الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة غرداية

University of Ghardaïa

N° d'ordre :

N° de série :

كلية العلوم والتكنولوجيا

Faculty of Science and Technology

قسم الآلية والكهروميكانيك

Department of Automation and Electromechanics

Thesis presented for the graduation of

MASTER'S DEGREE

Field: Science and Technology

Die: Automatic

Specialty: Automatic and Systems

Theme

PolliDrone: *Smart Drone for Precision Pollination of Date Palms*

Presented by

Belaid Younes

Moussalmal Youcef

Publicly defended on: 15/12/2025 In front of the jury:

BEKKAR Belgacem	MCA	Univ. Ghardaïa	President
Biteur Kada	MCB	Univ. Ghardaïa	Examiner
HACENE Nacer	MCA	Univ. Ghardaïa	Supervisor

Academic year: 2024/2025

Acknowledgements

First and foremost, we express our deep gratitude to Almighty God for granting us the strength, patience, and perseverance to complete this humble work.

We would like to extend our sincere thanks to our families for their constant support, encouragement, and understanding throughout all these years of study. Their sacrifices and unfailing confidence in us have been a continuous source of motivation.

Our special thanks go to our supervisor, **Dr. Nacer HACENE**, for his guidance, availability, and constructive remarks during the preparation of this thesis. His scientific rigor, valuable advice, and patience have greatly contributed to the quality of this work.

We also address our gratitude to the members of the jury for having accepted to evaluate this thesis, and for the time and effort they devoted to reviewing and assessing our work

Dedication

To our beloved parents,
for their endless love, sacrifices, and unwavering support,
who believed in us even when we doubted ourselves.

To our families,
who have been a constant source of strength, patience, and encouragement
through every step of this journey.

To our teachers and mentors,
who guided us, inspired our curiosity,
and helped shape our path as engineers.

To our friends and colleagues,
for their companionship, motivation, and shared moments of struggle and success.

This work is dedicated to all of you.

Abstract

This thesis presents the development of an autonomous quadcopter designed specifically for date-palm pollination. The work integrates a complete nonlinear dynamic model, a cascaded control architecture for stable low-speed flight, and a lightweight autonomous navigation strategy adapted to irregular oasis environments. A YOLOv8-based perception system enables real-time detection of palm flowers, while a custom hardware platform—combining an ESP32 flight controller, Raspberry Pi Zero 2 W, and a pollen pump—forms the operational prototype. Simulation and preliminary experimental results show that the proposed system can achieve precise hovering, reliable flower detection, and safe maneuvering near palm crowns. The findings demonstrate the technical feasibility of UAV-based pollination and highlight its potential to reduce labor demands, improve safety, and enhance consistency in date-palm agriculture.

keywords: Autonomous Quadcopter, Date-Palm Pollination, YOLOv8 Perception, PID Control, Autonomous Navigation

Résumé

Ce mémoire présente le développement d'un quadricoptère autonome dédié à la pollinisation des palmiers-dattiers. Le travail inclut l'établissement d'un modèle dynamique non linéaire complet, la conception d'une architecture de commande en cascade assurant un vol stable à basse vitesse, ainsi qu'une stratégie de navigation autonome légère adaptée aux environnements irréguliers des oasis. Un système de perception basé sur YOLOv8 permet la détection en temps réel des inflorescences, tandis qu'une plateforme matérielle personnalisée—associant un contrôleur de vol ESP32, un Raspberry Pi Zero 2 W et une pompe de dispersion —constitue le prototype fonctionnel. Les résultats de simulation et les premiers essais confirment la capacité du système à maintenir un vol précis, détecter de manière fiable les fleurs et se déplacer en toute sécurité autour des couronnes de palmiers. L'étude démontre la faisabilité technique d'une pollinisation automatisée par drone et son potentiel pour réduire la main-d'œuvre, améliorer la sécurité et uniformiser la qualité de pollinisation.

Mots-clés : Quadricoptère autonome, pollinisation des palmiers dattiers, perception YOLOv8, contrôle PID, navigation autonome

ملخص

تقدّم هذه المذكرة تطوير طائرة رباعية المراوح تعمل بشكل ذاتي بهدف تلقيح نخيل التمر. يشمل العمل بناء نموذج ديناميكي لاخطّي كامل، وتصميم هيكل تحكم متسلسل يضمن ثبات الطائرة عند السرعات المنخفضة، إضافة إلى استراتيجية ملاحة ذاتية خفيفة تتناسب مع الترتيب غير المنتظم لأشجار النخيل في الواحات. يعتمد النظام على رؤية حاسوبية قائمة على YOLOv8 للكشف الفوري عن العراجين، بينما يجمع النموذج التجريبي بين متحكم طيران من نوع ESP32 وحاسوب Raspberry Pi Zero 2 W ومضخة نثر لقاح. أظهرت نتائج المحاكاة والاختبارات الأولية قدرة النظام على التحويم بدقة، واكتشاف الأزهار بشكل موثوق، والتنقل بأمان بالقرب من قمة النخلة. وتبيّن الدراسة أن التلقيح باستخدام الطائرات بدون طيار حلّ قابل للتطبيق، مع إمكانية تقليل الاعتماد على اليد العاملة وتحسين السلامة وجودة التلقيح في زراعة نخيل التمر.

الكلمات المفتاحية: طائرة رباعية المراوح ذاتية القيادة، تلقيح أشجار النخيل، إدراك YOLOv8، التحكم PID، الملاحة الذاتية

Table of Contents

List of Figures

List of Tables

List of Symbols

Core System & Robotics

- **UAV** – *Unmanned Aerial Vehicle*: A generic term for any aircraft without an onboard human pilot (your quadcopter platform).
- **UAS** – *Unmanned Aerial System*: The UAV plus all supporting elements: ground control, communication links, sensors, etc.
- **VTOL** – *Vertical Take-Off and Landing*: Ability to take off and land vertically (like quadcopters, as opposed to fixed-wing aircraft).
- **DOF** – *Degree of Freedom*: Independent motion axes of the drone: 3 translational (x, y, z) + 3 rotational (roll, pitch, yaw).
- **IMU** – *Inertial Measurement Unit*: Sensor module combining accelerometer, gyroscope (and often magnetometer) used for attitude and motion estimation.
- **GPS** – *Global Positioning System*: Satellite-based positioning system used to estimate the drone's global position.
- **RTK** – *Real-Time Kinematic*: High-precision GPS technique using a base–rover correction link to achieve centimeter-level position accuracy.
- **GNSS** – *Global Navigation Satellite System*: General term covering GPS, GLONASS, Galileo, etc., used for satellite-based positioning.

Control & Dynamics

- **PID** – *Proportional–Integral–Derivative* (controller): Classical feedback controller used in your cascaded attitude, rate and altitude loops.
- **PD** – *Proportional–Derivative* (controller): PID variant without integral action, often used for fast inner angular-rate loops.
- **PI** – *Proportional–Integral* (controller): Used where steady-state error must be removed (e.g. altitude or velocity control).

- **FLC** – *Fuzzy Logic Control*: Intelligent control using linguistic rules instead of explicit mathematical models.
- **NNC** – *Neural Network Control*: Control strategy where a neural network learns the mapping from states/errors to control actions.
- **NN–PID** – *Neural Network–Based PID Tuning*: Hybrid scheme where a neural network adapts PID gains online rather than generating torque directly.
- **FSM** – *Finite State Machine*: High-level logic structure used to sequence mission states (Idle, Takeoff, Row Scan, Pollination, RTL, etc.).

Propulsion, Power & Electronics

- **BLDC** – *Brushless Direct Current (Motor)*: High-efficiency electric motor type used to drive each propeller of the quadcopter.
- **ESC** – *Electronic Speed Controller*: Power electronics module that converts battery DC into three-phase AC for the BLDC motor and controls its speed.
- **PWM** – *Pulse-Width Modulation*: Technique used by the ESC and microcontroller to encode command magnitude in the duty cycle of a digital signal.
- **PDB** – *Power Distribution Board*: PCB that distributes battery voltage to ESCs and regulators and often includes fusing and current sensing.
- **Li-Po** – *Lithium-Polymer (Battery)*: Rechargeable battery chemistry providing high energy density for the drone.
- **DC** – *Direct Current*: Electric current that flows in one direction; used for the battery bus and electronics.
- **AC** – *Alternating Current*: Time-varying current used internally inside BLDC stator windings (generated by the ESC).
- **EMF** – *Electromotive Force*: General term for induced voltage; you use *back-EMF* to describe the voltage induced in BLDC windings by rotor motion.

- **UART** – *Universal Asynchronous Receiver–Transmitter*: Serial communication interface used between ESP32 and Raspberry Pi.

Sensors & Perception

- **ToF** – *Time-of-Flight (Sensor)*: Distance measurement principle used by the VL53L1X: distance inferred from the travel time of emitted light.
- **VL53L1X** – *ST VL53L1X ToF Sensor*: Specific short-range laser distance sensor used for obstacle detection around the drone.
- **BMP388** – *Bosch BMP388 Barometric Sensor*: High-resolution barometer used for altitude estimation.
- **IMU (ICM-20948)** – *Inertial Measurement Unit ICM-20948*: 9-axis sensor (accelerometer, gyroscope, magnetometer) used for attitude and heading estimation.
- **FOV** – *Field of View*: Angular extent of the image captured by the camera; constrains how much of the canopy is visible at a given altitude.

Artificial Intelligence & Computer Vision

- **AI** – *Artificial Intelligence*: General term for algorithms that mimic intelligent decision-making, including vision and learning models.
- **CV** – *Computer Vision*: Subfield of AI that processes and interprets images and video (flower detection, tree localization, etc.).
- **CNN** – *Convolutional Neural Network*: Deep-learning architecture specialized for images; backbone of your detection pipeline.
- **YOLO** – *You Only Look Once*: Real-time, single-stage object detection family of models; you use **YOLOv8** to detect flowers and palm crowns.
- **YOLOv8** – *You Only Look Once, Version 8*: Specific modern YOLO architecture used as detector; anchor-free, fast enough for on-board inference.
- **mAP** – *mean Average Precision*: Standard object-detection metric combining precision–recall performance across thresholds.

Navigation, Planning & Autonomy

- **CPP** – *Coverage Path Planning*: Planning strategy ensuring full coverage of the orchard (e.g. boustrophedon / lawnmower patterns).
- **RTL** – *Return to Launch*: Safety mode where the UAV autonomously navigates back to the take-off point and lands.
- **GNF / NED** (if you use it) – *Global Navigation Frame / North–East–Down*: Inertial reference frame used to express positions and velocities in the world.

Agriculture & Remote Sensing

- **NDVI** – *Normalized Difference Vegetation Index*: Vegetation index computed from multispectral images to assess crop health; mentioned in the UAV-in-agriculture context.
- **FAO** – *Food and Agriculture Organization (of the United Nations)*: Source for global statistics on date-palm production and pollination.
- **IDC** – *International Data Corporation*: Market research firm providing growth and market size data for agricultural UAVs.

Computing & Communication

- **RPi** – *Raspberry Pi*: Family of single-board computers; you specifically use the **Raspberry Pi Zero 2 W:** as the high-level companion computer.
- **SBC** – *Single Board Computer*: Complete computer on one PCB (e.g. Raspberry Pi) used for AI, planning, and logging.
- **Wi-Fi** – *Wireless Fidelity*: Wireless networking standard used for communication between drone and ground app (via RPi hotspot).
- **RC** – *Remote Control*: Manual radio link (FlySky transmitter + receiver) used for manual mode and safety override.

General Introduction

General Introduction

1. Background and Motivation

Date palm (Phoenix dactylifera) is one of the most ancient domesticated fruit trees, with a well-established history in the agricultural tradition of the Middle East, North Africa, and other dry parts of the world. Not only is it an important source of food, but also an economic lifeline for various rural populations. Date palm yield and quality are largely dependent on successful pollination, a biological process that enables fertilization and subsequent fruit set to occur.

Pollination of date palm has been done manually by experienced laborers who ascend the trees and dust the female flowers with pollen. Although successful, this practice has many drawbacks. It is time-consuming, needing a high volume of labor during a limited pollination period, and it subjects workers to unsafe conditions. Manual pollination also leads to irregular pollen distribution that adversely affects fruit set percentages and overall yield quality.

Over the past several years, precision agriculture has revolutionized conventional farming through the integration of sophisticated technologies like remote sensing, robotics, and artificial intelligence. Among them, unmanned aerial vehicles (UAVs), also known as drones, have become very popular because they can reach hard-to-access areas and perform complicated farming tasks like spraying, seeding, and crop monitoring [6].

The combination of AI and UAVs facilitates autonomous flight, real-time decision-making, and task accuracy. In the pollination of date palms, the method can break the limitations of manual practices by automating the processes of flower detection, navigation, and pollen delivery. AI-based vision system autonomous quadcopters can achieve even pollination, save labor costs, and enhance productivity.

2. Problem Definition

Hand pollination of date palms confronts a number of interrelated challenges that affect productivity and sustainability:

- **Labor Shortage and Costs:** Pollination requires a huge labor force of trained personnel, who may not be available or are costly at peak times. This reliance raises operating expenses and exposes production to timely delays [5].

- **Time Sensitivity:** Date palm flowers are sensitive to pollination for a short duration, even for a few days in some cases. Late or inefficient pollination may result in poor fruit set and yield losses [4].
- **Environmental and Safety Hazards:** Manual pollination puts workers at risk of falls, heat stroke, and other work-related hazards associated with climbing tall trees in adverse environments.
- **Variable Pollination Quality:** Low or uneven pollen coverage leads to lower fruit quality and yield, decreasing farm profitability [3].

3. Thesis Organization

The subsequent chapters constitute the foundation of this thesis, with each chapter covering essential elements of the AI-powered date palm pollination via a quadcopter drone study:

- **Chapter I – Quadcopter Drone Overview in Agriculture:** Introduces UAV applications within precision agriculture and their relevance to pollination tasks and highlights the limitations of manual date-palm pollination and the motivation for UAV-based solutions.
- **Chapter II – Mathematical Modeling of the Quadcopter Drone:** Derives the nonlinear translational and rotational dynamics governing quadcopter motion and defines the coordinate frames, aerodynamic forces, and kinematic transformations and establishes the mathematical basis required for subsequent control system.
- **Chapter III – Design of Control System & Simulation:** Introduces a cascaded PID architecture for stabilizing and regulating the quadcopter's motion, formulating rate, attitude, velocity, and position loops using model-based reasoning. Finally Evaluates controller behavior through simulation of the full nonlinear dynamic model.
- **Chapter IV – Autonomous Navigation:** Presents the navigation strategy enabling tree-to-tree movement in irregular oasis layouts, explaining local positioning and approach behavior near palm crowns. And formalizes the mission flow governing autonomous pollination operations.
- **Chapter V – Visual Detection of Flowers and Palm Trees:** Introduces the training, and performance evaluation of YOLO-based computer-vision framework to detect palm crowns and flower clusters.
- **Chapter VI – Hardware Development and System Integration:** Summarizes the integration of computation, actuation subsystems, structural assembly, sensor placement, and subsystem interaction into a unified platform.

Chapter I

I. Quadcopter Drone Overview in Agriculture

I.1 Introduction

The evolution of technology in the recent decades has propelled the evolution of agriculture as a high-tech business, or precision agriculture. UAVs have become a key tool in these advances, with a broad range of applications that simplify farm operations, optimize resource utilization, and boost agricultural production. UAVs offer the agility and accuracy required to address intricate agricultural challenges, ranging from crop health monitoring to performing precision interventions such as pollination.

Date palms (Phoenix dactylifera), a key crop in arid areas, are highly dependent on artificial pollination because they are dioecious. Manual pollination, while traditional, is increasingly impractical due to rising labor costs and limited availability during critical periods. UAVs provide a viable alternative by facilitating targeted pollen delivery with less human intervention and danger.

UAV technologies are presented in this chapter, their types are discussed, and the reason quadcopters have been chosen for date palm pollination is explained with proper statistics and research results.

I.2 Applications of UAVs in Precision Agriculture

Precision agriculture is a farm management technique based on technology that optimizes agricultural inputs such as water, fertilizers, and pesticides to achieve maximum production and sustainability. UAVs are now an integral part of this practice, offering capabilities that address the problems faced by modern agriculture.

I.2.1 Evolution of UAV Technology in Agriculture

The introduction of UAVs in agriculture has transformed traditional practices. Initial agricultural UAVs were simple devices used for imaging and monitoring. However, rapid advancements in robotics, AI, and sensor technology have dramatically expanded their uses. The utilization of AI-driven analytics, multispectral imaging, and autonomous flight planning has made UAVs a necessity for precision farming [6].

As per the 2022 report by the International Data Corporation (IDC), the market for agricultural unmanned aerial vehicles (UAVs) experienced a 24.5% year-over-year growth rate, while the forecast is anticipating over 6 billion USD in valuation by the year 2026. The expansion mirrors the rising demand for UAV-based technology to solve enormous challenges in agriculture [6].

I.2.2 Critical UAV Applications in Agriculture

I.2.2.1 Crop Health Monitoring

UAVs equipped with multispectral and thermal sensors provide an accurate glimpse into plant health. By interpreting vegetation indexes like the Normalized Difference Vegetation Index (NDVI), farmers can identify stressed, diseased, or nutrient-deficient areas.

A study in Iowa-USA showed that the use of UAV-based NDVI imaging enabled a 12% increase in corn yield through the allowance of timely intervention against pest infestation [7].

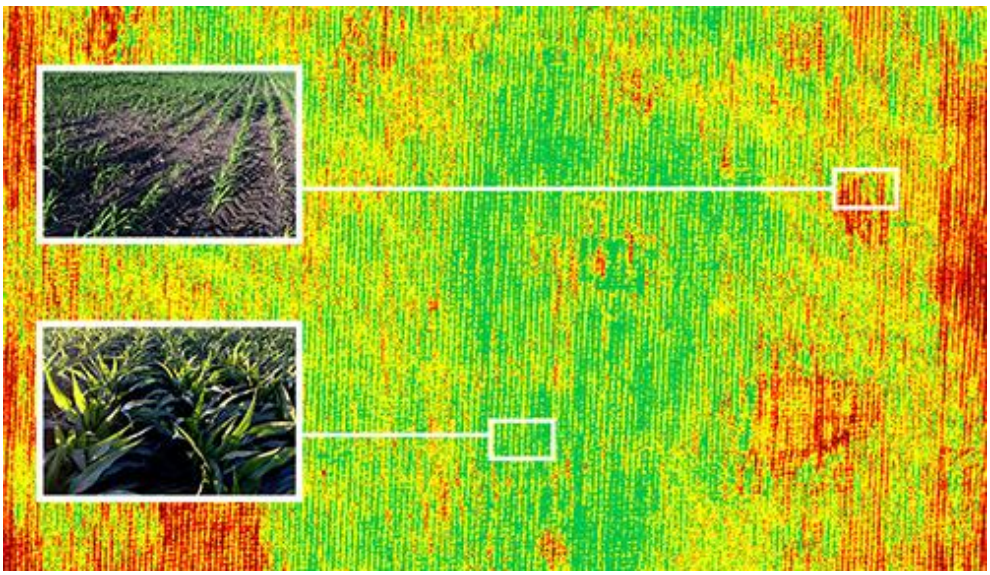

Figure I.1: Monitoring Drone Camera Scan Data [25]

I.2.2.2 Soil and Field Analysis

Prior to seeding, UAVs can generate 3D soil maps to identify soil characteristics such as moisture levels, compaction, and soil nutrients. These maps can be utilized by farmers to optimize seed placement and irrigation techniques.

Statistical Fact: UAV soil analysis has been shown to reduce irrigation water usage by 20–30%, according to research published in Agricultural Water Management [7], [8].

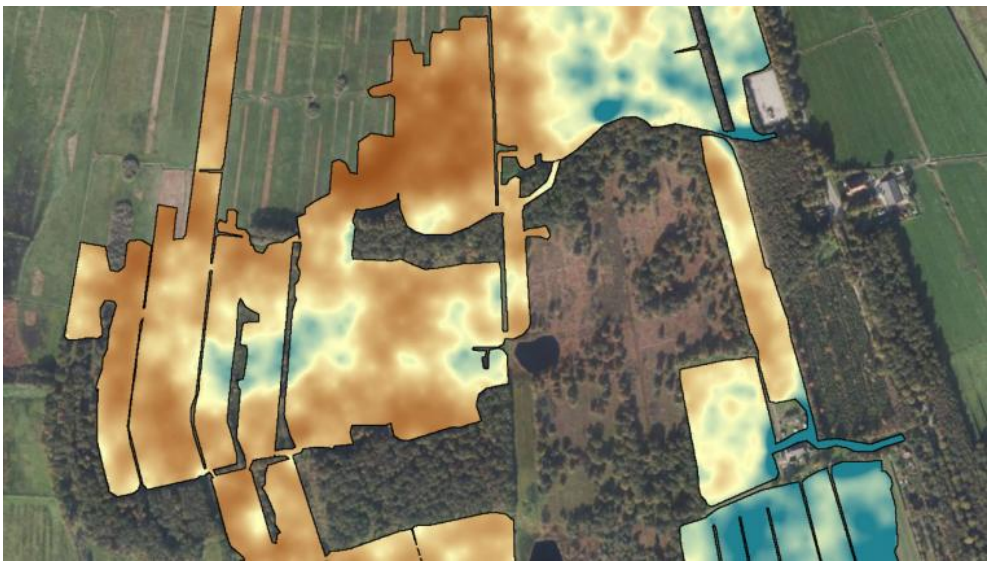

Figure I.2: Soil Mapping with Drones [26]

I.2.2.3 Irrigation Management

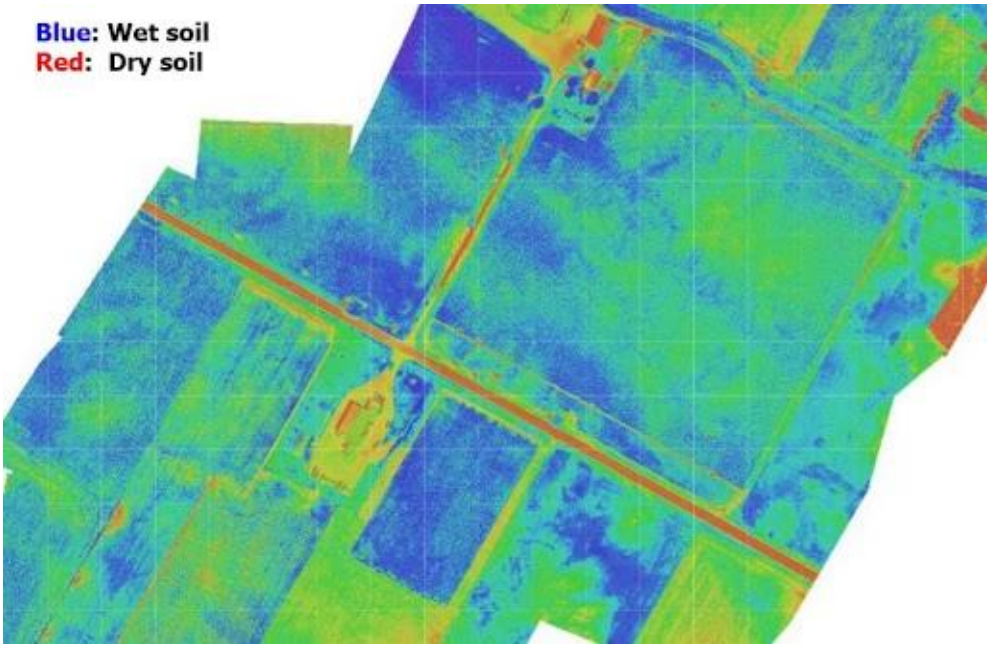

Figure I.3: Soil Moisture Levels Drone Data [25]

UAV-mounted thermal sensors can identify areas of water stress in plants, which can lead to better irrigation strategies.

Impact Example: In vineyards, UAVs reduce water usage by 25% without affecting grape quality, says the University of California, Davis [7], [8].

I.3 Pollination

Pollination is a foundation of farming productivity, enabling the reproduction of flowering plants and having a direct impact on crop yields. For a majority of crops, successful pollination establishes the quality, quantity, and commercial value of the harvest. In farming, pollination may take place naturally by wind, water, or insects, but for some crops, artificial assistance is necessary to achieve consistent and effective fertilization.

I.3.1 Pollination in Agriculture

Pollination directly contributes to approximately 35% of global food production and is essential for the yield of over 87 crop species worldwide. (Journal of Pollination Ecology, 2020). Its significance reaches beyond food security, maintaining biodiversity and ecosystem health [9].

- **Economic Impact:** High-value commodities like almonds, apples, cherries, and melons rely to a great extent on successful pollination. The economic value of pollinator services worldwide has been put at more than $200 billion per annum (Global Pollinator Report, 2021) [1].
- **Dependence on Natural Factors:** Conventional pollination depends on either wind or insects such as bees to move pollen among flowers. Yet, these strategies are progressively untrustworthy as a result of environmental perturbations, such as climate change, habitat destruction, and the application of pesticides. For instance, bee populations have fallen by 40% in North America in the last ten years (FAO Pollinator Initiative, 2019) [2].
- **Artificial Pollination:** To overcome the drawbacks of natural pollination, artificial techniques like manual pollination, mechanized platforms, and new technologies such as UAV-based systems have been introduced. These techniques are designed to increase precision, efficiency, and reliability.

I.3.2 Pollination in Date Palms

Date palms (Phoenix dactylifera) are a key crop in semi-arid and arid areas, playing an important role in food security and economies in South Asia, North Africa, and the Middle East. More than 100 million date palms are grown worldwide, with production of more than 9 million tons per year (FAO Statistical Yearbook, 2022).

I.3.2.1 Methods of pollination

Manual Pollination:

- **Process:** Pollen is gathered from male flowers and manually deposited onto female flowers by skilled workers using the help of poles, ladders, or by climbing onto trees [3].
- **Advantages:** Very accurate when performed by experienced personnel.
- **Challenges:**
  - **Labor Intensity:** Hand pollination is extremely labor and time-intensive, with one worker managing to pollinate only a few trees per day [3].
  - **Costs:** Labor expenses can reach up to $250 per hectare [3].
  - **Safety Hazards:** Laborers are at risk of falling and accidents during the climbing of high trees.
  - **Timing Problems:** Coordinating sufficient labor within the brief receptivity period of 3–5 days is generally challenging [3].

Mechanical Pollination:

- **Process:** Pollen is distributed over female flowers using machines such as air sprayers or blowers [3].
- **Advantages:** Quick and less laborious than hand methods, treating large areas rapidly [3].
- **Challenges:**
  - **Low Precision:** Mechanized systems are incapable of targeting individual flowers and, therefore, result in irregular pollination [3].
  - **Wastage of Pollen:** 30–40% pollen wastage occurs due to inefficient delivery [3].
  - **Soil Damage:** Heavy machinery used in mechanical pollination can compact soil, reducing porosity and aeration, which negatively affects root growth and water infiltration [3].
  - **Large Operating Surface:** Mechanical pollination equipment often requires significant space for operation, which can be limiting in tightly spaced orchards or dense plantations and maneuvering large machines between trees may lead to accidental damage to crops and inefficient use of resources [3].
  - **High Initial Cost:** The machinery is expensive and cannot be afforded by small farmers [3].

- **Maintenance:** Periodic maintenance and skilled operation are necessary, which increases the operational cost [3].

I.3.3 The Need for Innovation

Addressing the challenges of traditional pollination requires innovative approaches to enhance productivity, reduce costs, and promote sustainable date palm farming. Addressing such problems requires the development of novel technologies and techniques that can overcome the limitations of traditional approaches:

- **High Accuracy Required:** Traditional pollination methods often result in significant pollen wastage due to inconsistent distribution. There is a need for innovations to provide precise targeting of blossoms with maximum fertilization success and resource conservation.
- **Increased Efficiency:** Manual pollination's labor-intensive nature and the low precision of mechanical methods equate to inefficiencies, especially in large plantations. New technologies must aim to reduce pollination time while increasing coverage without compromising quality.
- **Automation:** Automation is critical to overcoming labor shortages and ensuring consistency in operations. By automating repetitive tasks such as pollination, new systems can:
  - Eliminate the dependency on seasonal labor.
  - Provide consistent performance regardless of environmental or logistical challenges.
  - Allow farmers to focus on other critical farm management activities.
- **Cost Savings:** Higher labor expenses and the necessity for skilled workers make manual pollination economically unrealistic for the majority of growers. Innovative systems must offer cost-effective alternatives that result in the same or improved pollination outcomes.
- **Enhanced Safety:** Traditional pollination methods habitually expose workers to risks, such as falling from high trees or being injured while handling heavy machinery. A safer approach is necessary to protect the workforce without lowering productivity.
- **Environmental Sustainability:** Current methods can be disruptive to soil, erosive, and waste natural resources. New methods must focus on reducing the environmental impact by limiting pollen wastage, preserving soil health, and conserving biodiversity.

- **Adaptability to Dense Plantations:** Mechanical pollination systems traditionally used are hard to maneuver through close spaces in plantations with high tree densities. Solutions must be adaptable to offer ideal pollination even in high-density orchards.
- **Versatility Across Crops and Conditions:** Agriculture is comprised of an extensive range of crops, each having its specific pollination needs. A successful innovation must be compatible with a wide range of crops, environmental situations, and regional demands, and thus a flexible instrument for various farming systems.

I.4 Why UAVs?

Modern agriculture faces increasing pressures to enhance productivity at the cost of sustainability and labor productivity. For date palms, traditional pollination methods typically do not work due to labor needs, inefficiencies, and environmental challenges. UAVs provide a revolutionary approach by adopting precision, efficiency, and green technology to reduce these constraints. This section presents reasons why UAVs are applied in agriculture through comparative studies, case studies.

