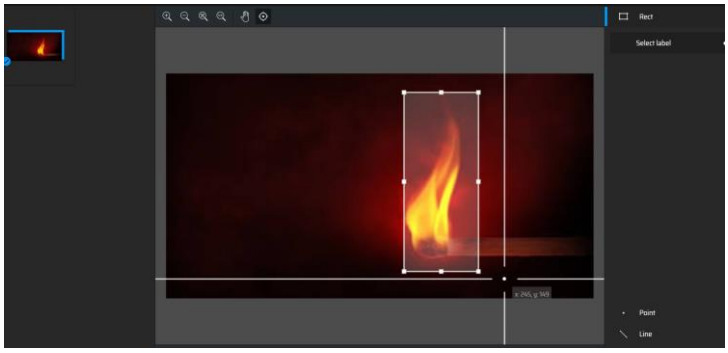



*Figure 5. 11 Crop the images*

- *Each cropped flame image, create a corresponding label file (.txt) with the same name and specify the coordinates of the flame with bounding box coordinates (x1, y1, x2, y2).*

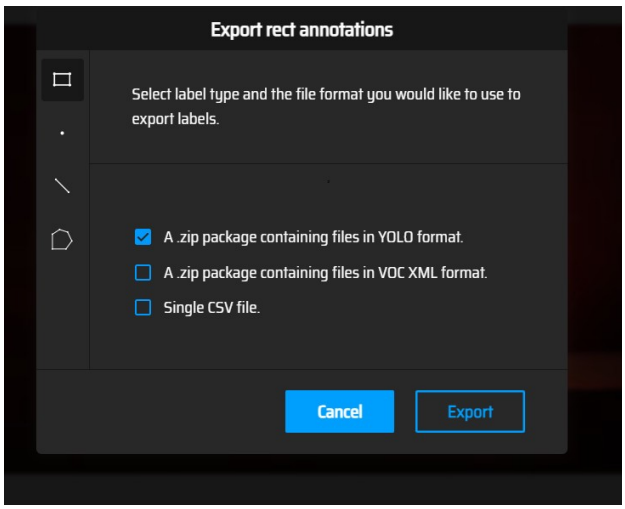



*Figure 5. 12 Final step of process*

## Training model with preparing data:

- *Download github of YOLO*

  %cd /content/drive/MyDrive !git clone
  https://github.com/WongKinYiu/yolov9

- *Install required library:*

  %cd /content/drive/MyDrive/yolov9
  !pip install -r requirements.txt

- *Download weight pretrain:*

  %cd /content/drive/MyDrive/yolov9 !wget -P /content/drive/MyDrive/yolov9
  https://github.com/WongKinYiu/yolov9/releases/download/v0.1/yolov9-e.pt

- *Training model with custom data:*

  # Upload data on Drive, prepare file data.yaml saved in Yolov9
  # File configured model to train

  %cd /content/drive/MyDrive/yolov9

  !cp/content/drive/MyDrive/yolov9/models/detect/yolov9-e.yaml
  /content/drive/MyDrive/yolov9/models/detect/yolov9-e-custome.y

#Train yolov9 models

```
!python train_dual.py --workers 8 --device 0 --batch 4 --data /content/drive/MyDrive/yolov9/data.yaml --img 640 --cfg /content/drive/MyDrive/yolov9/models/detect/yolov9-e-custome.yaml --weights /content/drive/MyDrive/yolov9/yolov9-e.pt --name yolov9-c --hyp /content/drive/MyDrive/yolov9/data/hyps/hyp.scratch-high.yaml --min-items 0 --epochs 50 --close-mosaic 15
```

For capturing and displaying the images**,** GStreamer pipelines are used for the aforementioned purposes with the ability to customize input signal parameters. Characteristic parameters include:

- **Sensor ID:** This specifies the ID of the camera sensor to use.
- **Capture Resolution:** This defines the width and height of the captured video frame.
- **Display Resolution:** This defines the width and height of the displayed video frame.
- **Frame Rate:** This specifies the desired frame rate of the video stream.
- **Flip Method:** This controls how the captured video is flipped.

Here some codes for illustrating:

```
def gstreamer_pipeline(
    sensor_id=0,
    capture_width=1920,
    capture_height=1080,
    display_width=720,
    display_height=480,
    framerate=30,
    flip_method=0,
):
```

# CHAPTER 6: CONTROLLER PROGRAMMING

## 6.1. Simulation results

### 6.1.1. HITL

Hardware-in-the-Loop (HITL or HIL) is a simulation mode in which normal PX4 firmware is run on real flight controller hardware. This approach has the benefit of testing most of the actual flight code on the real hardware. With Hardware-in-the-Loop (HITL) simulation the normal PX4 firmware is run on real hardware. JMAVSim or Gazebo (running on a development computer) are connected to the flight controller hardware via USB/UART.

- **HITL Configuration:** Selected in QGroundControl, simulates flight controller without real sensors.
- **Simulator (jMAVSim or Gazebo):** Connects to flight controller via USB.
- **Connection to PX4:** Bridges MAVLink messages through QGroundControl (UDP)
- **Optional:** Joystick/Gamepad hardware connects via QGroundControl (serial).

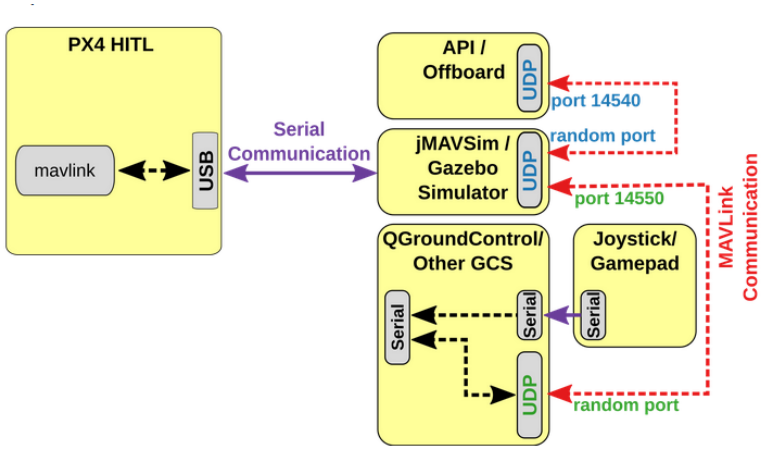



*Figure 6. 1 Diagram of how HITL simulation works [24]*

All of steps for running HIL simulation:

*- Connect the USB type-C between the computer and Pixhawk.*

*- Install the environment for preparing Gazebo's simulations.*

*- Proceed to simulate Gazebo*

*- Plug in FTDI cable to connect between Jetson and Pixhawk and run code's manufacture.*

*- Calibrate parameters of Qgroundcontrol, especially for MAV_1_CONFIG to allow transmit MAVLink throughout TELEM 2 Port*

*- Make sure accurate the transmission rate and address for PX4.launch*

*- Command to run the program:* ***roslaunch mavros PX4 launch***

*- Command to run offboard:* ***rosrun*** *<File code_offboard>*

### 6.1.2. Hardware setup

- *[24]*USB cable: Establish connection QGroundControl and Pixhawk through USB cable.

*Figure 6. 2 Connection between Computer and Pixhawk*

- FTDI cable: Connect Companion computer to TELEM2 port of Pixhawk for transferring data via MAVROS.

