

Design and Development of an Autonomous Aerial Manipulator

A report for the final phase submitted in fulfillment of the requirements for completion
of the Dual Degree Project

submitted by

Rudraksh Kuchiya
(20d110021)

Under the guidance of
Prof. Sukumar Srikant
Prof. Vivek Sangwan



Department of Mechanical Engineering
Indian Institute of Technology Bombay
Mumbai - 400076, India

June, 2025

Dissertation Approval

This dissertation entitled **Design and Development of an Autonomous Aerial Manipulator** by **Rudraksh Kuchiya**, Roll No. **20d110021**, is approved for the Dual Degree (BTech + MTech) from the Indian Institute of Technology Bombay.

.....
Prof. Sukumar Srikant Systems and Control Engineering	Prof. Vivek Sangwan Mechanical Engineering
.....
Prof. Vivek Natarajan Systems and Control Engineering	Dr. Arya Hemendra Aerospace Engineering

June, 2025

IIT Bombay

Declaration

I declare that this written submission represents my ideas in my own words. Where others' ideas and words have been included, I have adequately cited and referenced the original source. I declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated, or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause disciplinary action by the Institute and can also evoke penal action from the source which has thus not been properly cited or from whom proper permission has not been taken when needed.

Rudraksh Kuchiya

Roll No.: 20d110021

Date: 23 June 2025

Place: IIT Bombay

Abstract

This dissertation presents the comprehensive development of an autonomous aerial manipulation system integrating a quadcopter platform with a 2-DOF robotic manipulator for precision manipulation tasks. The research addresses critical challenges in aerial robotics by combining hardware design optimization, advanced control systems, computer vision algorithms, and mechanical stabilization techniques to create a functional aerial manipulation platform capable of autonomous object detection, approach, and manipulation.

The system architecture comprises three primary subsystems: a custom 2-DOF manipulator arm with high-torque end-effector, a Pixhawk Cube Orange-controlled quadcopter platform with ArduPilot firmware, and an integrated vision system utilizing Intel RealSense depth cameras for ArUco marker detection and pose estimation. Key innovations include a 6-DOF vibration isolation system for camera stabilization, dual-link torque distribution mechanisms for improved manipulator stability, and advanced image deblurring algorithms specifically designed for drone-induced motion blur.

Experimental validation demonstrated successful autonomous target approach and hovering capabilities, with the vision system achieving 11.2 fps processing rates with deblurring enabled and centimeter-level positioning accuracy through RTK-GPS integration. However, field testing revealed critical limitations when operating at 90% thrust capacity with the manipulator attached, leading to system instability and highlighting the need for enhanced platform stability and load balancing mechanisms.

The research contributes novel approaches to aerial manipulation through integrated hardware-software solutions, providing a foundation for future developments in autonomous aerial robotics applications including satellite servicing testbeds, aerial construction, and remote manipulation tasks in hazardous environments.

Contents

Dissertation Approval	iii
Declaration	iv
Abstract	v
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Dissertation Organization	3
2 Literature Review	4
2.1 Historical Development	4
2.2 Aerial Platform Configurations	5
2.3 Manipulator Design and Integration	5
2.4 Control Systems and Methodologies	5
2.5 Sensor Integration and Perception	6
2.6 Current Challenges and Future Directions	6
I HARDWARE DEVELOPMENT	8
3 Geometric and Dynamic Constraints for 2-DOF Manipulator Integration	9
3.1 System Assumptions	9
3.2 Derivation of Dimensional Constraints	10
3.2.1 Ground Clearance for Takeoff and Landing	10
3.2.2 Propeller and Frame Avoidance	11
3.2.3 Aerodynamic Interference	11
3.2.4 Center of Mass and Stability	11
3.2.5 Torque and Control Considerations	12

3.3	Combined Dimensional Constraints	12
4	Two-DOF Manipulator Design	13
4.1	Version1 : Separate Torque and Balance Links	13
4.2	Version2 : Dual Link Torque Distribution	15
4.3	Version3: Manipulator Arm Below the Quadcopter Base	17
5	End-Effector Development	19
5.1	Version1 : Gear Driven Parallel Gripper	19
5.2	Version2 : High Torque Gripper	20
6	Actuator Integration and Structural Optimization	23
6.1	Servo Actuators for Torque	23
6.2	Weight Balancing & Structural Weight Optimization	24
7	Simulation and Testing	26
7.1	Gazebo Simulation	26
7.2	Vertical Arm Stabilization	28
7.3	Manual Flight Testing	29
II	QUADCOPTER AND FIRMWARE DEVELOPMENT	31
8	Kinematic and Dynamic Modeling of the System	32
8.1	Kinematic Model	32
8.1.1	Coordinate Frames and Generalized Coordinates	32
8.1.2	Velocity Relationships	33
8.2	Dynamic Model	34
8.2.1	Lagrange-D'Alembert Formulation	34
8.2.2	Kinetic and Potential Energy	34
8.2.3	Dynamics Equation	35
8.2.4	Control Inputs	36
9	Aerial Platform Design Specifications	37
9.1	Flight Controller Integration	37
9.2	Thruster System: Motor, ESC, and Propeller	38
9.3	GPS and Auxiliary Sensors Integration	38
9.4	Power Management	39

9.5 Hover Thrust learning	40
10 Firmware Development	42
10.1 Flight Controller and On Board Processor Communication	42
10.2 Data Flow Architecture	43
10.3 MAVROS and Node Architecture	44
10.4 Servo Actuation Control	44
10.5 On-ground Communication Framework	46
10.6 Parameters for Ardupilot Configuration	47
III VISION SYSTEM AND MANIPULATION	49
11 Camera and Processing	50
11.1 On-board Computation Resources	50
11.2 Power Management	51
11.3 Camera and Image Processing	52
11.3.1 Raspberry Pi Camera	52
11.3.2 Intel Realsense Depth Camera	54
11.4 Camera Shake De-blurring	54
11.4.1 Multi-Scale Blind Deconvolution with Shock Filtering	55
11.4.2 Drone-Specific Piecewise Deblurring with Adaptive Regularization .	58
11.5 Magnetic Interference (HERE3 GPS and RPI)	61
12 Vision-based Approach Algorithms	63
12.1 Local Target Approach	63
12.2 Global Target Approach	64
12.3 Contingencies	66
13 Camera Mounting & Vibration Isolation	68
13.1 Version1 : High frequency dampers	68
13.2 Version2 : Heavy Industrial dampers	69
13.3 Version3 : DOF Vibration Isolation	70
IV CONCLUSIONS AND FUTURE WORK	74
14 Autonomous Approach : Field Test	75

14.1 Arena Setup	75
14.2 Autonomous approach Objectives	76
14.3 Results	77
15 Conclusion	79
16 Future Work	81
Bibliography	85
Acknowledgements	87

List of Figures

4.1	Separate torque and balance link	14
4.2	Separate torque and balance link : Base and End mounts	14
4.3	Dual Link Torque Distribution Design	16
4.4	Arm Mounted Below	18
5.1	Gear Driven Parallel Gripper	20
5.2	High Torque Gripper	21
5.3	Comparison of End effectors: Gear driven and High Torque	22
6.1	Weight Balancing: Placing battery behind the gripper	24
7.1	Gazebo Simulation Environment	27
7.2	Vertical Stabilization of arm	28
7.3	Drone assembled with manipulator attached below and camera on the top	29
8.1	Configuration of the coordinates for the combined system	33
9.1	Flight Controller: Pixhawk Cube orange with Ardupilot 4.4.1	37
9.2	General power management on the quadcopter system	40
10.1	Data flow architecture between RPI and Pixhawk	44
10.2	MAVROS communication architecture	45
10.3	Ground Communication Framework	47
11.1	Initial power management architectures: Could not meet the requirements	51
11.2	Modified Power Management with on-board processor	51
11.3	Marker Detection from Realsense Camera feed	53
11.4	No detection with basic de-blurring tools: Test 1	57
11.5	No detection with basic de-blurring tools: Test 2	57
11.6	Marker detected with Advanced de-blurring tools: Test 1	60

11.7 Marker detected with Advanced de-blurring tools: Test 2	60
12.1 Local Approach, Marker in center, Approaching towards target	65
12.2 Local Approach, marker not centered	65
12.3 Global Approach, Going towards global position	67
13.1 High frequency dampers attached on the drone	69
13.2 (version1) High Frequency Dampers : Frequency is reduced but amplitude is still high in X and Y axes	70
13.3 (version2) Industrial Dampers : Amplitude gets reduced but vibrations are high across all three axes	71
13.4 6DOF Dampers Assembly on the quadcopter	72
13.5 (version3) 6-DOF Vibration Reduction : Low amplitude and low frequency across all axes	73
14.1 Arena Setup	76

List of Tables

9.1	Technical specifications of the quadcopter with manipulator	38
10.1	Comparison between MAV_CMD_DO_SET_SERVO and RC_CHANNELS_OVERRIDE	46
10.2	Important parameter values for DRONECAN and hover thrust configuration	48
11.1	Comparison of Frame Rates on RPI 3B and RPI 4B With and Without Deblurring	50
12.1	Comparison between Local and Global Target Approaches	66

Chapter 1

Introduction

1.1 Background

Over the past decade, unmanned aerial vehicles have evolved from experimental research platforms to sophisticated systems capable of complex autonomous operations across diverse industrial applications. The integration of robotic manipulation capabilities with aerial platforms represents a significant advancement in robotics, enabling drones to transition from passive observation tools to active manipulation systems capable of physical interaction with their environment.

Traditional aerial platforms are limited to sensing and navigation tasks, constraining their utility in applications requiring physical manipulation such as valve operation, object retrieval, or precision assembly operations. The development of aerial manipulation systems addresses this limitation by combining the mobility advantages of multirotor aircraft with the dexterous capabilities of robotic manipulators, creating mobile manipulation platforms with unprecedented operational flexibility.

In the context of space robotics, aerial manipulation platforms offer unique advantages as cost-effective testbeds for satellite servicing algorithms. The 6-DOF flight capabilities of multirotor systems closely simulate zero-gravity conditions, providing realistic testing environments for docking procedures, orbital manipulation tasks, and autonomous servicing operations without the prohibitive costs associated with space-based testing facilities.

Current aerial manipulation research faces several critical challenges including payload-induced instability, limited thrust margins, vibration-induced sensor

degradation, and the complexity of integrating multiple subsystems while maintaining flight safety. These challenges necessitate comprehensive approaches combining mechanical design optimization, advanced control algorithms, and robust sensor integration to achieve reliable autonomous operation.

1.2 Motivation

The motivation for this research stems from the growing demand for autonomous systems capable of performing complex manipulation tasks in environments where human access is limited, dangerous, or impractical. Current drone platforms predominantly serve data collection roles, but real-world applications increasingly require physical interaction capabilities for tasks such as infrastructure inspection, maintenance operations, and emergency response activities.

In space robotics applications, the development of reliable manipulation algorithms requires extensive testing on physical systems that can accurately simulate orbital conditions. Traditional ground-based robotic systems cannot replicate the dynamic, 6-DOF environment characteristic of space operations, making aerial manipulation platforms ideal testbeds for satellite servicing technologies, orbital debris removal systems, and autonomous assembly operations.

The potential for aerial manipulation extends beyond space applications to include cooperative drone operations where master platforms manage and recharge smaller manipulator units, enabling sustained autonomous operations in complex environments. This capability opens possibilities for long-term autonomy in applications such as disaster response, environmental monitoring, and industrial automation where continuous operation is critical.

The economic advantages of aerial manipulation systems include reduced operational costs compared to traditional ground-based or human-operated alternatives, increased operational safety through remote operation capabilities, and enhanced mission flexibility through rapid deployment and reconfiguration. These factors drive the need for robust, reliable aerial manipulation platforms capable of autonomous operation in dynamic environments.

1.3 Dissertation Organization

This dissertation follows a systematic development approach, progressing from fundamental hardware design through system integration to experimental validation and future development recommendations. The document is structured to provide comprehensive coverage of all major system components and their integration challenges.

Part I: Hardware Development begins with detailed analysis of the 2-DOF manipulator design evolution, covering three major design iterations from separate torque and balance links through dual-link torque distribution to the final below-base configuration. This section includes comprehensive end-effector development, actuator integration strategies, and structural optimization approaches for weight balancing and load distribution.

Part II: Quadcopter and Firmware Development covers the aerial platform specifications, including flight controller integration, propulsion system design, sensor integration strategies, and power management architectures. This section provides detailed coverage of firmware development including MAVROS integration, servo actuation control, and ground communication frameworks essential for autonomous operation.

Part III: Vision System and Manipulation addresses the computational requirements, camera selection and integration, advanced image processing algorithms including specialized deblurring techniques, and vision-based approach algorithms. This section includes detailed analysis of vibration isolation systems and autonomous field testing results demonstrating system capabilities and limitations.

Part IV: Conclusions and Future Work synthesizes the research findings, identifies key limitations discovered during experimental validation, and provides comprehensive recommendations for future development including hexcopter platform migration and advanced bearing integration for improved stability and load balancing capabilities.

Chapter 2

Literature Review

The field of aerial manipulation combines the mobility advantages of unmanned aerial vehicles (UAVs) with robotic manipulators, creating transformative systems for autonomous operations (1; 2). This literature review examines current research in autonomous aerial manipulation systems, focusing on technological developments, control methodologies, and system integration challenges (3).

2.1 Historical Development

Foundational Survey Literature

Bonyan Khamseh *et al.* provided one of the most influential surveys on aerial manipulation, establishing classification frameworks and key research directions (2). Mohiuddin *et al.* expanded this foundation with comprehensive coverage of single and multi-UAV systems, highlighting challenges in outdoor operations and energy management (1).

Emerging Research Directions

Recent survey work has identified cooperative aerial manipulation systems and continuum manipulators as promising approaches for addressing rigidity-related challenges in traditional systems (3; 10).

2.2 Aerial Platform Configurations

Quadrotor-Based Systems

Kim *et al.* demonstrated early aerial manipulation using a quadrotor with a two-DOF robotic arm, establishing key kinematic and dynamic modeling approaches (4). However, field testing revealed that quadrotor platforms often operate at thrust levels exceeding 90% of capacity when equipped with manipulation hardware, limiting control authority (4).

Multi-Rotor Configurations

Hexacopter platforms offer approximately 50% additional thrust capacity compared to equivalent quadcopters, enabling safer thrust margins while maintaining control authority (5). Over-actuated aerial platforms demonstrate capability for full six-DOF motions, including hovering at non-horizontal attitudes (8).

2.3 Manipulator Design and Integration

Degrees of Freedom

Two-DOF manipulators have proven effective for basic manipulation tasks while maintaining manageable system complexity and weight constraints (6). More sophisticated applications require higher-DOF systems, with 3-DOF and 4-DOF manipulators becoming increasingly common (7).

End-Effector Technologies

End-effector design has evolved to include proximity sensor-based grasping systems that can accommodate position errors within specified tolerances, typically achieving successful grasps within a 50 mm radius (9).

2.4 Control Systems and Methodologies

Coupled System Dynamics

The Lagrange–D’Alembert formulation is the standard approach for deriving system dynamics of coupled aerial manipulation platforms (13). This unified model enables controllers that manage complex interactions between flight dynamics and manipulator motion (13).

Vision-Based Control

ArUco marker-based systems provide centimeter-level positioning accuracy when integrated with image processing techniques (4). Advanced deblurring networks address motion blur challenges inherent in aerial platforms, improving marker detection reliability (11).

Model Predictive Control

Model predictive control (MPC) approaches have gained traction for their ability to handle system constraints and optimize trajectory performance in aerial manipulation tasks (7).

2.5 Sensor Integration and Perception

Multi-modal sensing, combining RGB cameras, depth sensors, and Inertial Measurement Unit (IMUs), has become standard practice in aerial manipulation applications (11). Real-time kinematic (RTK) GPS systems achieve centimeter-level positioning accuracy, enabling reliable autonomous operation (2).

Industrial applications continue to expand, focusing on inspection, maintenance, and material handling in constrained or hazardous environments (1).

2.6 Current Challenges and Future Directions

Power and Endurance Limitations

Energy constraints remain a fundamental limitation for aerial manipulation systems, since manipulation hardware significantly reduces flight endurance compared to standard UAVs (2; 1). Ongoing research into battery technology improvements and energy-efficient actuator designs aims to alleviate these challenges (10).