I.4.1 Key Advantage of UAVs

Unmanned Aerial Vehicles (UAVs) bring a transformative approach to agricultural practices, particularly in addressing the limitations of traditional pollination methods. Their integration into farming offers several unique advantages that enhance efficiency, safety, and sustainability:

- **High Precision and Accuracy:** UAVs integrated with advanced sensors and AI algorithms enable precise flower detection and targeted pollen delivery. Reducing pollen wastage, and ensuring optimal fertilization success while conserving resources.
- **Time Efficiency:** Automated UAV systems drastically reduce the time required for pollination. Compared to manual methods, UAVs can cover larger areas in significantly less time, making them ideal for large-scale plantations where timely pollination is critical.
- **Automation:** UAVs operate autonomously, reducing the need for direct human intervention. Through pre-programmed flight paths and automated pollen delivery systems, UAVs enhance operational efficiency and consistency.
- **Cost Savings:** By reducing reliance on manual labor, UAVs lower operational costs, offering a financially sustainable solution for farmers.

- **Safety:** UAVs eliminate the risks associated with traditional pollination methods, such as climbing tall trees or operating heavy equipment. By allowing remote operation, UAVs ensure a safer environment for agricultural workers.
- **Environmental Sustainability:**
  - Minimizing soil disturbance, which is common with ground-based machinery.
  - Reducing the ecological footprint through optimized pollen delivery and decreased resource wastage.
  - Supporting practices that promote biodiversity by replacing harmful mechanized processes.
- **Adaptability to Complex Plantations:** UAVs can navigate tight spaces and uneven terrains, making them particularly useful in densely planted orchards. Their ability to hover and maneuver efficiently ensures effective pollination even in challenging layouts.
- **Flexibility and Versatility:** UAVs can be programmed to handle various crops, each with unique pollination requirements. Their adaptability extends to different environmental conditions, making them versatile tools for diverse agricultural applications.
- **Scalability:** From small family farms to large commercial plantations, UAVs offer scalable solutions. Their deployment can be adjusted to match the size and specific needs of the farming operation.

I.4.2 Disadvantages of UAV-Based Pollination

While UAV-based pollination systems represent a massive stride in agricultural technology, they are not without limitations. Knowing the limitations is essential so we can address them to facilitate broader adoption. The following are the key limitations of UAV-based pollination systems:

- **High Initial Costs:** UAV technology is expensive to adopt. The procurement of drones, fitting them with sensors, dispensers, and navigation systems, and maintaining the machinery can be too expensive, particularly for small-scale farmers or farmers in the developing world.
  - Operators learning to operate UAV systems efficiently.
  - Hardware and software upgrading to match new emerging technologies.
  - Regular repairs and maintenance for dependable service.
- **Technical Expertise:** Operating the UAVs is not as simple as other farm equipment. It needs special coding, calibration, and diagnosing abilities:

  - Effective deployment of UAV systems necessitates comprehensive training programs for farmers and operators.
  - Expertise in the management of newer technologies such as AI, computer vision, and data analytics is generally required.
  - The lack of access to technical support and training may slow adoption in rural or underserved areas.
- **Short Flight Duration:** Battery-powered UAVs are only limited by brief operation durations, which last from 20 to 40 minutes per charge.
  - Regular recharging or battery replacement is required for extensive operations.
  - Extended missions can involve several UAVs, which raise costs and logistical intricacy.
- **Weather Dependency:** UAVs' functionality is very prone to environmental conditions, hence not as reliable in some cases.
  - Strong winds can interfere with flight stability and accuracy of pollen delivery.
  - Rain or high humidity can decrease the viability of pollen or harm UAV components.
  - Extreme temperatures can influence battery life and sensor accuracy.
- **Regulatory Barriers:** Regulatory and legal barriers to UAV usage can deter adoption and make operations more complex.
  - Obtaining permits to operate UAVs over agricultural farms.
  - Adhering to flight restrictions, e.g., altitude restrictions or no-fly zones.
  - Handling different regulations in countries or regions.
- **Environmental Considerations:** While UAVs reduce soil disturbance, their manufacture, use, and disposal are sources of their total environmental effect.
  - Manufacturing drones and batteries is an energy-consuming process.
  - Improper disposal of batteries can result in environmental pollution.
  - Energy-intensive operations increase carbon emissions in case non-renewable sources of energy are used.

I.4.3 Why Quadcopter Drones?

In selecting a UAV for agricultural activities such as pollination, the quadcopter is the most suitable based on a number of benefits.

- **Ability for Hovering:** Quadcopter drones have the ability to hover exactly in place, an essential requirement for operations such as targeted pollination and flower inspection.

- **Maneuverability:** Quadcopter drones are capable of dynamic movement such as vertical takeoff and landing (VTOL) that fixed-wing drones cannot perform. They can navigate complex terrains and reach hard-to-access areas.
- **Compact Configuration:** Quadcopter drones have a lightweight and compact configuration, making transportation and deployment easy. They are particularly useful for small- to medium-sized farms or remote settlements.
- **Ease of Control:** Sophisticated control and stabilization systems render quadcopters highly user-friendly to operators with little training. Their symmetric design makes dynamic modeling and control easier.
- **Cost-effectiveness:** Relative to one-rotor drones, quadcopters have a reduced maintenance expense and lower mechanical complexity.
- **Flexibility:** Quadcopter drones are also versatile and can be equipped with a variety of payloads ranging from pollen dispensers, multispectral cameras, to environmental sensors.

I.4.4 Case Study: UAV Pollination of Date Palms in Saudi Arabia

Saudi Arabia is among the largest producers of dates in the world, with more than 30 million date palm trees being grown every year. Conventional methods of pollination are very dependent on human labor, which is time-consuming and expensive. Apart from this, seasonal shortages of labor during peak pollination periods contribute to these challenges, leading to inefficiency and low production. To solve such challenges, Saudi agriculturalists and scientists carried out a pilot study to test the efficiency of UAV-based pollination systems [10] [11].

I.4.4.1 Performance Measures

The project compared UAV pollination and manual pollination on various parameters. The findings reflected remarkable enhancement in efficiency, reduction in cost, and increase in yield .

Metric	Manual Pollination	UAV Pollination	Improvement
Pollination Time	15 hours per hectare	4 hours per hectare	73% reduction
Labor Cost (USD/ha)	$250	$100	60% savings
Pollination Accuracy	65%	85%	+20%
Yield Increase	-	+15%	Significant

Table I.1: Comparative Performance of Manual vs. UAV-Based Pollination in Date Palms

I.4.4.2 Findings

The research identified a number of benefits of UAV-based pollination for date palm farming:

- **Efficiency Gains:** UAVs reduce pollination time by 73%, allowing farmers to cover more area within the short window of flower receptivity [11].
- **Cost Savings:** Labor cost decreased by 60%, largely attributed to the lesser reliance on skilled manual labor [11].
- **Improved Yield:** Improved accuracy in pollination resulted in a 15% increase in yield with an improvement in fruit quality and market price [11].
- **Environmental Benefits:** Precise targeting prevented wastage of pollen and minimized soil disturbance, promoting sustainable agriculture [11].

I.4.4.3 Disadvantages

Although UAV-based pollination was extremely promising, studies also identified challenges:

- **High Initial Capital:** The price of UAV systems, along with dispensers and sensors, was too high for small farmers to afford [11].
- **Technical Expertise:** Continuous operation demanded technical expertise in UAV operations, calibration, and maintenance [11].
- **Weather Sensitivity:** Hot and windy weather affected UAV operations, necessitating modification to operational schedules [11].

I.5 Summary

The development of unmanned aerial vehicles (UAVs) has had a profound impact on precision agriculture by solving issues of labor shortages, inefficiencies inherent in conventional approaches to farming, and environmental costs associated with extensive agricultural practices. The chapter has detailed the imperative role played by UAVs, in this case quadcopters, in enhancing agricultural practices, citing their importance in pollination—a fundamental mechanism for crops like date palms.

Manual and mechanical pollination methods have been used by growers for decades but are increasingly unsustainable due to costly fees, labor dependencies, and environmental hazards. UAVs offer a disruptive solution, combining automation, precision, and scalability. Leveraging advanced technologies like AI-driven vision systems, multispectral sensors, and adaptive flight controls, quadcopters can achieve targeted and effective pollination, minimize wastage of resources, and enhance quality of yield.

Yet, the chapter also acknowledges the constraints that UAV systems have, like high initial investment, technical sophistication, as well as environmental dependency. Surmounting these obstacles entails ongoing development of drone technology, formulation of regulatory structures, and implementation of training schemes on operators.

In Summary, UAVs represent a major step forward toward sustainable and productive agriculture. Their adoption in agriculture is not only a response to current problems but also a step in the direction of future agricultural systems that emphasize precision, productivity, and sustainability. The next chapters will offer more technical, operational, and practical information on the design and implementation of UAV-based pollination systems, opening the way for their broader adoption and optimization.

Chapter II

II. Mathematical Modeling of Quadcopter Drone

II.1 Introduction

Quadcopters or quadrotor drones are an emerging technology that has made forays into several domains like agriculture, surveillance, transportation, and entertainment. Their four rotors, which are symmetrically designed, give them greater control over movement and stability compared to traditional aircraft. This design aspect enables quadcopters to hover motionless in air, launch and land vertically, and perform complex maneuvers, which makes them a necessity for applications that demand flexibility and quick response.

The growing prevalence of quadcopters highlights the need to have effective mathematical models for interpreting and predicting their dynamic responses. Unlike fixed-wing vehicles, quadcopters experience an inherent instability that requires them to make constant adjustments in order to achieve stability. Mathematical modeling is the foundation for solving this problem and, by extension, enables researchers and engineers to simulate flight dynamics, implement efficient control strategies, and increase the vehicle's overall performance efficiency [12].

By facilitating a definite mathematical explanation of the forces, torques, and resulting motions, it becomes feasible to analyze the performance of the quadcopter under hovering, navigation through obstacles, or response to external disturbance such as wind. Subsequently, the findings from such analyses have an important bearing on the design of robust and optimal quadcopter systems, allowing for advances in theoretical research as well as practical applications.

II.2 Quadcopter Description and Reference Frames

II.2.1 Quadcopter Movements and Working Principle

The quadcopter's flight is achieved by the independent control of its four rotors. By varying the rotors' speeds of rotation, the quadcopter can hover in place, move forward or backward, or sideways, spin, and perform complex aerial stunts. These motions are governed by the physical principles of thrust, torque, and angular momentum, and are controlled by onboard systems using information received from an array of sensors [12].

Rotor Placement and Rotation Directions

- **Rotor Placement:**
  - The four rotors are positioned at the vertices of a square frame, equidistant from the center.
  - They are typically labeled as follows:
    - ✓ Front-left (Rotor 1)
    - ✓ Front-right (Rotor 2)
    - ✓ Rear-right (Rotor 3)
    - ✓ Rear-left (Rotor 4)
- **Rotor Rotation Directions:**
  - To maintain stability and cancel out torque, adjacent rotors rotate in opposite directions:
    - ✓ Rotor 1 (Front-left): Counterclockwise (CCW)
    - ✓ Rotor 2 (Front-right): Clockwise (CW)
    - ✓ Rotor 3 (Rear-right): Counterclockwise (CCW)
    - ✓ Rotor 4 (Rear-left): Clockwise (CW)
  - This alternating rotation pattern ensures that the net torque is balanced during hover or symmetrical movements, preventing unintended spinning.

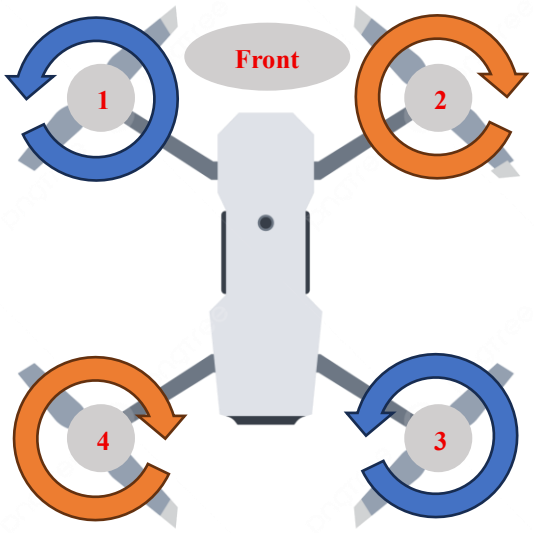

Figure II.1: Quadcopter Rotor Rotations

Movements Description

- **Hovering:**
  - All four rotors operate at identical velocities, producing a cumulative thrust that counteracts the force of gravity.
  - The quadcopter is suspended in the air.
- **Forward and Backward Movement:**
  - Changing the speeds of the front and rear rotors produces a tilt around the lateral axis.
  - Forward tilting enhances rear motors thrust, propelling the quadcopter forward.
  - Tilting backwards reduces rear motors thrust, propelling the quadcopter backwards.
- **Sideways Movement:**
  - Altering the speeds of the left and right rotors causes the quadcopter to rotate around its longitudinal axis.
  - Greater thrust on the right side tilts the quadcopter to the left, causing it to move laterally.
- **Rotational Motion:**
  - Achieved by creating a torque imbalance between the clockwise (CW) and counterclockwise (CCW) rotors.
  - Increasing the speed of the CW rotors relative to the CCW rotors causes the quadcopter to turn right, and vice versa for a left turn.
- **Ascension and Descension (Control of Thrust):**
  - Spinning all rotors faster creates more thrust than weight, then the quadcopter will ascend. Reducing rotor speeds makes the quadcopter descend.

Working Principle

- **Thrust Generation:**
  - All rotors generate thrust by spinning their propeller.
  - The level of thrust is a function of both the rotational speed and on the aerodynamic characteristics of the propeller.
- **Torque Balance:**
  - Adjacent rotors rotate in opposite directions (counterclockwise/clockwise) to counter the net torque, thereby providing stability while hovering.
  - Imbalances in rotor speed create a spinning movement.

- **Motion Control:** Directional motion (Left/Right and Forward/Backward) is achieved by tilting the frame of the quadcopter through differential thrust. Gyroscopic sensor inputs and accelerometers aid in tilt detection and stability compensation.

Modeling Assumptions

To ensure clarity and consistency in the mathematical modeling of the quadcopter drone system, the following assumptions are made:

- **Rigid Body Dynamics:**
  - The quadcopter is modeled as a rigid body with fixed mass and inertia.
  - Structural deformations and flexibilities of the frame are neglected.
- **Symmetry and Mass Distribution:**
  - The quadcopter has a symmetric configuration with equal arm lengths and identical motor properties.
  - The center of mass is located at the geometric center of the frame.
- **Aerodynamic Effects:**
  - Aerodynamic drag and lift forces are neglected due to the relatively low flight speed and small surface area.
  - Rotor-induced airflow interactions are not modeled explicitly.
- **Environmental Conditions**
  - The model assumes operation in calm weather with negligible wind disturbances.
  - Air density is considered constant and uniform.
- **Sensor and Actuator Idealization**
  - Sensors (e.g., IMU, GPS) are assumed to provide accurate and noise-free measurements.
  - Actuators (BLDC motors) respond instantaneously to control inputs without delay or saturation.
- **Motor and Propeller Modeling**
  - Each motor-propeller pair generates thrust proportional to the square of its angular velocity.
  - Motor dynamics are modeled using simplified electrical equations, assuming ideal commutation and negligible thermal effects.
- **Control Input Mapping**
  - Control inputs are mapped directly to motor speeds via an ideal ESC model.

- Sensorless control logic is assumed to detect zero-crossings accurately on the floating phase only.

II.2.2 Coordinate Systems

Coordinate systems play a primary function in explaining the position, orientation, and motion of the quadcopter. Establishing the link between the **Inertial Frame** and the **Body Frame** enables the creation of the necessary equations for simulation, control, and navigation.

Inertial Frame (O_i, x_i, y_i, z_i)

- **Definition:** A fixed, global reference frame aligned with the Earth's surface. It is often referred to as the "world frame."
- **Purpose:**
  - Provides an absolute reference point to describe the quadcopter's location and orientation.
  - Assumes a flat Earth approximation with gravity acting along the z_i-axis.
- **Axes:**
  - x_i : Aligned with a predefined horizontal direction, typically pointing to the geographic North.
  - y_i : Perpendicular to x_i , pointing East.
  - z_i: Vertical and upward, opposing the direction of gravity.

The inertial frame is crucial for defining where the quadcopter is relative to its environment.

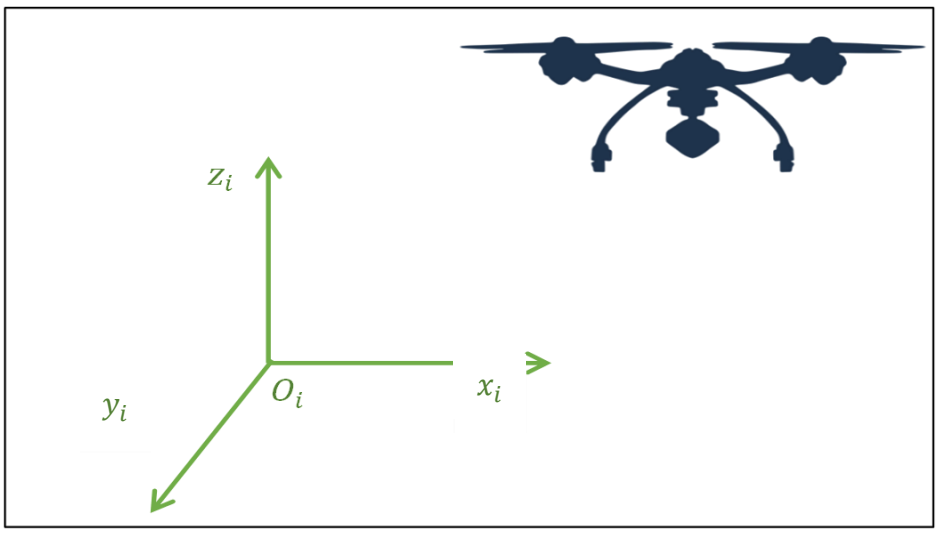

Figure II.2:Representation of Quadcopter in Inertial Frame

Body Frame (O_b, x_b, y_b, z_b)

- **Definition:** A moving reference frame fixed to the quadcopter's center of mass. It rotates and moves with the quadcopter. It simplifies the modeling of thrust and rotational forces.
- **Purpose:**
  - Represents forces, torques, and velocities in terms of the quadcopter's local orientation.
  - Useful for describing internal dynamics and sensor measurements.
- **Axes:**
  - x_b: Points forward along the quadcopter's nose.
  - y_b: Points laterally to the right side of the quadcopter.
  - z_b: Points upwards, perpendicular to the plane of the rotors.

Figure II.3:Quadcopter Body Frame

II.2.3 Euler Angles

Euler angles (ϕ, θ, ψ) are fundamental parameters for describing the orientation of a quadcopter in a three-dimensional space. They represent the rotation of the body frame of the quadcopter with respect to the inertial frame through a sequence of three angular rotations about specified axes.

What are Euler Angles?

Euler angles form a triplet of rotation parameters used to describe the orientation of a rigid body. They provide a thought model for describing the orientation of a quadcopter in motion by outlining its roll, pitch, and yaw with respect to the inertial frame of reference.

- **Roll (ϕ):**
  - Rotation about the body x_b-axis (front of the quadcopter).

- Adjusts the quadcopter's lateral tilt so that it banks either left or right.
- Positive roll ($+\phi$) tilts the right side of the quadcopter downwards.

- **Pitch (θ):**
  - Rotation about the body y_b-axis (direction towards the side of the quadcopter).
  - Alters the forward or backward tilt of the quadcopter.
  - Positive pitch ($+\theta$) tilts the nose of the quadcopter downward.
- **Yaw (ψ):**
  - Rotation about the body z_b-axis (vertical axis through the center of the quadcopter).
  - Changes the heading or directional orientation of the quadcopter.
  - Positive yaw ($+\psi$) rotates the quadcopter clockwise from the top view.

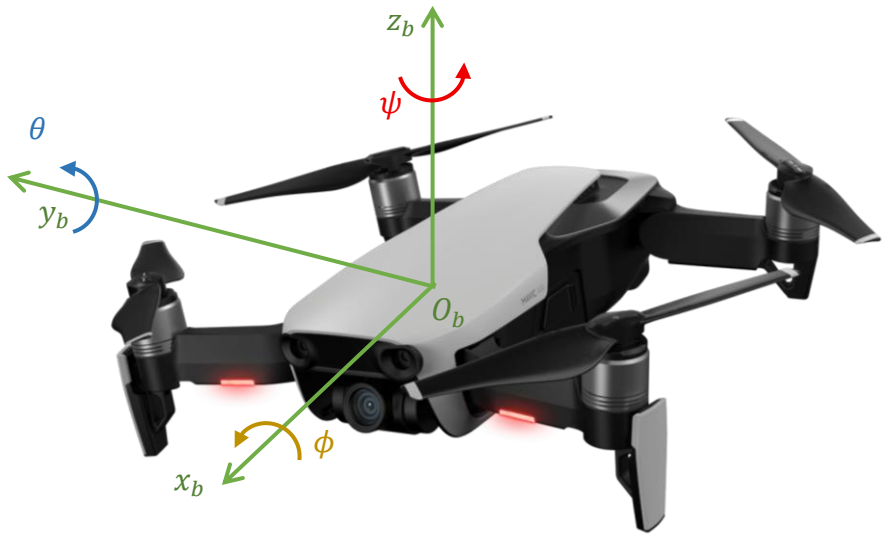

Figure II.4: Euler Angles in Quadcopter Body Frame

Euler angles (ϕ, θ, ψ) are employed to manage a quadcopter's movements and orientation. Roll(ϕ) manages lateral tilt, causing the quadcopter to move sideways by adjusting thrust in opposing motors. Pitch(θ) governs forward and backward tilt, causing the quadcopter to move forward and backward by deflecting thrust. Yaw(ψ) rotates the quadcopter around its vertical axis, changing its direction to face targets or execute curves. Collectively, these angles determine the orientation of the quadcopter, enabling it to make exact movements like hovering, spinning, or pursuing intricate flight trajectories.

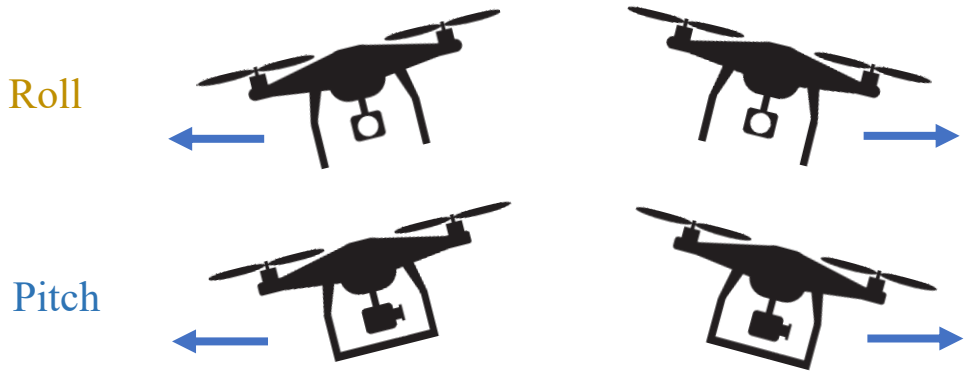

Figure II.5: Movement of Quadcopter

II.2.4 Transformations Between Frames

Mathematical modeling of quadcopter dynamics is typically founded upon body frame to inertial frame transformations and vice versa. Such transformations enable the precise description and analysis of forces, velocities, and positions in the relevant reference frame.

The Importance of Transformations

- **Force Alignment:** Gravity happens in the inertial frame, and thrust is generated in the body frame. The transformations align them.
- **Sensor Integration:** Accelerometers, for example, report measurements in the body frame and need to be transformed to the inertial frame to enable meaningful navigation.
- **Motion Analysis:** Positions and velocities in one frame must be transformed to the other for consistent calculations.

Rotation Matrices

Rotation matrices are mathematical constructs employed in representing the orientation of a quadcopter through vector transformations from the inertial frame of reference to the body frame of reference. The three Euler angles (ϕ, θ, ψ) form the foundation of the transformations therein, and as such, the orientation of the quadcopter can be represented within a three-dimensional space.

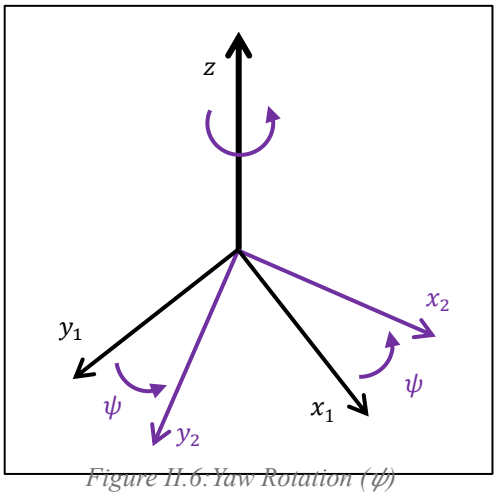

Figure II.6: Yaw Rotation (ψ)

- **Yaw Rotation (ψ)**: Rotation about the vertical (z_i) axis:

$$\mathcal{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.1)$$

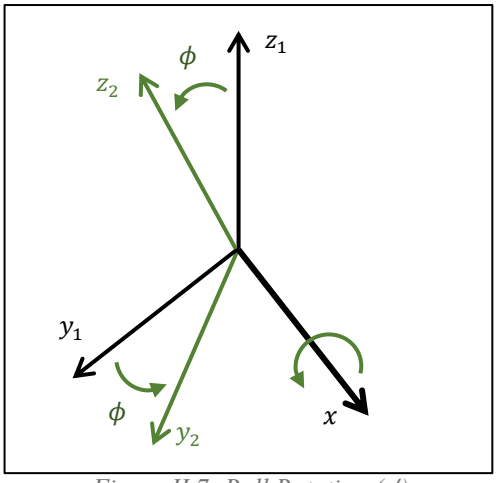

Figure II.7: Roll Rotation (ϕ)

- **Roll Rotation (ϕ):** Rotation about the longitudinal (x_b) axis:

$$\mathcal{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \qquad (2.2)$$

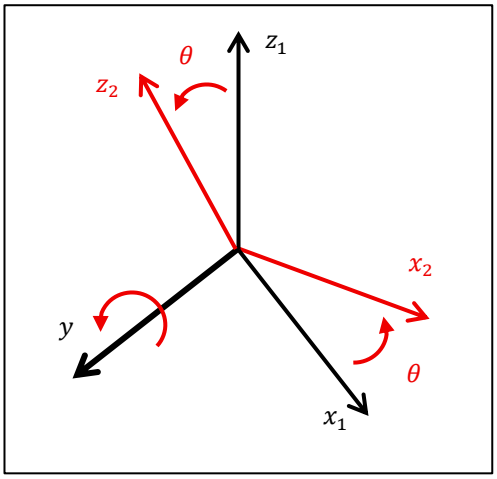

Figure II.8: Pitch Rotation (θ)

- **Pitch Rotation (θ):** Rotation about the intermediate (y_b) axis:

$$\mathcal{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad (2.3)$$

Combined Transformation

The combined rotation matrix $\mathcal{R}$ is the product of the individual matrices:

$$\mathcal{R} = \mathcal{R}_z(\psi).\mathcal{R}_y(\theta).\mathcal{R}_x(\phi) \qquad (2.4)$$

$$\mathcal{R} = \begin{bmatrix} \cos\psi.\cos\theta & \cos\psi.\sin\theta.\sin\phi - \sin\psi.\cos\phi & \cos\psi.\sin\theta.\cos\phi + \sin\psi.\sin\phi \\ \sin\psi.\cos\theta & \sin\psi.\sin\theta.\sin\phi + \cos\psi.\cos\phi & \sin\psi.\sin\theta.\cos\phi - \cos\psi.\sin\phi \\ -\sin\theta & \cos\theta.\sin\phi & \cos\theta.\cos\phi \end{bmatrix} \qquad (2.5)$$

Transformation Equations

The orientation of the quadcopter is described with rotation matrix $\boldsymbol{\mathcal{R}}$. It brings vectors from the body frame into the inertial frame. To get back, we use the transpose, $\boldsymbol{\mathcal{R}^T}$, because $\boldsymbol{\mathcal{R}}$ is an orthogonal matrix.

- **Body Frame to Inertial Frame:**

$$v_i = \mathcal{R}.v_b \qquad (2.6)$$

  - Converts vectors like thrust from the quadcopter's frame to the inertial reference frame.
- **Inertial Frame to Body Frame:**

$$v_b = \mathcal{R}^T.v_i \qquad (2.7)$$

  - Aligns external forces like gravity with the quadcopter's body frame.

II.3 Aerodynamic Forces and Propeller Dynamics

It is important to understand the various forces acting on a quadcopter for modeling and control with precision. The forces involved play a vital role in enabling the quadcopter to generate lift, be stable, and make controlled movements.

Here, we talk about the overall forces on a quadcopter with respect to the **inertial frame** $\boldsymbol{(O_i, x_i, y_i, z_i)}$, which is the universal reference frame for analyzing motion. Forces produced in the body frame of the quadcopter ($\boldsymbol{O_b, x_b, y_b, z_b}$) are mapped to the inertial frame by utilizing the rotation matrix ($\boldsymbol{\mathcal{R}}$) for consistency [15].

II.3.1 Forces Acting on a Quadcopter

Quadcopter dynamics are controlled by a few fundamental forces, which are classified as:

- **Gravitational Force (F_g):** The weight of the quadcopter from gravity is downwards along $-z_i$-axis in the inertial frame. In the body frame. Its components in the body frame depend on the quadcopter orientation.

$$\overrightarrow{F_g} = \begin{bmatrix} 0 \\ 0 \\ -m.g \end{bmatrix} \qquad (2.8)$$

Where:

- F_g: Gravitational force ($\boldsymbol{N}$).
- m: Quadcopter mass ($\boldsymbol{Kg}$).
- g: Gravitational acceleration ($\boldsymbol{9.81\ m/s^2}$).