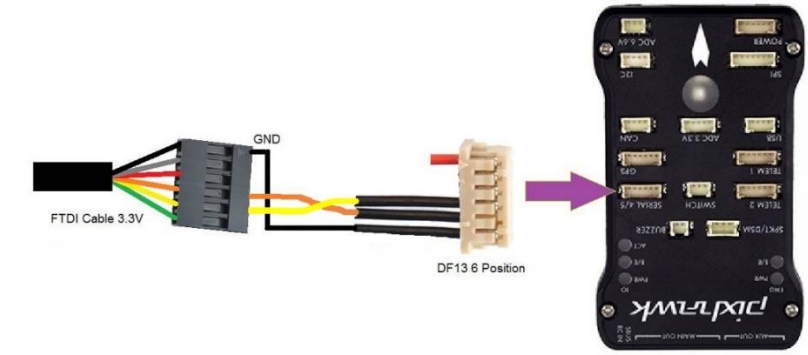



*Figure 6. 3 Connection between FTDI and TELEM2*

### 6.1.3. Software setup

*** Setting:**

- Download latest PX4 firmware and config the air frame
- Config the safety mode
- Open TELEM2*[15]* to connect the Companion Computer
- Guarantee Port for USB cable in QgroundControl is correct
- Verify that the IP address and braudrate in the PX4 file launch is accurate

* *[25]* **Running:**

- *For the fisrt step, run the simulation (Ensure QGroundControl is not running)*

  cd <PX4-Autopilot_clone>

  source/Tools/simulation/gazebo-classic/setup_gazebo.bash $(pwd) $(pwd)/build/px4_sitl_default

  gazebo/Tools/simulation/gazebo-classic/sitl_gazebo-classic/worlds/hitl_iris.world

- *Next step, run MAVROS:*

  roslaunch mavros px4.launch

- *Finally, start QGroundControl (Should Autoconnect to PX4 and Gazebo Classic)*

### 6.1.4. Overview of programming system

The system is built using a pub/sub model, which means that publishers create nodes and then send data through a topic. Subscribers listen to the topic and receive information whenever any data is published.

Here is a general model system is built based on pub/sub model:

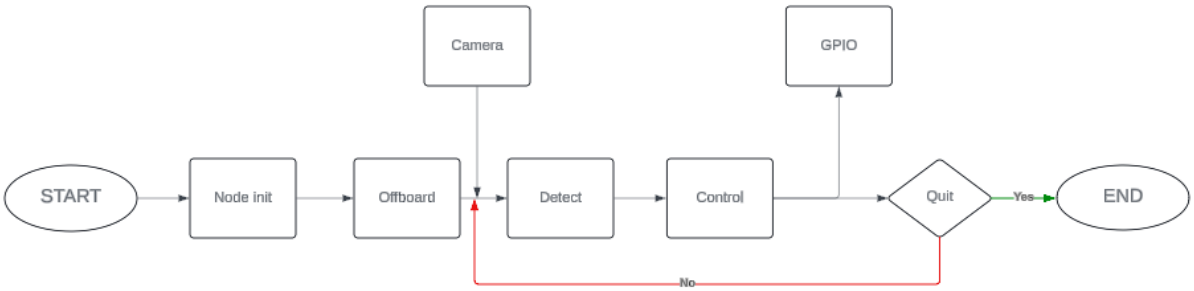



*Figure 6. 4 Graph of programming system*

#### *6.1.4.1. Node Init*

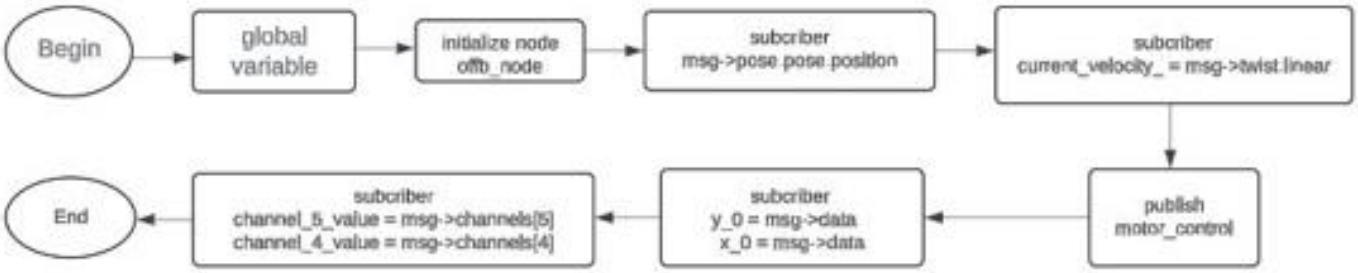



*Figure 6. 5 The graph of Node Init*

### 6.1.4.2. *Offboard*

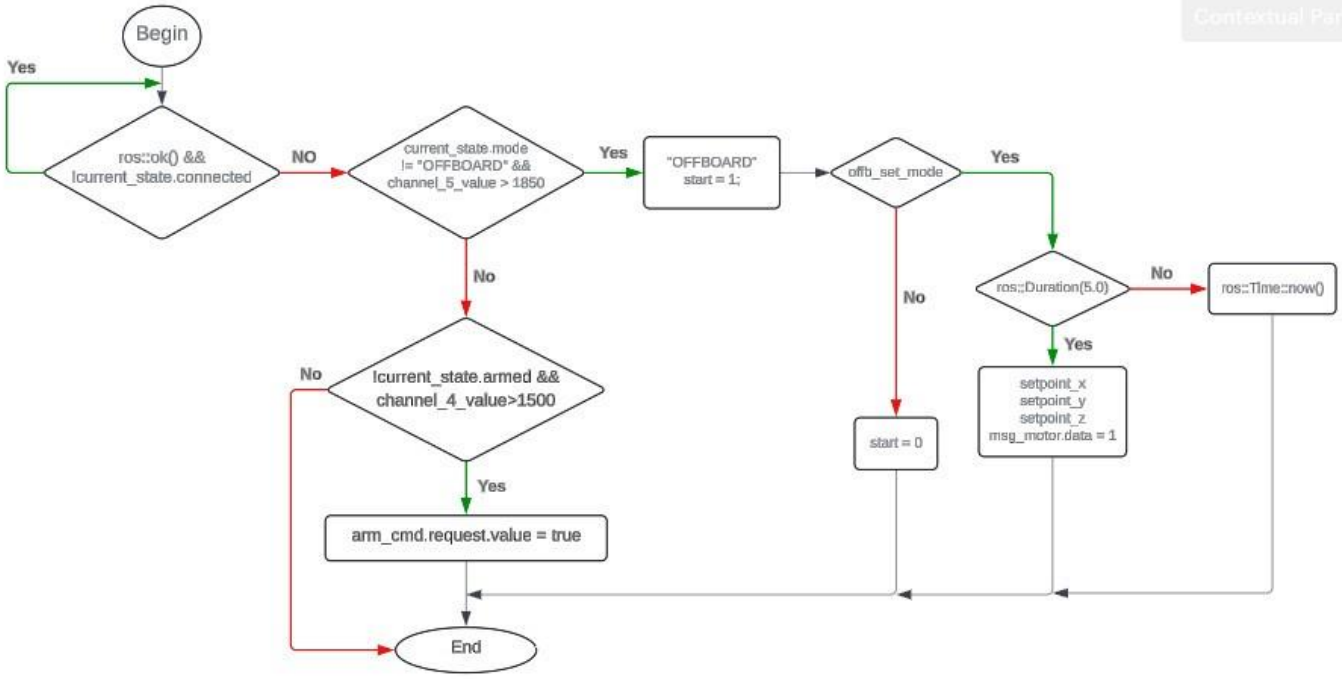



*Figure 6. 6 The Graph of Offboard Node*

### 6.1.4.3. *Camera*

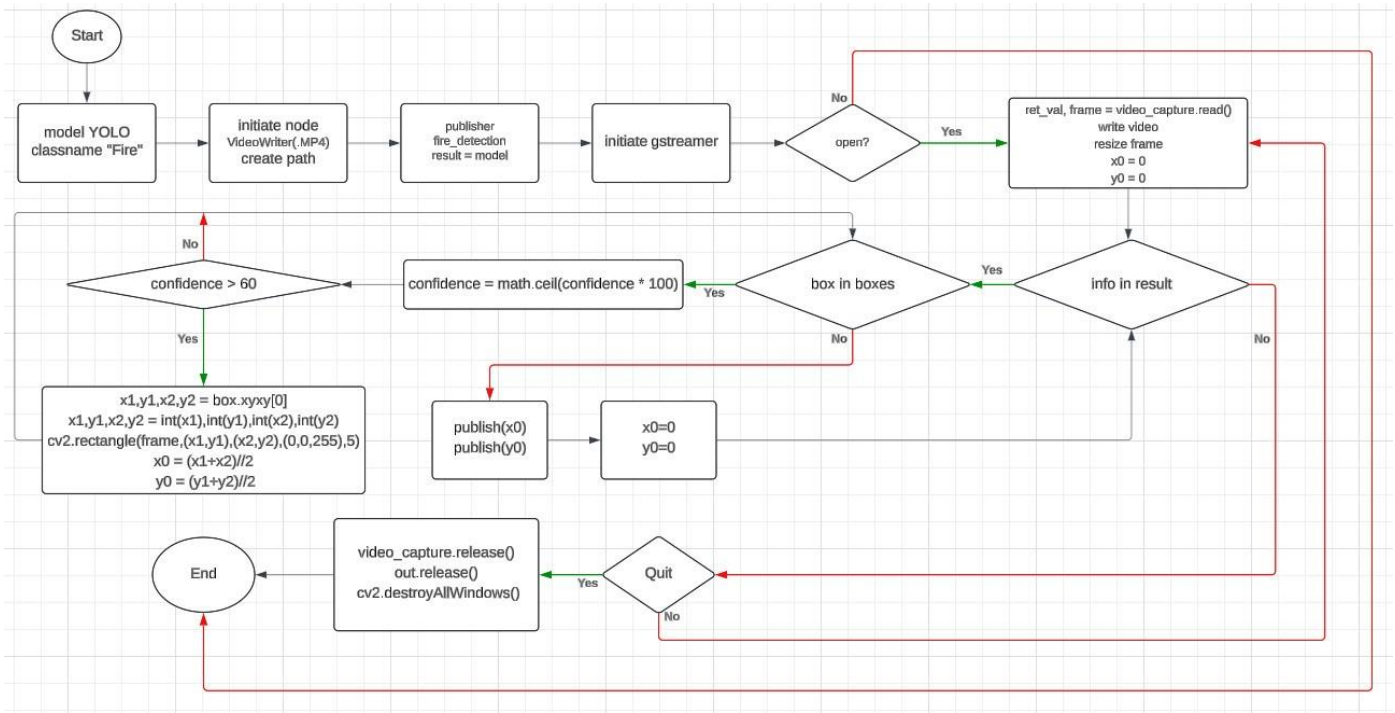



*Figure 6. 7 The Graph of Camera Nod*

### 6.1.4.4. *Detect*

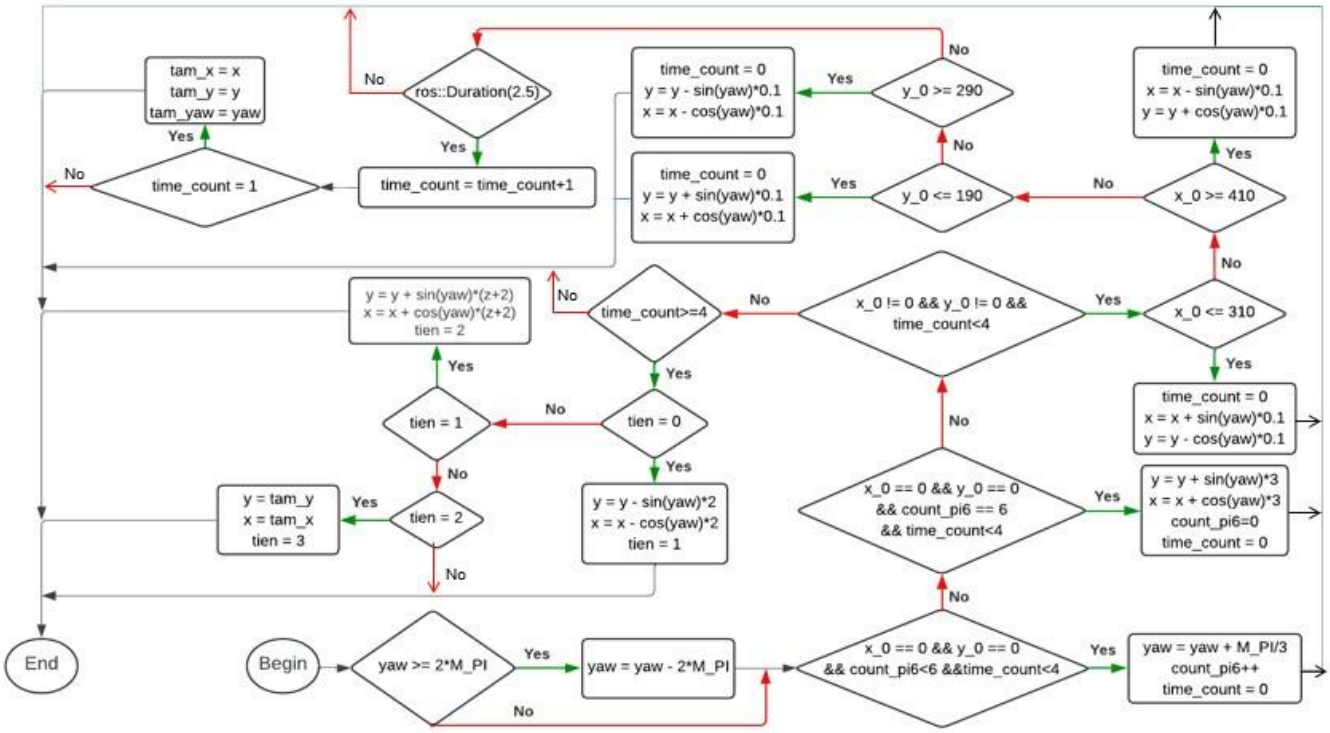



*Figure 6. 8 The graph for Detec node*

### 6.1.4.5. *Control*

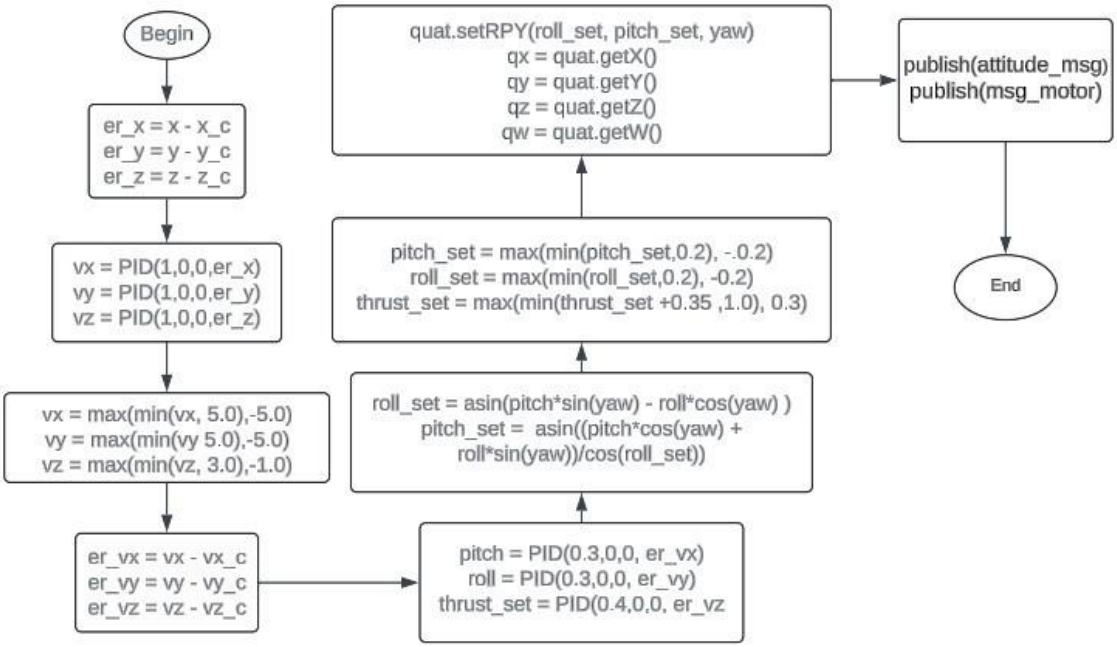



*Figure 6. 9 The graph for Control Node*

### *6.1.4.6. GPIO*

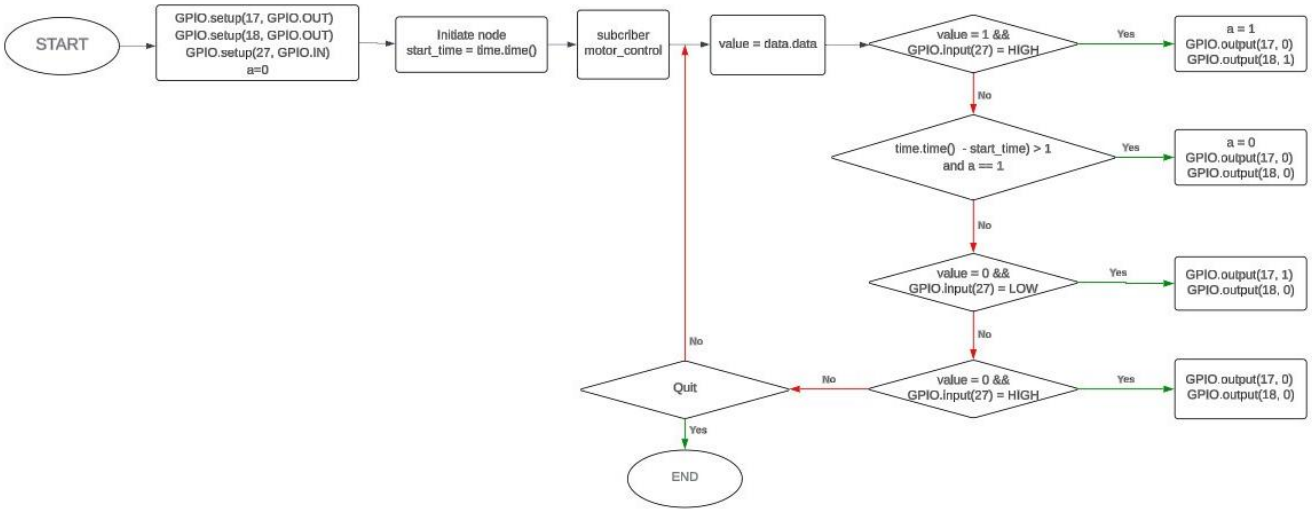



*Figure 6. 10 The graph for GPIO Node*

### 6.1.5. Image processing experiment

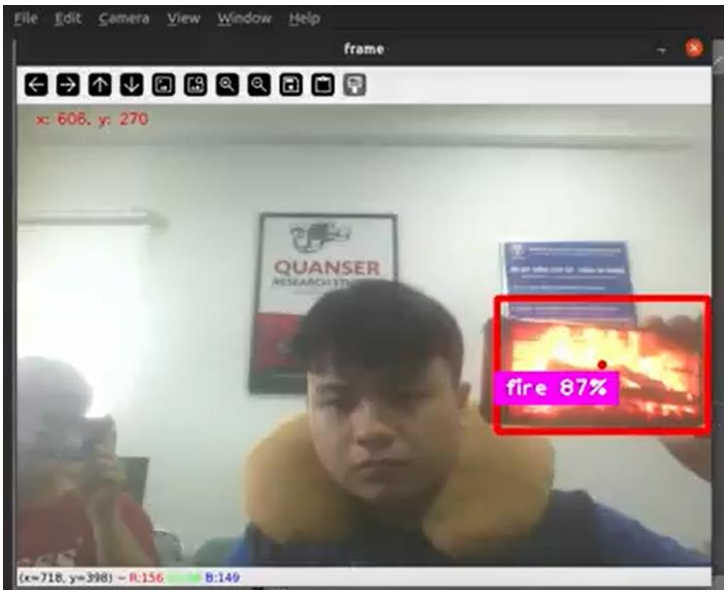



*Figure 6. 11 Experimental Result in Recognition Flame*

The text states that image processing is carried out using the YOLOv9 object detection model and a self-prepared dataset. This indicates that the image processing system is trained on a custom dataset of images containing fire and other objects to effectively detect fire in real-world scenarios.

The text introduces the challenge of performance limitations when running the image processing program on the Jetson Nano hardware. Due to the Jetson Nano's hardware constraints, the processing and recognition tasks experience delays during program execution. Despite the performance delay observed on the Jetson Nano, the text emphasizes that the impact is minor. This suggests that the delay does not significantly hinder the overall functionality of the image processing system for fire detection.