Outdoor Operation Challenges

The transition from controlled indoor environments to outdoor operations introduces wind disturbance, varying lighting conditions, and GPS reliability issues (1; 2). Adaptive control approaches and robust perception systems are essential to maintain performance under these variable field conditions (4).

Multi-Robot Coordination

Collaborative aerial manipulation using multiple UAVs presents significant potential for complex tasks but brings challenges in coordination, communication requirements, and

safety considerations (1; 7). Future work must address decentralized planning algorithms and fault-tolerant communication architectures to enable reliable multi-robot cooperation (5).

The literature reveals that aerial manipulation has evolved from early proof-of-concept demonstrations to sophisticated systems capable of complex autonomous operations (2). Key advances in platform design, control methodologies, and sensor integration have enabled increasingly capable systems (1), though significant challenges remain in energy management, outdoor operation, and multi-robot coordination (3).

Part I

HARDWARE DEVELOPMENT

Chapter 3

Geometric and Dynamic Constraints for 2-DOF Manipulator Integration

The integration of a 2-degree-of-freedom (2-DOF) robotic manipulator on the underside of a quadcopter introduces significant design challenges related to the manipulator's physical dimensions. To ensure safe takeoff, landing, and stable flight without interference between the manipulator and the quadcopter's components, dimensional constraints must be derived. This chapter presents a detailed derivation of the constraints on the manipulator's arm lengths, considering the quadcopter's geometry, operational dynamics, and stability requirements. The manipulator is assumed to be planar, with two revolute joints, mounted at the quadcopter's center of mass (COM).

3.1 System Assumptions

The following assumptions guide the derivation:

- **Quadcopter Geometry:** The quadcopter has a symmetrical frame with four rotors, each at a distance d from the COM. The landing gear height is h , and the body width is w .
- **Manipulator Configuration:** The 2-DOF manipulator consists of two links with lengths l_1 and l_2 , masses m_1 and m_2 , and joint angles θ_1 and θ_2 . It is mounted at $z = -h_m$ (below the quadcopter's body, typically $h_m \leq h$).
- **Coordinate System:** The quadcopter's COM is at the origin $(0, 0, 0)$, with the z -axis upward and the manipulator operating in the x - z plane.

- **Operational Requirements:** The manipulator must not contact the ground during takeoff or landing, avoid the propellers and frame during operation, and minimize destabilizing effects on the quadcopter.

3.2 Derivation of Dimensional Constraints

3.2.1 Ground Clearance for Takeoff and Landing

To prevent the manipulator from contacting the ground during takeoff and landing, its lowest point must remain above the ground ($z > 0$).

The end-effector position in the x - z plane, relative to the mounting point at $(0, -h_m)$, is derived using forward kinematics:

$$x_e = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \quad (3.1)$$

$$z_e = -h_m + l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \quad (3.2)$$

The manipulator must satisfy $z_e > 0$ for all $\theta_1, \theta_2 \in [-\pi, \pi]$. The lowest point occurs when the manipulator is fully extended downward ($\theta_1 = \theta_2 = -\pi/2$):

$$z_e = -h_m - l_1 - l_2 \quad (3.3)$$

To ensure no ground contact:

$$-h_m - l_1 - l_2 > 0 \implies l_1 + l_2 < h_m \quad (3.4)$$

Since the mounting height h_m is typically less than or equal to the landing gear height h , we obtain:

$$l_1 + l_2 < h \quad (3.5)$$

Constraint 1: The total length of the manipulator arms must be less than the landing gear height:

$$l_1 + l_2 < h \quad (3.6)$$

A stowed configuration (e.g., folded upward with $\theta_1 = \theta_2 = \pi/2$) minimizes vertical extent:

$$z_e = -h_m + l_1 + l_2 \quad (3.7)$$

This configuration reduces ground contact risk but still requires $z_e > 0$.

3.2.2 Propeller and Frame Avoidance

During flight, the manipulator must not interfere with the quadcopter's propellers or frame.

The maximum reach in the x - y plane occurs when the manipulator is fully extended horizontally ($\theta_1 = \theta_2 = 0$):

$$x_e = l_1 + l_2 \quad (3.8)$$

The propellers are located at a distance d from the COM. To avoid interference:

$$l_1 + l_2 < d \quad (3.9)$$

Additionally, the manipulator must not collide with the quadcopter's body, which has a width w . Thus:

$$l_1 + l_2 < \frac{w}{2} \quad (3.10)$$

Constraint 2: The total arm length must be less than the minimum of the propeller distance and half the body width:

$$l_1 + l_2 < \min(d, \frac{w}{2}) \quad (3.11)$$

3.2.3 Aerodynamic Interference

The manipulator's motion can generate aerodynamic disturbances (e.g., drag or turbulence). To minimize interference with the rotor downwash, the manipulator should operate below the rotor plane ($z = 0$):

$$z_e = -h_m + l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) < 0 \quad (3.12)$$

This condition is generally satisfied if $l_1 + l_2 < h_m$, aligning with the ground clearance constraint.

3.2.4 Center of Mass and Stability

The manipulator's motion shifts the system's COM, affecting flight stability. The COM position is:

$$x_{\text{COM}} = \frac{m_1(l_1/2) \cos(\theta_1) + m_2[l_1 \cos(\theta_1) + (l_2/2) \cos(\theta_1 + \theta_2)]}{m_q + m_1 + m_2} \quad (3.13)$$

$$z_{\text{COM}} = \frac{m_1(-h_m + (l_1/2) \sin(\theta_1)) + m_2(-h_m + l_1 \sin(\theta_1) + (l_2/2) \sin(\theta_1 + \theta_2))}{m_q + m_1 + m_2} \quad (3.14)$$

where m_q is the quadcopter's mass. To maintain stability, the COM shift is limited to acceptable tolerances δ_x, δ_z :

$$|x_{\text{COM}}| < \delta_x, \quad |z_{\text{COM}}| < \delta_z \quad (3.15)$$

Assuming small manipulator masses ($m_1, m_2 \ll m_q$), the maximum COM shift occurs when the arms are fully extended:

$$l_1 + l_2 < \frac{\delta_x(m_q + m_1 + m_2)}{\max(m_1, m_2)} \quad (3.16)$$

Constraint 3: The arm lengths should satisfy:

$$l_1 + l_2 < \frac{\delta_x(m_q + m_1 + m_2)}{\max(m_1, m_2)} \quad (3.17)$$

3.2.5 Torque and Control Considerations

The manipulator's motion induces reaction torques on the quadcopter, proportional to the moments of inertia:

$$I_1 = \frac{1}{3}m_1l_1^2, \quad I_2 = \frac{1}{3}m_2l_2^2 \quad (3.18)$$

To minimize control effort, the arm lengths are constrained by the quadcopter's control authority:

$$l_1, l_2 < l_{\text{max}} \quad (3.19)$$

where l_{max} depends on the maximum rotor thrust and control capabilities.

3.3 Combined Dimensional Constraints

Combining all constraints, the manipulator arm lengths l_1 and l_2 must satisfy:

$$l_1 + l_2 < \min \left(h, d, \frac{w}{2}, \frac{\delta_x(m_q + m_1 + m_2)}{\max(m_1, m_2)}, l_{\text{max}} \right) \quad (3.20)$$

Additionally, individual arm lengths are constrained:

$$l_1, l_2 < l_{\text{max}} \quad (3.21)$$

Chapter 4

Two-DOF Manipulator Design

4.1 Version1 : Separate Torque and Balance Links

The first iteration of the manipulator design utilizes a two-degree-of-freedom (2-DOF) configuration, developed for aerial manipulation tasks. The design features a parallel link arrangement in the primary arm segment, consisting of two structural arms connecting the base servo to the secondary joint. While both links run in parallel and maintain the geometric integrity of the structure, they serve distinct functional roles.

Torque Link

The torque link is the active load-bearing member in this configuration. It is directly mounted to the shaft of the servo motor and is responsible for transmitting all actuation torque to the next joint in the manipulator chain. As the sole carrier of the driving force, this link experiences continuous cyclic loading during operation, which can result in uneven load distribution, increased localized stress, and accelerated mechanical wear over time. This asymmetry not only impacts the mechanical durability of the system but also imposes limitations on the payload capacity and response precision of the manipulator.

Balance Link

The balance link, while structurally identical in appearance, is passively connected and does not contribute to torque transmission. Its primary role is to provide geometric stability and resist out-of-plane deformation. The passive link helps prevent the arm from twisting under its own weight or due to dynamic forces during drone flight, thereby improving spatial alignment during motion.

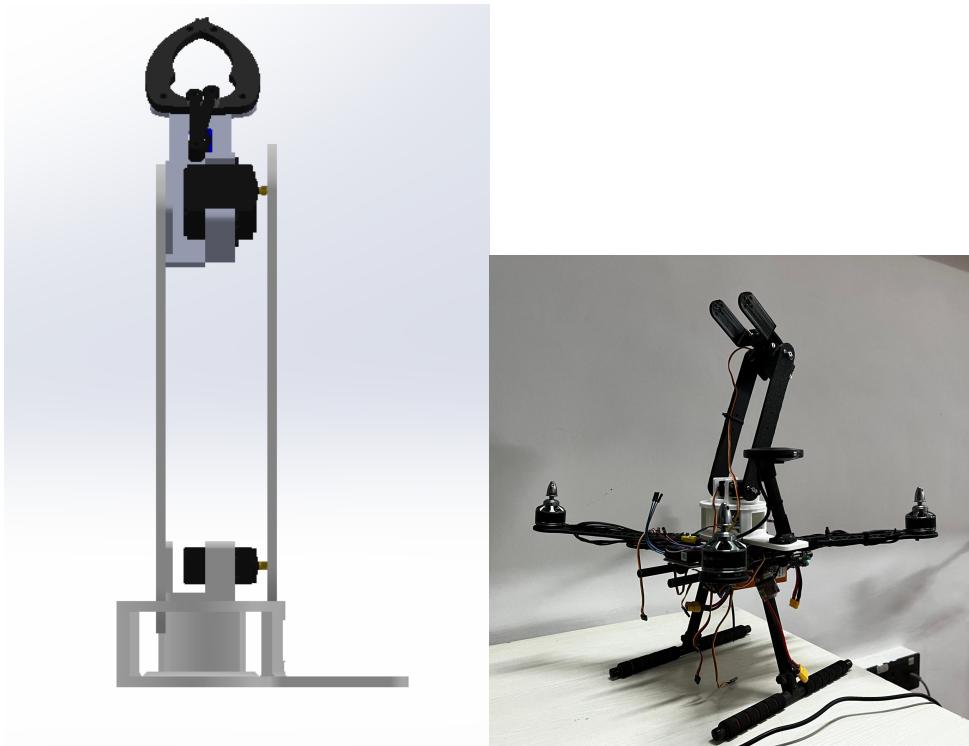


FIGURE 4.1: Separate torque and balance link



FIGURE 4.2: Separate torque and balance link : Base and End mounts

Limitations of the Design

While the use of parallel links appears to offer visual symmetry and some stabilization, this design proved inefficient in practical applications for several reasons:

- *Unequal Load Distribution:* Since only one link transmits torque, the system lacks balanced load sharing. This creates asymmetric loading at both the servo shaft and the joints, which leads to uneven wear and potential misalignment over extended usage.

- *Reduced Mechanical Efficiency:* The structural redundancy of the passive link adds weight without contributing to actuation. In aerial platforms where weight is a critical constraint, this leads to sub-optimal use of payload capacity.
- *Induced Vibrations:* The asymmetric torque transmission introduces micro-vibrations during motion, especially under dynamic loads or high-speed maneuvers. These vibrations propagate through the manipulator, affecting end-effector stability and control accuracy.
- *Limitations in Torque Handling:* The concentration of mechanical stress on a single link restricts the maximum torque the system can handle, which in turn limits the manipulator's ability to interact with heavier or more resistant objects.

In conclusion, while Version 1 served as a functional prototype for basic manipulation tasks, its performance was constrained by design inefficiencies that emerged during practical testing. The next iteration focused on improving torque distribution, structural symmetry, and dynamic stability to enhance overall performance.

4.2 Version2 : Dual Link Torque Distribution

Building on the insights gained from the limitations of Version 1, the second iteration of the manipulator arm introduces a significant structural refinement: mechanical coupling between both parallel links of the primary arm. As shown in Figure [X], this version features a rigid cross-member connecting the two links midway along their span. This design shift addresses the shortcomings of asymmetric torque transmission and improves system-wide mechanical performance.

Coupled Link Configuration

Unlike the previous version where only one link transmitted torque and the other acted merely as a stabilizer, Version 2 integrates both links into the torque path. The mechanical coupling between the two arms—achieved through a rigid cross-brace—ensures that torque applied by the servo motor is evenly distributed across both links. This transforms the arm into a unified structural element capable of balanced load sharing.

Benefits of Torque Distribution

The even distribution of torque between the two links leads to several important advantages:

- *Reduced Stress Concentration:* By sharing the actuation load across both sides of the arm, the design minimizes localized stress points on the servo shaft and structural joints. This extends the mechanical lifespan of the system and enables safer handling of higher payloads.
- *Improved Mechanical Efficiency:* The use of both links for torque transmission increases the effective stiffness of the arm. This results in more efficient force delivery to the end-effector with reduced torsional losses or deformation.

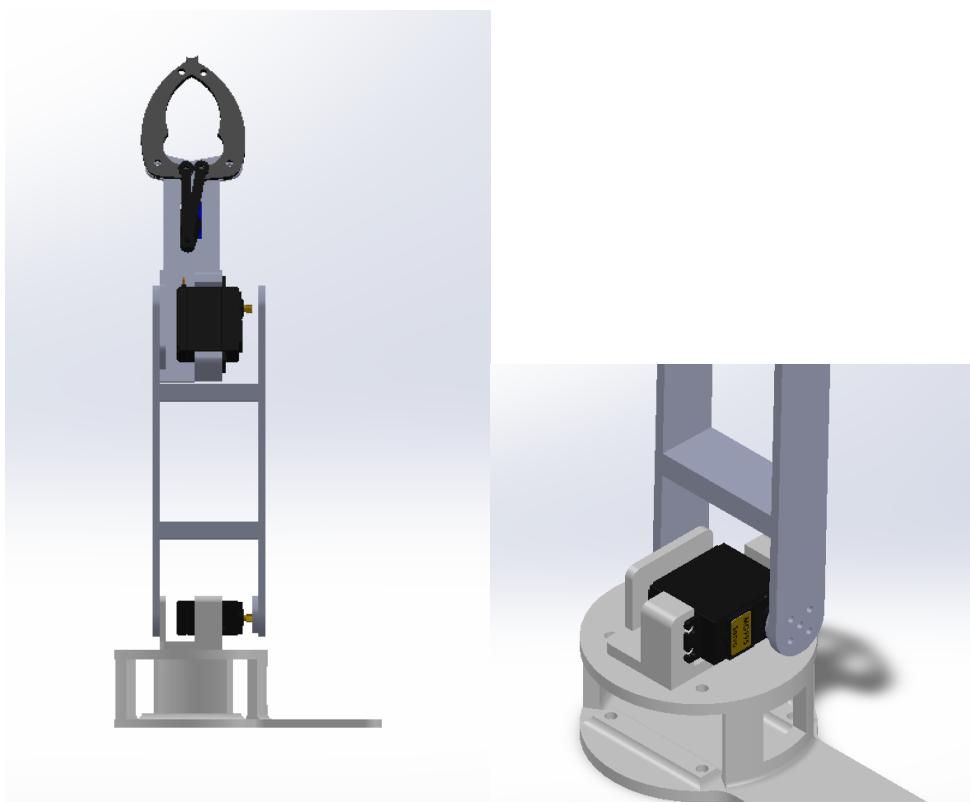


FIGURE 4.3: Dual Link Torque Distribution Design

Enhanced Stability and Vibration Damping

A major design motivation for this configuration was to counteract the vibrations and dynamic instability observed in Version 1. The symmetrically coupled links provide a closed structural loop that:

- *Increases Torsional Rigidity:* The rigid cross-connection acts as a brace that restricts twisting motion between the links, enhancing overall torsional stiffness.
- *Suppresses Oscillations:* During drone flight or manipulator motion, vibrations are absorbed and distributed across a broader structure rather than propagating linearly

down a single link. This significantly reduces oscillations reaching the gripper or affecting precision tasks. *Improves End-Effector Control:* With reduced vibration transmission, the manipulator maintains a steadier pose during actuation, which is critical for aerial manipulation tasks that require precision under dynamic conditions

4.3 Version3: Manipulator Arm Below the Quadcopter Base

In the subsequent design iteration, the manipulator arm was relocated beneath the base of the quadcopter. This structural change was motivated by a combination of imaging performance issues, spatial constraints, and thrust requirements.