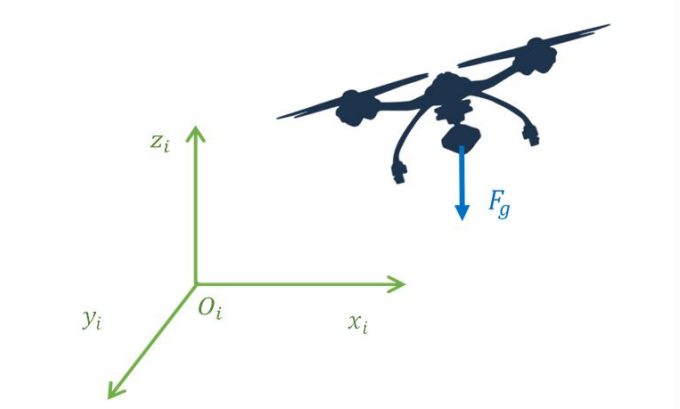

Figure II.9: Gravitational Force of Quadcopter

- **Thrust Force (T):** is the positive force created by the rotating propellers along $+z_b$-axis of the body frame, providing the lift to counteract gravity and enable vertical flight or hover. Total thrust for the four-rotor quadcopter is the total of the individual propeller contributions.

$$\overrightarrow{F_T} = \mathcal{R}.\begin{bmatrix} 0 \\ 0 \\ T_{Total} \end{bmatrix} \qquad (2.9)$$

$$T_{Total} = k_T.\sum_{i=1}^{4} {\omega_i}^2 = k_T.({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2) \qquad (2.10)$$

Where:

- k_T: Thrust coefficient ($\boldsymbol{N.s^2/rad^2}$).

- ω_i: Angular velocity of rotor i ($\boldsymbol{rad/s}$).

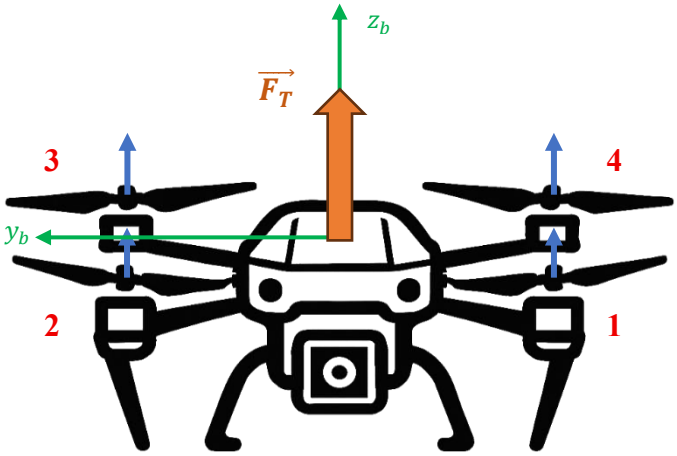

Figure II.10: Thrust Force of Quadcopter

- **Aerodynamic Drag Force ($\boldsymbol{F_D}$):** Aerodynamic drag opposes the movement of the quadcopter through the air. It acts on the entire quadcopter, including propeller blades, and is a function of orientation and velocity.

$$\overrightarrow{F_D} = -\frac{1}{2}\rho C_d A. \|\vec{V}\|.\vec{V} \qquad (2.11)$$

Where:

- F_D: Drag force ($\boldsymbol{N}$).
- C_d: Translational drag coefficient (***dimensionless***).
- ρ: Air density ($\boldsymbol{kg/m^3}$).
- A: Effective Reference area ($\boldsymbol{m^2}$) (typically frontal area for quadcopter).
- $\vec{V} = \begin{bmatrix} \dot{x}_\iota \\ \dot{y}_\iota \\ \dot{z}_\iota \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$: Linear velocity vector in the inertial frame ($\boldsymbol{m/s}$).

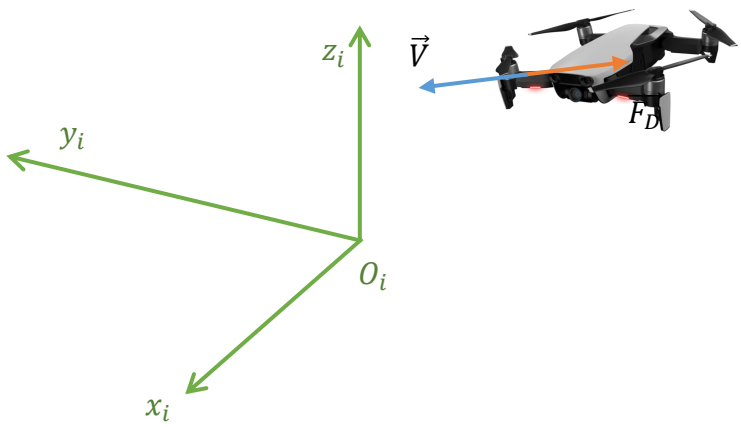

Figure II.11: Drag Force of Quadcopter

II.3.2 Torques Acting on a Quadcopter

The torques on a quadcopter are relevant to its rotational dynamics and control its motion in pitch, roll, and yaw. Torques are created by the forces produced by the propellers and consist of elements of thrust-related torque, drag-related torque, and gyroscopic effects. These torques are viewed with respect to the **body frame** $(\boldsymbol{O_b, x_b, y_b, z_b})$ since it provides the most natural description of the rotational motion of the quadcopter [15].

- **Thrust-Induced Torque ($\boldsymbol{\tau_T}$):** Thrust torques result from the asymmetrical thrust produced by the four propellers. By changing the rotational speed of each propeller, the quadcopter generates torques around its principal axes, allowing for rotations associated with pitch, roll, or yaw.

$$\overrightarrow{\tau_T} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l.\left((T_2 + T_3) - (T_1 + T_4)\right) \\ l.\left((T_3 + T_4) - (T_1 + T_2)\right) \\ -k_D.\sum_{i=1}^{4}(-1)^i.{\omega_i}^2 \end{bmatrix} = \begin{bmatrix} l.k_T.\left(({\omega_2}^2 + {\omega_3}^2) - ({\omega_1}^2 + {\omega_4}^2)\right) \\ l.k_T.\left(({\omega_3}^2 + {\omega_4}^2) - ({\omega_1}^2 + {\omega_2}^2)\right) \\ k_D.\left(({\omega_1}^2 + {\omega_3}^2) - ({\omega_4}^2 + {\omega_4}^2)\right) \end{bmatrix} \quad (2.12)$$

Where:

- $\overrightarrow{\tau_T}$: Thrust -Induced Torque $(\boldsymbol{N.m})$.
- l: Distance from the quadcopter's center to each propeller $(\boldsymbol{m})$.
- T_i: Thrust generated by the propeller n° i $(\boldsymbol{N})$.
- ω_i: Angular velocity of the propeller n° i $(\boldsymbol{rad/s})$.
- k_D: Torque Drag coefficient $(\boldsymbol{N.m.s^2/rad^2})$.

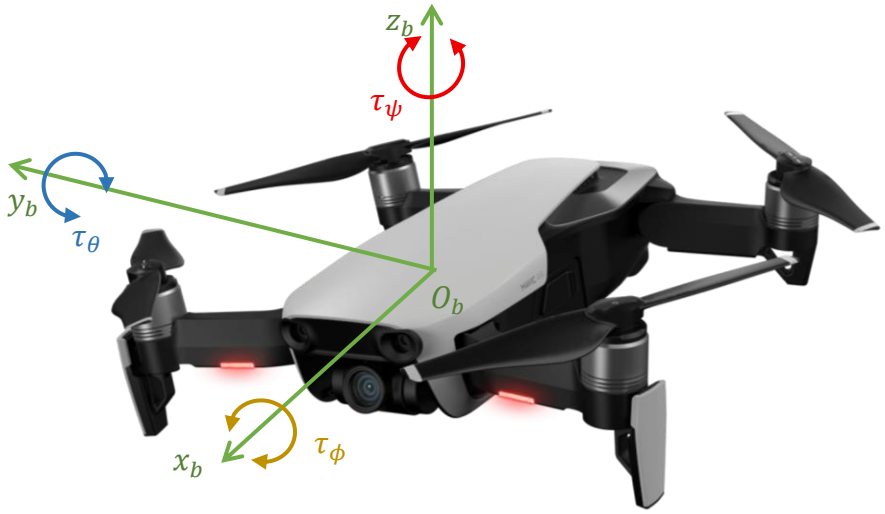

Figure II.12: Thrust-Induced Torque

- **Aerodynamic Drag Torque ($\boldsymbol{\tau_a}$):** Aerodynamic drag torque constitutes the rotational damping torque as a result of the viscous drag of the air resisting the angular movement of the quadcopter. When the structure of the quadcopter rotates around its axes (roll, pitch, yaw) the torques resist the movement as a result of the resistance of the air to rotation.

$$\overrightarrow{\tau_a} = D_a . \overrightarrow{\omega_b} = \begin{bmatrix} -\mathcal{B}_\phi & 0 & 0 \\ 0 & -\mathcal{B}_\theta & 0 \\ 0 & 0 & -\mathcal{B}_\psi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} -\mathcal{B}_\phi . p \\ -\mathcal{B}_\theta . q \\ -\mathcal{B}_\theta . r \end{bmatrix} \qquad (2.13)$$

Where:

- $\overrightarrow{\tau_a}$: Drag-Induced Torque ($\boldsymbol{N.m}$).
- p, q, r: Components of Angular velocity vector $\overrightarrow{\boldsymbol{\omega_b}}$ of the quadcopter in Body Frame ($\boldsymbol{rad/s}$).
- D_a: Aerodynamic Damping Coefficients Matrix ($\boldsymbol{N.m.s/rad}$).

II.4 Kinematics and Dynamics of the Quadcopter

The dynamics of a quadcopter are controlled by the basic principles of rigid-body dynamics, both translational and rotational in nature. Because of its six degrees of freedom—three translational and three rotational movements—and the capacity of its four rotors to produce forces and torques in a body-fixed coordinate frame, a mathematical model involving both reference frames must be developed: the inertial frame and the body frame. This section derives the complete set of nonlinear equations of motion for the quadcopter from Newton–Euler mechanics [13] [17].

The translational dynamics are expressed in the inertial frame, describing the position and velocity of the vehicle relative to the Earth, while the rotational dynamics are expressed in the body frame, describing the orientation and angular rates of the vehicle. The derivations thoroughly incorporate the interdependence of translational and rotational motions, given the orientation of the quadcopter determines the thrust direction and, by extension, its trajectory. Such equations constitute the fundamental framework for simulation, analysis, and control system design [13] [17].

II.4.1 State Variables

The movement of a quadcopter in a three-dimensional space can be fully accounted for by a collection of state variables that define its translational and rotational states. They are the position, velocity, orientation, and the angular velocity of the quadcopter's center of mass.

- **Translational State Variables (in Inertial Frame):**
  - **Position vector:**

$$\overrightarrow{r(t)} = \begin{bmatrix} x_i(t) \\ y_i(t) \\ z_i(t) \end{bmatrix} \qquad (2.14)$$

Represents the location of the quadcopter's center of mass relative to the origin of the inertial frame.

Units: meters ($\boldsymbol{m}$).

  - **Linear velocity:**

$$\overrightarrow{\dot{r}(t)} = \overrightarrow{V(t)} = \begin{bmatrix} \dot{x}_\iota(t) \\ \dot{y}_\iota(t) \\ \dot{z}_\iota(t) \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad (2.15)$$

Time derivative of position, representing the velocity of the center of mass in inertial frame.

Units: meters per second ($\boldsymbol{m/s}$).

- **Rotational State Variables:**
  - **Orientation (Euler angles):**

$$\Theta(t) = \begin{bmatrix} \phi(t) \\ \theta(t) \\ \psi(t) \end{bmatrix} \qquad (2.16)$$

Defines the orientation of the body frame $\boldsymbol{B}$ relative to $\boldsymbol{I}$, using the $\boldsymbol{Z-Y-X}$ (yaw–pitch–roll) convention:

- ✓ $\boldsymbol{\phi(t)}$: Roll angle (rotation about $\boldsymbol{x_B}$)
- ✓ $\boldsymbol{\theta(t)}$: Pitch angle (rotation about $\boldsymbol{y_B}$)
- ✓ $\boldsymbol{\psi(t)}$: Yaw angle (rotation about $\boldsymbol{z_B}$),

Units: radians ($\boldsymbol{rad}$).

  - **Angular velocity (in body frame):**

$$\omega_b(t) = \begin{bmatrix} \dot{x_B}(t) \\ \dot{y_B}(t) \\ \dot{z_B}(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ q(t) \\ r(t) \end{bmatrix} \qquad (2.17)$$

Represents the instantaneous rotation rates about $\boldsymbol{x_B}, \boldsymbol{y_B}$ and $\boldsymbol{z_B}$ axes.

Units: radians per second ($\boldsymbol{rad/s}$).

II.4.2 Equations of Motion with Diagrams

The motion of a quadcopter is controlled by the equations of motion for a rigid body under the influence of external torques and forces. The full set of equations comprises both translational motion of the center of mass within the inertial frame and rotational motion around the center of mass within the body frame [15].

II.4.2.1 Translational Motion (in Inertial Frame)

From **Newton's second law**, the translational motion is described by [13]:

$$\boldsymbol{m\ddot{r}} = \sum \boldsymbol{F_{external}} \qquad (2.18)$$

$$m\ddot{r} = F_g + F_T + F_D + F_d(t) \qquad (2.19)$$

From (2.8), (2.9) and (2.11):

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -m.g \end{bmatrix} + \mathcal{R}.\begin{bmatrix} 0 \\ 0 \\ k_T.(\omega_1{}^2 + \omega_2{}^2 + \omega_3{}^2 + \omega_4{}^2) \end{bmatrix} - \frac{1}{2}\rho C_d A.\|\vec{V}\|.V + F_d(t) \qquad (2.20)$$

Where:

- $F_d(t)$: External disturbance forces (wind gusts, impulsive pushes) $(\boldsymbol{N})$
- $\ddot{r} = \dot{V} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}$: linear acceleration $(\boldsymbol{m/s^2})$

So:

$$m\dot{V} = \begin{bmatrix} 0 \\ 0 \\ -m.g \end{bmatrix} + \mathcal{R}.\begin{bmatrix} 0 \\ 0 \\ k_T.(\omega_1{}^2 + \omega_2{}^2 + \omega_3{}^2 + \omega_4{}^2) \end{bmatrix} - \frac{1}{2}\rho C_d A.\|\vec{V}\|.V + F_d(t) \qquad (2.21)$$

$$\dot{V} = \frac{1}{m}.\left[-\frac{1}{2}\rho C_d A.\|\vec{V}\|.V + \mathcal{R}.\begin{bmatrix} 0 \\ 0 \\ k_T.(\omega_1{}^2 + \omega_2{}^2 + \omega_3{}^2 + \omega_4{}^2) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ m.g \end{bmatrix} + F_d(t)\right] \qquad (2.22)$$

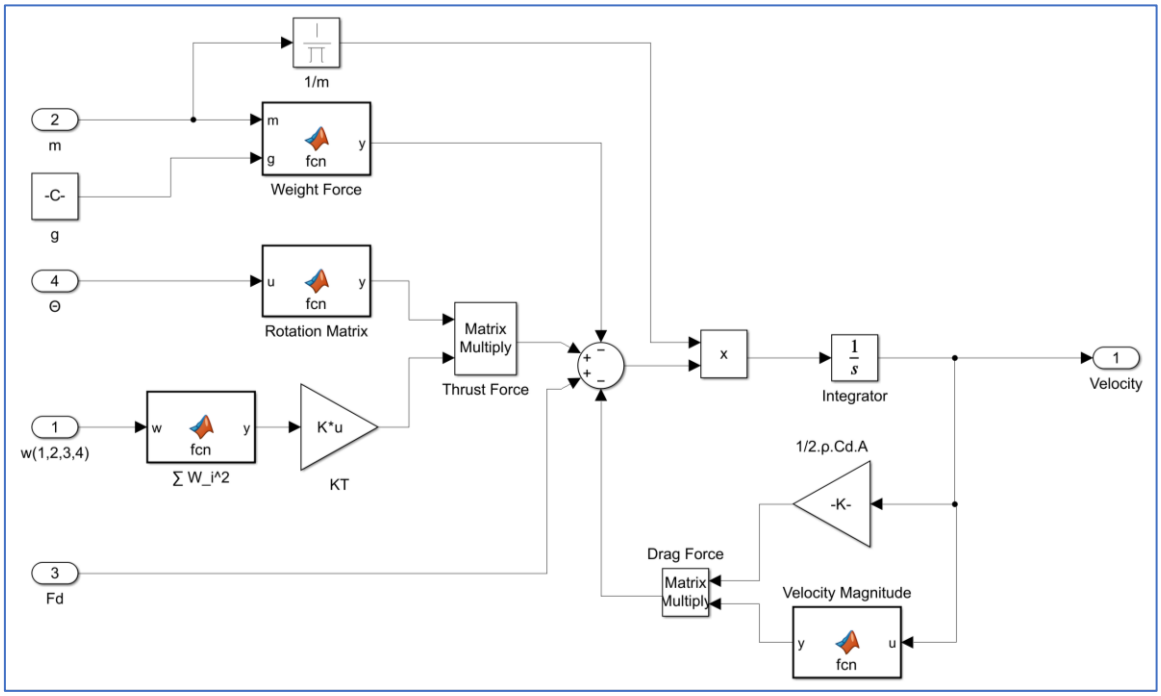

Figure II.13: Quadcopter Translational Motion Model

II.4.2.2 Rotational Motion (in Body Frame)

According to **Euler's rotational equations**, the rotational motion about the center of mass is [13]:

$$\boldsymbol{I\dot{\omega}_b + \omega_b \times (I\omega_b) = \tau_{Total}} \qquad (2.23)$$

$$I\dot{\omega}_b + \omega_b \times (I\omega_b) = \tau_T + \tau_a + \tau_g + \tau_d(t) \qquad (2.24)$$

From (2.12) *and* (2.13):

$$I\dot{\omega}_b + \omega_b \times (I\omega_b) = \begin{bmatrix} l.k_T.\left((\omega_2{}^2 + \omega_3{}^2) - (\omega_1{}^2 + \omega_4{}^2)\right) \\ l.k_T.\left((\omega_3{}^2 + \omega_4{}^2) - (\omega_1{}^2 + \omega_2{}^2)\right) \\ k_D.\left((\omega_1{}^2 + \omega_3{}^2) - (\omega_4{}^2 + \omega_4{}^2)\right) \end{bmatrix} - D_a.\omega_b + \tau_d(t) \qquad (2.25)$$

$$\dot{\omega}_b = I^{-1}.\left[\begin{bmatrix} l.k_T.\left((\omega_2{}^2 + \omega_3{}^2) - (\omega_1{}^2 + \omega_4{}^2)\right) \\ l.k_T.\left((\omega_3{}^2 + \omega_4{}^2) - (\omega_1{}^2 + \omega_2{}^2)\right) \\ k_D.\left((\omega_1{}^2 + \omega_3{}^2) - (\omega_4{}^2 + \omega_4{}^2)\right) \end{bmatrix} - \omega_b \times (I\omega_b) - D_a.\omega_b\right] + \tau_d(t) \qquad (2.26)$$

Where:

- $\tau_d(t)$: External Disturbance Torques (wind moment, collision) ($\boldsymbol{N.m}$).
- $\dot{\omega}_b = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}$: Angular acceleration ($\boldsymbol{rad/s^2}$).
- I: Inertia matrix ($\boldsymbol{Kg.m^2}$).

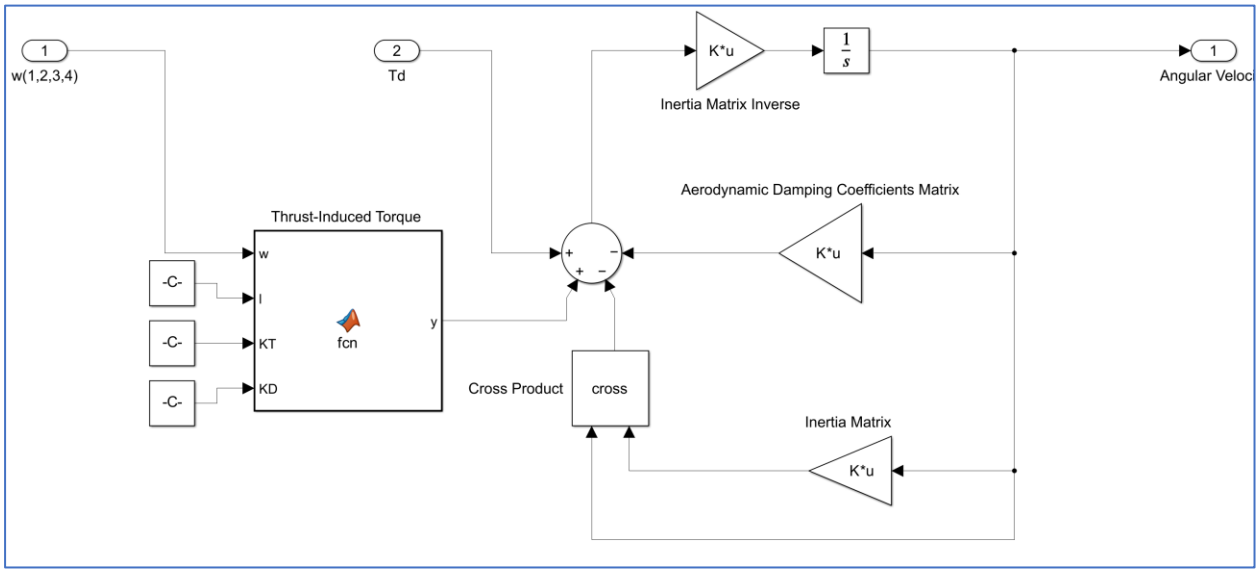

Figure II.14: Quadcopter Rotational Motion Model

II.4.2.3 Kinematic Relationships (Kinematic Coupling):

Translational and rotational dynamics are coupled via the orientation $\boldsymbol{\Theta}$. And since $\boldsymbol{\omega_b}$ is expressed in body frame $\boldsymbol{B}$ and $\dot{\boldsymbol{\Theta}}$ is expressed in inertial frame $\boldsymbol{I}$, they are related by a transformation matrix $\boldsymbol{W(\Theta)}$:

$$\dot{\Theta} = W(\Theta).\omega_b \qquad (2.27)$$

Where:

$$W(\Theta) = \begin{bmatrix} 1 & \sin\phi \,.\, \tan\theta & \cos\phi \,.\, \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \qquad (2.28)$$

$\boldsymbol{W(\Theta)}$ represents the nonlinear relationship between the rates of the Euler angles and the body frame angular velocity. It is a fundamental relationship in terms of describing rotational kinematics in measurable quantities.

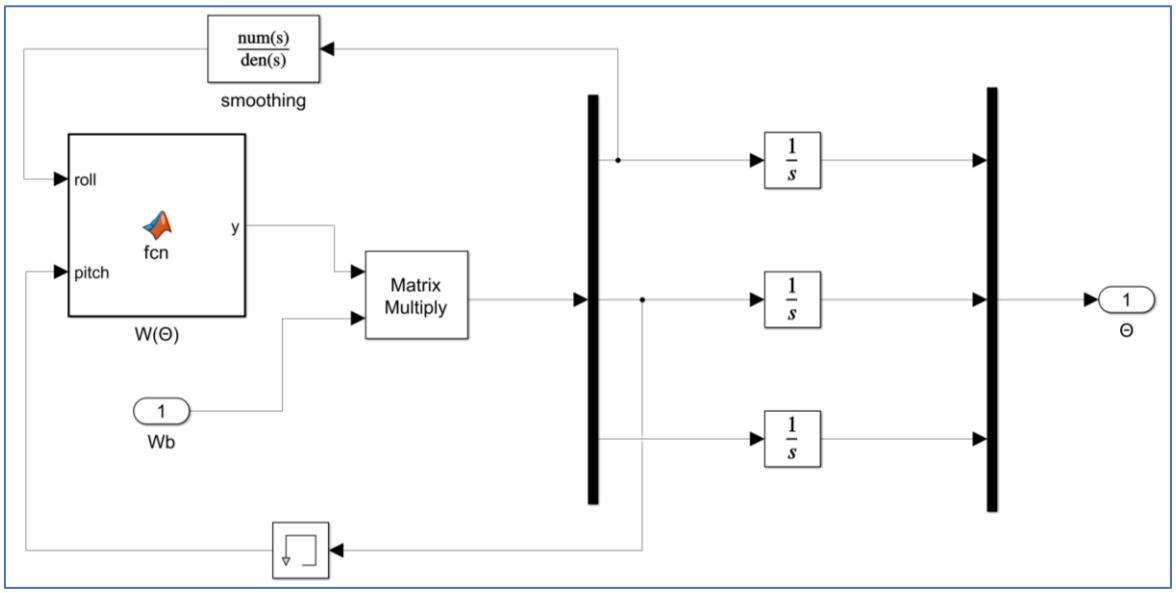

Figure II.15: Quadcopter Euler Angles

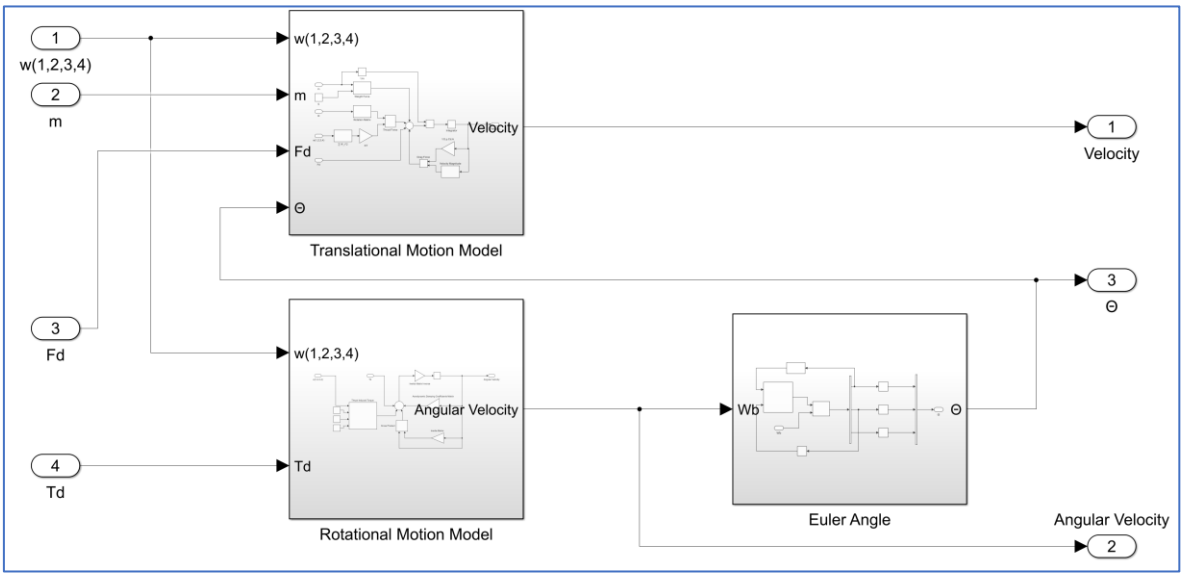

Figure II.16: Quadcopter Full Model

II.5 Summary

This chapter developed the complete mathematical model of the quadcopter drone, forming the analytical foundation upon which control strategies can be designed and validated. Starting from fundamental assumptions of rigid-body dynamics, the translational and rotational equations of motion were derived using Newton–Euler formalism. The relationships between the inertial and body frames were established through Euler-angle transformations, enabling a unified description of motion, attitude, and orientation. The resultant nonlinear model accurately represents the coupling between forces and moments generated by the four rotors and the vehicle's overall motion in three-dimensional space.

The comprehensive quadcopter model established here serves two key purposes. First, it provides the physical insight necessary to understand how control inputs—such as throttle, roll, pitch, and yaw commands—affect the drone's motion and stability. Second, it offers a reliable platform for simulation and controller synthesis, enabling accurate prediction of system behavior under feedback control.

With this mathematical foundation in place, the next chapter focuses on control system design. Chapter III will apply the derived model to formulate and analyze feedback control architectures—beginning with classical PID controllers and extending to intelligent and adaptive control techniques—to ensure stable, precise, and responsive flight performance suitable for autonomous date-palm pollination missions.

Chapter III

III. Design of Control System & Simulation

III.1 Introduction

The control of a quadcopter represents a classical yet challenging problem in nonlinear dynamics and feedback regulation. As established in Chapter II, the vehicle exhibits strong coupling between translational and rotational motions, under actuation in the inertial coordinates, and intrinsic open-loop instability. These characteristics necessitate the development of a structured feedback strategy capable of shaping the system's response, rejecting disturbances, and ensuring precise tracking of commanded motions. Within the broader field of aerial robotics, hierarchical or cascaded architectures have emerged as an effective means of addressing these challenges by exploiting the natural disparity between the fast rotational dynamics and the slower translational behavior [14].

In this theoretical context, proportional–integral–derivative (PID) controllers remain a widely adopted solution, not solely for their simplicity but for their compatibility with the linearized representations of multirotor dynamics near hover. When arranged in a cascaded structure, PID controllers provide a tractable framework for stabilizing angular rates, regulating attitude, and commanding translational accelerations. Their behavior can be systematically characterized in terms of classical control concepts such as bandwidth separation, damping, and error convergence, which allows rigorous analysis despite the underlying nonlinearities of the system [14].

Building on these principles, this chapter introduces a complete cascaded PID control framework for the quadcopter model. The formulation begins with the innermost angular-rate loop, proceeds to the attitude-regulation layer, and finally extends to velocity and position control. Each layer is synthesized using model-based reasoning grounded in the dynamical equations previously derived, enabling explicit relationships between controller gains and the system's physical parameters.