### 6.1.6. Mathematical Model for Quadcopter Fire Detection

#### 6.1.6.1. Find the relationship between the movement of the quadcopter and the change in pixels:

Simulate the parameters for calculation as follows:

- A is camera perspective with wide angle: 120º
- BC is the width of the displayed image frame: about 720 pixels
- The observed object is on segment AB
- Z is the distance between the camera and the object: n (meter)

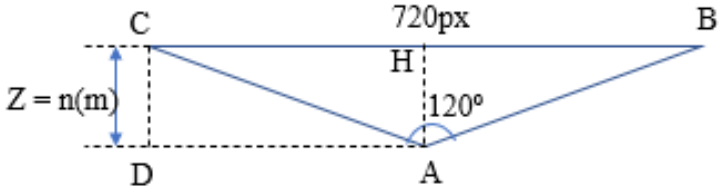



*Figure 6. 12 The geometry illustrate relationship between Z and pixel*

Calculation:

$$C\hat{A}D = \frac{180^0 - 120^0}{2} = 30^0$$

$$DA = \frac{CD}{\tan(30^0)} = n\sqrt{3}(m)$$

$$CH = \frac{CB}{2} = 360(px)$$

=> The relationship between the movement of the quadcopter and the pixel position is: $n\sqrt{3}$ (m) and 360 (px)

=> 10 pixels correspond to a left-right movement of 0.05n (m)

#### 6.1.6.2. Find the relationship between Z distance and quadcopter height

Simulate the parameters for calculation as follows:

- A is camera
- AK is quadcopter height (m) (can be read by pixhawk)
- L is the observed object
- Z is the distance between the camera and the object: n (meter)

- Angle LAK is fix with 45º

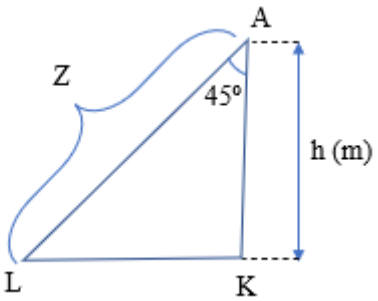



*Figure 6. 13 The geometry illustrate relationship between Z and h*

Caculation:

$$LK = \tan(45^0) * h = h(m)$$
$$AK = \frac{h}{\cos(45^0)} = h\sqrt{2}(m)$$
$$CH = CB/2 = 360(px)$$