1. *Vibration-Induced Imaging Errors-* Initially, the onboard camera—mounted below the drone—was positioned far from the system’s center of mass. This made it particularly susceptible to amplified vibrations during flight, resulting in motion blur that rendered ArUco marker detection unreliable. Mounting the manipulator arm below helped reduce the transmission of structural vibrations to the camera, leading to improved image clarity and more robust marker detection. A detailed analysis of vibration effects is presented in a later section.
2. *Physical Interference with Propellers -* In the earlier configuration, the manipulator arm was positioned above the drone body. However, this placement led to frequent collisions or near-interference with the spinning propellers, especially during dynamic maneuvers or when the manipulator moved. The restricted vertical clearance significantly limited the arm’s range of motion and posed a safety risk to both the hardware and surrounding environment.
3. *Accommodating Larger Propellers for Increased Payload -* The addition of the manipulator arm and onboard battery increased the drone’s overall takeoff weight, requiring higher thrust for stable flight. To meet this requirement, larger propellers were integrated into the design. Consequently, the arm had to be moved downward to create sufficient clearance and prevent interference with the extended propeller blades.

This design change ensured better mechanical integration, improved flight safety, and more reliable vision-based manipulation—all essential for stable aerial operation with onboard manipulation capability.

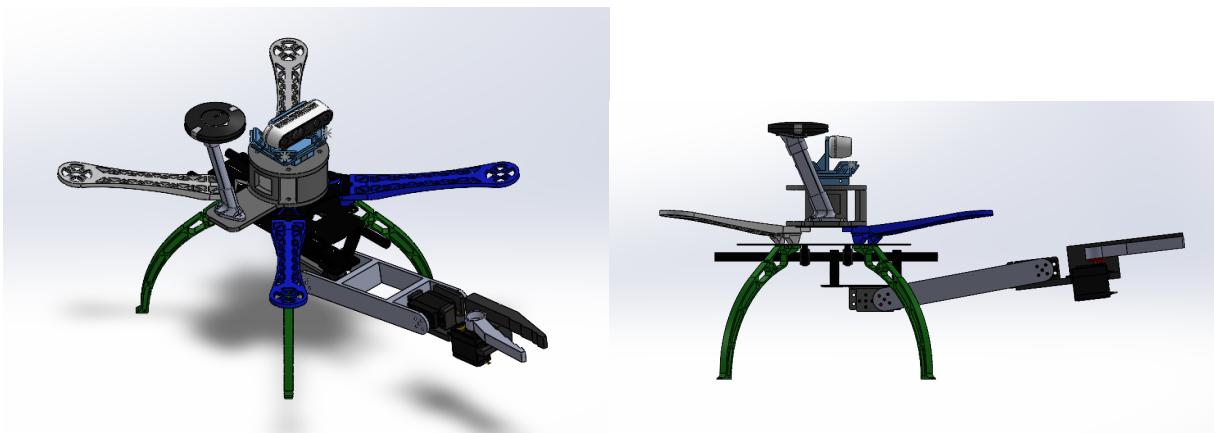


FIGURE 4.4: Arm Mounted Below

Chapter 5

End-Effector Development

5.1 Version1 : Gear Driven Parallel Gripper

The initial design of the end-effector features a gear-driven parallel gripper. This version was aimed at providing a basic gripping mechanism with synchronized finger motion, using a single actuator to drive both jaws.

Design Description

The gripper employs two oppositely placed fingers mounted on intermeshing spur gears. One of the gears is directly actuated by a servo motor, while the other rotates in the opposite direction through gear meshing. This configuration allows both fingers to open and close simultaneously in a parallel motion, offering geometric simplicity and easy control.

Limitations and Performance Issues

While this mechanism successfully achieves symmetrical movement, it presented several critical drawbacks during testing and use:

- *Low Torque Transmission:* The torque generated by the servo is first transferred to a gear and then shared with the second finger via meshing. This indirect transmission introduces losses due to backlash and friction within the gear pair, significantly reducing the effective gripping force. The lightweight plastic gears used in early prototypes further limited the torque handling capacity.
- *Reduced Gripping Surface Area:* The contact surfaces of the fingers were designed as narrow linear profiles, as visible in the figure. This limited surface area was

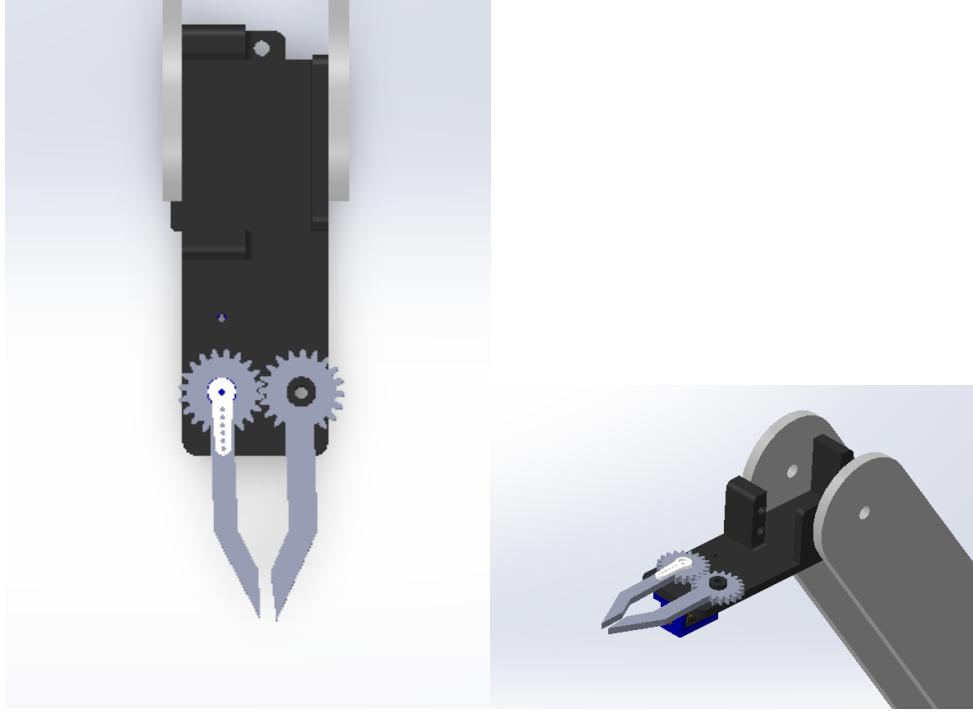


FIGURE 5.1: Gear Driven Parallel Gripper

insufficient for establishing secure contact with irregular or slightly curved objects. As a result, the frictional force generated at the interface was often inadequate to hold or manipulate payloads reliably.

- *Slippage and Object Instability:* Due to the reduced frictional force and low torque availability, the gripper frequently experienced object slippage during dynamic movements, especially when the drone was in motion or under acceleration.
- *High Sensitivity to Vibration:* The minimal holding force also made the gripper vulnerable to vibrational disturbances from the drone or manipulator arm. This often led to a loss of grip or unintended object drops during flight tests.

5.2 Version2 : High Torque Gripper

To overcome the limitations observed in the gear-driven gripper design, the second version of the end-effector adopts a high torque, single-actuation gripper mechanism.. This design prioritizes gripping strength, structural simplicity, and enhanced surface contact to enable more reliable object manipulation in dynamic environments.

Design Description

The Version 2 gripper features an asymmetric jaw configuration—one jaw remains fixed while the other is actuated by a high-torque servo motor. Specifically, the system integrates the MG995 servo motor, which provides a stall torque of up to 10 kg·cm at 4.8V. The increased torque capacity directly translates to greater gripping force, allowing the gripper to securely grasp a wider range of objects.

The gripping surface has also been significantly upgraded. The moving jaw is equipped with a wider, stepped profile, offering multiple points of contact and improved conformity to irregular object surfaces. This profile increases the effective contact area and introduces geometric features that enhance frictional grip, reducing the chance of slippage.

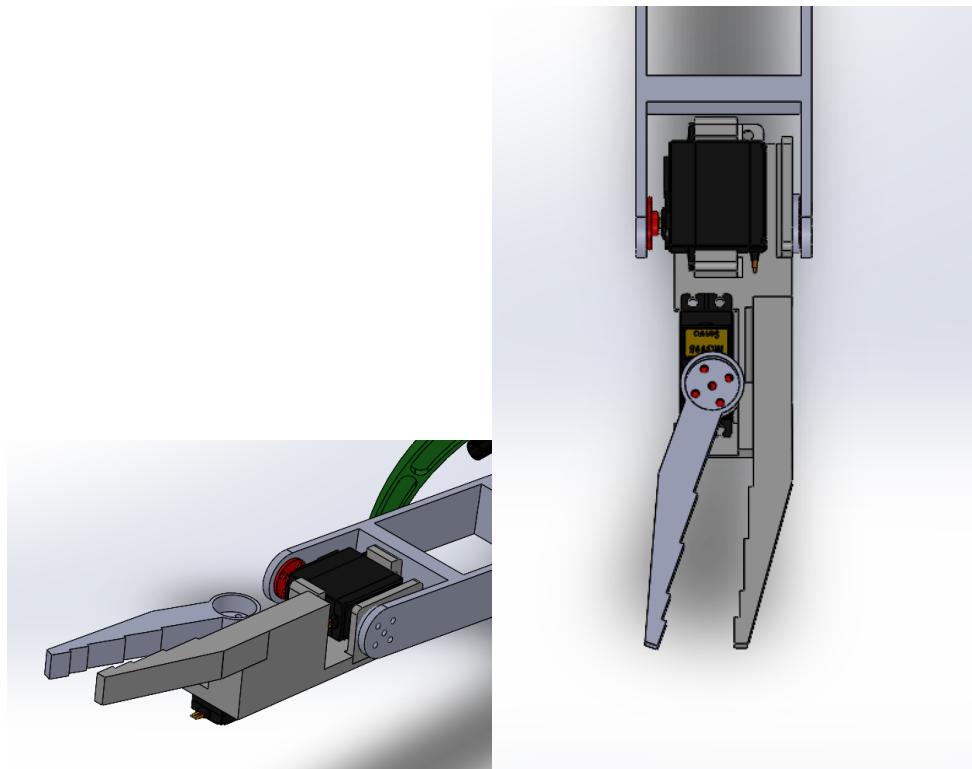


FIGURE 5.2: High Torque Gripper

Advantages Over Previous Design

This version addresses all major issues identified in the gear-driven gripper (Version 1):

- *Improved Torque Transmission:* The MG995 servo delivers torque directly to the moving jaw without intermediate gear losses. This eliminates issues of backlash and indirect force transfer, resulting in a more responsive and reliable grip.
- *Increased Contact Area:* The gripping surfaces are now broader and incorporate stepped features to accommodate varying object geometries. This design

significantly increases the normal force distribution, thereby improving grip reliability.

- *Higher Frictional Grip:* The stepped profile of the moving jaw acts as a mechanical interlock for irregular objects, increasing friction and preventing slipping during manipulation or drone movement.
- *Reduced Vibration Sensitivity:* With stronger gripping force and better surface conformity, the new gripper is less sensitive to vibration and dynamic loads, addressing a key vulnerability of the earlier parallel gear-driven version.

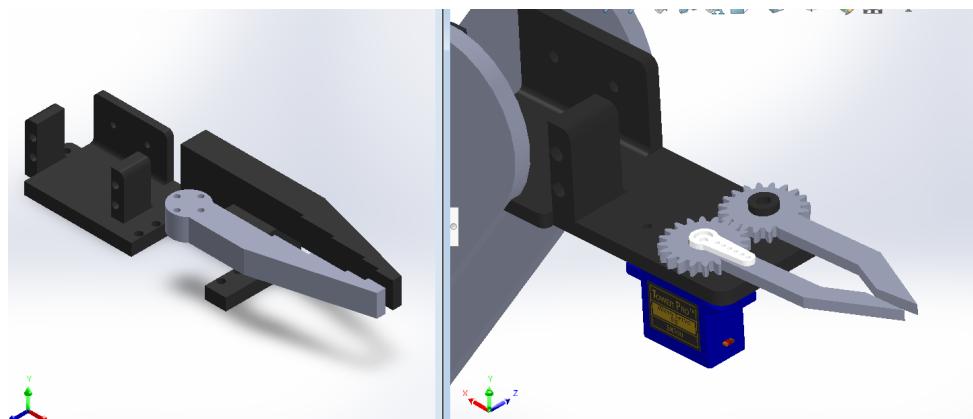


FIGURE 5.3: Comparison of End effectors: Gear driven and High Torque

Chapter 6

Actuator Integration and Structural Optimization

Efficient actuator selection and careful structural balancing were essential to ensure stable flight performance and precise manipulation during aerial operations. This chapter outlines the integration of servo actuators for torque delivery and the structural design optimizations carried out to manage weight, balance, and inertial compensation.

6.1 Servo Actuators for Torque

The manipulator arm required a robust actuation mechanism capable of delivering high torque while maintaining mechanical stability. Initially, the Waveshare dual-sided axle servo was selected due to its symmetric load-handling capabilities. However, due to limited library support and integration difficulties, we opted to simplify the setup and proceeded with MG995 servos—commonly used, single-sided axle servos providing a stall torque of 10 kg-cm at 4.8V input.

While the MG995 met the torque requirement, the single-sided axle configuration introduced structural challenges. As the load on the manipulator increased, the uneven distribution of force led to excessive stress on the servo axle, causing vibrations and reduced positional accuracy.

To address this, custom-designed mounts were fabricated (see figure below). These mounts distributed the arm’s weight more evenly across the servo structure, reducing the axial load and enhancing both servo life and operational smoothness. This modification significantly mitigated vibration-induced instabilities during manipulator motion.

6.2 Weight Balancing & Structural Weight Optimization

The dynamic movement of the manipulator arm introduced significant shifts in the system's center of mass (CoM), as well as variations in moments of inertia during flight. These changes affected the drone's stability and control response, particularly during hovering and sudden maneuvers.

The only significant additional mass onboard—aside from the manipulator—was the battery pack (300g). To counterbalance the arm's weight, the battery was strategically mounted at the rear end of the quadcopter, restoring the system's center of mass closer to the geometric center and ensuring better balance during static and dynamic conditions.

In addition to hardware-level compensation, the flight firmware played a crucial role in managing inertia changes. The onboard IMU, magnetometer, and GPS data were processed through an Extended Kalman Filter (EKF) to estimate the system state accurately. This sensor fusion allowed the position hold and altitude hold modes to compensate for moderate inertial shifts effectively. However, these corrections were bounded by the physical limits of the motors and propellers.

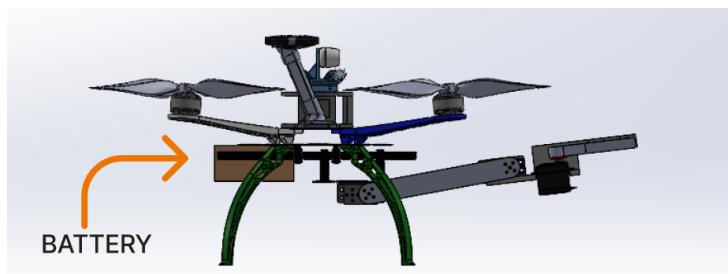


FIGURE 6.1: Weight Balancing: Placing battery behind the gripper

To meet the increased thrust demands, larger 13-inch propellers were introduced. However, frame constraints prevented the use of larger sizes. Thus, further optimizations focused on weight reduction across the structure. Key actions included:

1. Perforation of manipulator arm components to minimize mass without compromising structural integrity.
2. Optimized routing of wiring and efficient use of materials to eliminate excess weight.

3. Careful placement of electronics and payload components to ensure minimal asymmetry in weight distribution.

These cumulative efforts resulted in a more agile and balanced flight system, capable of maintaining stability even with onboard manipulation.

Chapter 7

Simulation and Testing

A systematic simulation and testing process was adopted to validate the aerial manipulation platform across different stages of development. While simulation provided an initial framework for testing perception and control algorithms, limitations in replicating realistic aerodynamics and hardware behavior prompted a shift toward real-world testing. This chapter elaborates on the initial simulation strategy, challenges encountered, and detailed results from physical testing—including stabilization analysis and task-based flight trials.