The last sections of the chapter examine the behavior of the proposed control laws through simulation of the full nonlinear quadcopter model. These theoretical assessments illustrate the closed-loop stability properties, transient dynamics, and robustness of the cascaded PID scheme under a variety of reference trajectories and perturbations.

III.2 Control Architecture

The control architecture of the quadcopter is organized in a **Hierarchical Cascade Structure** composed of two feedback loops and one actuator allocation stage. This configuration ensures fast and stable attitude response, effective disturbance rejection, and simplified tuning of control gains [14] [17].

The hierarchy consists of the following layers:

- **Outer Attitude Control Loop :**
  The outer loop regulates the drone's roll (ϕ), pitch (θ), and yaw (ψ) angles. It compares the desired attitude references (ϕ_d, θ_d, ψ_d) with the measured angles from the IMU and generates the corresponding desired angular rates (p_d, q_d, r_d). This loop operates at a moderate frequency and ensures smooth orientation tracking and stable hovering.
- **Inner Angular-Rate Control Loop:**
  The inner loop directly stabilizes the body angular rates (p, q, r) around each principal axis. By acting on the faster rotational dynamics of the drone, this loop provides rapid disturbance damping and precise control over angular velocity. It produces corrective torque components ($\Delta_\phi, \Delta_\theta, \Delta_\psi$) that determine how much each rotor must accelerate or decelerate to maintain stability.
- **Motor Mixing and ESC Command Allocation:**
  The torque corrections and total collective thrust are combined in the motor mixer to generate four individual ESC throttle signals $[u_1, u_2, u_3, u_4]$. Each ESC regulates the corresponding motor's rotational speed ω_i, which produces the required thrust and reactive torque to realize the control commands.

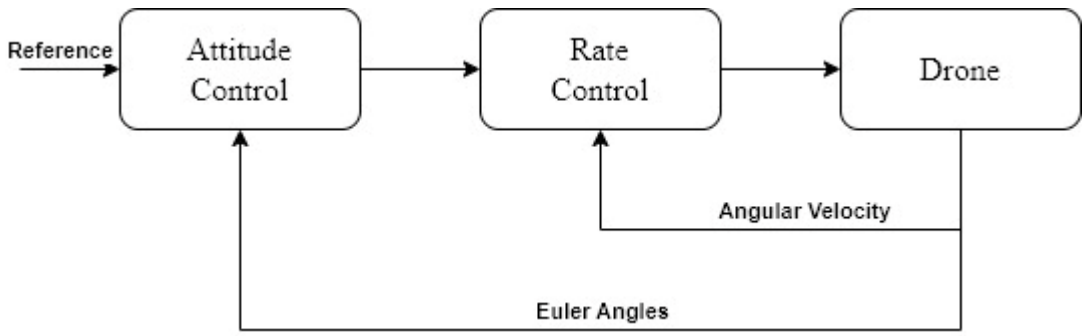

Figure III.1: Control Architecture Overview

III.2.1 Feedback Inputs

The effectiveness of the control system depends on accurate and timely feedback from the drone's onboard sensors. These feedback signals represent the state of the vehicle's motion and are essential for computing corrective control actions. In this work, the primary feedback variables are the attitude angles and angular rates, both derived from the **Inertial Measurement Unit (IMU)**.

Attitude Angles

The attitude of the quadcopter is defined by three Euler angles:

- **Roll (ϕ):** rotation around the body x-axis, representing lateral tilt.
- **Pitch (θ):** rotation around the body y-axis, representing longitudinal tilt.
- **Yaw (ψ):** rotation around the body z-axis, representing heading.

These angles are estimated using data fusion from the IMU's gyroscope, accelerometer, and magnetometer, typically through a complementary or Kalman filter. The filtered outputs provide smooth and reliable measurements for the outer attitude control loop.

Angular Rates

The instantaneous rotational velocities about the body axes are measured directly by the gyroscope as: $[p, q, r]^T$, where:

- p: roll rate (x-axis),
- q: pitch rate (y-axis),
- r: yaw rate (z-axis).

III.3 Classical Control Design (PID Controllers)

III.3.1 Description of PID Control

The Proportional–Integral–Derivative (PID) controller is one of the most widely used control strategies in engineering systems, including quadcopters. Its popularity stems from its intuitive structure, ease of implementation, and effectiveness in regulating both linear and mildly nonlinear systems. In the context of UAVs, PID controllers are typically deployed for attitude stabilization,

altitude regulation, and trajectory tracking, particularly when computational resources are limited and real-time responsiveness is critical [15] [17].

Conceptual Overview

At its core, a PID controller continuously computes an error signal $e(t)$, defined as the difference between a desired reference $r(t)$ and the measured system output $y(t)$:

$$e(t) = r(t) - y(t) \qquad (3.1)$$

The controller then generates a control input $u(t)$ by combining three terms:

Component	Description	Mathematical Term	Effect on System Behavior
Proportional (P)	Reacts to the current error. Provides immediate corrective action proportional to the magnitude of the error.	$\boldsymbol{K_p e(t)}$	Reduces rise time. Improves responsiveness. May cause overshoot if too high.
Integral (I)	Accumulates past errors over time.	$\boldsymbol{K_i \int e(t)dt}$	Eliminates steady-state error. May increase overshoot and settling time.
Derivative (D)	Predicts future error by observing its rate of change.	$\boldsymbol{K_d \frac{de(t)}{dt}}$	Improves stability. Reduces overshoot. Sensitive to noise.

Table III.1: Components of the PID Controller

The total control law is the sum of these three components:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \qquad (3.2)$$

Each gain K_p, K_i, K_d must be carefully tuned to balance responsiveness, stability, and robustness.

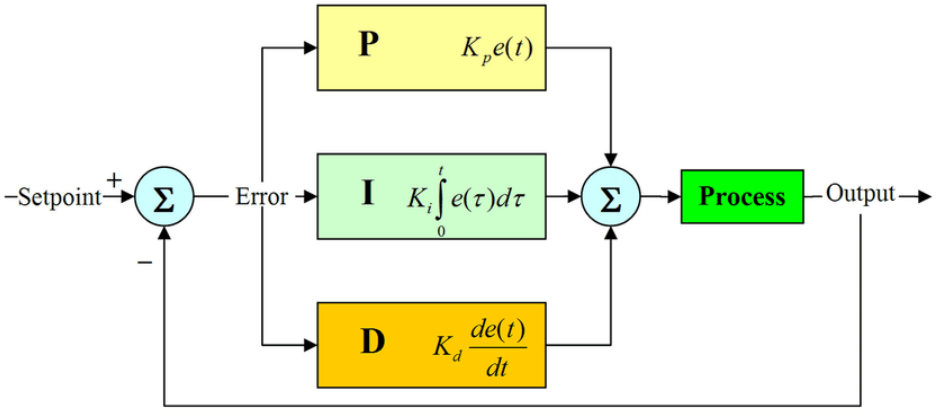

Figure III.2: PID Controller Structure [31]

Advantages and Limitations

Advantages	Limitations
• The PID controller has a clear structure and is easy to understand, implement, and modify. • Requires minimal processing resources • Effective for small-angle stabilization and hover control • Well-understood tuning methods • Smooth and Stable Response	• Performance degrades in highly nonlinear or time-varying conditions • Sensitive to gain tuning and sensor noise • Lacks adaptability to changing dynamics (e.g., payload variation, wind)

Table III.2: Advantages and Limitations of PID Control

PID Control in Quadcopter Systems

A quadcopter is a nonlinear, underactuated, and coupled system in which four rotors control six degrees of freedom. However, near hover, its rotational dynamics — roll (ϕ), pitch (θ), and yaw (ψ) — can be approximated as decoupled linear subsystems, allowing independent PID controllers for each rotational axis.

The PID algorithm offers several benefits for UAV attitude control:

- **Computational simplicity:** easy to implement on low-cost microcontrollers
- **Ease of tuning:** gains can be empirically adjusted during test flights

- **Smooth actuation:** generates continuous throttle signals compatible with ESCs

III.3.2 Inner Rate Control Loop

The inner rate control loop stabilizes the quadcopter's angular velocities about its principal axes—roll (ϕ), pitch (θ), and yaw (ψ). Operating at high frequency, it ensures rapid response to disturbances and attitude commands generated by the outer loop. It receives desired angular rates (p_d, q_d, r_d) and compares them to measured rates from the IMU, generating torque commands $(\tau_\phi, \tau_\theta, \tau_\psi)$ for motor mixing.

By operating at high frequency, the rate loop quickly rejects disturbances and linearizes the vehicle's rotational behavior, allowing the slower outer attitude loop to command smooth and predictable motion.

This loop is critical for maintaining flight stability during aggressive maneuvers, gust rejection, and precision hovering.

III.3.2.1 Measured-Rate Based Derivative Action

Derivative action in classical PID controllers is intended to anticipate system behavior by responding to the rate of change of the error signal. However, direct differentiation of the error often amplifies measurement noise, leading to undesirable oscillations and degraded control performance. To mitigate this, a measured-rate based derivative action is employed, where the derivative term is computed from the plant feedback rate rather than from the error. This approach enhances robustness and reduces sensitivity to high-frequency disturbances.

Thus, the modified control law becomes:

$$u(t) = K_p e(t) + K_i \int e(t)\, dt - K_d \frac{dy(t)}{dt} \qquad (3.3)$$

This design choice provides several practical and stability advantages:

Avoids Derivative Kick: Differentiating the error would include the derivative of the reference rate command. Any step or sudden change in ω_{ref} produces a large, undesirable spike in torque.

Provides Physical Damping: The term $-K_d \dot{\omega}$ acts like viscous damping on the rotational motion. This directly counteracts fast rotational accelerations and improves stability and disturbance rejection.

Reduces Loop Interaction: Derivative on the error couples the derivative action to the outer-loop commands. Using measured rate derivative keeps the derivative term purely feedback-based, preserving clean bandwidth separation between outer and inner loops.

More Robust to Reference Changes: Rate commands can change abruptly (pilot inputs, autopilot maneuvers). Derivative on measured rate is unaffected by these discontinuities, whereas derivative on error would amplify them.

Simpler and More Stable in Implementation: Gyro signals are already band-limited and easy to filter. Differentiating the reference requires extra filtering and synchronization. Measured rate derivative is therefore more reliable in embedded flight code.

III.3.2.2 Control Law Formulation

The control objective is to generate the torque τ_i that drives the measured angular velocity to follow its desired value.

For each axis, the control error is defined as:

$$e_{\omega_b}(t) = \omega_{b_d}(t) - \omega_b(t) \qquad (3.4)$$

Where $\omega \in \{p, q, r\}$ corresponds to roll, pitch, and yaw rates respectively.

The PID control law is:

$$\tau(t) = K_p^{\omega_b} e_{\omega_b}(t) + K_i^{\omega_b} \int e_{\omega_b}(t)dt - K_d^{\omega_b} \frac{d\omega_b(t)}{dt} \qquad (3.5)$$

The resulting control torque τ_i is distributed among the four rotors through the **motor mixer**, generating the necessary differential thrusts.

III.3.2.3 Model Based Tuning

The rotational motion of a rigid body is described by:

$$I \cdot \dot{\omega}_b(t) = \tau - \omega_b(t) \times (I \cdot \omega_b(t)) \qquad (3.6)$$

Where:

- I: 3×3 inertia matrix (diagonal for symmetric quadcopters)

- $\omega_b(t) = [p, q, r]^T$: angular velocity vector in the body frame
- $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T$: control torques
- $\omega_b(t) \times (I \cdot \omega_b(t))$: gyroscopic coupling term

This equation captures how torques produce angular accelerations, while also accounting for nonlinear coupling between axes due to rotation.

Axis-Wise Simplification (Small-Angle Assumption)

For control design — especially in hover or near-hover conditions — we often assume:

- The quadcopter is symmetric → I is diagonal:

$$I = diag(I_x, I_y, I_z) \qquad (3.7)$$

- Angular velocities are small → gyroscopic coupling $\boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega})$ is negligible

Under these assumptions, the decoupled rotational dynamics of the quadcopter around each body axis $i \in \{\phi, \theta, \psi\}$ are described by:

$$I_i \, \dot{\omega}_b(t) = \tau(t) \;\leftrightarrow\; \dot{\omega}_b(t) = \frac{\tau(t)}{I_i} \qquad (3.8)$$

where:

- I_i: is the moment of inertia about $i - axis$ $(\boldsymbol{kg \cdot m^2})$
- $\omega_b(t)$: is the body angular rate (p, q, r) $(\boldsymbol{rad/s})$
- $\tau(t)$: is the control torque produced by the propellers $(\boldsymbol{N \cdot m})$

Each axis behaves like a **first-order integrator** from torque input to angular velocity output. This is the foundation for designing PID controllers that regulate p, q, r using torque commands.

$$s.\Omega_b(s) = \frac{T(s)}{I_i} \qquad (3.9)$$

The open-loop transfer functions of the simplified rotational dynamics becomes:

$$G(s) = \frac{\Omega_b(s)}{T(s)} = \frac{1}{I_i.s} \qquad (3.10)$$

The PID controller in the Laplace domain is:

$$T(s) = \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)E(s) - K_d^{\omega_b}s.\Omega_b(s) \qquad (3.11)$$

Where:

$$E(s) = \Omega_{b_d}(s) - \Omega_b(s) \qquad (3.12)$$

Combining (3.4.9) and (3.4.10)

$$\Omega_b(s) = G(s).\left(\left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)E(s) - K_d^{\omega_b}s.\Omega_b(s)\right) \qquad (3.13)$$

$$\Omega_b(s) = \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)G(s)E(s) - K_d^{\omega_b}s.G(s).\Omega_b(s) \qquad (3.14)$$

From (3.4.11)

$$\Omega_b(s) = \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)G(s)\left[\Omega_{b_d}(s) - \Omega_b(s)\right] + K_d^{\omega_b}s.G(s).\Omega_b(s) \qquad (3.15)$$

$$\Omega_b(s) = \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)G(s).\Omega_{b_d}(s) - \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s} + K_d^{\omega_b}s\right)G(s).\Omega_b(s) \qquad (3.16)$$

$$\left[1 + \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s} + K_d^{\omega_b}s\right)G(s)\right].\Omega_b(s) = \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)G(s).\Omega_{b_d}(s) \qquad (3.17)$$

$$H(s) = \frac{\Omega_b(s)}{\Omega_{b_d}(s)} = \frac{\left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)G(s)}{1 + \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s} + K_d^{\omega_b}s\right)G(s)} \qquad (3.18)$$

From (3.10)

$$H(s) = \frac{\Omega_b(s)}{\Omega_{b_d}(s)} = \frac{\left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s}\right)\frac{1}{I_i.s}}{1 + \left(K_p^{\omega_b} + \frac{K_i^{\omega_b}}{s} + K_d^{\omega_b}s\right)\frac{1}{I_i.s}} \qquad (3.19)$$

Multiplying numerator and denominator by $I_i.s^2$ to remove fractions:

$$H(s) = \frac{\Omega_b(s)}{\Omega_{b_d}(s)} = \frac{K_p^{\omega_b}s + K_i^{\omega_b}}{I_i.s^2 + K_d^{\omega_b}s^2 + K_p^{\omega_b}s + K_i^{\omega_b}} \qquad (3.20)$$

$$H(s) = \frac{\Omega_b(s)}{\Omega_{b_d}(s)} = \frac{K_p^{\omega_b}s + K_i^{\omega_b}}{\left(I_i + K_d^{\omega_b}\right)s^2 + K_p^{\omega_b}s + K_i^{\omega_b}} \qquad (3.21)$$

Dividing the Numerator and Denominator by $\left(I_i + K_d^{\omega_b}\right)$:

$$H(s) = \frac{\Omega_b(s)}{\Omega_{b_d}(s)} = \frac{\frac{K_p^{\omega_b}}{\left(I_i + K_d^{\omega_b}\right)}s + \frac{K_i^{\omega_b}}{\left(I_i + K_d^{\omega_b}\right)}}{s^2 + \frac{K_p^{\omega_b}}{\left(I_i + K_d^{\omega_b}\right)}s + \frac{K_i^{\omega_b}}{\left(I_i + K_d^{\omega_b}\right)}} \qquad (3.22)$$

Closed-Loop Characteristic Polynomial

The denominator polynomial

$$P_H(s) = s^2 + \frac{K_p^{\omega_b}}{\left(I_i + K_d^{\omega_b}\right)}s + \frac{K_i^{\omega_b}}{\left(I_i + K_d^{\omega_b}\right)} \qquad (3.23)$$

Compare with the standard normalized second-order system:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \qquad (3.24)$$

By coefficient matching, the **model-based tuning relationships** are obtained:

$$\begin{cases} K_p^{\omega_b} = 2\zeta_{\omega_b}\omega_{n_{\omega_b}}\left(I_i + K_d^{\omega_b}\right) \\ K_i^{\omega_b} = \omega_{n_{\omega_b}}^2\left(I_i + K_d^{\omega_b}\right) \end{cases} \qquad (3.25)$$

Where:

- $\omega_{n_{\omega_b}}$: Desired Natural Frequency (defines speed of response),
- ζ_{ω_b}: Desired Damping Ratio (defines overshoot and settling behavior).
- I_i: The Moment of Inertia about $i - axis$ $(\boldsymbol{kg \cdot m^2})$

III.3.3 Outer Attitude Control Loop

The outer attitude loop converts a quaternion attitude error into a desired body-rate command for the high-bandwidth inner rate loop. For many multirotor applications a single proportional term is sufficient at the outer level: it produces smooth, non-noisy rate commands, is simple to tune, and preserves the strict bandwidth separation that guarantees cascade stability. Before forming the control law, the quaternion error must be signed consistently so the controller always commands the shortest rotation.

III.3.3.1 Control Law Formulation

Let φ_d be the desired attitude and φ the current attitude estimate. The error is defined:

$$e_{\varphi}(t) = \varphi_d(t) - \varphi(t) \qquad (3.26)$$

Using the attitude error signal, the purely proportional outer law is written per axis as

$$\omega_{b_d}(t) = K_p^{\theta}\, e_{\varphi}(t) \qquad (3.27)$$

Where:

- $\omega_{b_d}(t)$: is the desired body-rate vector sent to the inner loop.

III.3.3.2 Model Based Tuning

The design of the outer attitude loop is guided by a simplified dynamical model that captures the relationship between the Attitude and the Angular rate.

For small attitude deviations, the error $\varphi_e(t)$ evolves according to the linearized kinematic relation:

$$\omega_b(t) = \dot{\varphi}(t) \qquad (3.28)$$

In Laplace Domaine:

$$\omega_b(s) = s.\varphi(s) \;\leftrightarrow\; \frac{\varphi(s)}{\omega_b(s)} = \frac{1}{s} \qquad (3.29)$$

From (3.27) and (3.26):

$$\omega_b(s) = K_p^{\theta}\left(\varphi_d(s) - \varphi(s)\right) \qquad (3.30)$$

From (3.29)

$$\varphi(s) = \frac{K_p^\theta}{s}\left(\varphi_d(s) - \varphi(s)\right) \qquad (3.31)$$

Then:

$$\varphi(s)\left(1 + \frac{K_p^\theta}{s}\right) = \frac{K_p^\theta}{s}.\varphi_d(s) \qquad (3.32)$$

$$\varphi(s)\left(s + K_p^\theta\right) = K_p^\theta.\varphi_d(s) \qquad (3.33)$$

This is a **first-order system**

$$H_s(s) = \frac{\varphi(s)}{\varphi_d(s)} = \frac{1}{1 + \frac{1}{K_p^\theta}s} \qquad (3.34)$$

$$pole = -K_p^\theta \qquad (3.35)$$

Time constant:

$$\tau_\theta = \frac{1}{K_p^\theta} \qquad (3.36)$$

To get a desired response speed (bandwidth) ω_{nP}:

$$K_p^\theta = \omega_{n\theta}, \qquad with: \quad \omega_{n\theta} = \frac{1}{\tau_\theta} \qquad (3.37)$$

III.4 Position Control

Position control constitutes the outermost layer of the quadcopter's flight control hierarchy, responsible for regulating translational motion in three-dimensional space. While attitude control ensures orientation stability and angular rate damping, position control governs the drone's displacement along the Cartesian axes. This hierarchical separation is essential in agricultural applications such as date palm pollination, where the vehicle must not only hover stably but also navigate precisely between flower clusters under environmental disturbances.

The design philosophy adopted in this thesis is a **cascaded control architecture**, in which the position loop operates as the outer controller and the velocity loop serves as the inner controller. The outer loop compares the desired position with the measured position, generating reference velocities in yaw-aligned axes (forward, right, and vertical). These reference velocities are then

tracked by the inner velocity loop, which produces commanded accelerations in the world frame. Finally, the commanded accelerations are mapped into total thrust and a desired quaternion attitude, ensuring seamless integration with the lower-level attitude and motor controllers [14].

In the context of precision pollination, such a control framework enables the quadcopter to achieve sub-meter accuracy in positioning, maintain stable hovering near target flowers, and adapt to external factors such as wind gusts or foliage interference. The following subsections detail the structure and mathematical formulation of the outer position loop and the inner velocity loop.

III.4.1 Outer Position Control Loop

The outer position control loop represents the highest layer of the translational control hierarchy. Its primary objective is to regulate the quadcopter's position in three-dimensional space by generating a desired inertial-frame velocity command. Because quadcopters are underactuated systems—exerting thrust only along the body z-axis—the outer loop cannot directly generate forces or attitudes. Instead, it formulates a velocity reference that is subsequently tracked by the inner velocity loop, which operates at a higher bandwidth and manages rapid acceleration dynamics.

III.4.1.1 Control Law Formulation

Position Error: The control process begins by computing the position error in the inertial frame:

$$\mathrm{e}_P(t) = \mathrm{P}_d(t) - \mathrm{P(t)} \qquad (3.38)$$

Where:

- $P_d(t) = [x_d(t), y_d(t), z_d(t)]^\top$: The desired position
- $P(t) = [x(t), y(t), z(t)]^\top$:The current measured position.

Proportional Position Controller: A proportional control law maps the position error into the desired inertial velocity:

$$V(t) = K_p^P \mathrm{e}_P(t) \qquad (3.39)$$

Where:

$$K_p^P = \mathrm{diag}(K_p^x, K_p^y, K_p^z) \qquad (3.40)$$

Velocity Saturation: To prevent excessive or infeasible velocity commands:

$$V(t) = \text{sat}_{V_{max}}(V(t)) \qquad (3.41)$$

Where:

- V_{max}: is the maximum allowable linear velocity.

This ensures safe, smooth operation and prevents destabilization of downstream loops.

III.4.1.2 Model Based Tuning

The position dynamics are described by:

$$V(t) = \dot{P}(t) \qquad (3.42)$$

The open-loop transfer functions of the dynamics becomes:

$$G_P(s) = \frac{P(s)}{V(s)} = \frac{1}{s} \qquad (3.43)$$

The PID controller in the Laplace domain is:

$$V(s) = K_p^P E_P(s) \qquad (3.44)$$

$$C_P(s) = \frac{V(s)}{E_P(s)} = K_p^P \qquad (3.45)$$

Where:

$$E_P(s) = P_{\text{d}}(s) - P(s) \qquad (3.46)$$

Combining (3.38) and (3.40)

$$P(s) = G_P(s).C_P(s).E_P(s) \qquad (3.47)$$

From (3.41)

$$P(s) = G_P(s).C_P(s).[P_{\text{d}}(s) - P(s)] \qquad (3.48)$$

$$P(s)[1 + G_P(s).C_P(s)] = G_P(s).C_P(s).P_{\text{d}}(s) \qquad (3.49)$$

$$H_{\text{s}}(s) = \frac{P(s)}{P_{\text{d}}(s)} = \frac{G_P(s).C_P(s)}{1 + G_P(s).C_P(s)} \qquad (3.50)$$

The open-loop transfer function

$$L_P(s) = G_P(s).C_P(s) \qquad (3.51)$$

From (3.38)

$$L_P(s) = \frac{C_P(s)}{s} \qquad (3.52)$$

Substituting (3.47) back into (3.45)

$$H_s(s) = \frac{P(s)}{P_d(s)} = \frac{\frac{C_P(s)}{s}}{1 + \frac{C_P(s)}{s}} \qquad (3.53)$$

To translate the above physical model into explicit control equations, we substitute $C_P(s)$ into the closed-loop transfer function:

$$H_s(s) = \frac{P(s)}{P_d(s)} = \frac{\frac{K_p^P}{s}}{1 + \frac{K_p^P}{s}} = \frac{K_p^P}{s + K_p^P} \qquad (3.54)$$

This is a **first-order system**

$$H_s(s) = \frac{P(s)}{P_d(s)} = \frac{1}{1 + \frac{1}{K_p^P}s} \qquad (3.55)$$

$$pole = -K_p^P \qquad (3.56)$$

Time constant:

$$\tau_P = \frac{1}{K_p^P} \qquad (3.57)$$

To get a desired response speed (bandwidth) ω_{n_P}:

$$K_p^P = \omega_{n_P}, \qquad with: \quad \omega_{n_P} = \frac{1}{\tau_P} \qquad (3.58)$$

III.4.2 Inner Velocity Control Loop

The inner velocity control loop ensures that the quadcopter tracks the reference velocity $V_d(t)$ generated by the outer position loop. Operating at higher bandwidth, this loop provides rapid correction of velocity deviations and robust disturbance rejection. To achieve this, a **PI–D structure** is adopted: a proportional–integral (PI) controller acts on the velocity error, while a derivative (D) term is applied directly to the measured velocity. This formulation improves stability by damping oscillations and ensures accurate velocity tracking with minimal steady-state error.

III.4.2.1 Control Law Formulation

Velocity Error (Inertial Frame):

$$e_V(t) = V_d(t) - V(t) \qquad (3.59)$$

PI Control on Error:

$$a_{PI}(t) = K_p^V \, e_V(t) + K_i^V \int e_V(t)\, dt \qquad (3.60)$$

D Control on Measured Velocity:

$$a_D(t) = -K_d^V \, \frac{dV(t)}{dt} \qquad (3.61)$$

Commanded Acceleration (Inertial Frame):

$$a(t) = a_{PI}(t) + a_D(t) + g(t) \qquad (3.62)$$

Feedforward Compensation:

$$g(t) = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \qquad (3.63)$$

To maintain hover or execute altitude changes, the controller must offset this force by generating an upward thrust

III.4.2.2 Model Based Tuning

In Small Angles, the body-frame translational dynamics along each axis can be simplified as:

$$\dot{V}_i(t) = \frac{1}{m} F_i(t), \qquad i \in \{x, y, z\} \qquad (3.64)$$

Where:

- m is the quadcopter mass,
- F_i is the force along axis

The open-loop transfer functions of the dynamics becomes:

$$G_{V_i}(s) = \frac{V_i(s)}{F_i(s)} = \frac{1}{m.s} \qquad (3.65)$$

The PID controller in the Laplace domain is:

$$a_i(s) = \left(K_p^{V_i} + \frac{K_i^{V_i}}{s}\right) E_{V_i}(s) - K_d^{V_i} s V_i(s) \qquad (3.66)$$

Where:

$$E_{V_i}(s) = V_{d_i}(s) - V_i(s) \qquad (3.67)$$

We have:

$$m.a_i(s) = F_i(s) \qquad (3.68)$$

So:

$$F_i(s) = m\left(\left(K_p^{V_i} + \frac{K_i^{V_i}}{s}\right) E_{V_i}(s) - K_d^{V_i} s V_i(s)\right) \qquad (3.69)$$

From (3.60)

$$V_i(s) = G_{V_i}(s).F_i(s) \qquad (3.70)$$

From (3.64)

$$V_i(s) = G_{V_i}.m\left(\left(K_p^{V_i} + \frac{K_i^{V_i}}{s}\right) E_{V_i}(s) - K_d^{V_i} s V_i(s)\right) \qquad (3.71)$$

$$V_i(s) = G_{V_i}.m\left(\left(K_p^{V_i} + \frac{K_i^{V_i}}{s}\right)\left(V_{d_i}(s) - V_i(s)\right) - K_d^{V_i} s V_i(s)\right) \qquad (3.72)$$

From (3.60)

$$V_i(s) = \frac{1}{s}\left(\left(K_p^{V_i} + \frac{K_i^{V_i}}{s}\right) V_{d_i}(s) - \left(K_p^{V_i} + \frac{K_i^{V_i}}{s} + K_d^{V_i} s\right) V_i(s)\right) \qquad (3.73)$$

$$V_i(s) = \left(\frac{K_p^{V_i} s + K_i^{V_i}}{s^2}\right) V_{d_i}(s) - \left(\frac{K_d^{V_i} s^2 + K_p^{V_i} s + K_i^{V_i}}{s^2}\right) V_i(s) \qquad (3.74)$$

$$\left(1 + \frac{K_d^{V_i} s^2 + K_p^{V_i} s + K_i^{V_i}}{s^2}\right) V_i(s) = \left(\frac{K_p^{V_i} s + K_i^{V_i}}{s^2}\right) V_{d_i}(s) \qquad (3.75)$$

$$\left(\frac{\left(1+K_d^{V_i}\right)s^2+K_p^{V_i}s+K_i^{V_i}}{s^2}\right)V_i(s)=\left(\frac{K_p^{V_i}s+K_i^{V_i}}{s^2}\right)V_{d_i}(s) \qquad (3.76)$$

The closed Loop Transfer Function

$$H_{V_i}(s)=\frac{V_i(s)}{V_{d_i}(s)}=\frac{K_p^{V_i}s+K_i^{V_i}}{\left(1+K_d^{V_i}\right)s^2+K_p^{V_i}s+K_i^{V_i}} \qquad (3.77)$$

Dividing the Numerator and Denominator by $\left(1+K_d^{V_i}\right)$

$$H_{V_i}(s)=\frac{V_i(s)}{V_{d_i}(s)}=\frac{\frac{K_p^{V_i}}{\left(1+K_d^{V_i}\right)}s+\frac{K_i^{V_i}}{\left(1+K_d^{V_i}\right)}}{s^2+\frac{K_p^{V_i}}{\left(1+K_d^{V_i}\right)}s+\frac{K_i^{V_i}}{\left(1+K_d^{V_i}\right)}} \qquad (3.78)$$

Closed-Loop Characteristic Polynomial

The denominator polynomial

$$P_{V_i}(s)=s^2+\frac{K_p^{V_i}}{\left(1+K_d^{V_i}\right)}s+\frac{K_i^{V_i}}{\left(1+K_d^{V_i}\right)} \qquad (3.79)$$

Compare with the standard normalized second-order system:

$$s^2+2\zeta\omega_n s+\omega_n^2=0 \qquad (3.80)$$

By coefficient matching, the **model-based tuning relationships** are obtained:

$$\begin{cases}K_p^{V_i}=2\zeta_{V_i}\omega_{n_{V_i}}\left(1+K_d^{V_i}\right)\\ K_i^{V_i}=\omega_{n_{V_i}}^2\left(1+K_d^{V_i}\right)\end{cases} \qquad (3.81)$$

III.5 Full Control System

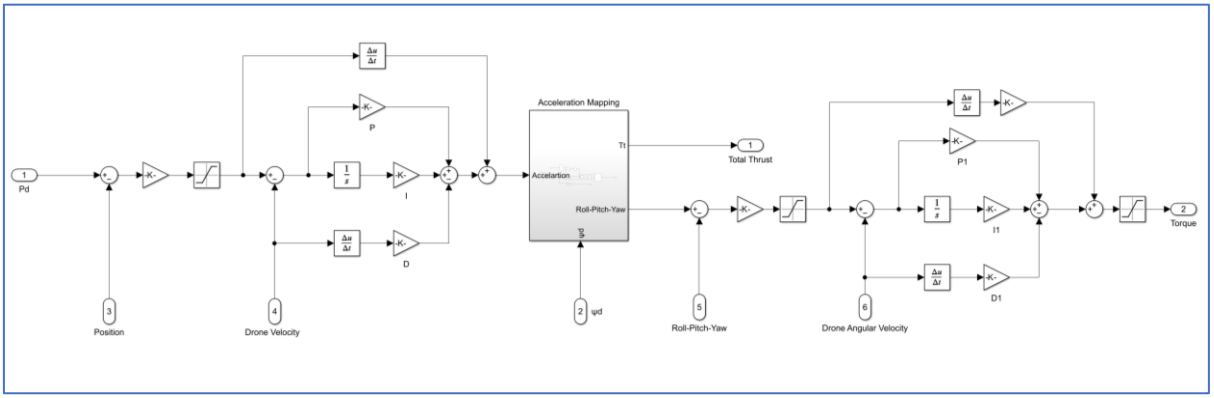

Figure III.3: Quadcopter Full Control Model

III.6 Simulation & Testing

This chapter presents the simulation and testing of the designed quadcopter control system using the Proportional–Integral–Derivative (PID) controller. The purpose of these simulations is to validate the controller's ability to stabilize the quadcopter and maintain desired flight conditions under different operating scenarios.