So the relationship between the Z distance and quadcopter height is: n=h$\sqrt{2}$ (m)

Thus: 10 pixels correspond to a left-right movement of 0.05n meters = 0.07h (m)

#### 6.1.6.3. The relationship between the movement forward-backward of the quadcopter and the change in pixels

- A is camera perspective with wide angle: 65º
- IP is the width of the displayed image frame: about 400 pixels width
- The observed object is on segment JP
- AJ(Z) is the distance between the camera and the object: n (meter)

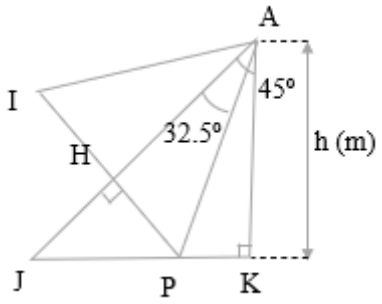



*Figure 6. 14 The geometry illustrate relationship between move back-foward and pixels*

Calculation:

$$AJ = h\sqrt{2}(m)$$
$$A\hat{J}P = 45^0$$
$$HP = JP * \sin(45^0) = \frac{JP}{\sqrt{2}}(m)$$
$$JP = JK - PK = h * \tan(45^0) - h * \tan(12,5^0) = 0,78h(m)$$
$$HP = IP / 2 = 200(px)$$

Found the relationship between the movement of the quadcopter and the pixel position is:

$0.78h/\sqrt{2}$ (m) = 0.55h and 200 (px)

So: 10 pixels correspond to a forward-backward movement is 0.028h (m)

#### 6.1.6.4. The equation describe relation between yaw angle and position x,y

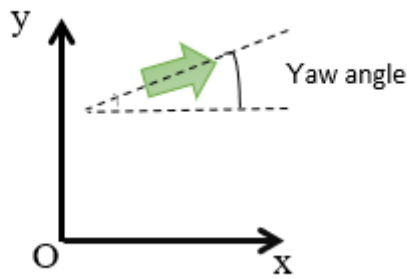



*Figure 6. 15 Coordinate X-axis and Y-axis*

- The angle between the quadcopter's orientation and the positive x-axis is yaw angle

- Formula for calculating x, y coordinates for quadcopter movement to the left to cause a 10-pixel displacement in the captured image

$$x\_d = x\_c - \sin(yaw) * 0,032h(m)$$
$$y\_d = y\_c + \cos(yaw) * 0,032h(m)$$

- Formula for calculating x, y coordinates for quadcopter movement to the right to cause a 10-pixel displacement in the captured image

$$x\_d = x\_c + \sin(yaw) * 0,032h$$
$$y\_d = y\_c - \cos(yaw) * 0,032h$$

- Formula for calculating x, y coordinates for quadcopter move forward to cause a 10-pixel displacement in the captured image

$$x\_d = x\_c + \sin(yaw) * 0,028h(m)$$
$$y\_d = y\_c + \cos(yaw) * 0,028h(m)$$

- Formula for calculating x, y coordinates for move backward to cause a 10-pixel displacement in the captured image

$$x\_d = x\_c - \sin(yaw) * 0,028h$$
$$y\_d = y\_c - \cos(yaw) * 0,028h$$

### 6.1.7. HITL simulation results

The graph data comes from log files. These files are saved in .bin format on the SD card of your Pixhawk 6C. They contain flight information like sensor data, actuator commands, and vehicle state.

These are six graphs that represent roll, pitch, yaw, X, Y, and Z, respectively:

- In the yaw graph, the quadcopter rotates 60 degrees to the left immediately upon takeoff due to the lack of fire detection. Once the camera detects the fire at 39 minutes 30 seconds, the quadcopter ceases its rotation.
- In the X and Y graphs, from 39 minutes 22 seconds to 39 minutes 30 seconds, the quadcopter moves towards a previously set waypoint along the X-axis. At 39 minutes 30 seconds, the camera detects the fire on the left side of the screen, causing the quadcopter to move to the left. At 39 minutes 50 seconds, the fire is detected on the right side of the screen, prompting the quadcopter to move to the right again. This pattern of alternating left and right movements continues until 40 minutes 10 seconds, when the fire is centered in the frame. These movements occur while the quadcopter is at a yaw angle of 110 degrees, so the left and right movements also affect the Y-axis, causing a slight shift in the Y graph. Upon centering the fire, the quadcopter spends about 7 seconds refining its aim and calculating the distance to the fire area. Then, it flies directly to the fire location and drops the payload at the coordinates (7, 0.4, 0).