7.1 Gazebo Simulation

The development process began by building a simulation environment in Gazebo to test the manipulator’s control logic, ArUco marker detection, and motion strategies in a safe and repeatable setting. A complete URDF model of the quadcopter and 2-DOF manipulator was created, incorporating appropriate joint definitions, link inertias, and control plugins. This setup allowed early testing of vision-based manipulation strategies without risking hardware damage.

However, as development progressed, significant limitations of the Gazebo physics engine became apparent. One of the most critical issues was the inability to accurately simulate real-world thrust generation. In Gazebo, rotor thrust is generally modeled as a simplified pseudo-force applied at the ends of the quadcopter arms. This approximation does not account for complex aerodynamic behaviors such as wind resistance, propeller wash interactions, or the distribution of thrust vectors across dynamic flight envelopes. As a

result, the simulated flight dynamics deviated significantly from real-world behavior, especially during aggressive maneuvers or when external forces acted on the system.

Furthermore, the simulation architecture was designed to bypass the onboard flight controller, sending direct PWM values to Electronic Speed Controllers (ESCs) using custom ROS nodes. While this setup worked effectively in simulation, it lacked the robust safety features of traditional flight stacks. Key capabilities such as altitude hold, position hold, failsafe triggers, and manual override were lost. This meant that the platform could no longer respond predictably to failures or disturbances, posing substantial risks for real-world deployment.

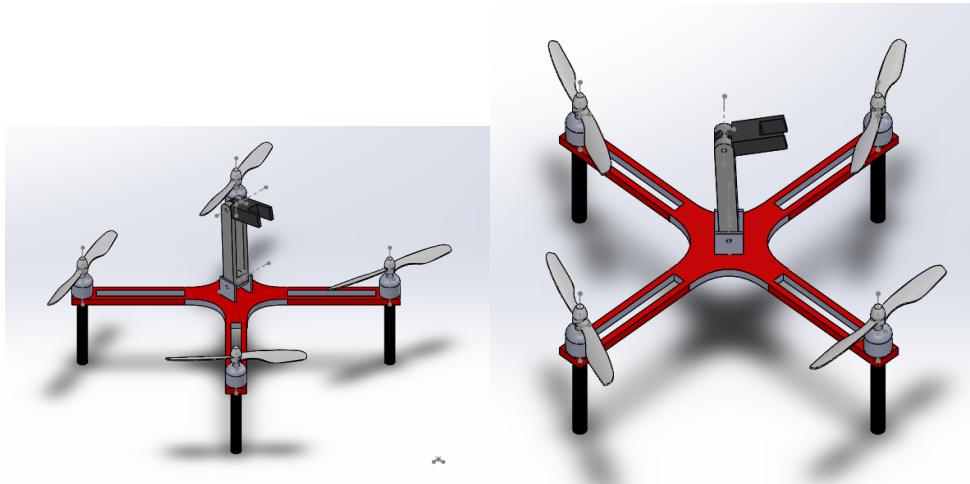


FIGURE 7.1: Gazebo Simulation Environment

Additionally, using a custom low-level control approach in simulation required a highly accurate representation of the system's mass properties. Replicating the exact inertia and center of mass of the physical platform proved to be particularly challenging, especially considering the non-linear dynamics of the manipulator during motion. These discrepancies led to significant difficulties in tuning controller gains, as values that worked well in simulation were unstable in real-world conditions.

In light of these challenges, the simulation phase was concluded early. It was decided to transition to hardware testing while relying on the Pixhawk flight controller's onboard low-level controllers (running ArduPilot) for all flight operations. This shift enabled a more accurate assessment of real-world behavior while retaining safety-critical features and access to proven flight control modes.

7.2 Vertical Arm Stabilization

Following the transition to hardware testing, one of the primary objectives was to ensure that the manipulator remained stable during quadcopter motion, especially in tasks that required high precision such as docking or inspection.

In aerial systems, especially multirotors, pitch and roll variations are inherent during flight. These changes can adversely affect the orientation of any mounted payload. When a manipulator is mounted on the platform, even small angular disturbances can propagate to the end-effector, severely degrading task performance.

To address this, a stabilization mechanism was implemented using the Pixhawk's onboard IMU data. The attitude angles—primarily pitch—were read in real time, and corrective commands were passed to the manipulator's servos to maintain a vertical orientation of the arm. This active compensation ensured that the end-effector stayed aligned with the target during flight, reducing error in visual detection and mechanical gripping.



FIGURE 7.2: Vertical Stabilization of arm

This vertical stabilization strategy effectively decoupled the arm's orientation from the drone's attitude, enabling more accurate and reliable manipulation during aerial operation.

7.3 Manual Flight Testing

Manual testing was conducted to evaluate real-world behavior under realistic operating conditions. These tests focused on both general flight performance with the manipulator and task-specific trials involving object interaction.



FIGURE 7.3: Drone assembled with manipulator attached below and camera on the top

General Flight Tests

In position hold and altitude hold modes, the manipulator was actuated to assess its impact on flight stability. It was observed that significant movements of the arm, particularly when extended away from the body, resulted in noticeable shifts in the center of mass and inertia of the system. The quadcopter responded by dynamically altering its altitude and pitch to compensate and restore equilibrium.

While the onboard control algorithms—including EKF-based state estimation—handled minor CoM shifts well, larger movements caused altitude oscillations and delayed stabilization. This behavior posed a challenge for consistent mission execution, especially when manipulating heavier payloads or operating in cluttered environments.

Gripping Task: Juice Box Pickup

A manual gripping test was also conducted to demonstrate practical manipulation. The quadcopter, held in stabilized flight, was commanded to grasp a small juice box using the onboard manipulator. During this operation, two critical observations were made:

1. *Propeller Clearance:* The initial manipulator placement above the quadcopter resulted in minimal clearance between the end-effector and rotating propellers. This posed a high risk of collision, especially when the drone made reactive adjustments or experienced drift. This finding validated the need to relocate the manipulator arm below the quadcopter base, as detailed in Chapter 3.
2. *Hover Instability During Gripping:* The quadcopter showed transient instability when the arm grasped the object. The addition of the juice box altered the system's mass and thrust requirements, causing a short period of wobble and altitude variation. This behavior was mitigated by tuning the Hover Thrust Learning parameters, enabling the flight controller to adapt more rapidly to changing payload conditions and maintain consistent altitude.

The insights gained from these manual flight tests played a crucial role in informing design revisions, control architecture decisions, and safety measures implemented in the final system. While simulation was valuable for initial development, real-world testing proved essential for achieving a reliable and safe aerial manipulation platform.

Part II

QUADCOPTER AND FIRMWARE DEVELOPMENT

Chapter 8

Kinematic and Dynamic Modeling of the System

This chapter presents the kinematic and dynamic models of a quadcopter with a 2-DOF manipulator, as derived in (13), to provide a comprehensive understanding of the system's behavior for precise control and manipulation tasks.

8.1 Kinematic Model

The kinematic model describes the motion relationships between the quadcopter and its attached 2-DOF manipulator without considering forces. The system is modeled as a unified entity with generalized coordinates defined to capture the position and orientation of both components.

8.1.1 Coordinate Frames and Generalized Coordinates

The inertial coordinate frame is denoted as O_I , while the body-fixed coordinate frame of the quadcopter is O_b , and those of the manipulator links are O_1 and O_2 for links 1 and 2, respectively, located at their centers of mass. The generalized coordinates are defined as:

$$q = \begin{bmatrix} p^T & \Phi^T & \eta^T \end{bmatrix}^T,$$

where $p = [x \ y \ z]^T$ is the position of the quadcopter's center of mass in the inertial frame, $\Phi = [\phi \ \theta \ \psi]^T$ represents the Euler angles (roll, pitch, yaw), and $\eta = [\eta_1 \ \eta_2]^T$

denotes the joint angles of the 2-DOF manipulator, defined about the y -axis of the quadcopter's body-fixed frame.

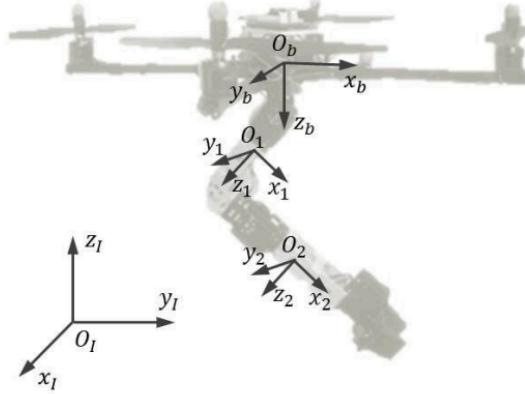


FIGURE 8.1: Configuration of the coordinates for the combined system

8.1.2 Velocity Relationships

The translational velocity of the quadcopter in the inertial frame, \dot{p} , is related to its body-fixed velocity, \dot{p}^b , via the rotation matrix $R_t \in SO(3)$:

$$\dot{p} = R_t \dot{p}^b.$$

The angular velocity of the quadcopter in the inertial frame, $\omega = [\omega_x \ \omega_y \ \omega_z]$, and body-fixed frame, $\omega^b = [\omega_x^b \ \omega_y^b \ \omega_z^b]$, are related by:

$$\omega = R_t \omega^b, \quad \omega = T \dot{\Phi}, \quad \omega^b = R_t^T T \dot{\Phi} = Q \dot{\Phi},$$

where T is the transformation matrix mapping Euler angle rates to angular velocities, and $Q = R_t^T T$.

For the manipulator links, the position of the center of mass of link i ($i = 1, 2$) in the inertial frame, p_i , is:

$$p_i = p + R_t p_i^b,$$

where p_i^b is the position in the body-fixed frame O_b . The translational and angular velocities of each link are:

$$\dot{p}_i^b = J_t \dot{\eta}, \quad \omega_i^b = J_r \dot{\eta},$$

where $J_t, J_r \in \mathbb{R}^{2 \times 2}$ are the translational and rotational Jacobian matrices, respectively. The velocities in the inertial frame are:

$$\dot{p}_i = \dot{p} + \dot{R}_t p_i^b + R_t \dot{p}_i^b = \dot{p} + \bar{\omega}_b R_t p_i^b + R_t J_t \dot{\eta},$$

$$\omega_i = \omega + R_t J_r \dot{\eta},$$

where $\bar{\omega}_b$ is the skew-symmetric matrix of ω_b . These relationships are expressed in matrix form as:

$$\dot{p} = M_{t,b} \dot{q}, \quad \omega = M_{r,b} \dot{q}, \quad \dot{p}_i = M_{t,i} \dot{q}, \quad \omega_i = M_{r,i} \dot{q},$$

with appropriately defined matrices $M_{t,b}$, $M_{r,b}$, $M_{t,i}$, and $M_{r,i}$, enabling the mapping of generalized coordinate velocities to the velocities of the quadcopter and manipulator links.

8.2 Dynamic Model

The dynamic model accounts for the forces and torques acting on the combined system, crucial for designing controllers that ensure stability and precise manipulation.

8.2.1 Lagrange-D'Alembert Formulation

The dynamics are derived using the Lagrange-D'Alembert equation:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \tau + \tau_{\text{ext}},$$

where $\mathcal{L} = \mathcal{K} - \mathcal{U}$ is the Lagrangian, \mathcal{K} is the total kinetic energy, \mathcal{U} is the total potential energy, τ is the generalized force, and τ_{ext} represents external disturbances.

8.2.2 Kinetic and Potential Energy

The total kinetic energy is the sum of the quadcopter's and manipulator links' contributions:

$$\mathcal{K} = \mathcal{K}_b + \sum_{i=1}^2 \mathcal{K}_i,$$

$$\mathcal{K}_b = \frac{1}{2} \dot{p}^T m_b \dot{p} + \frac{1}{2} \dot{\Phi}^T T^T R_t I_b R_t^T T \dot{\Phi},$$

$$\mathcal{K}_i = \frac{1}{2} \dot{p}_i^T m_i \dot{p}_i + \frac{1}{2} \omega_i^T (R_t R_i) I_i (R_t R_i)^T \omega_i,$$

where m_b and m_i are the masses of the quadcopter and link i , respectively, and I_b and I_i are their inertia matrices. The total potential energy is:

$$\mathcal{U} = m_b g e_3^T p + \sum_{i=1}^2 m_i g e_3^T (p + R_t p_i^b),$$

where g is the gravitational acceleration, and $e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is the unit vector along the z-axis.

8.2.3 Dynamics Equation

Substituting the kinetic and potential energies into the Lagrangian, the dynamics equation is:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_{\text{ext}},$$

where $M(q) \in \mathbb{R}^{8 \times 8}$ is the positive definite, symmetric inertia matrix, $C(q, \dot{q})$ is the Coriolis matrix, and $G(q)$ accounts for gravitational effects. The inertia matrix is:

$$M(q) = M_{t,b}^T m_b M_{t,b} + M_{r,b}^T R_t I_b R_t^T M_{r,b} + \sum_{i=1}^2 \left(M_{t,i}^T m_i M_{t,i} + M_{r,i}^T (R_t R_i) I_i (R_t R_i)^T M_{r,i} \right),$$

where R_i is the transformation matrix from link i to the quadcopter's body-fixed frame. The Coriolis matrix elements are computed as:

$$c_{kj} = \sum_{i=1}^8 \frac{1}{2} \left\{ \frac{\partial m_{kj}}{\partial q_i} - \frac{\partial m_{kj}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right\},$$

and the gravity term is:

$$G(q) = \frac{\partial \mathcal{U}}{\partial q}.$$

8.2.4 Control Inputs

The generalized forces τ include the quadcopter's total thrust F , torques τ_x, τ_y, τ_z , and manipulator joint torques τ_η . These are related to τ via:

$$\begin{bmatrix} F \\ \tau_x \\ \tau_y \\ \tau_z \\ \tau_\eta \end{bmatrix} = \begin{bmatrix} R_t(3,3) & 0 & 0 \\ 0 & Q^{-1} & 0 \\ 0 & 0 & I_{2 \times 2} \end{bmatrix}^{-1} \begin{bmatrix} \tau(3) \\ \vdots \\ \tau(8) \end{bmatrix},$$

with desired roll and pitch angles derived as:

$$\begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} = \frac{1}{F} \begin{bmatrix} \cos \psi & \sin \psi \\ \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} \tau(1) \\ \tau(2) \end{bmatrix},$$

assuming small roll and pitch angles.

Chapter 9

Aerial Platform Design Specifications

This chapter outlines the core hardware components and control strategies implemented on the aerial manipulation platform. These include the integration of the flight controller, thrust system configuration, sensor suite for navigation and localization, power management architecture, and implementation of hover thrust learning for dynamic weight compensation. Each subsystem has been selected and tuned to ensure a balance between payload capacity, stability, accuracy, and flight endurance.

9.1 Flight Controller Integration

The system is controlled by the Pixhawk Cube Orange flight controller, running ArduPilot firmware version 4.4.1. ArduPilot offers a versatile and extensible platform capable of advanced flight modes such as Loiter, AltHold, and Guided, while also supporting direct MAVROS integration for offboard control.



FIGURE 9.1: Flight Controller: Pixhawk Cube orange with Ardupilot 4.4.1

To power the Pixhawk reliably, we employed a Battery Eliminator Circuit (BEC) that draws power from the main LiPo battery and regulates the output to stable 5V and 3.3V rails for the flight controller and connected peripherals. This architecture ensures that the Pixhawk remains unaffected by voltage fluctuations during aggressive maneuvers or motor current spikes, thereby maintaining consistent sensor readings and control signal processing.

9.2 Thruster System: Motor, ESC, and Propeller

The propulsion system comprises a set of four brushless DC motors by T-Motor, each rated at 470 KV, controlled by 30A electronic speed controllers (ESCs) and fitted with 13" × 4.7" propellers. The design parameters and measured thrust data are detailed below:

Parameter	Value
Flight Time	10–12 mins
Battery Capacity	5200 mAh
Quadcopter Weight	1400 g
Manipulator Weight	320 g
Battery Weight	600 g
Total Weight	2320 g
Motor KV Rating	470
Prop Size	13"
Prop Pitch	4.7"
ESC	30A
Max Thrust (per motor)	950 g
Combined Max Thrust	3800 g

TABLE 9.1: Technical specifications of the quadcopter with manipulator

The configuration provides sufficient thrust-to-weight ratio (1.6:1 at full throttle), ensuring safe takeoff, hover, and flight operations even with dynamic payload changes due to manipulator motion. The inclusion of larger 13" propellers was crucial to accommodate the increased weight from the manipulator (320 g) and onboard systems.