The tests are performed on the complete nonlinear dynamic model of the quadcopter developed in Chapter II. The control structure implemented is the cascaded PID configuration, consisting of an inner loop for attitude stabilization and an outer loop for altitude control. Simulations are conducted in the MATLAB/Simulink environment with appropriate parameters corresponding to the selected A2212/13T brushless DC motors and quadcopter configuration.

To comprehensively assess controller performance, several simulation scenarios are considered, including hovering stability, altitude step response, attitude control, wind disturbance, payload variation, load disturbance, and sensor noise with delay. Each scenario provides insight into a particular aspect of controller robustness and accuracy. Key performance metrics such as rise time, settling time, overshoot, steady-state error, and control effort are used to evaluate the results.

The subsequent sections describe each simulation scenario, the applied inputs, and the corresponding system responses through time-domain plots and performance analysis.

III.6.1 Simulation Setup and Parameters

To validate the control system performance and dynamic behavior of the quadcopter, a high-fidelity simulation environment was constructed using MATLAB/Simulink. The drone model is based on the DJI F450 frame, selected for its widespread use in research and prototyping. The

simulation incorporates aerodynamic effects, actuator dynamics, and nonlinear feedback loops to ensure realistic system response.

III.6.2 Physical Configuration

Parameter	Value	Description
Frame Type	DJI F450	Quadcopter frame type
Arm Length	$0.225\ m$	Center-to-rotor distance
Payload	$0.15\ kg$	Additional payload mass
Mass	$1.00\ kg$	Total mass (frame + payload)
g	$9.81\ m/s^2$	Gravitational acceleration
$\boldsymbol{\rho}$	$1.225\ kg/m^3$	Air density at sea level

Table III.3: Quadcopter Physical Parameters

III.6.3 Aerodynamic and Actuator Parameters

Parameter	Value	Description
$\boldsymbol{K_T}$	$1.33 \times 10^{-5}\ \mathrm{s^2/rad^2}$	Thrust coefficient
$\boldsymbol{K_D}$	$1.14 \times 10^{-7}\ \mathrm{N \cdot m \cdot s^2/rad^2}$	Yaw drag coefficient
$\boldsymbol{C_D}$	$\begin{bmatrix}0.25\\0.25\\0.40\end{bmatrix}$	Translational drag coefficients (x, y, z)
$\boldsymbol{A}$	$0.03\ m^2$	Frontal area per axis
$\boldsymbol{D_a}$	$\begin{bmatrix}1&0&0\\0&1&0\\0&0&2\end{bmatrix} \times 10^{-4}\ N.m.s/rad$	Angular damping matrix
$\boldsymbol{K_Q}$	$3.1 \times 10^{-7} N.m.s^2$	Propeller Drag Torque

Table III.4: Quadcopter Aerodynamic and Actuator Parameters

III.6.4 Inertial Properties

The quadcopter's rotational inertia is modeled using a diagonal inertia tensor:

$$\boldsymbol{I} = \begin{bmatrix} 8.0 \times 10^{-3} & 0 & 0 \\ 0 & 8.0 \times 10^{-3} & 0 \\ 0 & 0 & 1.6 \times 10^{-2} \end{bmatrix} kg \cdot m^2 \qquad (3.82)$$

Rotor gyroscopic effects are included via rotor inertia:

$$Jr = 3.5 \times 10^{-5}\, kg \cdot m^2 \qquad (3.83)$$

III.6.5 Hover and Rotor Speed Conditions

The hover thrust per rotor is computed as:

$$T_{\text{hover}} = \frac{m \cdot g}{4} = \frac{1.00 \cdot 9.81}{4} \approx 2.45\ \text{N} \qquad (3.84)$$

Corresponding hover angular velocity:

$$\omega_{\text{hover}} = \sqrt{\frac{T_{\text{hover}}}{K_T}} \approx 429\ \text{rad/s} \qquad (3.85)$$

Maximum rotor speed:

$$\omega_{\text{max}} = \frac{7600 \cdot 2\pi}{60} \approx 796\ \text{rad/s} \qquad (3.86)$$

III.6.6 Control Loop Dynamics

The quadcopter control system employs cascaded PID architectures for both altitude and attitude regulation. This separation improves responsiveness and stability by allowing fast inner-loop damping and slower outer-loop tracking.

Axis	Loop	Controlled Variable	PID Gains
$x-y$ Position	Outer	$x-y$	$K_p = 1$
	Inner	$u-v$	$\begin{cases} K_p = 2.8 \\ K_i = 2 \\ K_d = 1 \end{cases}$
Altitude	Outer	z	$K_p = 1$
	Inner	w	$\begin{cases} K_p = 5.6 \\ K_i = 8 \\ K_d = 1 \end{cases}$
Roll - Pitch	Outer	$\phi-\theta$	$K_p = 5.56$
	Inner	$p-q$	$\begin{cases} K_p = 0.315 \\ K_i = 5.625 \\ K_d = 0.001 \end{cases}$
Yaw	Outer	ψ	$K_p = 2.78$
	Inner	r	$\begin{cases} K_p = 0.434 \\ K_i = 3.1 \\ K_d = 0.015 \end{cases}$

Table III.5: PID Controllers Gain Values

III.6.7 Simulation Model

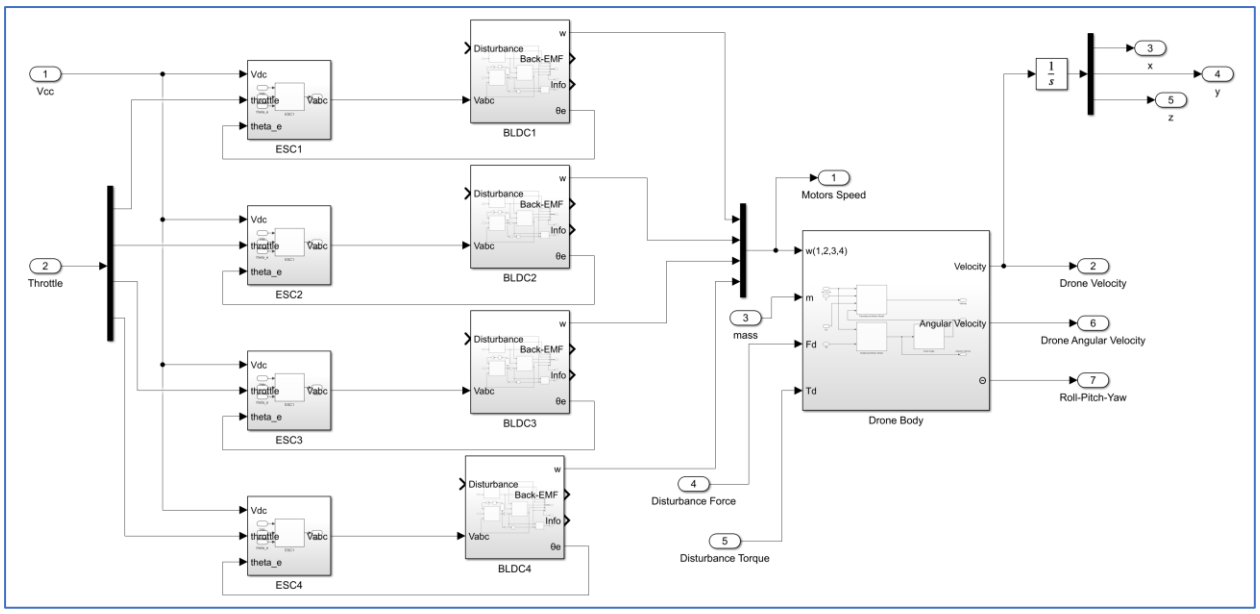

Figure III.4: Quadcopter Simulation Mode

III.6.8 Simulation Scenarios and Results

This section presents the simulation scenarios used to evaluate the performance and robustness of PID control system. Each test focuses on a specific aspect of the quadcopter's behavior, including hovering, altitude regulation, attitude stabilization, and response to disturbances such as wind, payload variation, noise, and delays.

All simulations were executed in MATLAB/Simulink with a total simulation duration of 40 s.

Initial conditions:

- Position: $\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} (m)$
- Attitude angles: $\begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} (°)$
- Battery Voltage: $V_{DC_0} = 11.1\ V$

III.6.8.1 Altitude Step Response

To evaluate the performance of the outer-loop PI controller during altitude changes and measure time-domain performance indices.

Simulation Setup

- Altitude command step: z_{ref} changes from 0.0 m to 2.0 m at $t = 5$ s.
- Attitude angles: $\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ (°)
- No external disturbances or noise.

PID Control Response

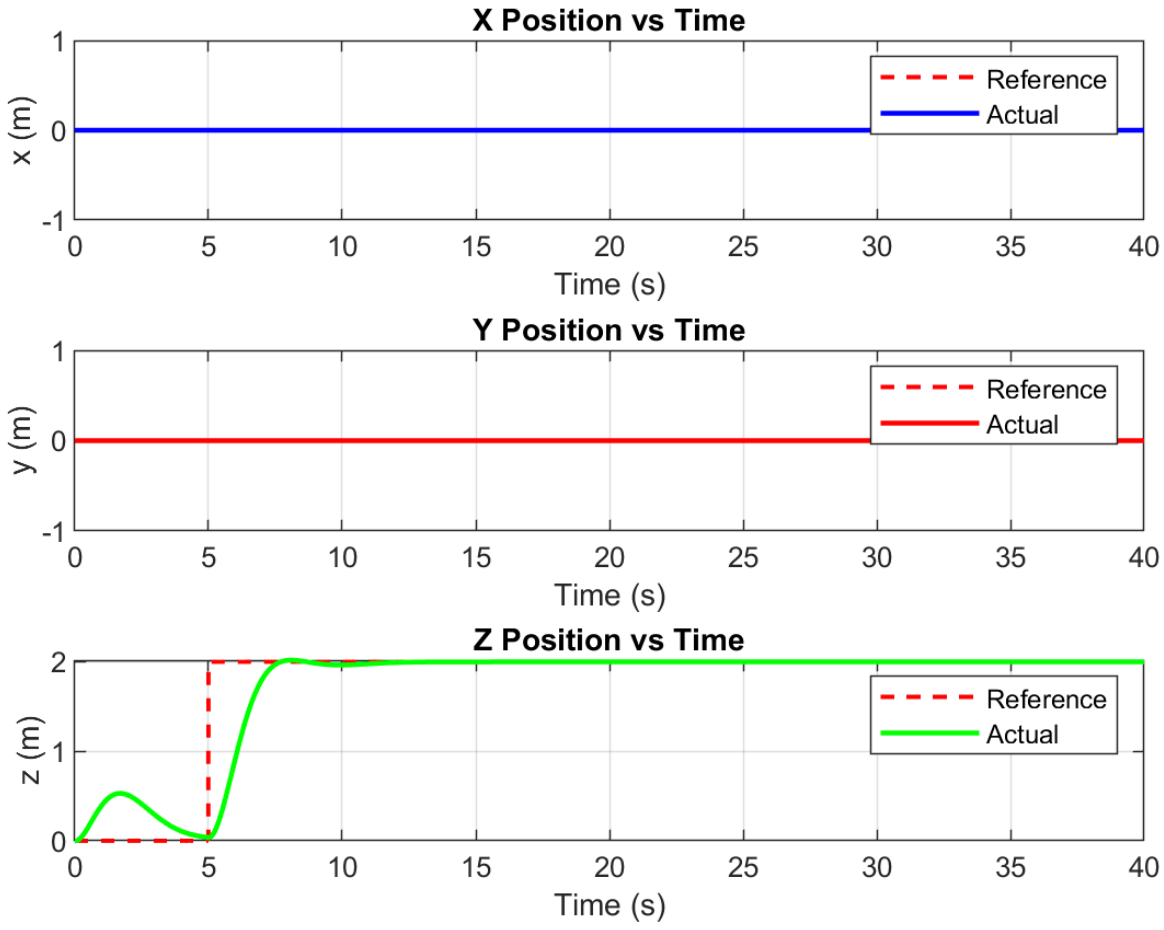

Figure III.5: Simulation Position Response

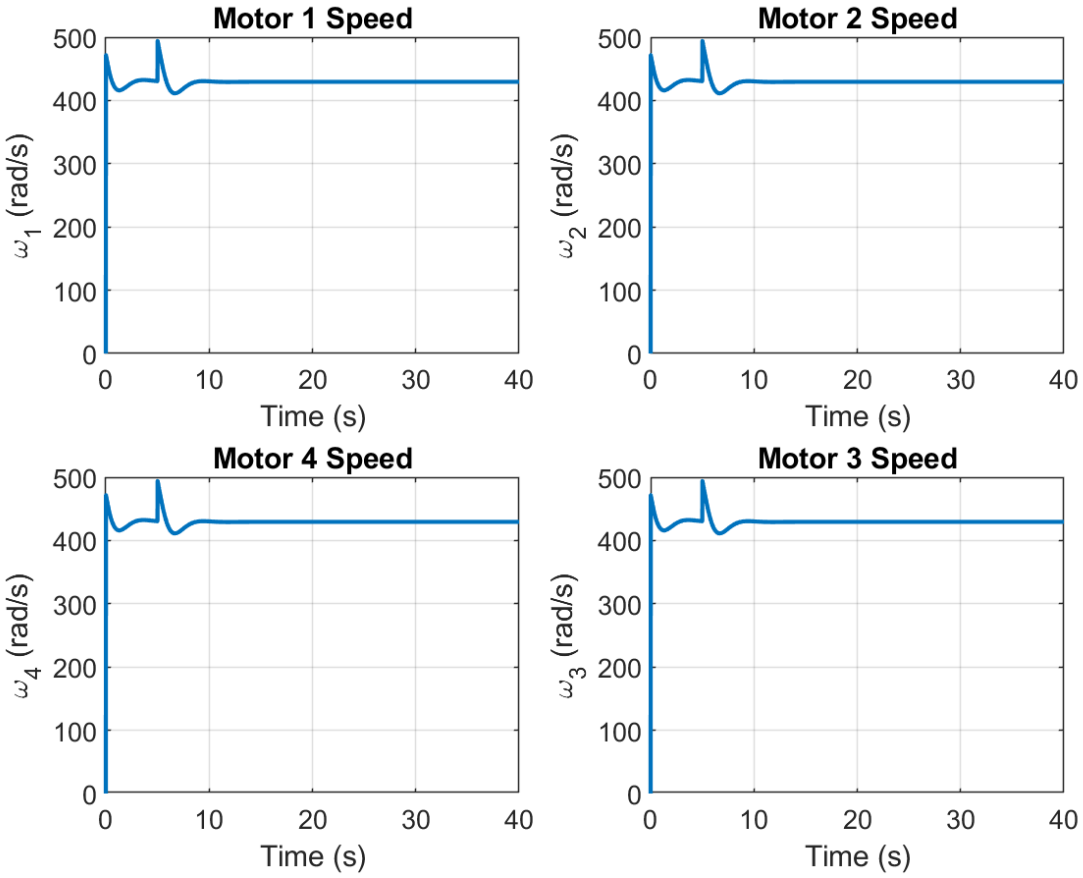

Figure III.6: Simulation Motor Speeds

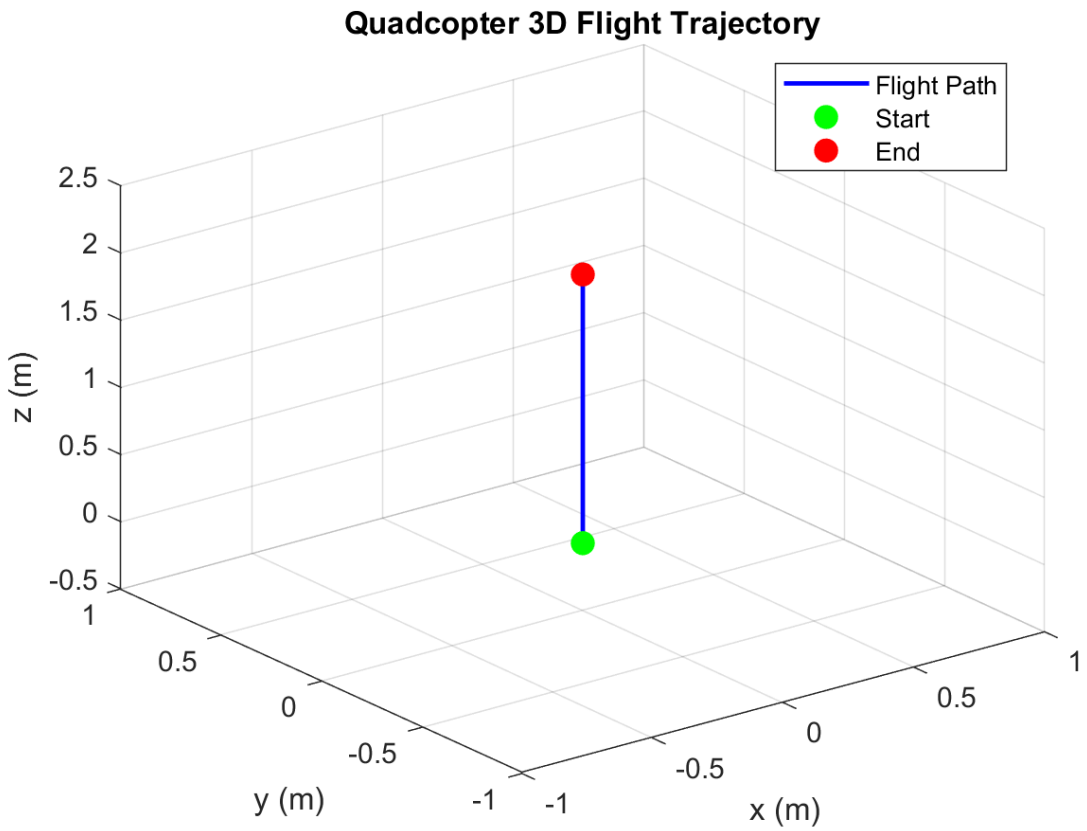

Figure III.7: Simulation Trajectory Response

III.6.8.2 Attitude Control Test

To verify the inner-loop PD controllers' performance for roll, pitch, and yaw stabilization during commanded orientation changes.

Simulation Setup

- Roll command: $\phi_{ref} = +10°$(1–5 s).
- Pitch command: $\theta_{ref} = -10°$(8–12 s).
- Yaw command: $\psi_{ref} = +15°$(from 15 s onward).
- Altitude reference fixed at 1.0 m.

PID Control Response

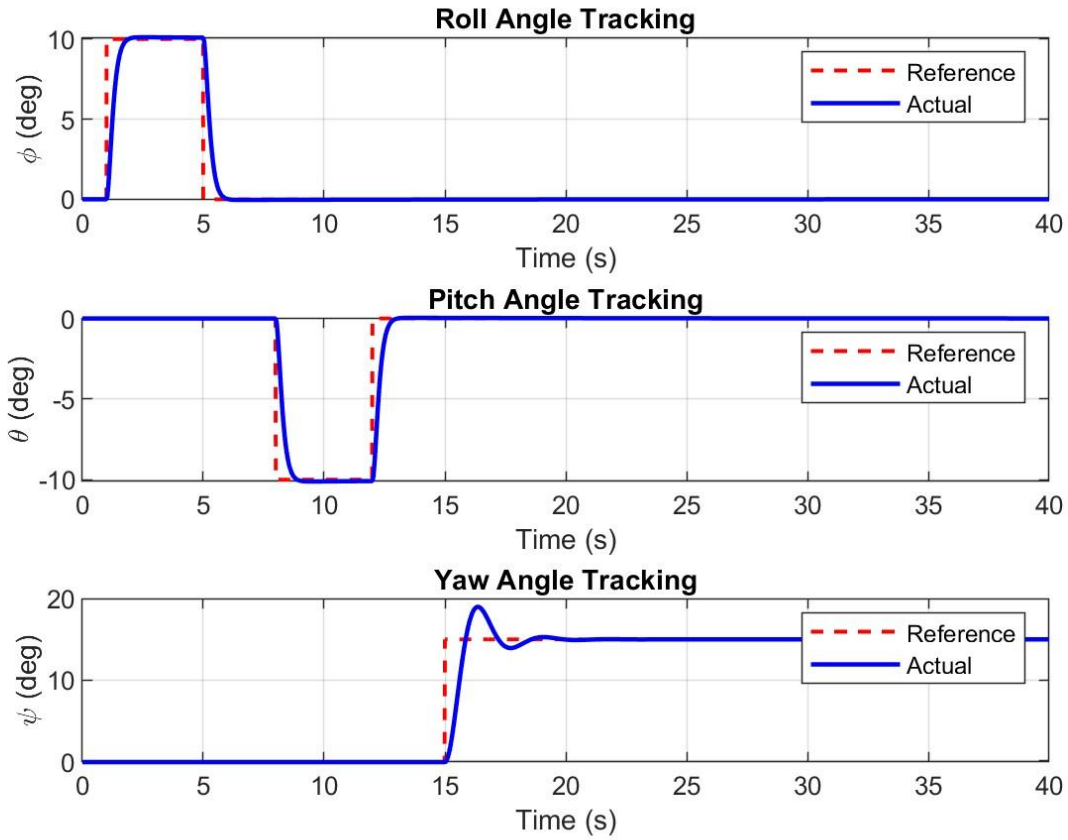

Figure III.8: Simulation Attitude Response

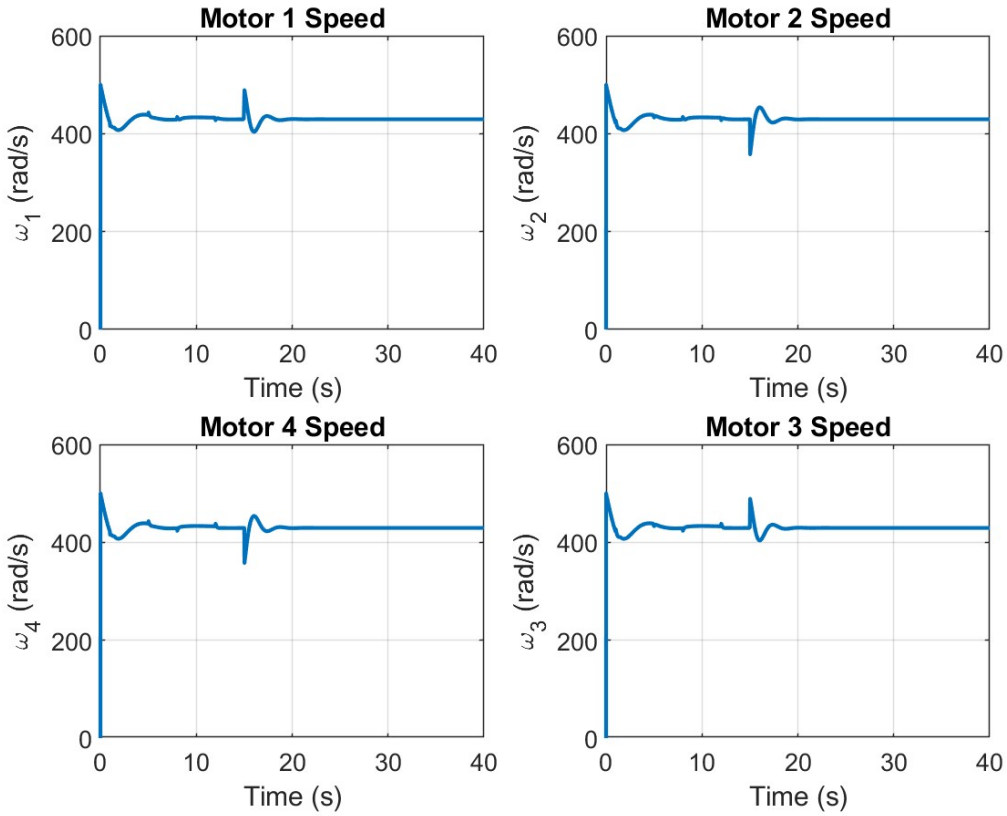

Figure III.9: Simulation Motor Speeds

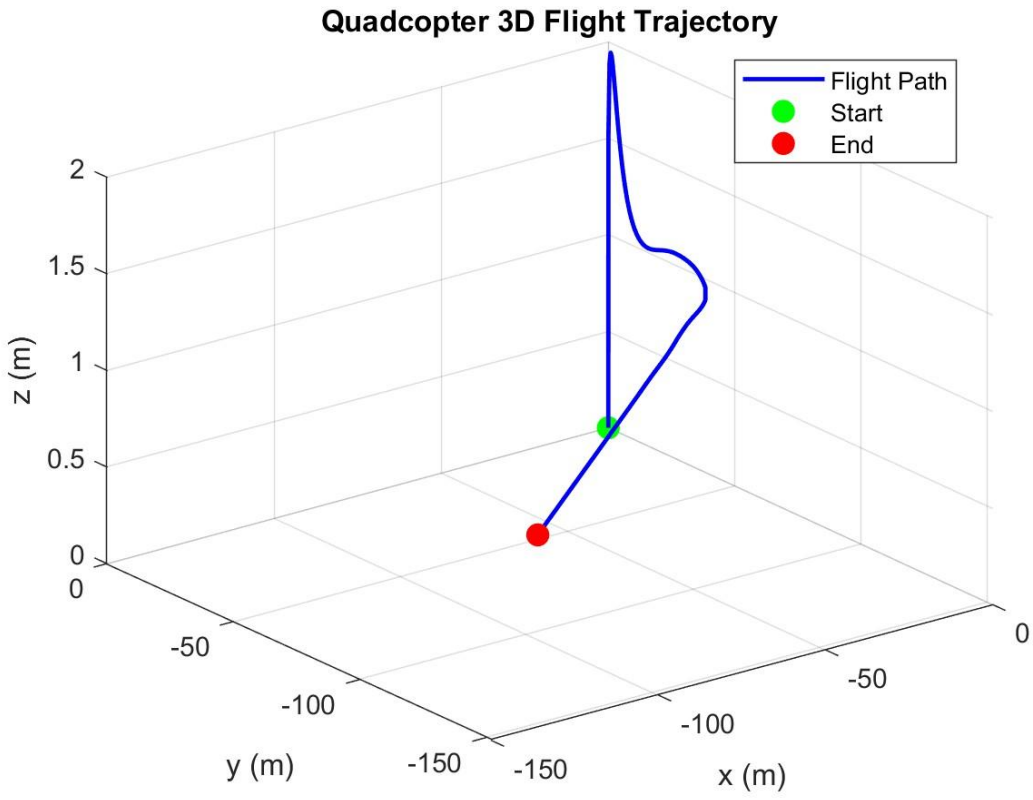

Figure III.10: Simulation Trajectory Response

III.6.8.3 Trajectory Tracking

Test coordinated control of altitude and attitude for path following.