*Figure 6. 16 Graph for Roll Angle in HITL Simulation*


*Figure 6. 17 Graph for Pitch Angle in HITL Simulation*

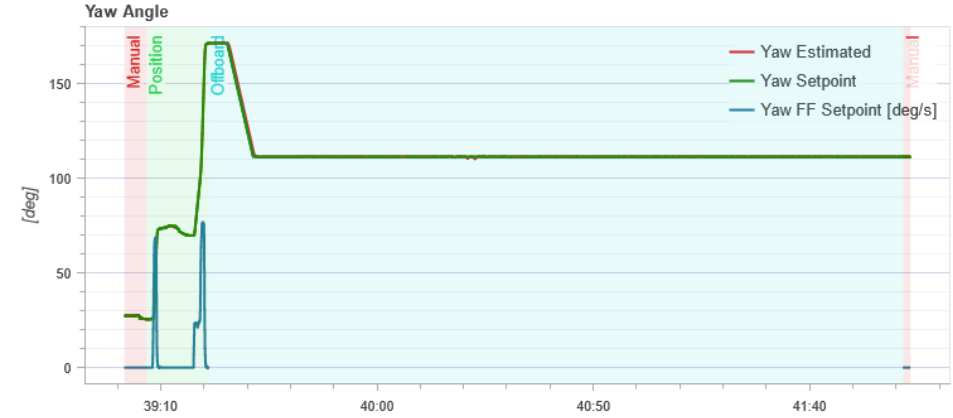



*Figure 6. 18 Graph for Yaw Angle in HITL Simulation*

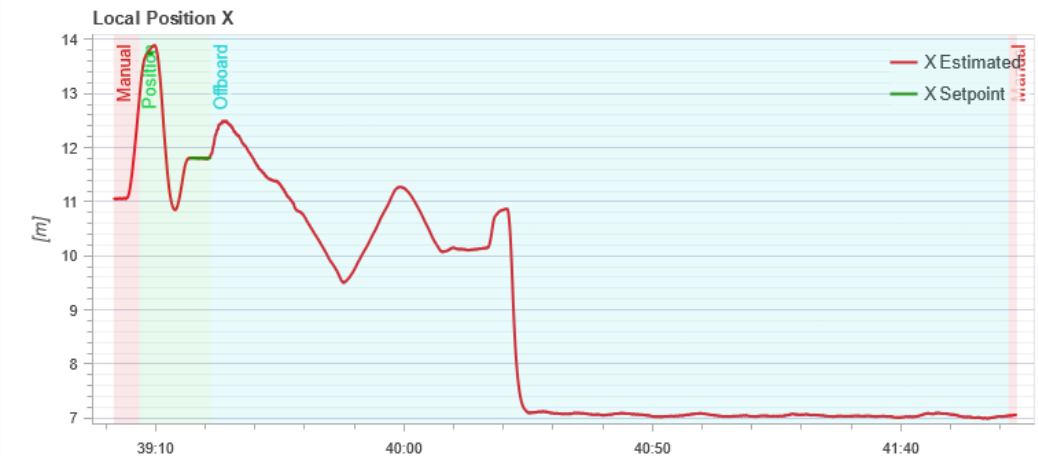



*Figure 6. 19 Graph for X-Position in HITL Simulation*

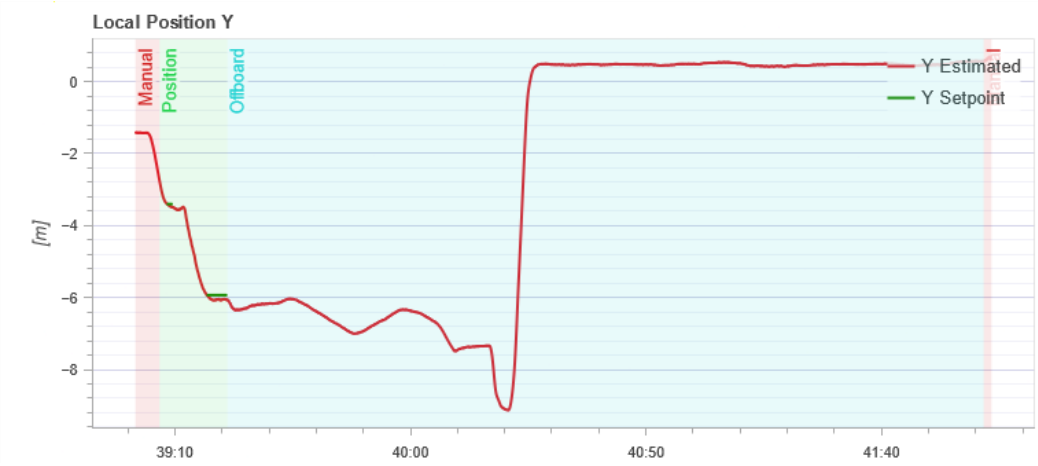



*Figure 6. 20 Graph for Y-Position in HITL Simulation*

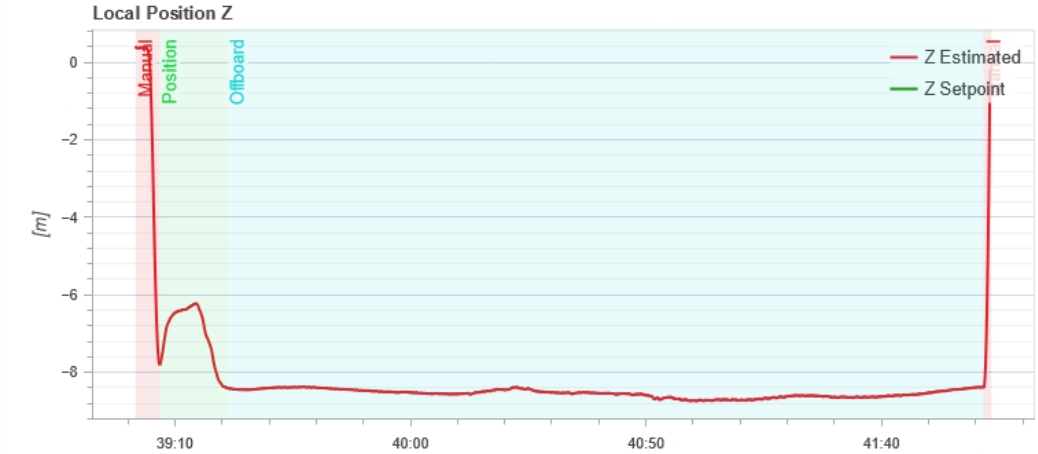



*Figure 6. 21 Graph for Z-Position in HITL Simulation*

## 6.2. Experimental Results

### 6.2.1. Overview of Hardware

After a period of research and development, following hardware revisions, the team has completed the design and assembly of the Firefighting Drone.

*Figure 6. 22 Drone after has been assembled*

Key drone specifications:

| Specifications | Values |
|---|---|
| Weight | 4.2 kg |
| Dimensions | Height: 647 mm |
| Maximum thrust of 4 motors | 8,2 kg |
| Diagonal distance: | 661 mm |
| Flight time | 5 minutes |

*Table 6. 1 Specifications of Firefighting Quadcopter*

### 6.2.2 Graph for parameters in real flight

The team created a small fire using a torch. The drone will rotate on the Yaw axis to locate the fire. Once detected, it will move to position the fire in the center of the frame. The torch will be moved horizontally left and right for about 2 meters to test the drone's ability to track the fire. When the fire is stabilized for a period of time, the drone will move towards the fire and drop the ball. Finally, the drone will return to its original position.