9.3 GPS and Auxiliary Sensors Integration

The quadcopter's localization and orientation are governed by data from the Here 3+ GPS module, integrated with the internal IMU and compass of the Pixhawk Cube Orange. This

sensor suite provides a robust state estimate via an Extended Kalman Filter (EKF), which fuses noisy sensor data into a consistent and accurate state estimate.

In standard operation, the system achieves positional accuracy in the range of 0.2–0.5 meters. However, for precision-demanding tasks such as contact-based manipulation or visual servoing, such error margins are unacceptable.

To address this, we implemented Real-Time Kinematics (RTK) using a fixed RTK Base Station and a mobile rover (GPS receiver on the drone). The base station, after being geolocated with high satellite fidelity, continuously computes differential corrections that are sent to the rover, allowing it to achieve centimeter-level accuracy. This reduces the horizontal error to 0.1–0.15 meters, enabling far more precise control and feedback during manipulation tasks.

9.4 Power Management

The aerial platform employs a carefully distributed power system to support various high-power and low-power subsystems. The main 4S LiPo battery (5200 mAh, 600 g) serves as the primary power source. The power flow and regulation are organized as follows:

1. *Main Power Distribution*: The battery feeds into a Power Distribution Board (PDB), which supplies power to the ESCs and motors directly.
2. *BEC Output for Flight Controller*: A dedicated BEC circuit ensures regulated 5V and 3.3V outputs to the Pixhawk, minimizing risk of brownouts during sudden current draw.
3. *Servo Powering*: Servos are powered through the signal rail connections on the Pixhawk. The servo signal pins provide a steady 5V supply sufficient to operate the MG995 actuators used in the manipulator.
4. *Peripheral Devices*: GPS, buzzer, LEDs, and basic telemetry modules are powered by the Pixhawk through its auxiliary ports.
5. *Onboard Computer (Raspberry Pi)*: Originally, the Raspberry Pi (used for visual processing) was powered by the main battery via a step-down module. However, due to power fluctuations affecting performance, a dedicated power source was later introduced (discussed in Section 11.2).

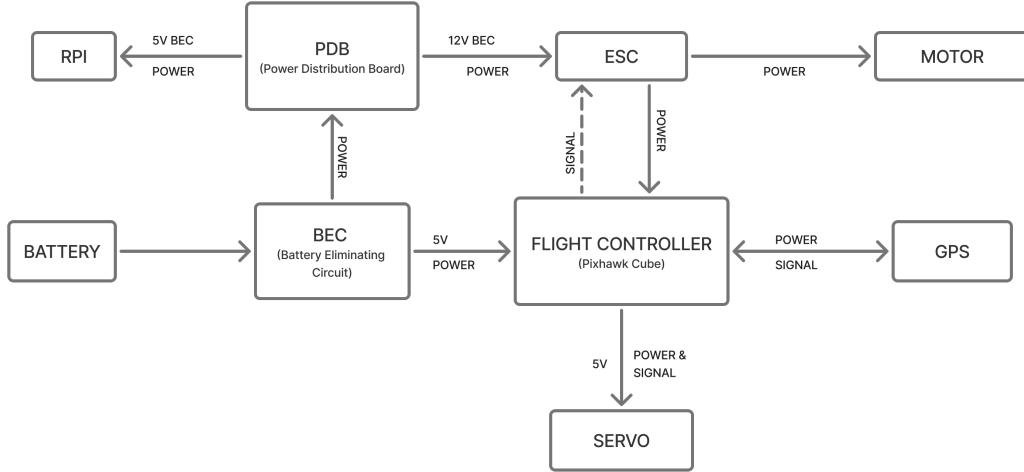


FIGURE 9.2: General power management on the quadcopter system

This modular power distribution approach enhances system stability, allows for efficient fault isolation, and ensures that critical components such as the flight controller and servos remain powered during power fluctuations.

9.5 Hover Thrust learning

Hover thrust learning was implemented to ensure consistent altitude performance across changing payload conditions—especially during manipulation tasks where the center of mass and moment of inertia may shift significantly.

Activation and Conditions

Hover thrust learning is controlled by the parameter `MOT_THST_HOVER_LEARN`. A value of:

- 1 enables learning in volatile RAM,
- 2 enables learning and saves the final result to EEPROM after disarming.

The feature activates only in altitude-controlled modes such as AltHold, Loiter, or PosHold and requires a stable hover condition with minimal stick inputs.

Internal Process

The algorithm utilizes:

- **Barometer** to measure altitude and identify vertical drift,
- **IMU** (accelerometer and gyroscope) to detect motion and attitude changes,
- **Motor output data** to measure how much throttle is being used to maintain hover.

During stable flight, the autopilot performs a moving average of throttle outputs to filter out short-term fluctuations due to wind or pilot input. This average value is computed as the actual thrust required for stable hover and is then written to the parameter *MOT_THST_HOVER*.

Benefits

- Enables adaptive altitude control as manipulator position changes,
- Improves system responsiveness to mass distribution variations,
- Enhances flight stability and reduces oscillations in vertical axis.

When *MOT_THST_HOVER_LEARN* = 2, this learned value is stored permanently, ensuring that future missions start with an updated estimate, thus reducing the need for manual re-tuning after every payload or configuration change.

Chapter 10

Firmware Development

This chapter outlines the firmware architecture implemented for the aerial manipulator platform. It discusses the communication interface between the flight controller and the onboard computer, the structure of data flow through ROS and MAVROS, methods for servo actuation, the wireless ground communication setup, and the key firmware parameters configured on the Pixhawk running ArduPilot 4.4.1.

10.1 Flight Controller and On Board Processor Communication

The Pixhawk Cube Orange flight controller, operating with ArduPilot firmware version 4.4.1, communicates with the onboard processor, a Raspberry Pi (RPI), through an FTDI (USB-to-UART bridge) interface. This setup enables robust serial communication between the two devices using the MAVLink protocol.

The FTDI connector used is based on a standard Future Technology Devices International chip, which converts USB signals to TTL UART. This allows the Raspberry Pi to interface with the Pixhawk's telemetry port over a virtual serial interface (e.g., `/dev/ttyUSB0`). The connector carries standard serial signals including TX (Transmit), RX (Receive), GND (Ground), and optionally power (VCC) or flow control signals (DTR/CTS).

The choice of FTDI was motivated by the following:

- Reliable serial communication bridge between USB-based computers and UART-enabled devices.

- Support for debugging and serial logging during development.
- Voltage level shifting to protect devices operating at different logic levels (e.g., 3.3V vs. 5V).
- Optional access to bootloader mode for firmware updates or troubleshooting.

Note: During development, we encountered an issue where powering the Pixhawk via USB (from the RPi) while also supplying regulated power via a BEC led to redundant voltage paths, potentially causing over-voltage conditions. This was resolved by isolating the power source from the USB side during active missions.

10.2 Data Flow Architecture

The data architecture is based on the Robot Operating System (ROS) running on the Raspberry Pi, with MAVROS serving as the middleware interface between ROS nodes and the Pixhawk.

The workflow is as follows:

- Initialization Phase:
 - The Raspberry Pi connects to the Pixhawk over `/dev/ttyUSB0` or `/dev/ttyACM0` at a baud rate of 57600 or 115200.
 - MAVROS launches and begins serial communication using MAVLink.
 - MAVLink heartbeats are exchanged, and a persistent connection is established.
- Command and Telemetry Flow:
 - Python-based ROS nodes publish commands to relevant topics (e.g., `/mavros/mission/push`) or call services.
 - MAVROS converts these ROS messages into corresponding MAVLink messages.
 - These messages are transmitted to the Pixhawk over the serial link.
 - The Pixhawk interprets the messages and updates flight state, actuator outputs, or mission plans accordingly.

In return, telemetry data from the Pixhawk (such as IMU status, GPS coordinates, battery voltage, etc.) is converted by MAVROS into ROS topics, allowing real-time access and logging within the ROS ecosystem.

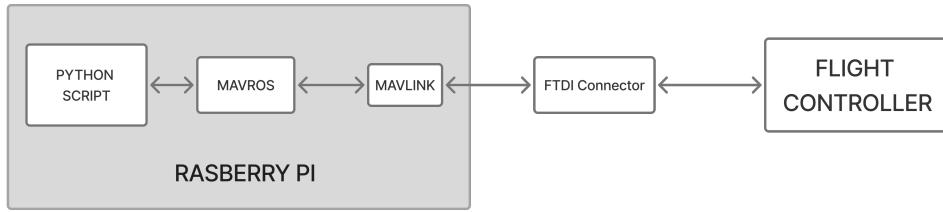


FIGURE 10.1: Data flow architecture between RPI and Pixhawk

10.3 MAVROS and Node Architecture

MAVROS, the ROS package that bridges MAVLink and ROS, forms the backbone of autonomous control in this system. MAVROS provides:

- Topics for reading telemetry and state estimates.
- Services for mission commands, takeoff/landing, servo actuation, etc.
- Actions for executing long-duration control routines like waypoint following or arm/disarm sequences.

The onboard ROS node architecture includes:

- A Python control node that processes vision input and calculates arm movements. Service clients for sending servo commands.
- Subscribers to sensor feedback (e.g., ArUco marker detection and IMU data).
- A main execution script that sequences the mission steps based on state feedback.

10.4 Servo Actuation Control

Servo actuation was critical for operating the onboard manipulator. Two approaches were explored:

1. **RC Channel Override (RC_CHANNELS_OVERRIDE)**

Initially, servos were connected directly to the Pixhawk and controlled via RCOVERRIDE topics in MAVROS. This method allows overriding any of the 18 RC input channels by publishing to `/mavros/rc/override`.

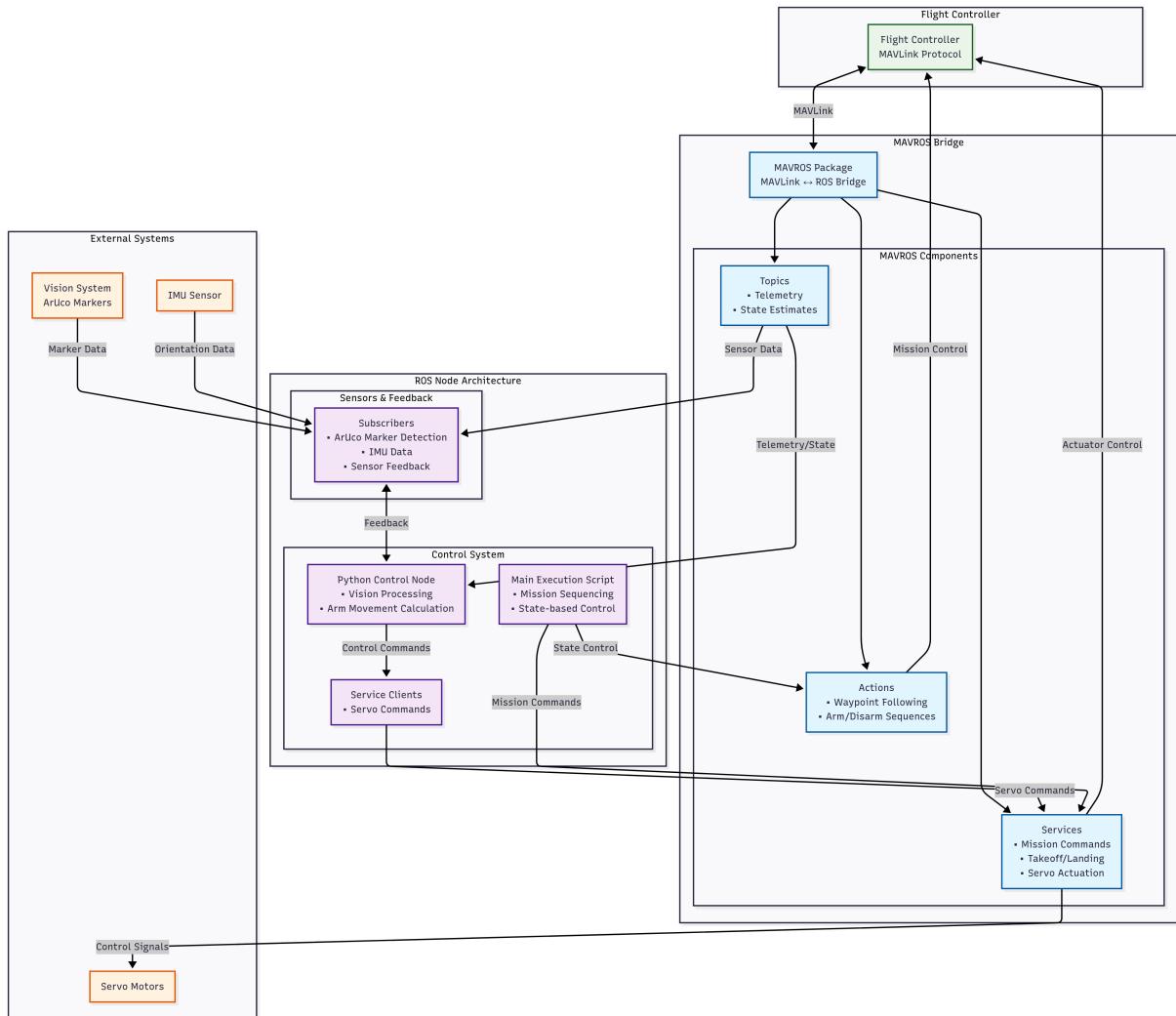


FIGURE 10.2: MAVROS communication architecture

- *Pros:* Quick to implement, worked well during manual ground tests.
- *Cons:* Interfered with flight control inputs when overriding channels used for throttle, pitch, or yaw. Not suitable for autonomous missions.

2. MAVLink Servo Command (MAV_CMD_DO_SET_SERVO)

To decouple servo control from flight channels, we adopted MAV_CMD_DO_SET_SERVO, a dedicated MAVLink command (183) that sets a specific PWM value on a servo output.

- *Advantages of RCOverride*

Feature	MAV_CMD_DO_SET_SERVO	RC_CHANNELS_OVERRIDE
Direct servo control	Yes	Indirect via RC
Interferes with flight control	No	Yes
Script-safe and stable	Yes	Risky in mission
Ideal use-case	Grippers, payloads	Manual input emulation

TABLE 10.1: Comparison between MAV_CMD_DO_SET_SERVO and RC_CHANNELS_OVERRIDE

- *Implementation Example*

```
rosservice call /mavros/cmd/command_long \
    "broadcast: false
    command: 183
    confirmation: 0
    param1: 9          # Servo number (AUX1)
    param2: 1500        # PWM value
    param3-7: 0"
```

This command remains effective until explicitly changed and does not affect the vehicle's attitude or flight control.

Important Note: For this method to work, the respective SERVOx_FUNCTION parameter must be set to -1 (manual override mode).

10.5 On-ground Communication Framework

For monitoring, debugging, and visualization during flight testing, a Wi-Fi-based communication setup was implemented:

- A local Wi-Fi network was established between the Raspberry Pi onboard and a ground laptop.
- This allowed SSH access to the RPi, ROS topic monitoring (*rqt_graph*, *rqt_image_view*), and camera streaming.
- The Wi-Fi module initially drew power from the drone's main battery, which led to power instability issues.

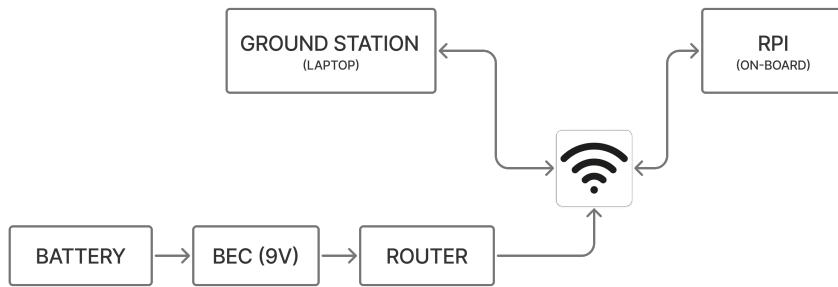


FIGURE 10.3: Ground Communication Framework

To resolve this, we introduced a dedicated buck converter to power the Wi-Fi module independently. This improved communication stability and allowed for continuous, real-time ground monitoring without affecting flight performance.

10.6 Parameters for Ardupilot Configuration

Several key parameters were configured in ArduPilot to enable and optimize autonomous operation, CAN-based peripherals, and servo control. A selection of important parameters includes:

These parameters were essential for integrating GPS, LED indicators, and peripheral sensors, and for maintaining robust communication and control flow across the system.