Simulation Setup

- Trajectory: Circular path (4m × 4m) at: 10s.
- Altitude reference fixed at 1.5 m.
- No external disturbances or noise.

PID Control Response

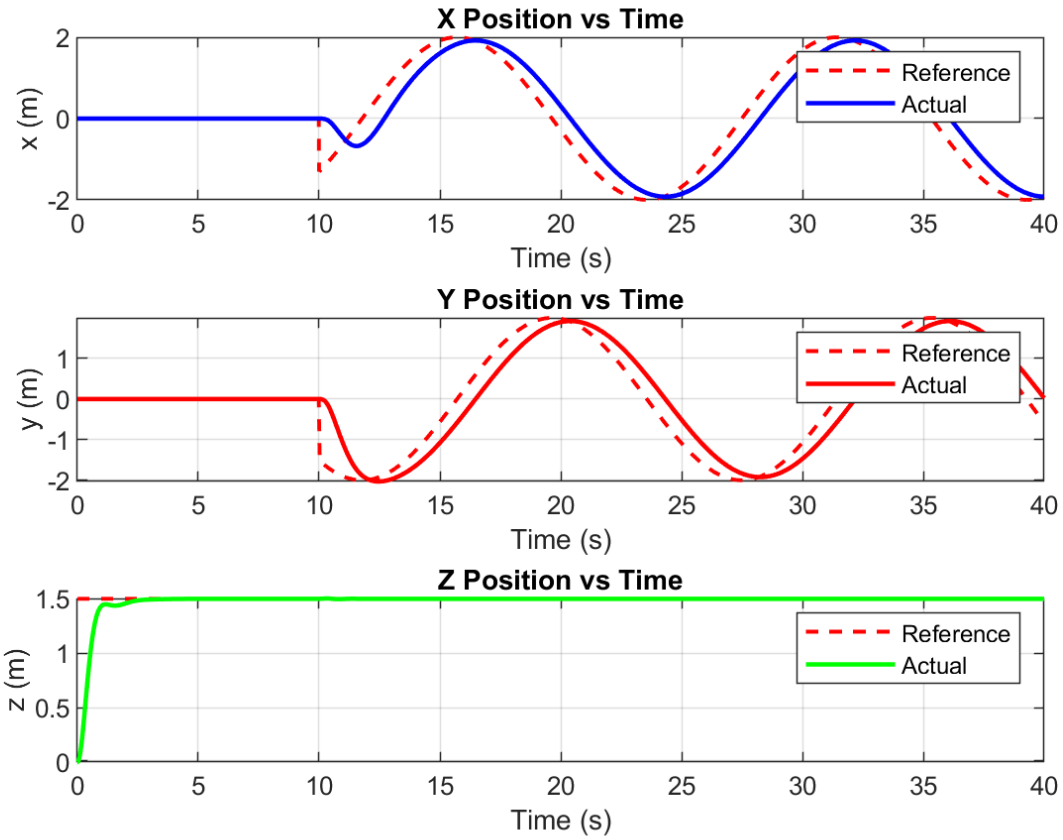

Figure III.11: Simulation Position Response

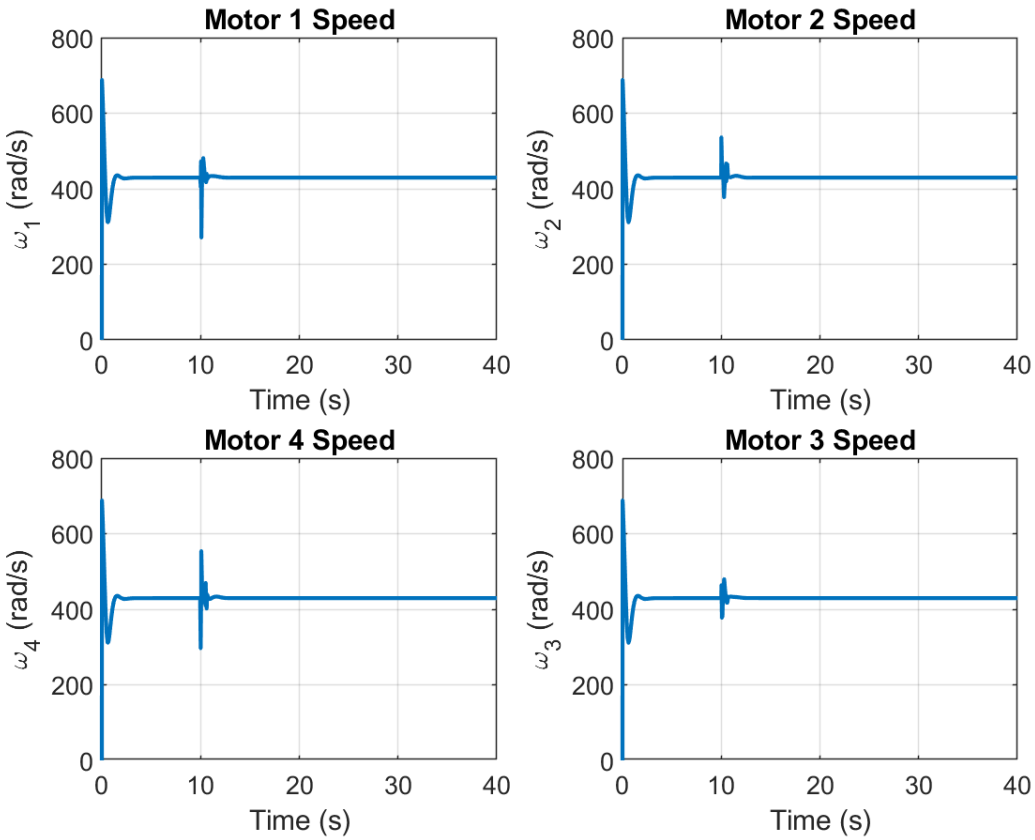

Figure III.12: Simulation Motor Speeds

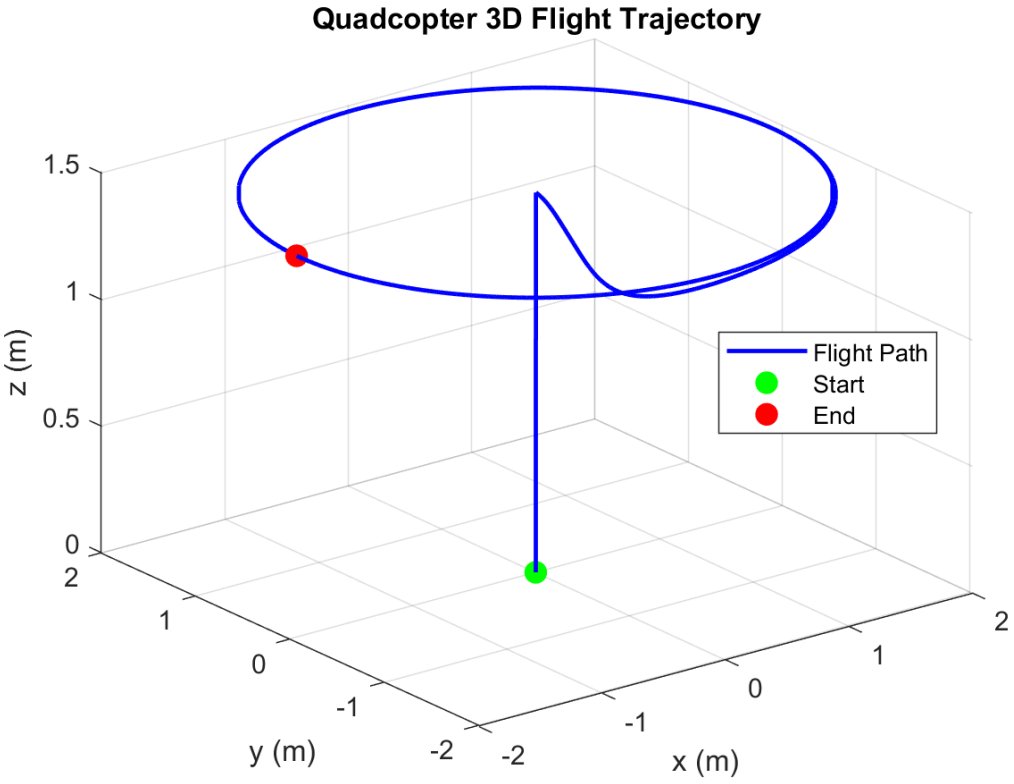

Figure III.13: Simulation Trajectory Response

III.6.8.4 Wind Disturbance Rejection

To evaluate the quadcopter's robustness when subjected to external aerodynamic disturbances such as wind gusts.

Simulation Setup

- Introduce wind torque/force.
- Altitude reference fixed at 1.5 m.

To evaluate robustness under external perturbations, two disturbance magnitudes were selected: an external forces and a rotational torque. The force is representative of the aerodynamic drag generated by an outdoor wind gust of approximately $6 - 10\ m/s$ acting on a small quadcopter

Similarly, the disturbance torque corresponds to the rotational moments produced by asymmetric aerodynamic loads, propeller imbalance, or crosswind-induced rolling motion.

PID Control Response

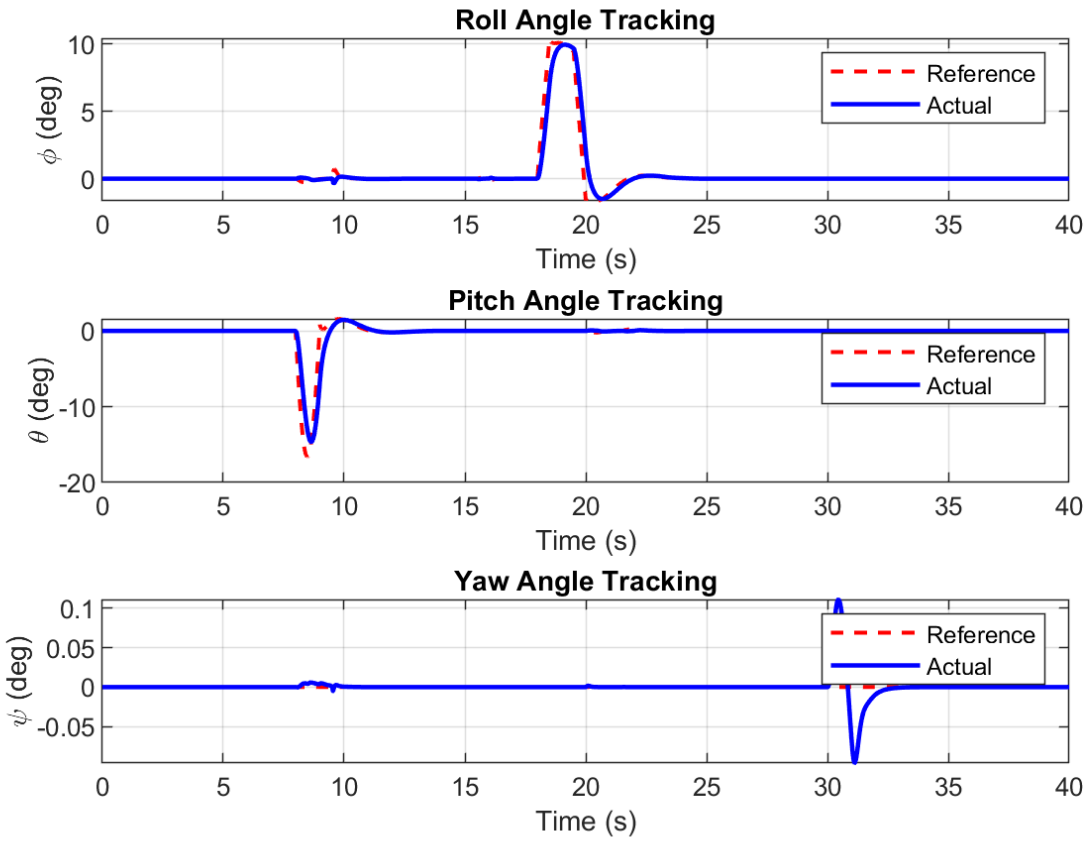

Figure III.14: Simulation Attitude Response

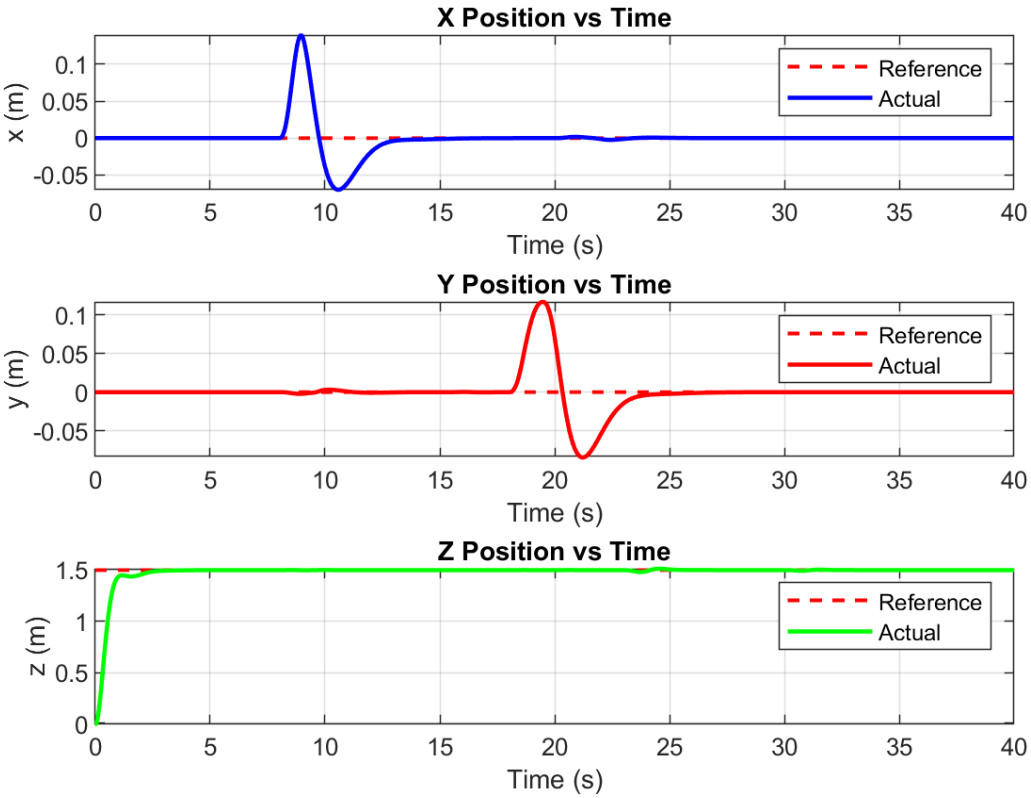

Figure III.15: Simulation Position Response

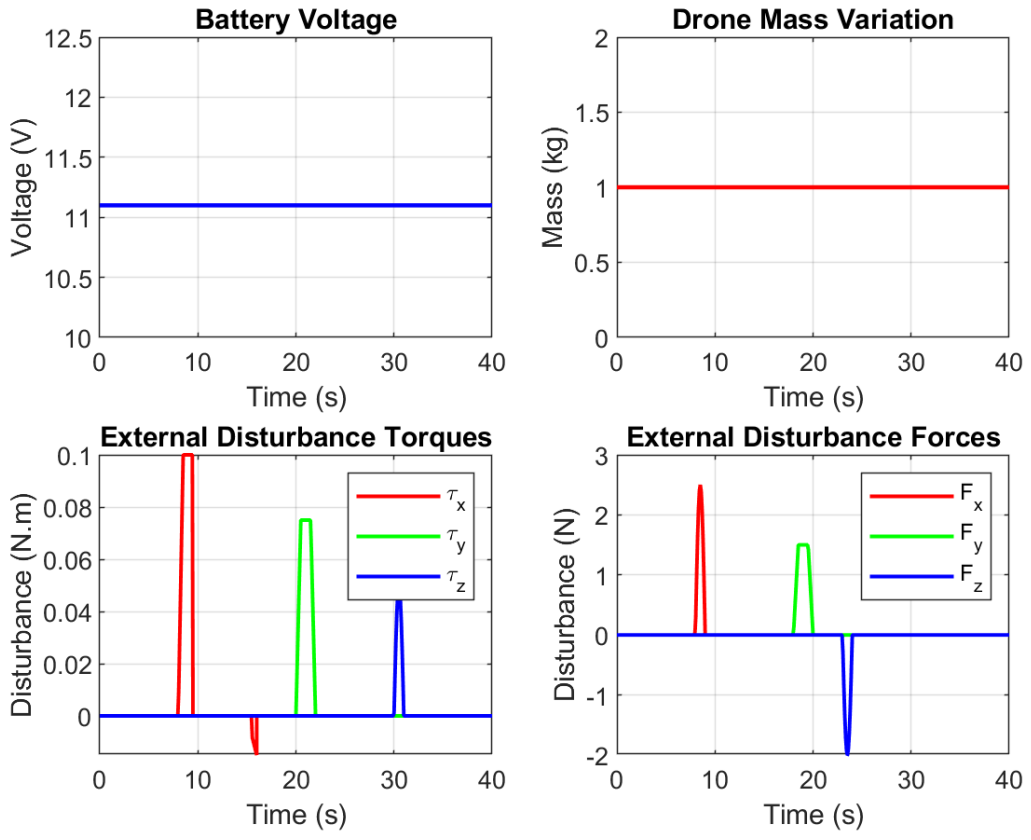

Figure III.16: Simulation Disturbances

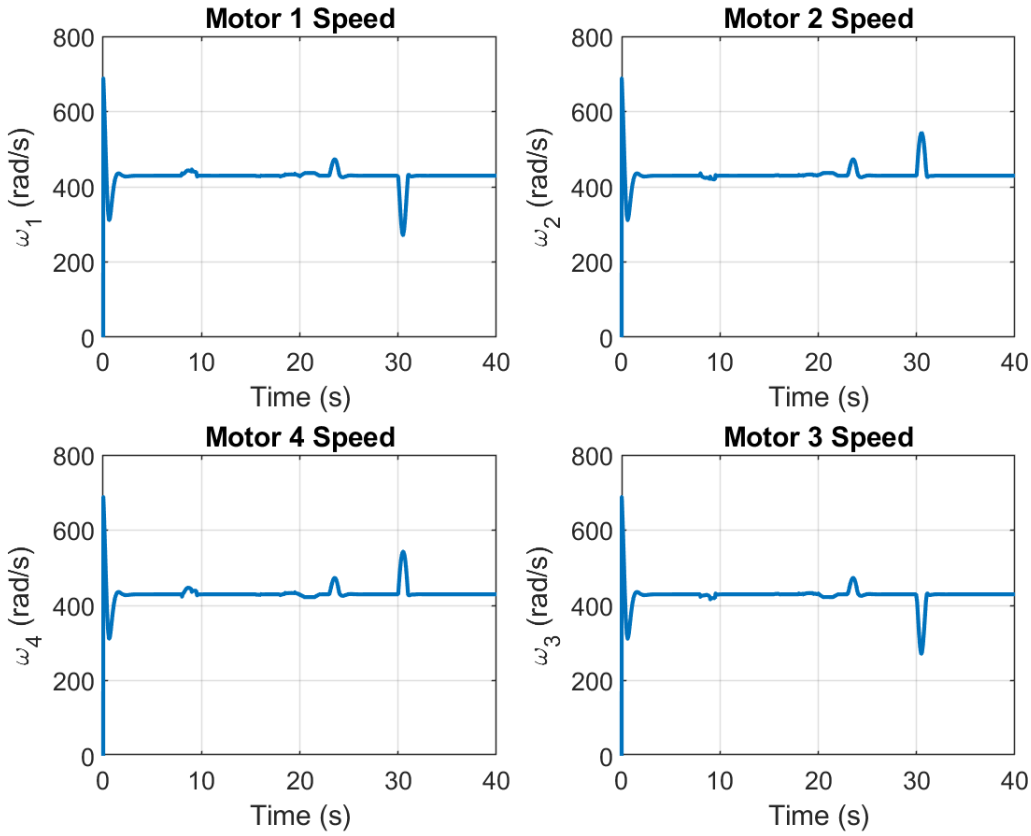

Figure III.17: Simulation Motor Speeds

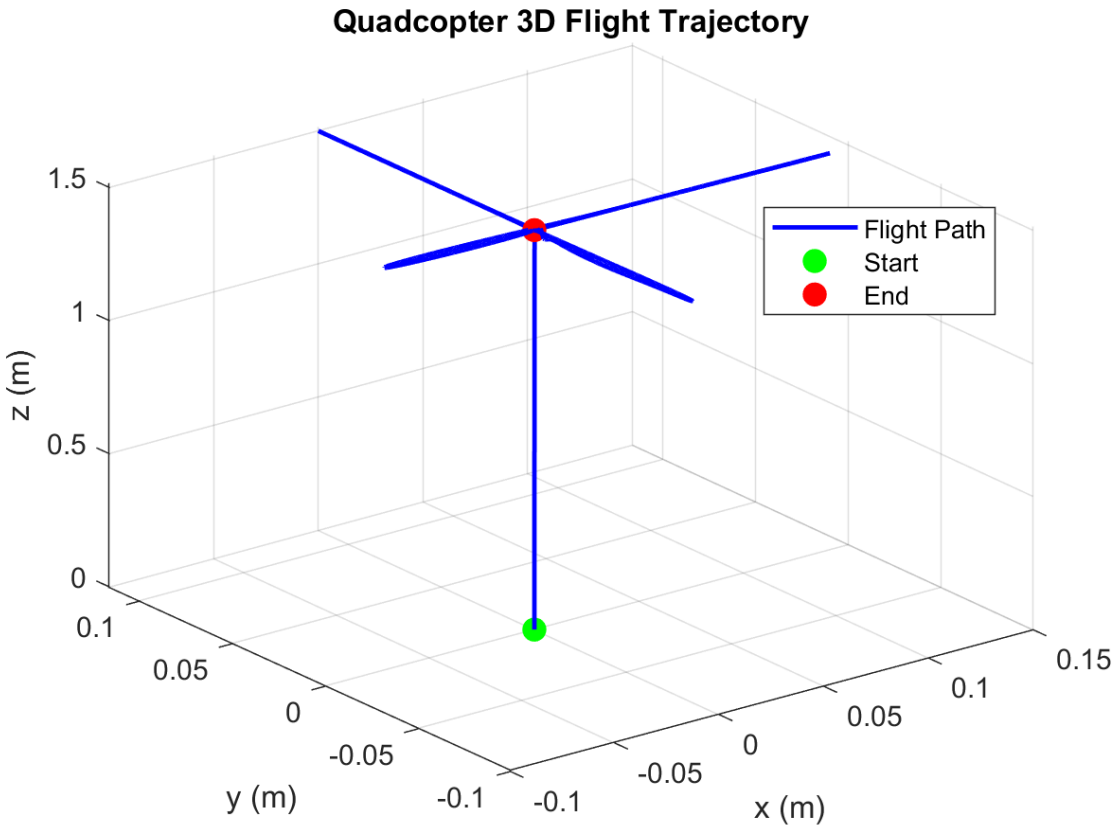

Figure III.18: Simulation Trajectory Response

III.6.8.5 Payload Effect Test

To assess the controller's ability to maintain stability and altitude when the quadcopter experiences both an increase and a decrease in total mass, simulating payload pickup and release.

Simulation Setup

- **Mass Increase (Payload Pickup):** At $t = 8\ s$, the vehicle's mass is increased by $500\ g$ to simulate attaching a small payload.
- **Mass Decrease (Payload Release):** At $t = 18\ s$, the added mass is removed, returning the system to its nominal mass.
- Altitude reference fixed at 1.5 m.
- No other disturbances are added.

PID Control Response

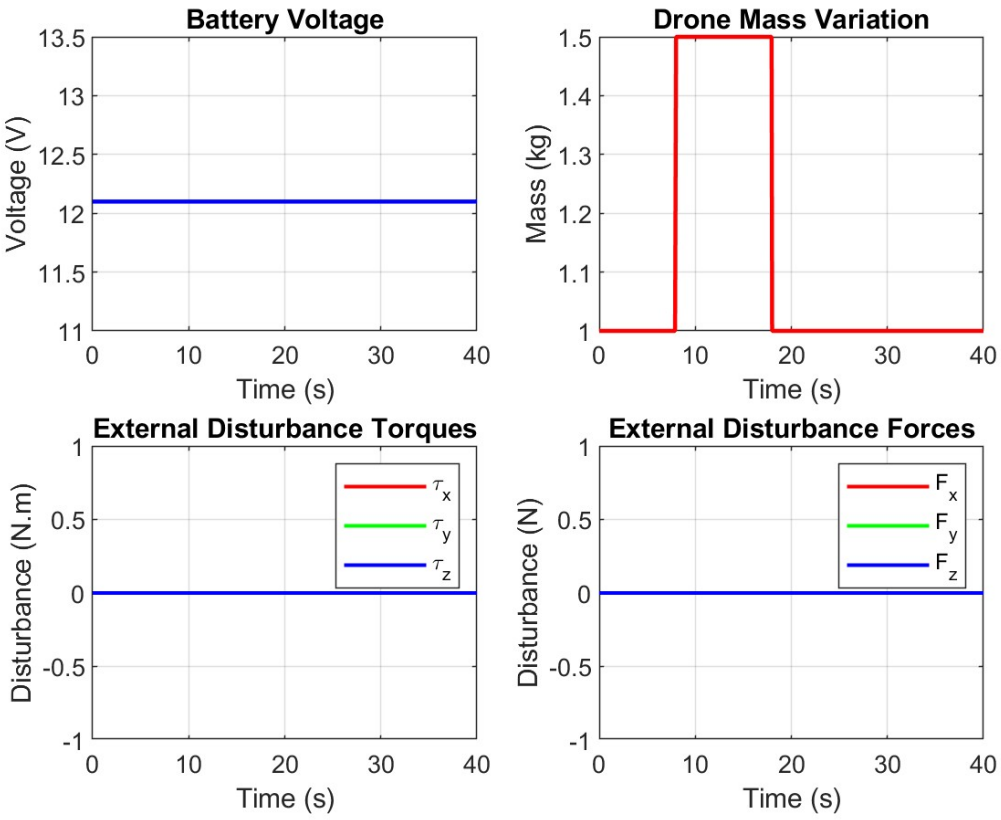

Figure III.19: Simulation Disturbances

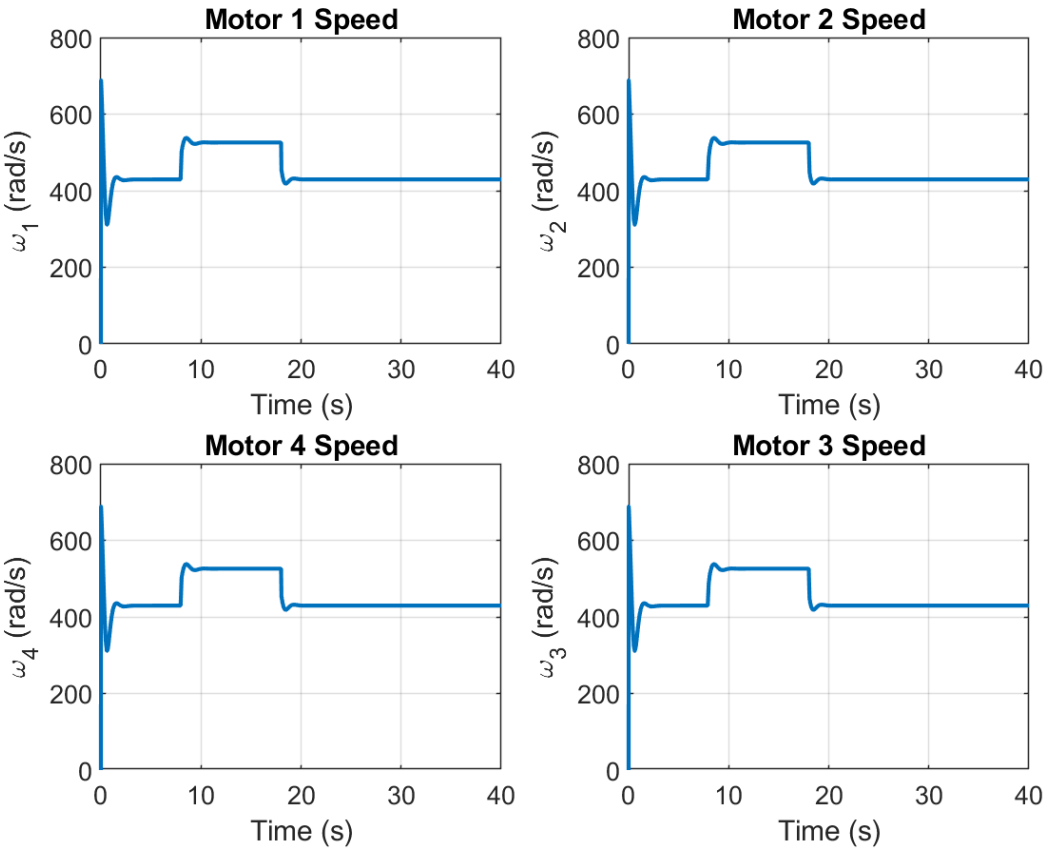

Figure III.20: Simulation Motor Speeds

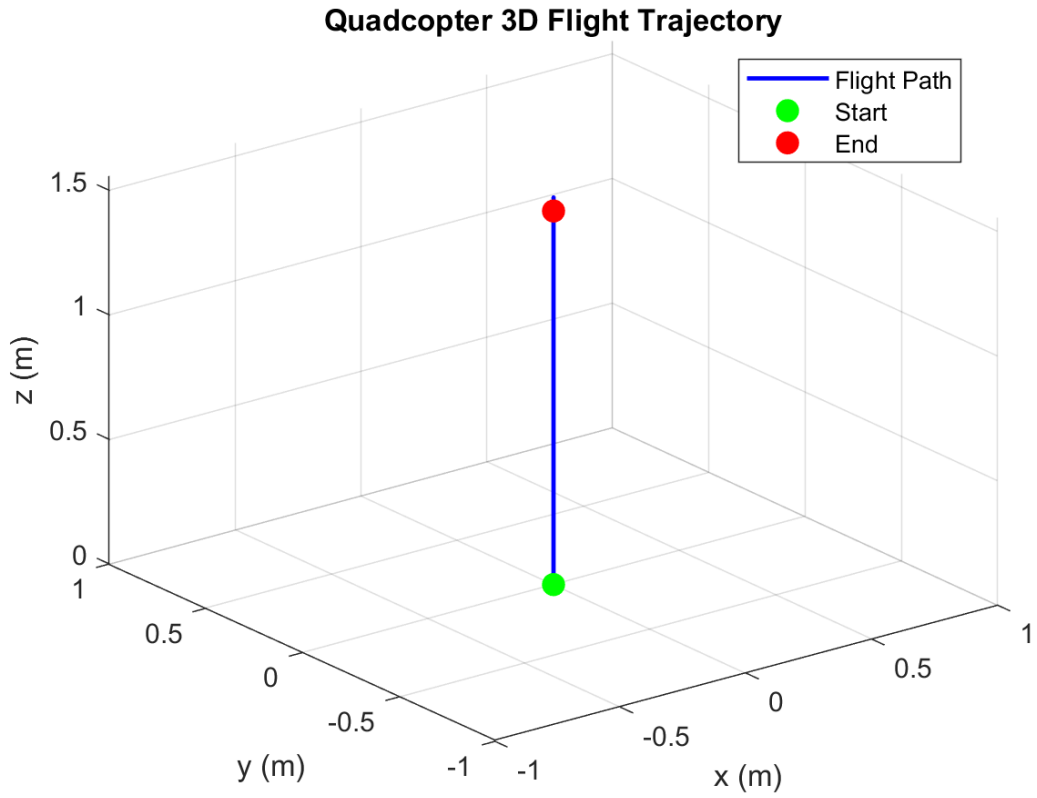

Figure III.21:Simulation Trajectory Response

III.6.8.6 Battery Voltage Drop Test

To evaluate the quadcopter's stability and control performance under decreasing battery voltage, which reduces the maximum available motor speed and thrust. This test simulates the natural discharge behavior of a Li-Po battery during flight.