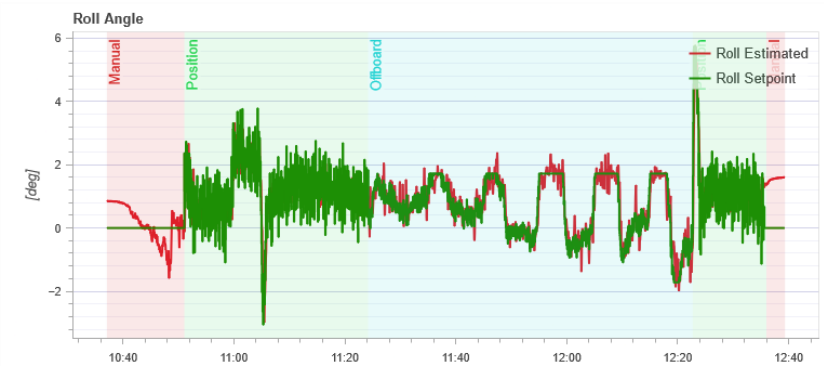



*Figure 6. 23 Roll Axis Response Curve in Real Flight*

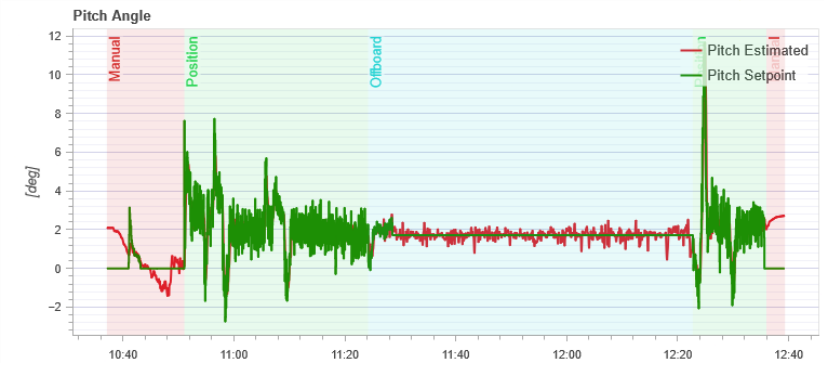



*Figure 6. 24 Pitch Axis Response Curve in Real Flight*

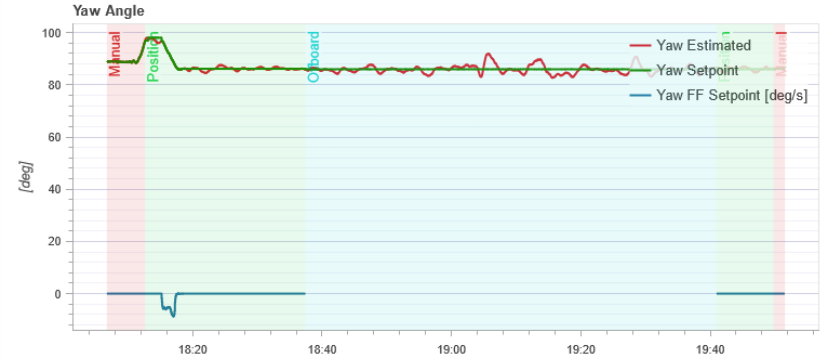



*Figure 6. 25 Yaw Axis Response Curve in Real Flight*

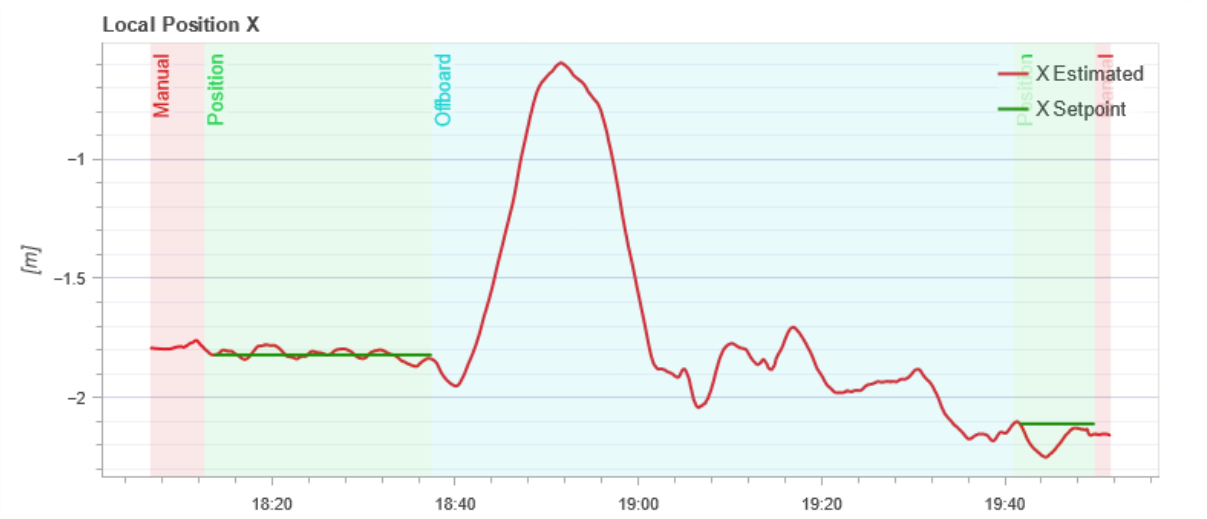



*Figure 6. 26 X-Position Response Curve in Real Flight*

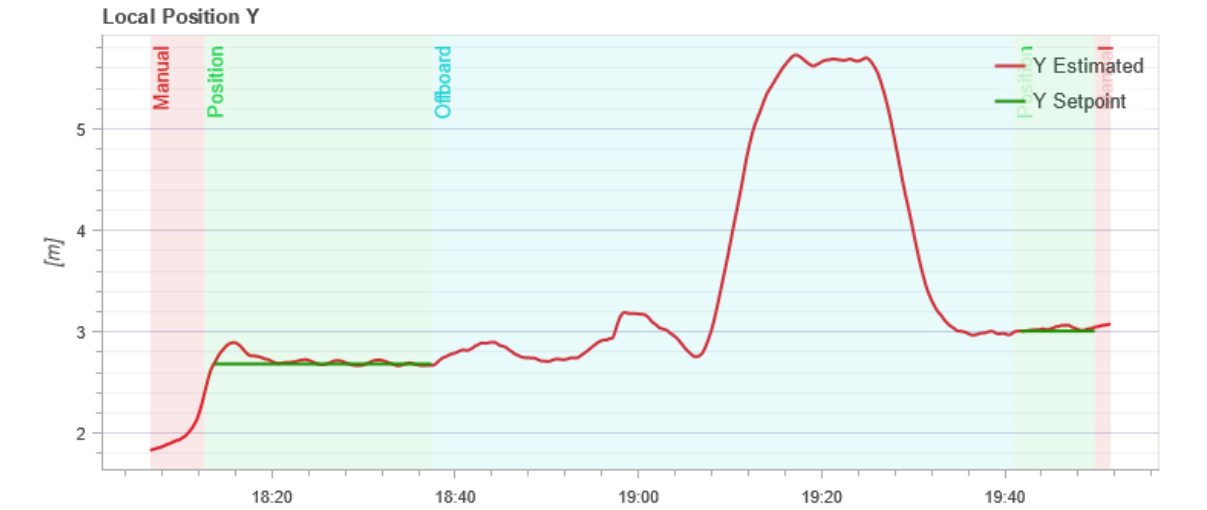



*Figure 6. 27 Y-Position Response Curve in Real Flight*

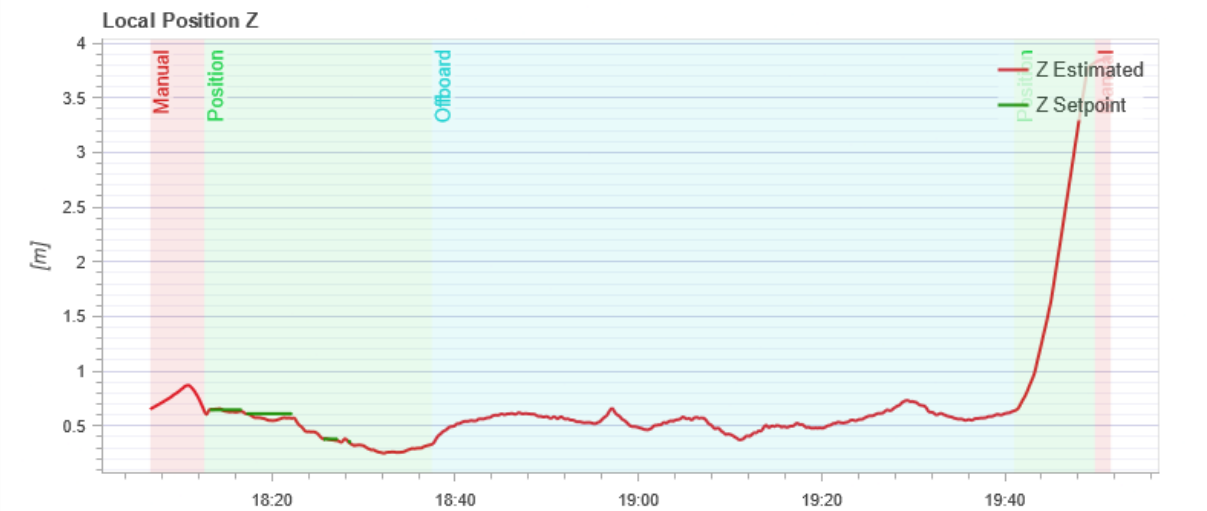



*Figure 6. 28 Z-Position Response Curve in Real Flight*

## 6.3. Conclusion

After a period of research, design, and fabrication of a fire-fighting drone utilizing a camera for fire detection and implementing MAVROS and PX4 Firmware for control:

- Due to limitations in carbon frame fabrication, the drone's primary structure was assembled using commercially available components. However, the gripper system was entirely designed and fabricated in-house. This gripper allows for flexible grasping and releasing of objects. The overall system exhibits minimal vibration during flight.

- The drone utilizes a combination of camera-based image processing and Jetson-powered algorithms to effectively detect and locate fires. Once a fire zone is identified, the drone autonomously initiates a fire suppression strategy the integrated GPS system ensures accurate positioning of the drone throughout the operation, enabling precise control and execution of the fire suppression maneuver.

- Despite the notable achievements, the project acknowledges certain limitations that could be addressed in future endeavors:
    - The current fire detection system may experience delays, potentially affecting the accuracy of fire location identification
    - The drone's flight duration and payload capacity are currently constrained, restricting its operational capabilities.
    - the drone's GPS signal is susceptible to interference from telemetry and Wi-Fi signals, potentially causing signal loss and hindering navigation.