Parameter	Value	Description
MOT_THST_HOVER_LEARN	2	Enables hover thrust learning and stores to EEPROM
CAN_D1_PROTOCOL	1	Enables DRONECAN protocol on CAN1
CAN_D2_PROTOCOL	1	Enables DRONECAN protocol on CAN2
CAN_P1_DRIVER	1	Activates CAN1 bus driver
CAN_P2_DRIVER	1	Activates CAN2 bus driver
GPS_TYPE	9	Sets GPS communication to DRONECAN
NTF_LED_TYPES	231	LED output behavior set to DRONECAN-compatible mode

TABLE 10.2: Important parameter values for DRONECAN and hover thrust configuration

Part III

VISION SYSTEM AND MANIPULATION

Chapter 11

Camera and Processing

11.1 On-board Computation Resources

The initial onboard processor employed for our system was the Raspberry Pi 3B, a quad-core processor with 1 GB RAM. Using this, we implemented a basic ArUco marker detection pipeline. However, the processing throughput was limited to approximately 5.6 frames per second (fps). Upon integrating additional image processing tasks such as pose estimation and early-stage deblurring techniques, the throughput further dropped to 2.8 fps, rendering the system inefficient for real-time applications.

To address these limitations, we upgraded to a Raspberry Pi 4B with 4 GB RAM. This significantly improved the computational bandwidth, enabling the execution of image sharpening, deblurring, marker detection, and MAVROS communication in parallel. With this setup:

- Without deblurring: 15 fps
- With deblurring: 11.2 fps

TABLE 11.1: Comparison of Frame Rates on RPI 3B and RPI 4B With and Without Deblurring

Device	RAM	Frame Rate (no deblurring)	Frame Rate (with deblurring)
RPI 3B	1 GB	5.6 fps	2.8 fps
RPI 4B	4 GB	15 fps	11.2 fps

11.2 Power Management

Initially, the Raspberry Pi was powered via a 5V BEC output from the Power Distribution Board (PDB), which in turn was connected to the drone's main battery. Although the BEC was rated for 2.5A, our system's demands were around 3A, which occasionally led to instability.

To resolve this, a dedicated 5V BEC was added in parallel, rated for 3A. However, during flight—when the motors drew high current—the voltage drop across the main battery's internal resistance caused the Raspberry Pi to power down intermittently.

Ultimately, we introduced a separate, smaller 2S LiPo battery solely to power the Raspberry Pi. This resolved voltage drop issues and stabilized onboard processing during flight operations.



FIGURE 11.1: Initial power management architectures: Could not meet the requirements

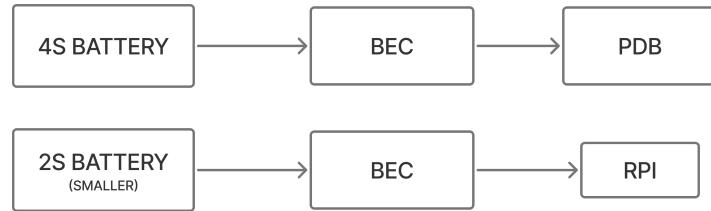


FIGURE 11.2: Modified Power Management with on-board processor

11.3 Camera and Image Processing

11.3.1 Raspberry Pi Camera

The Raspberry Pi camera used was a 2MP Sony sensor, functioning as a monocular camera. To enable 3D pose estimation from 2D image data, we calibrated the camera using OpenCV's `calibrateCamera` function with a fixed-size (15 cm x 15 cm) ArUco marker.

The **camera projection model** is

$$x_{image} = K[R||t]X_{world}$$

where,

- $X_{world} \in \mathbb{R}^3$ is the 3D point in the world (marker corners)
- $R \in SO(3), t \in \mathbb{R}^3$: extrinsic rotation and translation (pose)
- K : the intrinsic matrix

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

- f_x, f_y : focal lengths in pixel
- c_x, c_y : optical center

And, the **Distortion model** is

$$D = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]$$

where,

- k_1, k_2, k_3 : radial distortion
- p_1, p_2 : tangential distortion

The `camera_matrix` and `dist_coeffs` from the calibration file are passed to the pose estimation function. These parameters correct for lens distortion and map the 2D image points to 3D coordinates accurately. The pose estimation function solves the **Perspective-n-Point (PnP) problem**: Given the 3D coordinates of the marker corners (in the marker's own coordinate system) and their corresponding 2D image

points, along with the camera's intrinsic parameters, it computes the marker's position and orientation relative to the camera. Internally, this function solves the Perspective-n-Point (PnP) problem, where:

- Known: X_{world}, K, D
- Measured: x_{image}
- Estimated R, t such that

$$x_{image} = \text{project}(K, D, R, t, X_{world})$$

During operation, the video stream was processed as follows:

- Frames were converted to grayscale
- ArUco markers were detected
- The pose of the marker was estimated using `estimatePoseSingleMarkers`
- This yielded:
 - Rotation Vectors: Orientation of the marker
 - Translation Vectors: 3D position (X, Y, Z) of the marker in camera coordinates

This setup, though functional, was computationally expensive due to the extra steps involved in monocular depth estimation, resulting in a frame rate of 3.4 fps. Additionally, image blur during drone motion significantly reduced detection reliability.



FIGURE 11.3: Marker Detection from Realsense Camera feed

11.3.2 Intel Realsense Depth Camera

To overcome the limitations of the Raspberry Pi camera, we upgraded to an Intel RealSense D435i depth camera, capable of capturing synchronized color and depth frames.

- The VISUAL_GUIDE class was developed to manage camera operations
- Streams were configured at 640x480 resolution, 30 fps
- Real-time depth scale and intrinsic parameters were used to extract true 3D coordinates of the marker directly

The frame processing pipeline included:

- Blind deconvolution and sharpening
- ArUco marker detection
- Depth extraction and 3D localization
- Variance filtering and threshold checks for robustness

This allowed stable and reliable pose estimation at around 12 fps, sufficient to integrate with control algorithms.

11.4 Camera Shake De-blurring

Camera shake is a pervasive problem due to platform vibrations, wind, and rapid maneuvers. This shake introduces complex, non-uniform blur into captured images, severely degrading the visibility of fine details and the sharpness of edges. For computer vision tasks such as ArUco marker detection—which rely on crisp, high-contrast edges and well-defined geometric patterns—camera shake can drastically reduce detection accuracy or even prevent detection entirely. Therefore, robust deblurring algorithms are essential preprocessing steps to restore image quality, enhance edge definition, and enable reliable marker detection in challenging, real-world conditions.

11.4.1 Multi-Scale Blind Deconvolution with Shock Filtering

The first approach to camera shake removal is based on a multi-scale blind deconvolution pipeline, enhanced with edge-preserving shock filtering. The key steps are:

1. Blind Kernel Estimation

The blur kernel is estimated directly from the input image using a simplified blind deconvolution approach, assuming a uniform blur across the image. The blurred image $B(x, y)$ is modeled as:

$$B(x, y) = I(x, y) * k(x, y) + n(x, y)$$

where:

- $I(x, y)$: latent sharp image (unknown)
- $k(x, y)$: blur kernel (unknown)
- $*$: convolution operator
- $n(x, y)$: additive noise

Blind deconvolution solves for both I and k using an optimization approach:

$$\min_{I, k} \|B - I * k\|^2 + \lambda R(I)$$

Here, $R(I)$ is a regularization term (e.g., total variation) promoting natural images, and λ controls the regularization weight.

A uniform blur is assumed, simplifying the problem by treating the kernel as spatially invariant. Edge features in the image are leveraged for better kernel estimation since blur tends to smooth high gradients.

2. Multi-Scale Processing

To improve robustness and computational efficiency, deblurring is performed across an image pyramid:

$$\{B_1, B_2, B_3\}, \quad B_3 = \text{original}, \quad B_1 = \text{lowest resolution}$$

At each scale s , a blur kernel k_s is estimated and the image I_s is deblurred. The estimated kernel is refined as the algorithm progresses to higher resolutions.

After deconvolution at all scales, the final image is obtained by weighted averaging:

$$I_{\text{final}} = \sum_s w_s \cdot I_s$$

where weights w_s reflect detail confidence or noise suppression at each scale.

3. Shock Filtering

To enhance edges post-deblurring, a shock filter is applied. This is based on the nonlinear PDE:

$$\frac{\partial I}{\partial t} = -\text{sign}(\Delta I) \cdot |\nabla I|$$

where:

- ΔI : Laplacian (second derivative)
- ∇I : gradient magnitude

The sign of the Laplacian determines whether the pixel is inside or outside an edge, sharpening the transition accordingly while preserving structure. This helps in improving the detectability of fine features like ArUco marker corners.

4. Parameter Optimization:

Multiple combinations of kernel size, iteration count, and scale are tested in parallel, and the best result is selected based on a combined sharpness and edge density metric.

Due to the sensitivity of blind deconvolution, several parameter combinations are tested:

- Kernel sizes: 5×5 , 9×9 , 15×15
- Iteration counts
- Image scales

Each candidate output I_i is evaluated using:

- **Sharpness metric** (e.g., variance of Laplacian)
- **Edge density** (number of high gradient pixels)

A scoring function is defined as:

$$\text{Score}(I_i) = \alpha \cdot \text{Sharpness}(I_i) + \beta \cdot \text{EdgeDensity}(I_i)$$

The image with the highest score is selected as the final output.

Problems and Limitations:

- *Uniform Kernel Assumption*: The method assumes a single blur kernel for the entire image, which is often unrealistic for drone footage where blur can vary spatially due to complex vibrations.
- *Limited Adaptivity*: The kernel estimation is basic and does not account for local variations or frequency characteristics of drone-induced blur.
- *Noise Amplification*: While shock filtering enhances edges, it can also amplify noise, especially in low-contrast regions.
- *Detection Gaps*: In practice, some images still failed to yield reliable ArUco marker detection, especially under severe or non-uniform blur.



FIGURE 11.4: No detection with basic de-blurring tools: Test 1

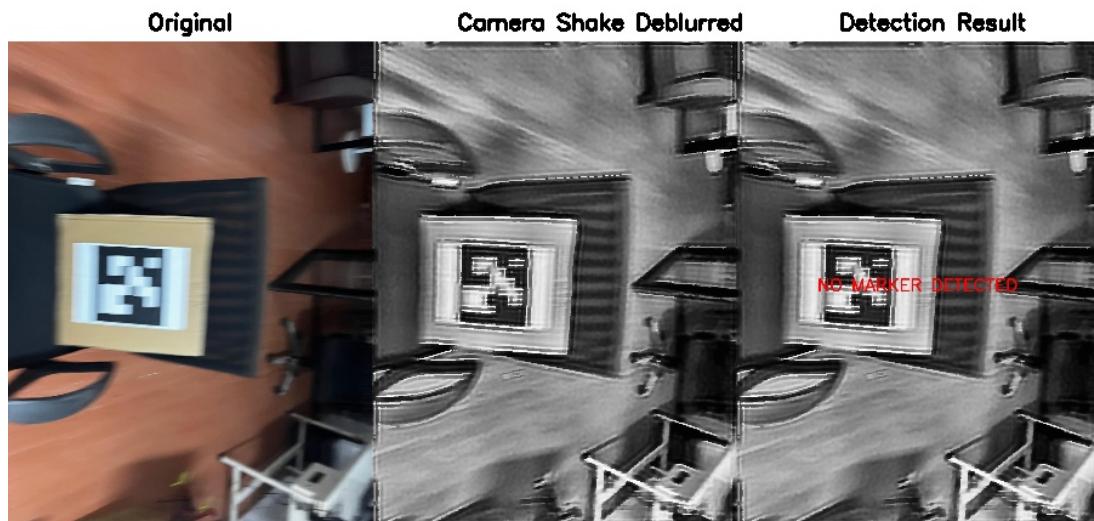


FIGURE 11.5: No detection with basic de-blurring tools: Test 2

11.4.2 Drone-Specific Piecewise Deblurring with Adaptive Regularization

To address the shortcomings of the first method, a more advanced, drone-specific deblurring pipeline was developed. The improvements include:

1. Piecewise Kernel Estimation

Instead of assuming a uniform blur across the image, the image I is divided into overlapping patches $\{I_i\}$, such that:

$$I = \bigcup_i I_i \quad \text{with overlap}$$

For each patch I_i , a local blur kernel K_i is estimated by solving the inverse problem:

$$I_i = K_i * S_i + n_i$$

where S_i is the latent sharp image and n_i is additive noise. This models **spatially varying blur**, a common artifact from high-frequency drone vibrations.

2. Frequency-Constrained Kernel Estimation

Typical drone vibrations occur at dominant frequency bands. Let $\mathcal{F}(K_i)$ denote the Fourier Transform of the kernel K_i . A regularization term is added to suppress energy outside expected frequency bands ω_d :

$$\min_{K_i} \|I_i - K_i * S_i\|^2 + \lambda \cdot \Phi(\mathcal{F}(K_i))$$

where $\Phi(\cdot)$ penalizes high-frequency components inconsistent with known vibration harmonics.

3. Vibration-Specific Preprocessing

Preprocessing enhances edges and suppresses low-frequency content irrelevant to deblurring:

$$I_{hp} = I - G_\sigma * I$$

where G_σ is a Gaussian filter. Additionally, edge enhancement (e.g., Sobel, Laplacian) is applied to improve kernel estimation and deblurring quality.

4. **Enhanced Richardson-Lucy Deconvolution** The latent image S is recovered using a modified Richardson–Lucy (RL) algorithm. The classical RL iteration is:

$$S^{(t+1)} = S^{(t)} \cdot \left(\frac{I}{K * S^{(t)}} * \tilde{K} \right)$$

where \tilde{K} is the flipped blur kernel. To reduce noise amplification, Total Variation (TV) regularization is introduced:

$$\min_S \{ \|I - K * S\|^2 + \beta \cdot \text{TV}(S) \}$$

with

$$\text{TV}(S) = \sum_{x,y} \sqrt{(\nabla_x S)^2 + (\nabla_y S)^2}$$

The regularization weight β is adaptively modulated based on local gradients, preserving edges while suppressing flat-region noise.

5. Multi-Scale and Multi-Parameter Optimization

The deblurring process is run over multiple configurations:

- Kernel sizes $k \in \{5, 9, 15, 21\}$
- Iterations $t \in \{10, 20, 30\}$
- Image scales $s \in \{0.5, 1.0, 2.0\}$

Each output is evaluated with a composite quality score:

$$Q = \alpha_1 \cdot \text{Sharpness} + \alpha_2 \cdot \text{Edge Density} + \alpha_3 \cdot \text{Gradient Variance} + \alpha_4 \cdot \text{Marker Detection Confidence}$$

The configuration maximizing Q is selected.

6. Optimized ArUco Detection

Marker detection parameters (e.g., thresholding constant, corner refinement method) are tuned for the deblurred images, improving ArUco marker detection success under degraded capture conditions.

Advantages Over Method 1:

- *Handles Non-Uniform Blur:* By estimating kernels regionally and incorporating drone vibration characteristics, the method can effectively deblur images with complex, spatially varying blur.

- *Better Noise Control*: Adaptive TV regularization reduces noise amplification, a common problem in standard deconvolution.
- *Physically Informed*: Frequency constraints ensure that estimated kernels are consistent with real drone motion, improving deblurring fidelity.
- *Superior Detection Performance*: The combination of better deblurring and optimized detection parameters leads to higher ArUco marker detection rates, even in challenging conditions.



FIGURE 11.6: Marker detected with Advanced de-blurring tools: Test 1

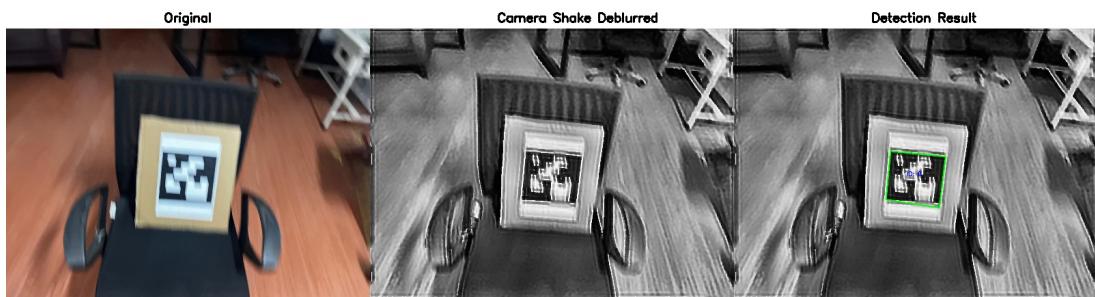


FIGURE 11.7: Marker detected with Advanced de-blurring tools: Test 2

11.5 Magnetic Interference (HERE3 GPS and RPI)

During the integration of the HERE3+ GPS module with the onboard Raspberry Pi (RPI), significant magnetic interference was observed, affecting the stability and reliability of sensor readings and orientation data. The interference was primarily attributed to the electromagnetic emissions from the RPI's high-speed data lines and processing circuitry, which were in close proximity to the GPS wiring and compass module.