Simulation Setup

- Initial voltage: **11.1 V** at $t = 0$ s.
- Final voltage: **9.0 V** at $t = 40$ s.
- Linear drop : $v(t) = 12 - 0.0525t$
- Sudden Voltage drop $7s - 10s$: $\pm 3V$
- Altitude reference fixed at 1.5 m.
- No other disturbances are added.

PID Control Response

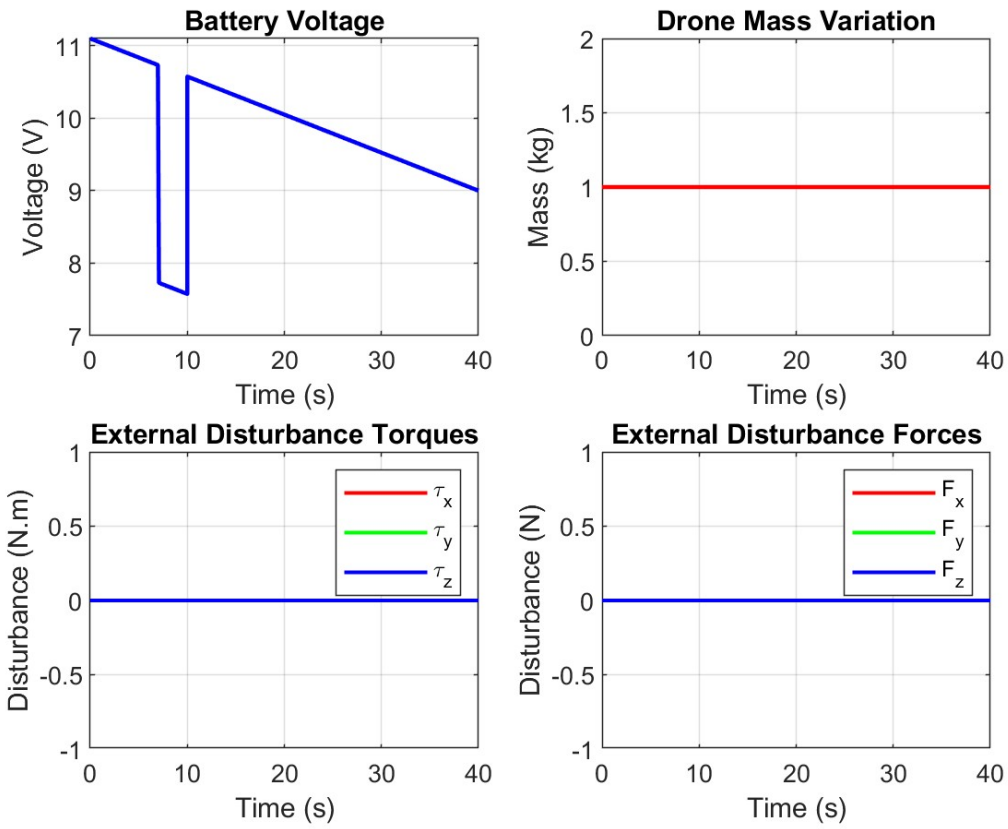

Figure III.22: Simulation Disturbances

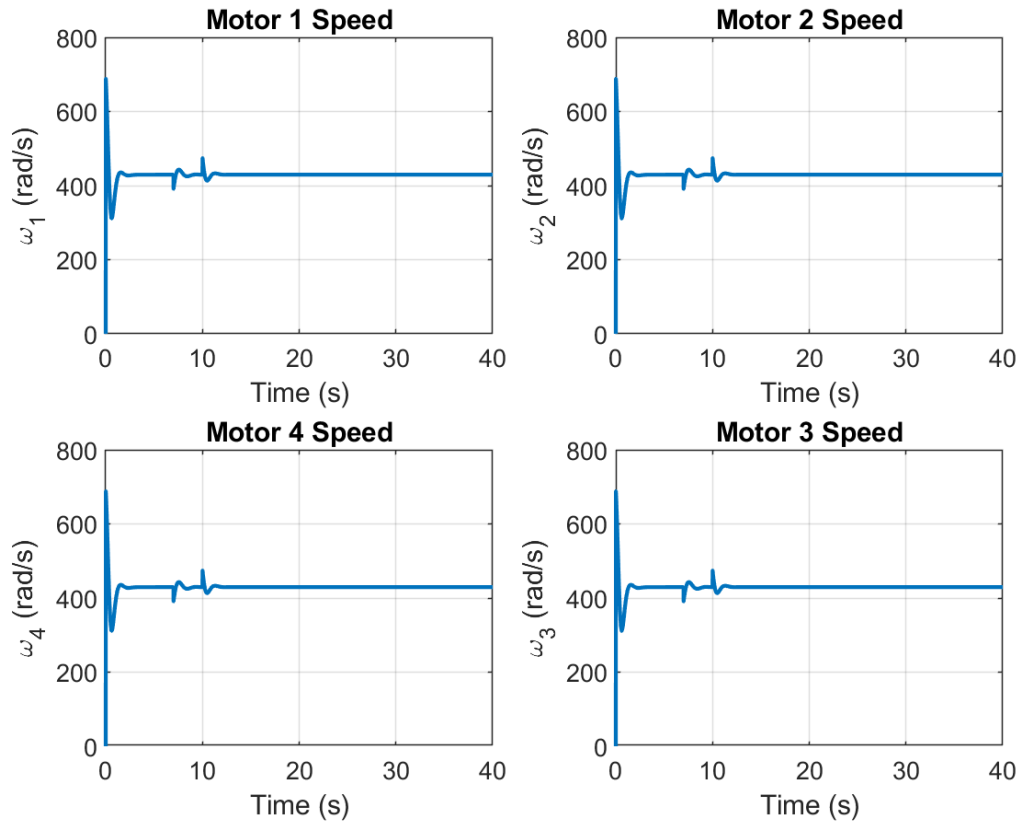

Figure III.23: Simulation Motor Speeds

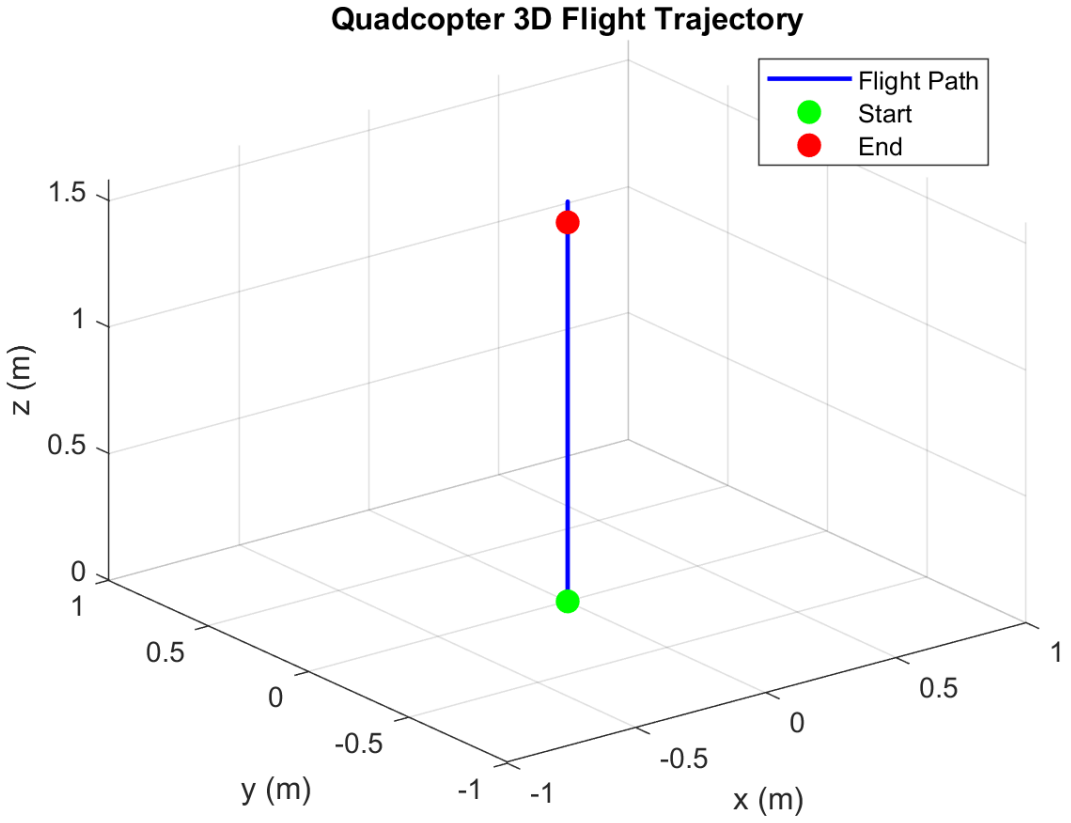

Figure III.24: Simulation Trajectory Response

III.6.8.7 Sensor Noise and Delay Test

To evaluate the quadcopter's control performance under realistic measurement imperfections, including sensor noise and feedback delay.

Simulation Setup

Gaussian Sensor Noise: Noise is added to each sensed variable:

- **Position noise:** standard deviation: $\sigma_z = 0.1\,m$ representing approximately 20% of nominal position.
- **Attitude noise:** standard deviation : $\sigma_{\phi,\theta,\psi} = 2°$ consistent with typical low-cost IMU sensors.
- **Feedback Delay:** A fixed delay is applied to the feedback signals to simulate sensor processing and latency: $\tau_{delay} = 0.03\,s\ (30\,ms)$ affecting position and attitude measurements.
- Altitude step command (0 m → 1 m) under noise and delay.

PID Control Response

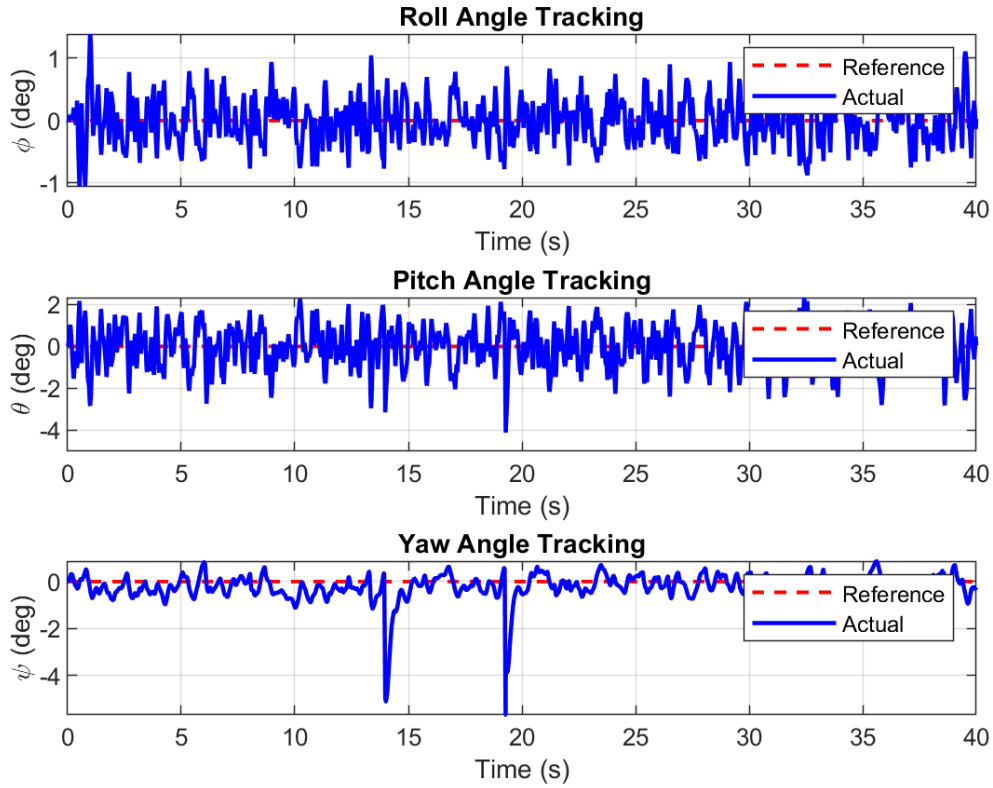

Figure III.25: Simulation Attitude Response

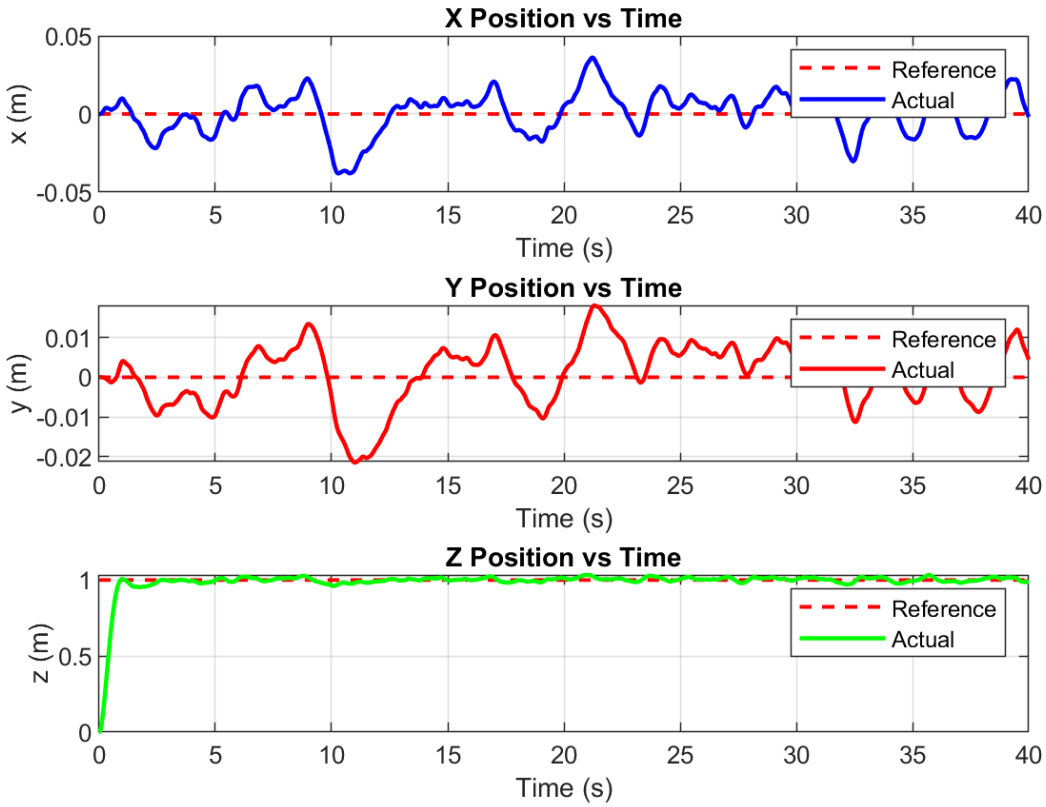

Figure III.26: Simulation Position Response

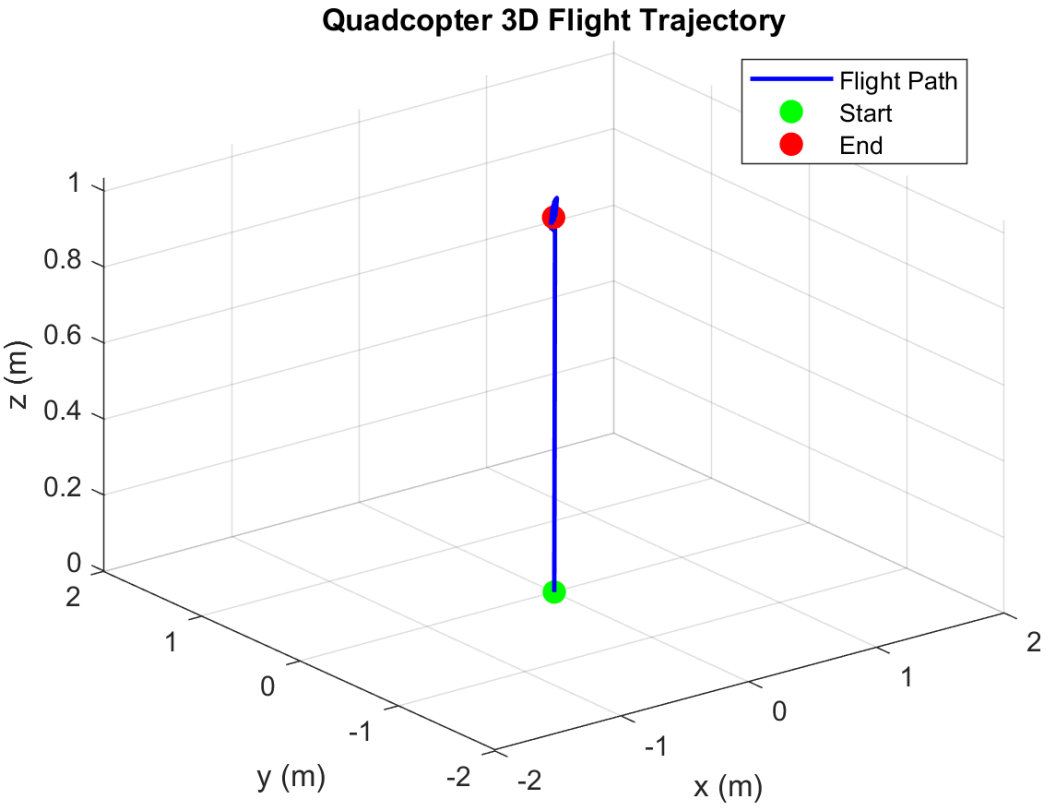

Figure III.27: Simulation Trajectory Response

III.7 Summary

This chapter introduces the cascaded control architecture required to stabilize and maneuver the quadcopter with high precision. Since quadrotors are inherently unstable and underactuated systems, the chapter begins by formulating the control objectives associated with pollination tasks: accurate hovering, smooth trajectory tracking, and rapid disturbance rejection. These objectives motivate a hierarchical control strategy that separates fast rotational dynamics from slower translational behavior.

The chapter then details the design of the inner rate controllers, outer attitude controllers, and the position/velocity regulation layers. Model-based tuning ensures that each controller operates at the appropriate bandwidth, enabling fast stabilization without causing interaction between control layers. This modularity simplifies implementation and ensures predictable dynamic responses.

The simulation and testing results presented in this chapter demonstrate the effectiveness and robustness of the cascaded PID control architecture applied to the quadcopter platform. Through a series of structured scenarios, the controller's performance was evaluated under normal operating conditions as well as under various disturbances and system uncertainties.

The results show that the PID controller is capable of achieving stable position regulation, precise attitude tracking, and smooth trajectory following. In nominal conditions, the system exhibits fast response, minimal overshoot, and small steady-state errors, confirming that the tuning of the altitude and attitude loops is appropriate for the quadcopter's dynamic characteristics.

Chapter IV

IV. Autonomous Navigation

IV.1 Introduction

For a drone to pollinate date palms autonomously, it must be able to move intelligently, understand its surroundings, and make decisions during flight. This capability is known as autonomous navigation. It goes beyond simple GPS waypoint flying: the drone must position itself close to tall palm crowns, approach flower clusters safely, avoid branches, and adapt to the natural irregularity of oasis plantations.

Unlike large industrial orchards, oasis farms often have mixed layouts. Trees may form partial rows, arcs, clusters, or even random distributions depending on water availability, terrain, and the age of the plantation. This irregularity means the drone cannot rely on strict geometric routes. Instead, it must navigate based on flexible planning, local sensing, and real-time adaptation.

This chapter explains the navigation concepts, the common approaches used in robotics, and the custom method adopted in this work. The focus is to achieve safe, simple, and reliable autonomous flight suitable for date-palm pollination.

IV.2 What Autonomous Navigation Means for a Pollination Drone

Autonomous navigation combines several tasks:

- **Understanding the environment**: Using GPS, sensors, and minimal mapping information.
- **Deciding where to fly next:** Visiting each palm tree or pollination zone in an efficient way.
- **Moving safely between obstacles:** Avoiding trunks, branches, fronds, and unexpected objects such as workers or tools.
- **Positioning precisely near flowers:** Slowing down, adjusting horizontally, and controlling altitude to reach pollination distance.
- **Reacting to uncertainty:** Wind, moving fronds, varying visibility, or slight GPS drift.

Navigation is therefore a behavioral system, not just a path planner. The drone must constantly update its decisions using sensor feedback and simple mission rules.

IV.3 Common Navigation Strategies and Their Limitations in Oases

To understand our approach, it is important to look at well-known navigation methods used in robotics.

- **GPS Waypoint Navigation:**

  The simplest method: move from one GPS point to another.

  Limitation: Palm crowns are tall and wide. A GPS point alone cannot ensure safe positioning near the crown, especially when flowers are hidden behind fronds.

- **Grid or Row-Based Coverage**

  Often used for spraying or mapping: the drone follows a straight grid pattern or row-by-row path.

  Limitation: Oasis fields are rarely perfect rows. Many sections have irregular spacing. Some palms lean. Others are placed wherever irrigation allows.

- **Full 3D Mapping and SLAM**

  Used in advanced robotics to reconstruct the environment.

  Limitation: SLAM requires heavy computing, powerful sensors, and large memory—too much for a lightweight pollination drone. It also struggles with repetitive textures such as palm leaves.

Classical navigation solutions fail because oasis fields present irregular geometry, dense overhead obstacles, and precision requirements that most systems do not handle well on small UAV hardware.

IV.4 Navigation Approach for Oasis Pollination

Instead of forcing the environment into a strict geometric model, this navigation approach is built around three simple layers that work together:

Layer 1: Global Tree-to-Tree Navigation (Big Movements)

The drone starts with a basic list of tree positions or pollination zones. These do not need to be in rows. They can be:

- roughly aligned,
- arranged in clusters,
- or completely irregular.

The drone groups nearby trees and plans an order that minimizes long unnecessary flights. This is similar to planning a short walking route inside a garden: visit nearby points first, then move on.

The global layer only decides:

- **which tree to visit next**,
- **at what altitude to travel**,
- **and when to return home**.

This layer also handles the geometrical Limits (borders) of the navigation area.

Layer 2: Smooth Local Navigation (Path Between Trees)

Between two trees, the drone generates a **smooth path**.

Not a straight line, and not a sharp turn—just a gentle movement that:

- keeps the drone clear of the tree crowns,
- avoids sudden attitude changes,
- and ensures good flight stability for the camera.

These “soft curves” make the flight:

- safer,
- more predictable,
- and easier for the low-level controller to track.

Altitude is usually kept slightly above the highest expected fronds to avoid collisions.

Layer 3: Flower-Level Positioning (Final Few Meters)

The most critical part of the mission happens when the drone reaches a specific tree. Here, navigation becomes reactive rather than planned.

The drone:

1. Slows down automatically.
2. Switches to camera-based alignment.
3. Adjusts sideways until the flower cluster is centered in view.
4. Descends slightly if needed.
5. Stabilizes and triggers the pollination mechanism.

Distance sensors around the drone add a **safety bubble**: if any frond is too close, the drone backs away or rises instantly.

This layer is inspired by how a human approaches a palm crown: slowly, carefully, using their eyes to judge distance.

IV.5 Mission Flow: How the Drone Executes Navigation

Here is a simplified view of the drone's mission behavior:

1. **Takeoff:** Rise to a safe altitude.
2. **Navigate Toward the First Tree:** Using global planning + smooth curves.
3. **Approach the Tree:** Slow down, position near the crown.
4. **Search and Align to Flower Clusters:** Camera and AI help the drone refine its position.

5. **Pollinate:** Stable hover, pollen release, climb back up.
6. **Move to the Next Tree:** Smooth transition to the next target.
7. **Repeat Until All Trees Are Completed**
8. **Return Home and Land**

IV.6 Summary

Chapter IV explains how the PolliDrone is able to move by itself inside a real date-palm oasis, without being “remote-controlled” tree by tree. It starts by clarifying what autonomous navigation means in this context: the drone must know which palm to visit next, how to move safely between tall crowns, and how to place itself in the right position and height to reach the flowers. The chapter also shows why classic techniques used in large, regular farms (straight grids, perfect rows, heavy 3D mapping) are not well adapted to small, irregular oases where trees can be in partial rows, arcs or clusters.

To solve this, the chapter proposes a simple three-layer navigation strategy. At the top level, a global tree-to-tree planner chooses an efficient order to visit all planned palms, grouping nearby trees but not requiring any strict row structure. At the intermediate level, the drone follows smooth, safe paths between trees, avoiding sharp turns and keeping a comfortable distance from crowns, which makes flight more stable and easier for the control loops. Finally, in the last few meters around each tree, a flower-level positioning layer takes over: the drone slows down, uses the camera and AI to locate flower clusters, makes small sideways and vertical corrections, and relies on distance sensors to maintain a safety bubble around itself during pollination.

In summary, Chapter IV shows that the navigation system is designed to be lightweight, robust, and realistic for oasis conditions. It does not try to build a perfect 3D map of the environment. Instead, it combines basic information about tree locations with real-time perception to ensure that every target tree is visited, obstacles are avoided, and the drone achieves high precision only where it truly matters: close to the date-palm flowers.

Chapter V

V. Visual Detection of Flowers and Palm Trees

V.1 Object detection concepts

In this work, object detection is based on a convolutional neural network (CNN) architecture. A CNN is a type of deep neural network specifically designed for image data, composed of successive convolution, non-linearity and pooling layers that automatically learn hierarchical visual features (edges, textures, shapes) directly from pixels. Thanks to weight sharing and local receptive fields, CNNs are well suited to large images and are now the standard backbone for most modern computer-vision tasks [23].

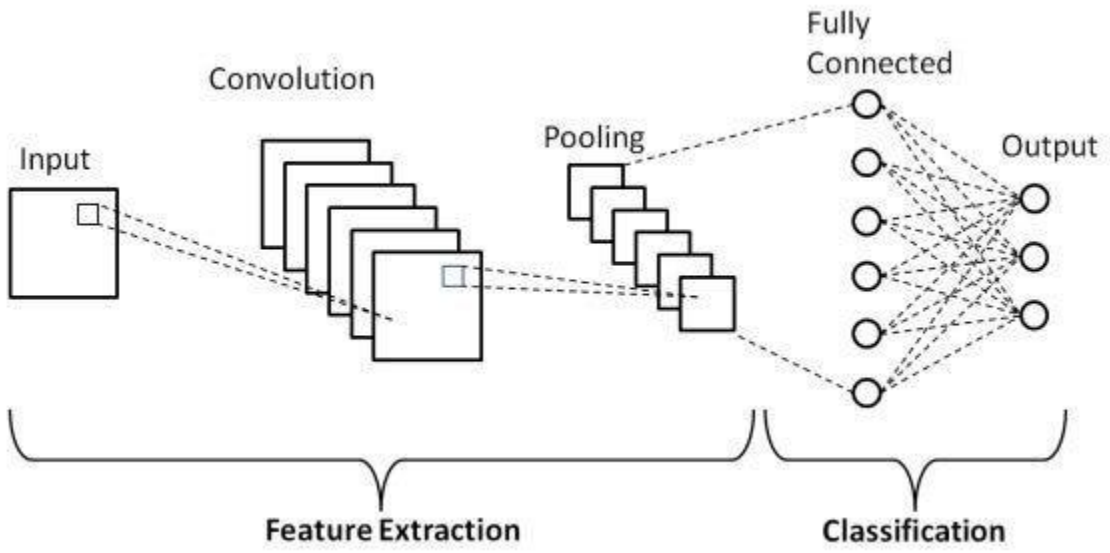

Figure V.1: A Comprehensive Guide to Convolutional Neural Networks [32]

The detector used in this thesis belongs to the YOLOv8 (“You Only Look Once”, version 8) family. YOLOv8 is a single-stage, real-time object detector that predicts bounding boxes and class probabilities in one forward pass of the network, without a separate region-proposal stage. It combines a CNN backbone for feature extraction with a lightweight detection head, and uses an anchor-free formulation where box centers, widths and heights are regressed directly from feature maps. This design provides a good compromise between accuracy and inference speed, which is essential for embedded deployment on the onboard computer of the pollination drone [23].