## 6.4. Development Orientation

Given the limitations mentioned above, in order to optimize performance and withstand the harsh conditions of high-temperature fire environments, the team needs to further research and improve the following shortcomings:

- The physical structure of firefighting drones should be reinforced to withstand the harsh conditions of a fire environment. This includes using heat-resistant materials for the drone's body and components

- Extended periods of operation. To address this, drones should be equipped with high-capacity batteries that can provide sufficient power for prolonged firefighting activities.
- Fine-tuning the algorithm and the appropriate PID parameters to ensure the Drone operates stably and processes faster.

## REFERENCES

*English*


[1]. B. Bilgin et al., "**Modeling and Analysis of Electric Motors: State-of-the-Art Review," in IEEE Transactions on Transportation Electrification**, vol. 5, no. 3, pp. 602-617, Sept. 2019, doi: 10.1109/TTE.2019.2931123.

[2]. Burchan Aydin, Emre Selvi , Jian Tao and Michael J. Starek, **Use of Fire-Extinguishing Balls for a Concept System of Drone-Assisted Wildfire Fighting**, https://doi.org/10.3390/drones3010017

[3]. David Perez-Saura Miguel Fernandez-Cortizas Rafael Perez-Segui, Pedro Arias-Perez, Pascual Campoy, **Urban Firefighting Drones: Precise Throwing from UAV**, Journal of Intelligent & Robotic Systems (2023) 108:66, https://doi.org/10.1007/s10846-023-01883-6.

[4]. Gabriele Barbaraci, Steve Stefanidis, **PWM signal generation for a quad-rotor with variable geometry arms**, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2020.107345.

[5]. H. Yin, Q. Wang and C. Sun, **"Position and attitude tracking control for a quadrotor UAV via double-loop controller**," 2017 29th Chinese Control and Decision Conference (CCDC), Chongqing, China, 2017, pp. 5358-5363, doi: 10.1109/CCDC.2017.7979449.

[6]. Hettiarachchi, Maheth. (2021). **Evaluation and optimization of the aerodynamics performance of a drone design**.

[7]. Mardan M, Esfandiari M, Sepehri N, **Attitude and position controller design and implementation for a quadrotor, International Journal of Advanced Robotic Systems**. 2017;14(3), doi:10.1177/1729881417709242.

[8]. Mihai-Alin STAMATE1, Adrian-Florin NICOLESCU2, Cristina PUPAZĂ, **MATHEMATICAL MODEL OF A MULTI-ROTOR DRONE PROTOTYPE CALCULATION ALGORITHM FOR MOTOR SELECTION**, ISSN 2067-9238.

[9]. R. B. Kalamkar1*, M. C. Ahire1, P. A. Ghadge1, S. A. Dhenge2 and M. S. Anarase, **Drone and its Applications in Agriculture**, *International Journal of Current Microbiology and Applied Sciences, ISSN: 2319-7706* Volume 9 Number 6 (2020).

[10]. Redouane Dargham, Adil Sayouti, Hicham Medromi**, Euler and Quaternion Parameterization in VTOL UAV Dynamics with Test Model Efficiency**, ISSN: 2249-0868.

[11]. Şahin Yıldırım, Nihat Çabuk, Veli Bakırcıoğlu, **Design and trajectory control of universal drone system**, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2019.07.062.

[12]. Yang Hyunsoo, Lee Yongseok, Jeon, Sang Yun, Lee Dongjun, **Multi-rotor drone tutorial: systems, mechanics, control and state estimation**, ISSN 1861-2784, https://doi.org/10.1007/s11370-017-0224-y.

[13]. Zdobyslaw Goraj, **DYNAMICS OF A HIGH ALTITUDE LONG ENDURANCE UAV**, Warsaw University of Technology, Warsaw, Poland, ICAS 2000 CONGRESS.

## OTHER REFERENCES

*[14]*. ArduCam, **16MP IMX519**, https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/16MP-IMX519/, 07/2024.

*[15]*. BetaFPV, **ELRS Nano Receiver**, betafpv.com/products/elrs-nano-receiver, 07/2024

*[16]*. Drone Nodes, **Power Distribution Board PDB**, https://dronenodes.com/pdb-power-distribution-board/, 07/2024.

*[17]*. Gazebo, https://gazebosim.org/home, 07/2024.

*[18]*. Genstatuu, **What is difference between X-Frame and H-Frame**?, https://www.genstattu.com/blog/what-is-difference-between-x-frame-and-h-frame, 07/2024.

*[19]*. Gstreamer, **What is GStreamer**? https://gstreamer.freedesktop.org/, 07/2024.

*[20]*. HobbyWing, **Skywalker ESC**, https://www.hobbywingdirect.com/products/skywalkeresc?variant=40777942073459 , 07/2024.

*[21]*. Holybro, **M10GPS**, https://holybro.com/products/m10-gps, 07/2024.

*[22]*. NVIDIA, **NVIDIA Jetson Nano**, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/, 07/2024*[*.

*[23]*. PX4 AutoPilot, **Open Source Autopilot for Drone Developers**, https://px4.io/, 07/2024.

*[24]*. PX4 Guide, **Hardware in the Loop Simulation (HITL)**, https://docs.px4.io/main/en/simulation/hitl.html. 07/2024*]*.

*[25]*. PX4 Guide, **Hardware in the Loop Simulation (HITL)**, https://docs.px4.io/main/en/simulation/hitl.html#simulator-specific-setup, 07/2024*[1*.

*[26]*. PX4 Guide, **Holybro Pixhawk 6C**, https://docs.px4.io/main/en/flight_controller/pixhawk6c.html, 07/2024.

*[27]*. PX4 Guide, **MAVLink Peripherals (GCS/OSD/Companion),** https://docs.px4.io/main/en/peripherals/mavlink_peripherals.html#telem2, 07/2024*[15]*.

*[28]*. QGC Guide, **QGroundControl User Guide**, https://docs.qgroundcontrol.com/master/en/qgc-user-guide/index.html, 07/2024.

*[29]*. QuanPhongRC, **GNB 5500mAh 4S 70C 14.8V LiPo Battery**, https://quanphongrc.vn/ct/may-bay-dieu-khien/4719/gnb-5500mah-4s-70c-14-8v-lipo-battery.html, 07/2024.

*[30]*. RadioMaster, **Pocket Radio Controller (M2)**, https://www.radiomasterrc.com/products/pocket-radio-controller-m2, 07/2024.

*[31]*. ROS, **ROS - Robot Operating System**, https://www.ros.org/, 07/2024.

*[32]*. SunSkyUSA, **SunnySky X2814 Brushless Motors**, https://sunnyskyusa.com/products/sunnysky-x2814-brushless-motors, 07/2024.

# APPENDIX

$x_e$: the inertial position (North)of the quadcopter along $i^e$ in $F^e$

$y_e$: the inertial position (East) of the quacopter along $j^e$ in $F^e$

h: the inertial position (Down) of the quacopter along $k^e$ in $F^e$

u: the body frame velocity measured along $i^b$ in $F^b$

v: the body frame velocity measured along $j^b$ in $F^b$

ω: the body frame velocity measured along $k^b$ in $F^b$

ϕ: the roll angle defined with respect $F^{v2}$

θ: the pitch angle defined with respect $F^{v1}$

ψ: the yaw angle defined with respect $F^v$

p: the roll rate measured along $i^b$ in $F^b$

q: the pitch rate measured along $j^b$ in $F^b$

r: the yaw rate measured along $k^b$ in $F^b$

Additionally:

[ϕ, θ, ψ]: attitude angle

h: altitude

[x, y] ᛫ horizontal position