Several factors contributed to this interference:

- *Electromagnetic Emissions from RPI:* The Raspberry Pi, particularly during high CPU and I/O activity, generates electromagnetic fields that can affect nearby sensitive magnetometers such as the one embedded in the HERE3+ GPS module.
- *Unshielded Cables:* The GPS and data cables connecting the modules were initially unshielded, making them susceptible to picking up EMI (electromagnetic interference) from the surrounding electronics.
- *Close Physical Placement:* The physical proximity between the RPI and the HERE3 GPS, especially the routing of data cables near power lines or the Raspberry Pi's processor, further amplified the interference.

Effects on System Operations. The interference led to the following operational issues:

- **Compass Inconsistency:** The onboard magnetometer showed erratic heading values, leading to unreliable yaw estimation and affecting the accuracy of autonomous navigation.
- **EKF Warnings:** The flight controller's Extended Kalman Filter (EKF) reported intermittent compass variance errors, resulting in degraded position hold and inconsistent waypoint behavior.
- **Mission Instability:** During autonomous missions, these fluctuations occasionally caused mode failsafes or inaccurate turns, which required manual intervention.

Mitigation Measures To counteract the magnetic interference, aluminium foil shielding was employed. The GPS wire and data lines connected to the RPI were carefully wrapped with multiple layers of aluminium foil to serve as a makeshift Faraday shield. This solution proved effective in the following ways:

- **Reduction in EMI Coupling:** The aluminium foil acted as a barrier, significantly reducing the electromagnetic field penetration from the RPI to the GPS wiring.

- Stabilized Compass Readings: After shielding, the compass data exhibited improved consistency and accuracy, which in turn enhanced overall flight stability.
- Restored Navigation Reliability: The frequency of EKF compass errors was greatly reduced, allowing successful execution of autonomous missions without unexpected deviation or failsafe triggers.

Chapter 12

Vision-based Approach Algorithms

In this chapter, we discuss the development of two vision-based target acquisition strategies—local target approach and global target approach—used to guide the drone for precision-based manipulation tasks using ArUco markers. Each approach has been designed to address a specific set of challenges related to perception, accuracy, and flight stability in real-world aerial robotic systems. We also detail the safety contingencies implemented to mitigate unexpected behaviors during flight.

12.1 Local Target Approach

The local target approach leverages real-time, continuous ArUco marker detection to navigate the drone incrementally toward a detected marker using local frame data only. This approach relies on camera-based feedback rather than any global positioning estimate and is particularly useful in confined indoor environments or close-range tasks.

Methodology:

1. The camera captures video frames, and ArUco detection is performed every 10th frame to reduce computational load and increase throughput.
2. Upon detection, the algorithm compares the detected marker’s center against the center of the image frame to decide which direction the drone must move (left/right, up/down).
3. A step size is chosen based on the displacement of the marker’s center from the image center. Larger displacements result in a larger movement step (0.7 m), while smaller displacements use finer steps (0.15 m).

4. Once the marker is centrally aligned, the drone moves forward along the Z-axis (towards the marker) until it reaches a depth offset of 0.5 meters, to allow space for dexterous manipulation.

Advantages:

- Enables real-time correction and visual servoing.
- Effective for tracking moving markers, as the motion is computed relative to the drone's current camera frame.

Limitations:

- Susceptible to vibrations and noisy detections.
- Frame-skipping may cause lag in fast-paced maneuvers.
- Does not use sensor fusion; hence, cumulative error or drift is not corrected over time.

```

Every 10th frame:
Detect ArUco marker
If marker detected:
    Calculate offset from image center
    If offset > high threshold:
        step_size = large
    Else:
        step_size = small
    Determine direction of movement
    Move drone laterally using step_size
    If marker is centered:
        Move drone forward to target depth

```

12.2 Global Target Approach

To address the limitations of the local approach, a global target approach was developed. This strategy focuses on using ArUco detection only to identify the global location of the target marker, after which the drone is navigated using a sensor-fused localization



FIGURE 12.1: Local Approach, Marker in center, Approaching towards target

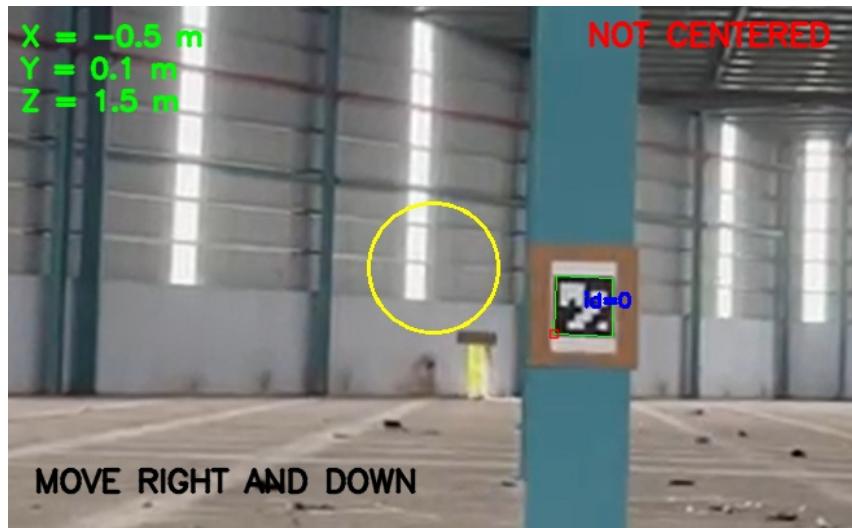


FIGURE 12.2: Local Approach, marker not centered

system based on IMU, GPS, and compass data passed through an Extended Kalman Filter (EKF).

Methodology:

1. The drone attempts ArUco marker detection from a stable hover.
2. Upon successful detection, the system verifies the marker's pose data to eliminate garbage values or detection errors using pre-defined thresholds.
3. The marker's position is transformed into a global reference frame.
4. This global position is then pushed as the next setpoint, and the drone autonomously navigates to it.

5. To reduce noise, the process is repeated three times and the average position is used as the final setpoint.

Aspect	Local Target Approach	Global Target Approach
Detection Frequency	Every 10th frame	Once per iteration
Noise Sensitivity	High	Low (with averaging and thresholds)
Uses Sensor Fusion	No	Yes (IMU + GPS + Compass + EKF)
Ability to Track Moving Target	Yes	No (best for stationary targets)
Vibration Resistance	Low	High
Mission Stability	Medium	High

TABLE 12.1: Comparison between Local and Global Target Approaches

The global approach is significantly more reliable in real-world conditions where vibration, environmental noise, and frame drops interfere with vision-based control. Although it cannot track dynamic targets in real time, it excels in stability and robustness, making it suitable for precise missions like docking or object delivery.

```

Initialize empty list of valid global poses
While less than 3 valid detections:
    Detect ArUco marker
    If marker is detected and pose is valid:
        Transform to global coordinates
        Store in list
    Compute average of all valid global poses
    Send averaged global setpoint to flight controller

```

12.3 Contingencies

To ensure flight and operational safety in a hardware-first testing environment, several contingency protocols were integrated:

- *Geometric Boundaries:* Virtual 3D boundaries were set in the drone's flight firmware to prevent crossing predefined spatial limits.



FIGURE 12.3: Global Approach, Going towards global position

- *Velocity and Jerk Limits:* Maximum linear and angular velocities were capped. Sudden high-magnitude command spikes were rejected.
- *Garbage Value Filtering:* Pose and orientation values from ArUco detection were verified against realistic limits (e.g., max depth, rotation angles).
- *Failsafe Return to Home (RTH):* Triggered in case of loss of visual target, GPS failure, or command timeout.
- *Emergency Land:* If an unrecoverable situation is detected (e.g., unstable altitude, multiple missed detection attempts), the drone executes a controlled emergency landing.
- *Kill Switch:* A manual override channel was configured on the transmitter to instantly disarm and shut down the system in case of critical failures.

Chapter 13

Camera Mounting & Vibration Isolation

With the addition of an onboard manipulator and the inherent dynamic nature of aerial flight, our camera system experienced significant motion-induced blur. Vibrations arose from multiple sources: motor-induced harmonics, frame resonances, air turbulence, and inertial shocks due to manipulator actuation. While software-level image stabilization and deblurring techniques were attempted, their effect was limited under real-time constraints and degraded frames. Hence, it became essential to implement mechanical vibration isolation for the cameras.

We iterated through three major design versions of the vibration damping system, with each step aimed at improving stability, frequency handling, and directional damping. These are described in the sections below.

13.1 Version1 : High frequency dampers

The first design implemented four lightweight high-frequency silicon dampers, with specifications:

- *Material*: Silicone (25° shore hardness)
- *Size*: Height 11.7 mm, Width 12 mm
- *Hole Diameter*: 3.5–4 mm
- *Mounting configuration*: Angled at 45° in stretched position toward the center of the mounting plate

These dampers were selected for their ability to attenuate high-frequency vibrations from the drone's motors. Their angled mounting helped in distributing damping forces across all three primary axes (X, Y, Z), attempting pseudo multi-axis damping through a simple structural geometry.

However, in practical testing:

- The amplitude of vibration in X and Y directions remained high.
- The dampers, though effective at isolating high-frequency components, were too soft to handle the combined camera and mount weight (97g).
- Excessive stretching and strain in the dampers introduced rebound oscillations, negating the damping benefits.

This version provided limited vibration suppression, specifically failing under low-frequency, large amplitude conditions introduced during manipulator actuation and flight transitions.



FIGURE 13.1: High frequency dampers attached on the drone

13.2 Version2 : Heavy Industrial dampers

To address the issues of weight handling and vertical oscillations, the next iteration used stiff industrial dampers, each with:

- *Size:* 21 mm × 17 mm
- *Inner Diameter:* 7 mm
- *Load Capacity:* 120–180 g per damper

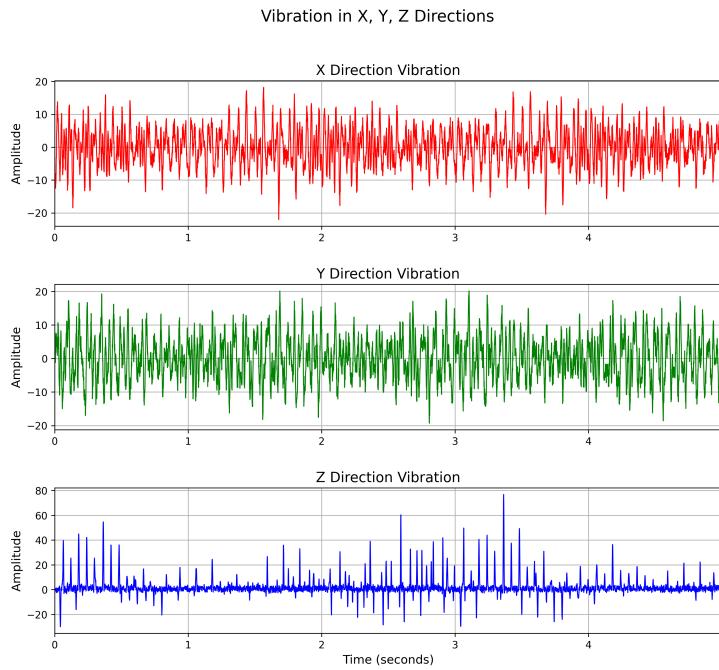


FIGURE 13.2: (version1) High Frequency Dampers : Frequency is reduced but amplitude is still high in X and Y axes

These dampers were mounted vertically and optimized for Z-direction (vertical) damping. The idea was to mitigate dominant vertical vibrations transferred from propeller thrust cycles.

Observations from testing:

- Improved performance in reducing Z-axis oscillations
- However, the stiff nature and vertical-only configuration failed to isolate lateral (X, Y) and torsional vibrations
- These dampers were less effective in filtering high-frequency noise, particularly those from frame harmonics and manipulator shocks

This version handled weight-induced compression better but sacrificed multi-directional and high-frequency isolation, leading to partial stabilization and residual image blur.

13.3 Version3 : DOF Vibration Isolation

Recognizing the need for full 6 degrees of freedom (DOF) isolation, we developed a custom mechanical structure using four pairs of high-frequency silicon dampers configured for multidirectional damping.

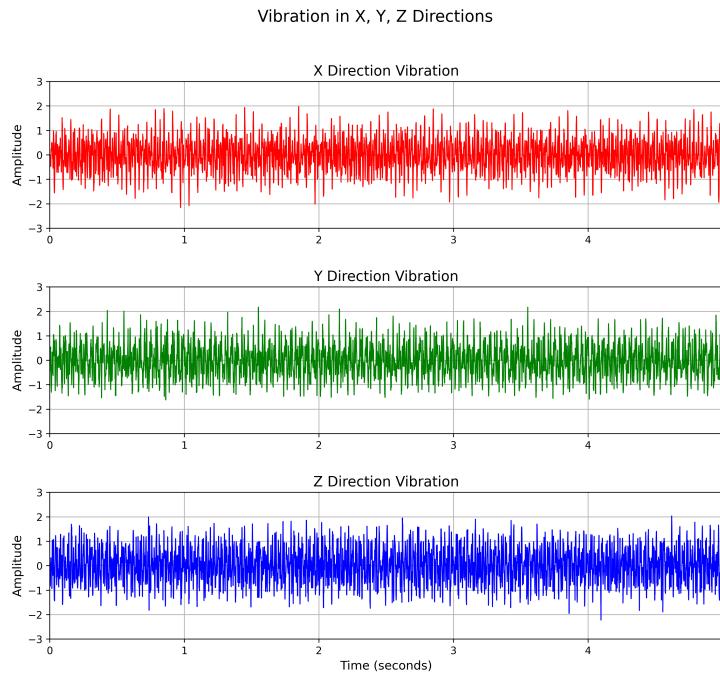


FIGURE 13.3: (version2) Industrial Dampers : Amplitude gets reduced but vibrations are high across all three axes

Design Concept:

- Each pair of dampers was placed symmetrically and angled inward at 45° , allowing the structure to respond to:
 - *Translational vibrations*: X, Y, and Z
 - *Rotational vibrations*: Pitch, Roll, and Yaw
- Undisturbed rest configuration: The dampers were not pre-stressed, maintaining symmetric elasticity and bidirectional response
- The structure suspended the camera mount in a floating frame supported by the 8 dampers (4 pairs), acting similarly to a gimbal suspension but passively compliant

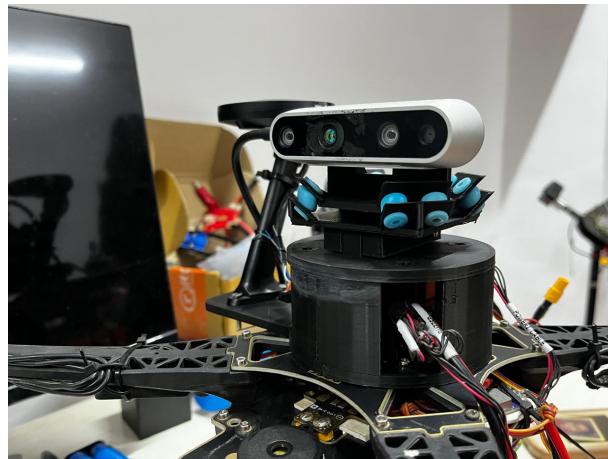


FIGURE 13.4: 6DOF Dampers Assembly on the quadcopter

How it Enables 6-DOF Vibration Isolation

Vibration Type	Damping Mechanism
X & Y (lateral)	Angled dampers provide cross-axial elasticity, absorbing sideways momentum
Z (vertical)	Paired dampers compress/stretch equally under vertical load, damping thrust pulses
Pitch & Roll	Differential compression in front vs. back or left vs. right pairs creates torque cancellation
Yaw (rotation about Z)	Twisting resistance from symmetric damper tension absorbs torsional pulses

This arrangement offers:

- Balanced load distribution
- High-frequency vibration filtering
- Low-frequency and large amplitude absorption
- All-axis dynamic response

Results:

- Significant improvement in video stabilization
- ArUco marker detection rates increased due to clearer frames
- Reduced computational load in downstream vision algorithms due to less image distortion
- Better robustness to manipulator movement and airframe shocks

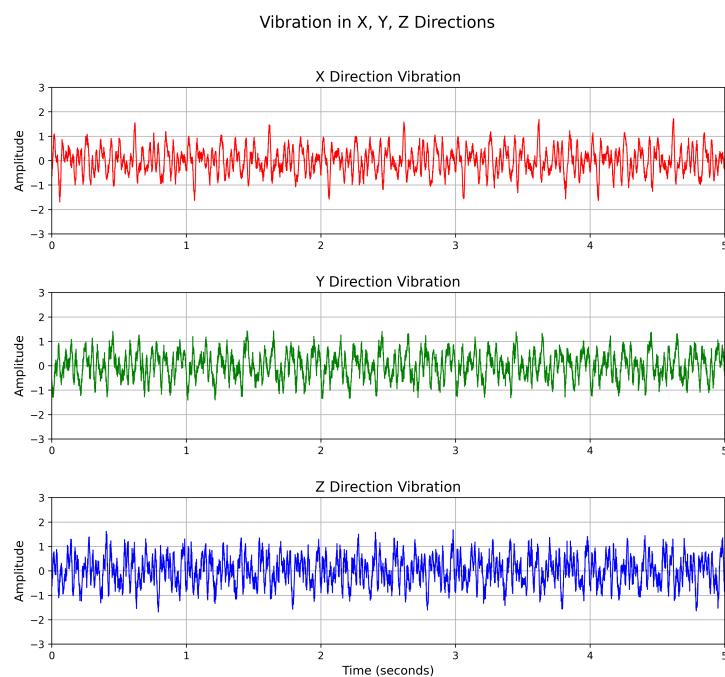


FIGURE 13.5: (version3) 6-DOF Vibration Reduction : Low amplitude and low frequency across all axes

Part IV

CONCLUSIONS AND FUTURE WORK

Chapter 14

Autonomous Approach : Field Test

After extensive system development, modular testing, and integration of vibration isolation, perception pipelines, and manipulator mechanisms, we transitioned to full-scale autonomous field trials. These tests were designed to validate the performance of the vision-based approach, autonomous navigation, and manipulator integration in a realistic operational scenario.