The vision system uses a YOLOv8 object detection model to find two types of objects in the camera image:

- date-palm flower clusters (class 0)
- palm trees / crowns (class 1)

The model is a convolutional neural network (CNN) with three main parts:

- a backbone that extracts features from the image at different scales
- a neck / feature pyramid that combines these features
- a detection head that outputs bounding boxes, class scores and an objectness score

The detector is anchor-free: instead of using predefined anchor boxes, it directly predicts the box center, width and height for each candidate location. This reduces complexity and is suitable for embedded hardware.

For onboard use we select a small YOLOv8 model (e.g. YOLOv8n / YOLOv8s). The choice is based on a trade-off: the model must be accurate enough to detect flowers, but also fast enough to run in real time on the Raspberry Pi Zero 2 W during flight.

V.2 Challenges: illumination, occlusion, variability [23]

Detecting flowers and palm crowns in a real orchard is more difficult than on standard datasets. The main problems are:

- **Illumination changes**
  - ✓ strong sunlight and hard shadows from the fronds
  - ✓ over-exposed and under-exposed regions in the same image
  - ✓ appearance of the same flower changing with time of day
- **Occlusion**
  - ✓ flower clusters partially hidden by leaves, branches or other bunches
  - ✓ only small visible parts of the cluster, with irregular shapes

- **Variability of objects**
    - ✓ flowers: different cultivars and different opening stages (closed, half open, fully open)
    - ✓ crowns: different pruning, shapes and densities
- **Background clutter**
    - ✓ trunks, nets, other vegetation and ground textures that may look similar to flowers at some scales
- **Scale changes**
    - ✓ when the UAV moves in altitude or distance, a cluster can be very small (few pixels) or very large in the image

In this project, the available dataset is **limited**: it is composed of **789 images** downloaded from online sources. The images are mostly captured under similar conditions (similar lighting and viewpoints) and do not fully cover all the variations listed above. This lack of diversity is a clear limitation and can reduce the generalization capability of the model to real orchard conditions.

To partially compensate for this, the training pipeline uses **strong data augmentation**, including random cropping, horizontal flipping, brightness and contrast changes, light blur and partial masking. The objective is to artificially create variations in illumination, scale and occlusion so that the YOLOv8 model becomes more robust, even though the original dataset is relatively small and not very varied.

V.3 Model Training and Dataset Preparation

Dataset collection

In this project, no new images were captured with the prototype drone.

Instead, the training data were taken from an existing public dataset available on Roboflow Universe, in the project “Pollination Drone”, version 5:

https://universe.roboflow.com/pollination-palm/pollination-drone/dataset/5 [24]

This dataset contains images of:

- date-palm flower clusters
- palm trees / crowns and trunks.

After download, the dataset used in this work includes 789 images (Figure V.2). Most of these images were captured under similar lighting and viewing conditions, so the variability in time of day, season and orchard location is limited. This small size and limited diversity is a clear limitation and reduces the expected generalization of the model to real field conditions.

Figure V.2: Sample Images from the Pollination Drone Dataset (Roboflow, v5)

V.4 Training pipeline: preprocessing, augmentation, splitting

Training is performed in Python using **Ultralytics YOLOv8**

Main settings:

- model: **YOLOv8n**, pre-trained on COCO
- input size: **640 × 640** pixels
- batch size: 16, up to **80 epochs**, with early stopping

Data augmentation is enabled in the model.train() call and includes HSV color shifts, rotation (±10°), translation, scaling, horizontal flips, **mosaic** and **mix-up**. These operations are applied on-the-fly to compensate for the small and not very diverse dataset.

The original Roboflow splits (train, valid, test) are used for baseline training and evaluation. In addition, a **5-fold cross-validation** is created by merging train and valid, then re-training YOLOv8n on different fold splits to obtain more reliable metrics on this small dataset.

Model selection criteria (size, inference speed, accuracy)

The detector is intended to run on a **Raspberry Pi Zero 2 W**, so the model must be compact and reasonably fast.

A lightweight **YOLOv8n** configuration is chosen to balance:

- **size** (number of parameters),
- **inference speed** on embedded hardware,
- **detection accuracy** (precision, recall, mAP) on flower clusters.

A slightly smaller but faster model is preferred to a larger model that would be too slow for real-time pollination.

Evaluation metrics (precision, recall, mAP)

Performance is evaluated using standard object detection metrics:

- **Precision**: fraction of predicted boxes that are correct (When the model says *flower/palm*, how often is it correct?).

- **Recall**: fraction of real objects that are detected (Of all real flowers/palms, how many did the model find?).

For this application, **high recall on flowers** is especially important (avoid missing receptive clusters), while good precision reduces unnecessary pollination attempts on non-flower regions.

V.5 Overfitting handling and generalization

Because the dataset is small (789 images) and comes from a single online source, there is a high risk of overfitting.

To limit this, the training uses:

- Strong data augmentation,
- Early stopping based on validation metrics,
- A small model (yolov8n) rather than a larger architecture,

V.6 The Trained Model Evaluation

Precision

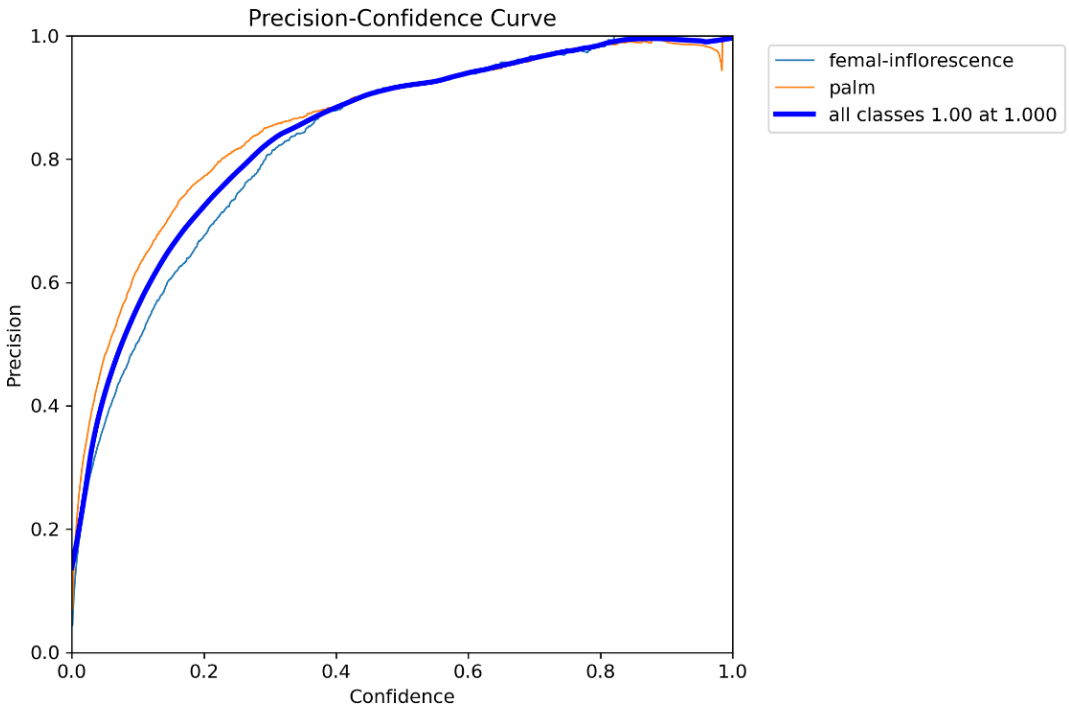

Figure V.3: Model Validation Precision Curve

Discussion:

- Precision increases as confidence increases because you're keeping only the predictions the model is most sure about.
- Palm has slightly higher precision at low–mid confidence ⇒ palm detections are cleaner even when the model isn't very confident.
- After about 0.4 confidence, both classes become very accurate, and near 0.8–1.0, precision is almost perfect.

✅ Meaning: If you want fewer false alarms, raising the confidence threshold works well.

Recall

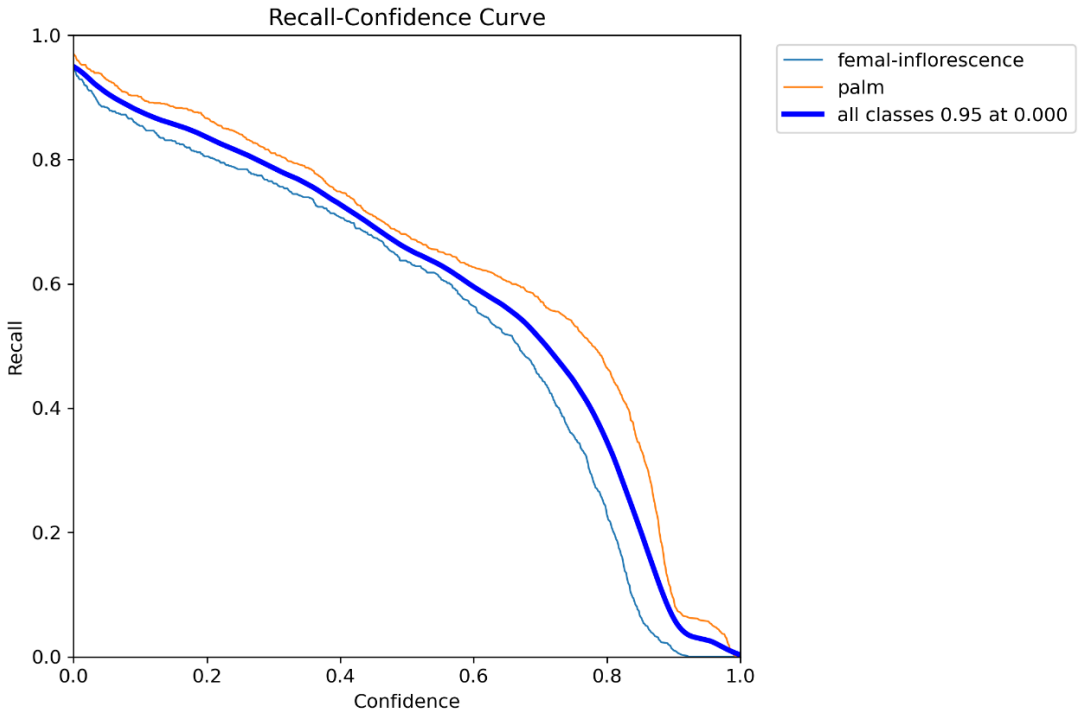

Figure V.4: Model Validation Recall

Discussion:

- Recall decreases as confidence increases because stricter thresholds throw away many detections, including correct ones.
- Palm keeps higher recall across thresholds ⇒ easier class for the model to detect consistently.
- Female-inflorescence drops faster, especially after ~0.6–0.7 ⇒ the model struggles to assign high confidence to this class, so it gets filtered out more.

✅ Meaning: High thresholds will make you miss many objects, especially female-inflorescence.

V.7 Summary

Chapter V explains how the drone “sees” palm trees and flowers and turns camera images into usable targets for pollination. It adopts an embedded YOLOv8 CNN detector that runs in real time on the Raspberry Pi Zero 2 W and detects two classes: date-palm flower clusters and palm crowns/trees.

The chapter first recalls core object-detection concepts (CNN backbone, feature pyramid, detection head) and then details the specific challenges of real orchards: strong illumination changes, hard shadows, partial occlusions by fronds, large intra-class variability (different cultivars and flower stages), background clutter and scale changes as the UAV altitude varies. These factors make robust detection significantly harder than on “clean” benchmark datasets and increase the risk of missed flowers or false positives in the field.

Since no in-house images were collected yet, the model is trained on a small Roboflow dataset (Figure V.2) (789 images) of palms and inflorescences (“Pollination Drone”, v5). To compensate for limited diversity, the training pipeline uses a YOLOv8n model pre-trained on COCO, resized inputs (640×640), and heavy data augmentation The dataset is split into train/Val/test plus 5-fold cross-validation, and the model is evaluated with standard metrics (precision, recall, mAP@0.5 and mAP@0.5:0.95).

Finally, the chapter stresses the risk of overfitting due to the small, biased dataset and the fact that images come from similar conditions. It mitigates this with augmentation, early stopping, regularization, and by deliberately choosing a small architecture (YOLOv8n instead of larger variants) to avoid memorization. Still, the detector is considered prototype-level: real orchard data collection and retraining are identified as essential next steps to reach production-grade robustness in varying light, occlusion, and canopy geometries.

Chapter VI

VI. Hardware Development and System Integration

VI.1 Introduction

This chapter presents the complete development of the prototype pollination drone, focusing on the design, assembly, and integration of all hardware subsystems required for autonomous field testing. The prototype serves as the first fully functional platform used to validate the guidance, perception, and pollination concepts introduced in earlier chapters.

The chapter covers the mechanical layout of the drone, including propulsion selection, structural mounting of electronics, and temporary fixtures used during early experimentation. It then details the power distribution prototype, which provides regulated 5 V and 3.3 V rails, fused power lines, and battery protection necessary for reliable operation. The integration of the sensor suite described with emphasis on orientation, field-of-view alignment, and electrical interfacing.

The development of the control electronics is summarized along with grounding and EMI-mitigation considerations. The prototype pollen-dispensing mechanism is also presented, covering the pump driver, tubing, and early flow-control trials. The chapter concludes with system-level wiring, calibration procedures, and initial integrated tests, leading to the first low-altitude flight experiments.

VI.2 Hardware

Prototype Platform

The prototype is built on a DJI F450 quadrotor frame used as a development kit. It is equipped with four A2212/13T 1000 kV BLDC motors paired with 10×4.5" propellers and driven by 30 A ESCs, powered from a 3S (11.1 V) LiPo battery.

This configuration provides a comfortable thrust margin for payload and sensor integration during development.

Figure VI.1: DJI F450 Quadcopter Frame

Figure VI.2: 10x4.5" Propellers, A2212/13T 1000KV Motors and 30A ESC

Figure VI.3: 3S li-Po 11.1V Battery

Power Distribution Board

The PDB distributes the 3S 11.1 V battery to the four 30 A ESCs using wide copper pours, with fuse and a current-sense shunt on the main input. Two buck converters generate regulated 5 V (Raspberry Pi Zero 2 W, camera, auxiliaries) and 3.3 V (ESP32, IMU, barometer, ToF), with simple LC filtering. High-current and signal grounds are tied at a single star point to reduce noise injection into the control electronics.

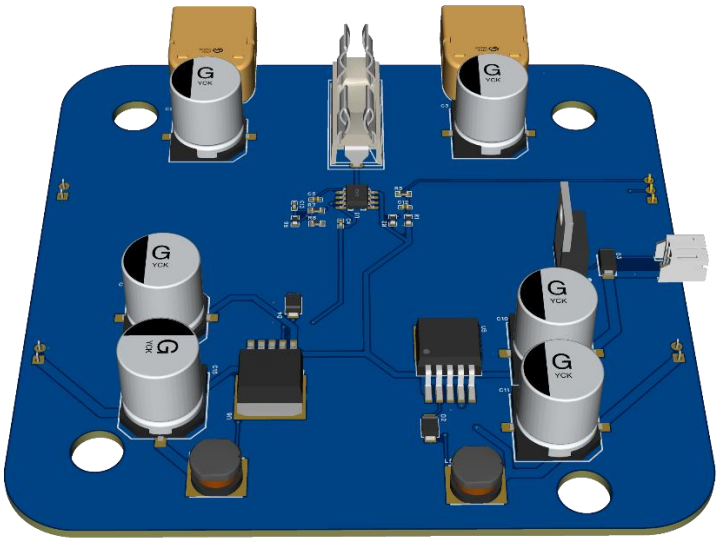

Figure VI.4: Power Distribution Board

Control Electronics

The control stack is based on:

- An **ESP32** microcontroller on a custom **carrier board**, handling state estimation, inner control loops and direct sensor interfacing

Figure VI.5: ESP-WROOM-32 38 Pins

- A **Raspberry Pi Zero 2 W** single-board computer as the high-level companion for AI vision and mission logic

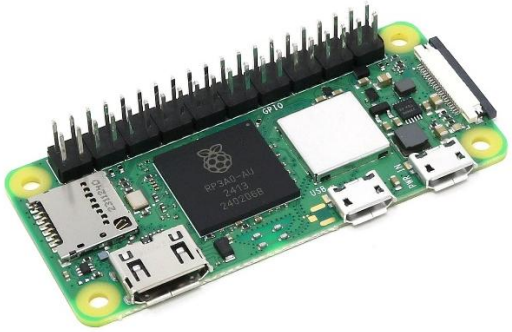

Figure VI.6: Raspberry Pi Zero 2 W

Sensor Integration

The navigation and environment sensing stack includes:

- An **ICM-20948** 9-axis IMU (3-axis accelerometer, gyroscope and magnetometer)

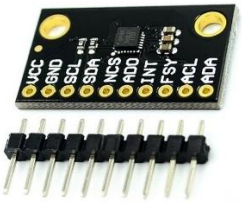

Figure VI.7: ICM-20948

- A **BMP388** barometric pressure sensor for altitude estimation

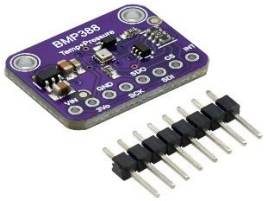

Figure VI.8: BMP388

- A **Radiolink TS100 GPS/compass** module for GNSS position and heading

Figure VI.9: Radiolink TS100 GPS/compass

- A **CSI (or ribbon) AI camera** connected to the **Raspberry Pi Zero 2 W** for vision-based perception and flower detection.

Figure VI.10: Raspberry Pi AI Camera

Remote Control and Wireless Interface

- **FlySky Remote Controller:** a **FlySky FS-i6** transmitter with an **FS-iA6B** receiver is included for manual flight, safety override, and early tuning. It provides a reliable 2.4 GHz link, with basic channel mapping for throttle, attitude, and mode switching.

Figure VI.11: FS-iA6B Receiver

Figure VI.12: FlySky FS-i6 Transmitter

- **Wi-Fi Control via RPi Hotspot:** the **Raspberry Pi Zero 2 W** provides a built-in Wi-Fi hotspot used by a **custom control app** for configuration and semi-autonomous testing. The app displays live telemetry (IMU, GPS, battery, ToF sensors) and allows sending simple high-level commands such as arming, mode selection, and pollination triggers.

Pollen Dispensing Prototype

The pollen-delivery system uses a 1L 3D printer tank and 12 V DC pump oriented towards the target inflorescences. The pump is driven by an ESP32-controlled MOSFET stage with PWM control, enabling adjustment of the effective flow rate.

Figure VI.13: 12V DC Pump

VI.3 Software Architecture

The software stack is divided between the **ESP32**, responsible for all real-time flight-critical functions, and the **Raspberry Pi Zero 2 W**, running the high-level perception and mission logic. Both units operate independently but exchange compact telemetry and command messages over a UART link.

ESP32 Flight Control Firmware

The ESP32 executes the time-critical control pipeline:

- IMU (ICM-20948) and barometer (BMP388) sampling
- Sensor fusion and attitude estimation
- Rate, attitude, and velocity control loops
- PWM generation for the four ESCs

- GPS data parsing (TS100)
- Pump actuation via PWM
- Failsafe routines and FlySky RC override

The ESP32 maintains stable flight even if the Raspberry Pi or Wi-Fi link fails.

Raspberry Pi High-Level Processing

The Raspberry Pi Zero 2 W acts as the autonomy and perception computer:

- AI camera inference for flower detection and classification
- Row-following and local pollination decision logic
- Obstacle awareness from ToF (VL53L1X) sampling
- Telemetry display and configuration through the Wi-Fi hotspot
- Data logging for flight and sensor analysis

It produces high-level navigation or pollination triggers but does not control the motors directly.

Inter-Processor Communication

A lightweight UART protocol synchronizes both processors:

- **ESP32 → RPi:** attitude, altitude, GPS state, battery voltage, system flags
- **RPi → ESP32:** pollination commands, mission events, row-tracking adjustments, app-based inputs.

Message rates and payloads are kept minimal to ensure deterministic behavior and low overhead.

Safety and Override Logic

Safety is prioritized by hardware and firmware structure:

- The **FlySky RC receiver** overrides all autonomous commands
- Loss of RPi or Wi-Fi control leaves the ESP32 running full stabilization

VI.4 Prototype

The final prototype integrates all mechanical, electronic, and sensing subsystems and includes several custom 3D-printed accessories designed specifically for agricultural operations and the constraints of palm-tree pollination. The prototype is equipped with:

- **3D-Printed Protective Cover:** A lightweight top cover to shield the ESP32 carrier board, wiring harness, barometer, and power electronics from dust, pollen, and direct sunlight.
- **3D-Printed Pollen Tank and Mount:** A 3D-printed pollen reservoir interfaces directly with the 12 V micro-pump.
- **3D-Printed Blade Guards (Propeller Protectors):** for safety during close-proximity around palms, the drone features 3D-printed blade guards fitted around each propeller.
- **3D-Printed Landing Gear:** Enhanced landing legs were printed to increase ground clearance.
- **3D-Printed Camera Mount:** The Raspberry Pi camera is mounted on a vibration-isolated 3D-printed Servo Motor Adjustable bracket.

Figure VI.14: Prototype Front View

Figure VI.15: Prototype Top View

Figure VI.16: Prototype Back View

VI.5 Summary

This chapter presented the complete prototype implementation of the pollination drone, covering both the hardware platform and the onboard software architecture. The DJI F450 frame, A2212 motors, 10×4.5" propellers, 30 A ESCs and 3S LiPo battery provide a robust development baseline, while the custom PDB, ESP32 carrier board, and integrated sensor suite form a coherent avionics stack tailored for autonomous navigation and pollination tasks. The addition of the pollen-dispensing module, FlySky manual controller, and Wi-Fi interface further supports flexible testing conditions and safe operation.

On the software side, responsibilities are clearly split: the ESP32 handles real-time sensing and control, ensuring stable flight under all conditions, while the Raspberry Pi Zero 2 W executes high-level perception, mission logic, and user interaction. This distributed architecture enables reliable autonomy without compromising safety or responsiveness.

Together, the hardware and software subsystems form a fully functional development prototype, establishing the foundation for the autonomous behaviors, mission strategies, and pollination performance evaluated in the following chapter.

Chapter VII

VII. General Summary

This work presented the complete development cycle of an autonomous pollination drone—from dynamic modeling and control design to perception, mission planning, and hardware prototyping. The objective was to investigate whether a compact quadcopter, equipped with AI-based flower detection and a lightweight dispensing mechanism, could operate as a reliable alternative to manual date-palm pollination.

The thesis first established a rigorous mathematical framework describing the quadcopter's nonlinear dynamics, rotor aerodynamics, and BLDC/ESC behavior. This model provided the basis for designing a cascaded PID control structure capable of maintaining stable attitude, altitude, and position regulation during low-speed, close-proximity maneuvers typical of pollination tasks.

Simulation results demonstrated that the proposed control architecture achieves robust tracking performance under wind, modeling uncertainties, and sensor noise—validating its suitability for precision operations around dense palm canopies. The integration of AI perception further enabled real-time detection and localization of palm inflorescences, allowing the drone to navigate autonomously and trigger pollen release with accuracy.

A hardware prototype was built using a DJI F450 frame, A2212/13T 1000 KV BLDCs, 30 A ESCs, 10×4.5″ propellers, and a 3S 11.1 V Li-ion battery, coupled with a Raspberry Pi Zero 2 W, ESP32 control carrier board, environmental and ranging sensors, and a custom pollen-dispensing module. The system validation confirmed that the overall architecture—power distribution, sensing pipeline, control loops, and high-level autonomy—can be effectively integrated on a light UAV platform.

Although full field-scale deployment was beyond the scope of this work, the results collectively show that autonomous pollination using UAVs is technically feasible and has the potential to reduce labor demand, standardize pollination quality, and improve worker safety in date-palm agriculture.

Future improvements should focus on:

- more advanced perception models robust to occlusion and harsh illumination;
- adaptive and AI-assisted control strategies for complex canopy navigation;

- more precise, low-loss pollen delivery mechanisms;
- long-duration endurance platforms and battery optimization;
- extensive real-world field trials to evaluate agronomic performance.

Overall, this thesis contributes a complete framework—modeling, control, perception, autonomy, and hardware integration—demonstrating that AI-driven UAV pollination is a promising and scalable approach for modern precision agriculture.

References

References

[1] Global Pollinator Report. (2021). *Decline in natural pollinator populations: Implications for agriculture.*

[2] FAO Statistical Yearbook. (2022). *Global date palm production statistics and innovations in cultivation.*

[3] Mohamed, R. E., et al. (2018). *Pollen distribution effect on date palm productivity.* Agricultural Sciences.

[4] Nixon, K., et al. (2016). *Date palm fruit set and the timing of pollination.* Horticulture Journal.

[5] Saudi Ministry of Agriculture. (2022). *Comparative cost analysis of pollination methods.*

[6] Market Research Future. (2023). *Agricultural drone market research report – Global forecast to 2028.*

[7] Dorling, K., Heinrichs, J., & Messier, G. G. (2017). *UAV systems for precision agriculture.* Computers and Electronics in Agriculture.

[8] Zhang, C., & Kovacs, J. M. (2012). *The application of UAVs in agriculture: A review.* Precision Agriculture.

[9] Journal of Pollination Ecology. (2021). *Emerging technologies in agricultural pollination: The case of UAV systems.*

[10] Al-Khalifa, H., Abdulhadi, N., & Al-Harbi, R. (2020). *Artificial pollination in date palms using drones.* Journal of Agricultural Science and Technology.

[11] King Faisal University. (2023). *UAV pollination field trials: Saudi Arabia 2020–2023.*

[12] Beard, R. W., & McLain, T. W. (2012). *Small unmanned aircraft: Theory and practice.* Princeton University Press.

[13] Bangura, M., & Mahony, R. (2012). *Nonlinear dynamic modeling for quadcopter UAVs.* IFAC Proceedings.

[14] Bouabdallah, S., Murrieri, P., & Siegwart, R. (2004). *Design and control of an indoor micro quadrotor.* IEEE International Conference on Robotics and Automation (ICRA).

[15] Castillo, P., Lozano, R., & Dzul, A. E. (2004). *Modelling and control of mini-flying machines.* International Journal of Robust and Nonlinear Control.

[16] Hoffmann, G. M., Huang, H., Waslander, S. L., & Tomlin, C. J. (2007). *Quadrotor helicopter flight dynamics and control experiment.* AIAA Guidance, Navigation, and Control Conference.

[17] Pounds, P., Mahony, R., & Corke, P. (2006). *Modelling and control of a quadrotor robot.* Australasian Conference on Robotics and Automation (ACRA).

[18] Pillay, P., & Krishnan, R. (1989). *Modeling, simulation, and analysis of permanent-magnet motor drives.* IEEE Transactions on Industry Applications.

[19] Hanselman, D. C. (2006). *Brushless permanent magnet motor design* (2nd ed.).

[20] Krishnan, R. (2010). *Permanent magnet synchronous and brushless DC motor drives.*

[21] Chen, J., & Pan, W. (2014). *Sensorless control of BLDC motor based on back-EMF detection.* IEEE Transactions on Power Electronics.

[22] Texas Instruments. (n.d.). *Sensorless BLDC motor control using back-EMF.*

[23] Santos, M., et al. (2021). *AI-powered drones for agricultural applications.* Precision Agriculture.

[24] Roboflow. (2023). *Date palm flower detection dataset.*

[25] *https://www.croptracker.com/blog/drone-technology-in-agriculture.html*

[26] *https://www.gim-international.com/content/article/soil-mapping-with-drones*

[27] *https://pngtree.com/free-png-vectors/professional-drone-vector*

[28] *https://www.vexels.com/png-svg/preview/211836/drone-quad-thin*

[29]*https://depositphotos.com/vector/drone-icon-eps-copter-symbol-front-and-side-view-114524458.html*

[30] https://pngimg.com/image/70842

[31] https://www.linkedin.com/posts/aashish-dilawari-54b66142_basics-of-a-pid-controller-as-the-name-suggests-activity-6483971550809780224-uCtA

[32] https://www.researchgate.net/figure/Basic-CNN-Architecture-CNN-based-automated-deep-learning-algorithms-have-excelled-at-the_fig4_374391696

Annex

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة غرداية
كلية العلوم والتكنولوجيا
قسم :الآلية والكهروميكانيك

Université de Ghardaïa
Faculté des Sciences
et de la technologie

شعبة :الآلية
تخصص: آلية و أنظمة

غرداية في : 2025/12/23

شهادة ترخيص بالتصحيح والايداع:

انا الاستاذ(ة) : بكار بلقاسم
بصفتي المشرف المسؤول عن تصحيح مذكرة تخرج (ليسانس/ماستر/دكتورا) المعنونة بـ:

PolliDrone: Smart Drone for Precision Pollination of Date Palms

من انجاز الطالب (الطلبة):

بلعيد يونس

موسى المال يوسف

التي نوقشت بتاريخ : 2025/12/15

اشهد ان الطالب/الطلبة قد قام /قاموا بالتعديلات والتصحيحات المطلوبة من طرف لجنة المناقشة وقد تم التحقق من ذلك من طرفنا
وقد استوفت جميع الشروط المطلوبة.

امضاء المسؤول عن التصحيح

بكار بلقاسم

مصادقة رئيس القسم

جامعة غرداية
قسم الآلية والكهروميكانيك-ا
كلية العلوم و التكنولوجيا

رئيس قسم الآلية والكهروميكانيك
عزاوي محمد