14.1 Arena Setup

To simulate a controlled field environment while still ensuring operator supervision and safety, we configured an indoor test arena with the following setup:

- An ArUco marker was mounted 1.5 meters above the ground on a vertical surface.
- The marker was placed 10 meters ahead of the drone's initial position, aligned on a horizontal plane.
- The drone's initial position was 2 meters above ground level at the starting point.

This setup ensured:

- A clean line of sight between the camera and the marker.
- Controlled lighting and obstruction-free detection.
- Sufficient depth range for the Intel RealSense D435i to deliver accurate positional data.

To allow detection, the drone was instructed to:

- Take off autonomously to 2 meters altitude.
- Move forward by 3 meters before initiating the marker detection and localization routine.
- Safety systems were enabled including:
 - Failsafe Return-To-Launch (RTL) in case of communication loss or script failure.
 - Emergency landing triggers based on unexpected drift, excessive tilt, or thrust saturation.
 - Predefined geofencing to limit positional drift beyond safe boundaries



FIGURE 14.1: Arena Setup

14.2 Autonomous approach Objectives

The high-level goal was to validate vision-based autonomous aerial manipulation in a real-world setting. The drone was tasked to:

1. Take off autonomously from the home position.
2. Fly toward a region where the marker becomes visible and begin tracking it.

3. Approach the marker using the local or global visual servoing logic.
4. Hover precisely at a virtual point 0.5 meters in front of the marker center.
5. Trigger the manipulator arm to grab an object (placed manually at this stage).
6. Return to launch (RTL) or base with the object securely held.

To evaluate the components in isolation and in combination, the process was broken down into three distinct test stages:

- *Stage 1*: Autonomous takeoff and manipulation test (static)
- *Stage 2*: Autonomous approach and return (without manipulator attached) *Stage 3*: Full autonomous approach, manipulation, and return (with manipulator attached)

14.3 Results

Stage 1: Autonomous Takeoff and Manipulation (Static Test)

In this preliminary test:

- The drone was statically mounted or tethered, and the manipulator was activated post-takeoff.
- This test confirmed that the drone could:
 - Maintain attitude stability during manipulator actuation.
 - Execute servo commands using MAV_CMD_DO_SET_SERVO reliably.
- *Outcome*: The system successfully handled the manipulator motion without interference to the onboard control or MAVROS node.

Stage 2: Autonomous Visual Approach and Return (Without Manipulator)

In this phase:

- The drone took off autonomously, moved ahead 3 meters, and began detecting the ArUco marker.

- Using local or global visual servoing, it approached the marker, aligned precisely, hovered at the designated 0.5-meter offset, and returned to base.
- The camera system performed stably, with vibrations sufficiently mitigated by the 6-DOF damping system.
- Detection rates were high, and sharp, well-deblurred frames enabled accurate pose estimation.

Outcome:

- The drone successfully performed vision-guided waypoint alignment, validating the real-time perception-action loop.
- No significant drift was observed, and the return trajectory followed the intended path.

Stage 3: Autonomous Manipulation with Onboard Gripper

In the final and most critical stage, the manipulator was attached and powered.

Challenges Observed:

- The combined weight and dynamic imbalance caused by the manipulator payload pushed the drone to operate at 90%
- This exceeded the safe operating margin (generally recommended $< 70\%$ for maneuverability and power headroom).
- As a result:
 - The drone struggled to stabilize in flight.
 - Any perturbation caused by manipulator motion led to loss of control authority.
 - Oscillations and instability rapidly grew, resulting in a crash and structural damage to the drone.

Technical Diagnosis:

- The flight controller's thrust margin was exhausted.
- Battery sag and power draw increased, affecting real-time computation and flight control loop frequency.
- IMU-based feedback loop was unable to accommodate inertial shifts caused by manipulator movement.

Chapter 15

Conclusion

This dissertation presents a comprehensive development of an autonomous aerial manipulation system that successfully integrates mechanical design, control systems, computer vision, and autonomous navigation capabilities into a functional platform capable of vision-guided manipulation tasks. The research demonstrates significant advances in several key areas of aerial robotics while identifying critical limitations that inform future development directions.

The mechanical design evolution through three major iterations successfully addressed fundamental challenges in aerial manipulation, progressing from asymmetric load distribution problems in the initial separate torque and balance link configuration to the improved dual-link torque distribution system that provided balanced load sharing and enhanced structural stability. The final below-base configuration effectively resolved propeller clearance issues while improving vibration isolation for the vision system, demonstrating the importance of iterative design approaches in complex robotic systems.

The vision system development achieved substantial improvements in real-time performance and robustness through advanced image processing techniques specifically designed for aerial platforms. The custom deblurring algorithms incorporating drone-specific vibration characteristics and frequency-constrained kernel estimation significantly improved ArUco marker detection reliability, while the 6-DOF vibration isolation system reduced camera shake to levels compatible with precision manipulation tasks.

System integration challenges were successfully addressed through comprehensive firmware development incorporating MAVROS communication frameworks, servo actuation control, and sensor fusion algorithms that enabled autonomous operation. The

implementation of hover thrust learning capabilities and RTK-GPS integration provided the precision and adaptability necessary for manipulation tasks, achieving centimeter-level positioning accuracy essential for contact-based operations.

However, experimental validation revealed critical limitations in the current quadcopter configuration when operating with the full manipulator payload. The system's operation at 90% thrust capacity created insufficient control authority margins, leading to instability during manipulator actuation and ultimately resulting in system failure during field testing. This finding highlights the fundamental importance of adequate thrust-to-weight ratios in aerial manipulation systems and identifies the primary limitation constraining current system capabilities.

The research contributes significant advances to the field of aerial manipulation through novel approaches to vibration isolation, specialized image processing for aerial platforms, and integrated hardware-software solutions for autonomous manipulation. The developed system provides a solid foundation for future aerial manipulation research while clearly identifying the critical areas requiring further development to achieve robust, reliable autonomous operation.

Chapter 16

Future Work

Future development of the aerial manipulation system should focus on three primary areas: platform configuration enhancement, mechanical system improvements, and advanced control integration. These developments will address the fundamental limitations identified during experimental validation while expanding system capabilities for more demanding manipulation tasks.

Transition to Hexcopter Configuration

The most critical improvement required is the transition from the current quadcopter to a hexcopter configuration to address the thrust limitations that caused system instability during field testing. Hexcopters provide several fundamental advantages over quadcopters for aerial manipulation applications, including increased payload capacity, enhanced stability, and critical redundancy for safe operation.

The six-motor configuration of hexcopters provides approximately 50% additional thrust capacity compared to equivalent quadcopters, enabling operation at safer thrust margins while maintaining adequate control authority for manipulation tasks. This increased thrust capacity will allow the system to operate at 60-70% maximum thrust rather than the problematic 90% experienced with the current quadcopter configuration, providing essential control margins for stable manipulation operations.

Enhanced stability characteristics of hexcopters result from improved force distribution across six thrust vectors rather than four, providing superior resistance to external disturbances and manipulator-induced center of mass shifts. The additional thrust vectors enable more precise attitude control and better compensation for dynamic loads introduced by manipulator motion, critical factors for maintaining stable flight during manipulation tasks.

Redundancy benefits of hexcopter configurations include the ability to maintain controlled flight even with single motor failure, a critical safety feature for aerial manipulation systems operating in complex environments. This redundancy is particularly important for manipulation tasks where sudden payload changes or environmental interactions could stress individual motors beyond normal operating limits.

The hexcopter configuration also provides improved yaw control authority through better torque distribution, enabling more precise orientation control essential for manipulation tasks requiring specific approach angles or fine positioning adjustments. Implementation should utilize a symmetric hexcopter design with evenly spaced 60-degree motor separation to optimize control authority distribution and maintain balanced thrust characteristics.

Advanced Bearing Integration for Load Balancing

A critical mechanical improvement required is the integration of precision bearing systems at manipulator joints to provide superior load balancing and reduce stress concentrations that currently limit system performance. The current servo-based actuation system experiences significant stress concentrations and limited load handling capabilities that restrict manipulation capacity and reduce system reliability.

High-precision radial and thrust bearing integration at each manipulator joint will distribute loads more evenly across the mechanical structure, reducing stress concentrations and enabling higher payload manipulation capabilities. Ball bearings or angular contact bearings should be integrated at the base joint to handle the primary structural loads while allowing smooth rotation with minimal friction and backlash.

The bearing integration should incorporate dual-bearing configurations with preload adjustment capabilities to eliminate play and maintain precise positioning under varying load conditions. This approach will significantly improve manipulator precision and repeatability while reducing wear on servo actuators by isolating them from direct load path stresses through the bearing support structure.

Thrust bearing integration at the base joint is particularly critical for handling axial loads introduced by manipulation forces and payload weights. These bearings should be sized to handle maximum anticipated manipulation loads with appropriate safety factors while maintaining low friction characteristics essential for precise servo control.

Advanced bearing materials such as ceramic or hybrid ceramic-steel bearings should be considered for critical joints to provide superior durability, reduced weight, and enhanced performance under dynamic loading conditions. These materials offer improved fatigue

resistance and reduced maintenance requirements compared to traditional steel bearings, important factors for aerial manipulation systems requiring high reliability.

Load Balancing and Dynamic Compensation Systems

Future development should incorporate active load balancing systems to maintain optimal center of mass positioning during manipulation operations. These systems should include movable counterweight mechanisms that automatically adjust based on manipulator position and payload characteristics to maintain stable flight conditions throughout manipulation sequences.

Battery positioning systems should be implemented using linear actuators or servo-driven mechanisms to provide active center of mass adjustment during flight operations. This approach will enable dynamic compensation for manipulator-induced balance changes while maintaining optimal weight distribution for stable flight characteristics.

Advanced sensor integration including load cells at critical joints will provide real-time feedback on manipulation forces and enable predictive load balancing adjustments. This sensor data should be integrated with flight control systems to provide coordinated responses to manipulation-induced disturbances, maintaining stable flight throughout manipulation operations.

Enhanced Control System Integration

Future control system development should incorporate model predictive control approaches specifically designed for aerial manipulation systems to provide superior performance under dynamic loading conditions. These controllers should account for manipulator dynamics and their coupling with flight dynamics to provide coordinated control of both flight and manipulation subsystems.

Machine learning approaches for adaptive control should be investigated to enable the system to learn optimal control strategies for different manipulation tasks and environmental conditions. These approaches could provide automatic tuning of control parameters based on operational experience and improve system performance over time through accumulated operational data.

Advanced vision system integration should incorporate multiple camera systems providing 360-degree coverage and redundant sensing capabilities for improved manipulation accuracy and safety. Stereo vision systems should be considered to provide enhanced depth perception and improved manipulation precision compared to the current monocular and depth camera configuration.

The integration of force feedback control capabilities will enable more sophisticated manipulation strategies including compliant manipulation and force-controlled assembly

operations. This capability requires integration of force/torque sensors with advanced control algorithms capable of managing the coupling between manipulation forces and flight dynamics.

These comprehensive improvements will transform the current aerial manipulation system into a robust, reliable platform capable of complex autonomous manipulation tasks while maintaining the safety and operational flexibility essential for practical applications in diverse environments.

Bibliography

- [1] Abdullah Mohiuddin, Taha Tarek, Yahya Zweiri, and Dongming Gan, “A Survey of Single and Multi-UAV Aerial Manipulation,” *Unmanned Systems*, vol. 8, no. 2, pp. 119–147, Apr. 2020.
- [2] H. Bonyan Khamseh, F. Janabi-Sharifi, and A. Abdessameud, “Aerial manipulation—a literature survey,” *Robotics and Autonomous Systems*, vol. 107, pp. 221–235, Sep. 2018.
- [3] Survey on Aerial Manipulator: System, Modeling, and Control, *Cambridge University Press*, Oct. 2019.
- [4] S. Kim, S. Choi, and H. J. Kim, “Aerial Manipulation Using a Quadrotor with a Two-DOF Robotic Arm,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, 2013, pp. 4990–4995.
- [5] G. M. Hoffmann et al., “Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 3261–3266.
- [6] “The Design and Simulation for Two-Link Manipulators PD Robust Controller under Uncertain Upper Bound Disturbance,” *Scientific.Net*, 2013.
- [7] M. Tognon et al., “Control-Aware Motion Planning for Task-Constrained Aerial Manipulation,” *IEEE Robotics and Automation Magazine*, vol. 25, no. 4, pp. 24–35, 2018.
- [8] Y. Chao et al., “Modeling and Sliding Mode Control of a Fully-actuated Multirotor with Tilted Propellers,” *Control Engineering Practice*, vol. 78, pp. 73–82, 2018.
- [9] “Active Stabilization of Unmanned Aerial Vehicle Imaging Platform,” *PML Tech. Report*.
- [10] T. de la Torre et al., “Tendon-Driven Continuum Robots for Aerial Manipulation—A Survey of Fabrication Methods,” *MDPI Drones*, vol. 8, no. 2, 2024.

- [11] J. Smith et al., “A Motion Deblurring Network for Enhancing UAV Image Quality in Bridge Inspection,” *MDPI Drones*, 2023.
- [12] “Deblurring Images Using the Blind Deconvolution Algorithm,” *MathWorks Documentation*, 2025.
- [13] Suseong Kim, Seungwon Choi, and H. Jin Kim. Aerial manipulation using a quadrotor with a two DOF robotic arm In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4990–4995, November 2013.

Acknowledgements

I would like to express my deepest gratitude to all those who supported and guided me throughout the course of this project.

First and foremost, I sincerely thank my advisor, Prof. Sukumar Srikant, for their invaluable guidance, consistent encouragement, and technical insights. Their mentorship played a pivotal role in shaping the direction of this work and pushing me to explore ideas rigorously and creatively.

Special thanks to my peers and teammates for their collaboration and constructive feedback during countless debugging sessions, field tests, and brainstorming discussions.

A word of appreciation goes to my family for their unconditional support, especially during long hours of development and testing. Their patience and encouragement have been instrumental in completing this project.

Finally, I thank all the contributors of open-source software tools and libraries, including ArduPilot, MAVROS, OpenCV, and the ROS ecosystem, without which this work would not have been possible.

This project has been a deeply enriching experience, blending research, engineering, and field execution—and I remain grateful to all who made it possible.

Rudraksh Kuchiya