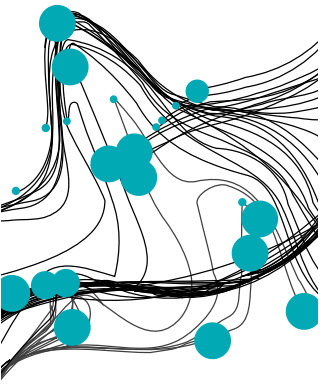
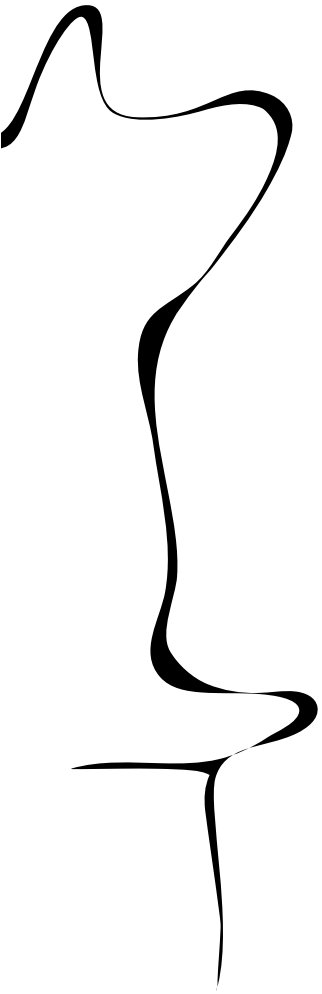


RAM ROBOTICS AND MECHATRONICS



ASSESSMENT OF THE TRAJECTORY CONTROL PERFORMANCE ON THE OMNIMORPH, AN OMNIDIRECTIONAL MORPHING MULTIROTOR UAV

A. (Ahmed) Tamer Salah Abdo Ali

MSC ASSIGNMENT

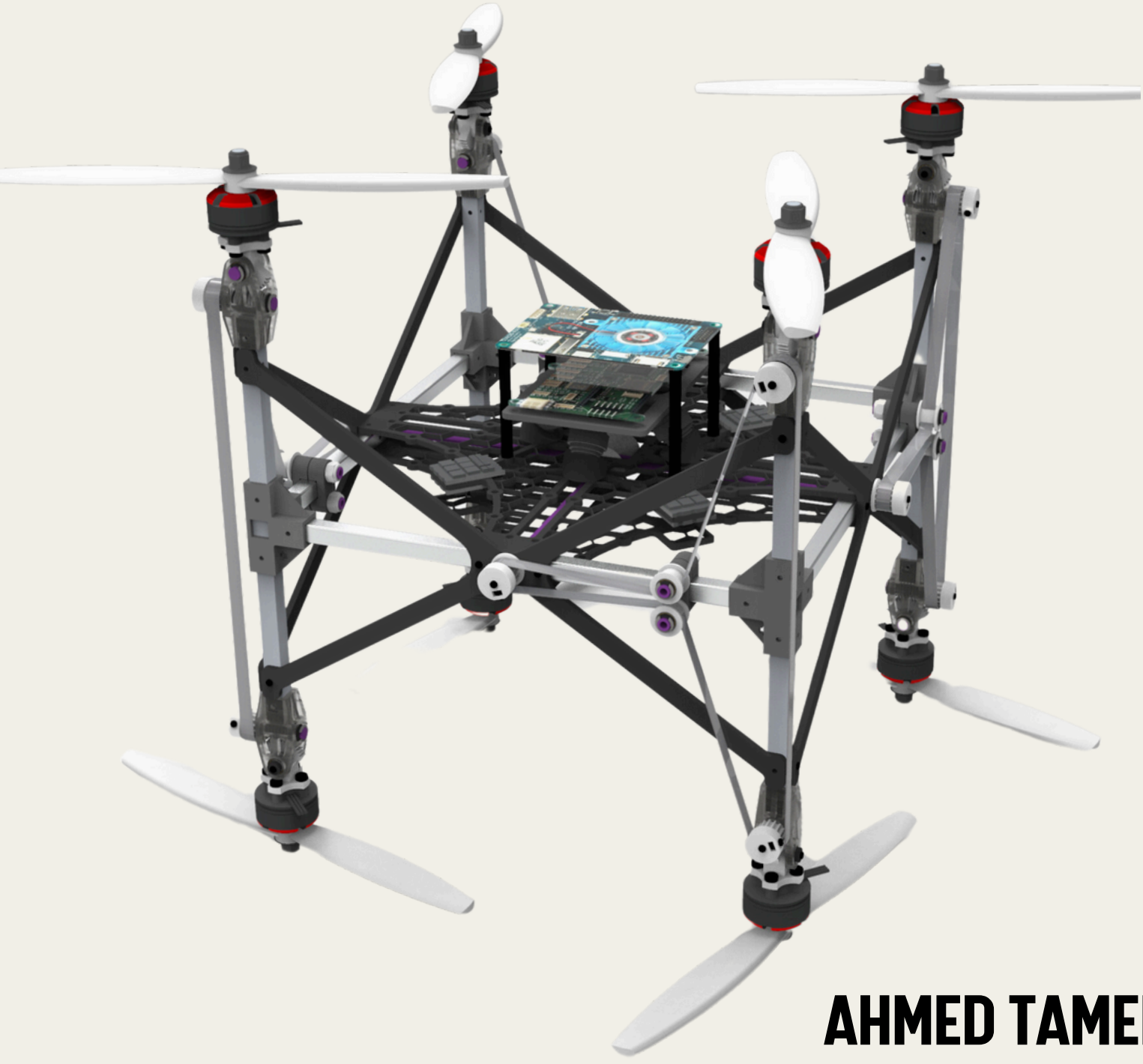
Committee:
prof. dr. ir. A. Franchi
C. Gabellieri, Ph.D
dr. F.C. Nex

August, 2024

054RaM2024
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

TRAJECTORY CONTROL PERFORMANCE OF THE OMNIMORPH



AHMED TAMER

Contents

List of Figures

Nomenclature

α	Tilting angle of the propellers.
α^*	Optimal tilting angle at the current time step.
$\ddot{q}_r$	Reference acceleration vector for the robot, including both translational and rotational accelerations.
ϵ_α	Maximum allowable rate of change for α.
sat	Saturation function to limit the values of errors and integrals.
μ	Gravity and Coriolis effects vector.
C, b	Constants defining input constraints.
e_p	Position error vector.
e_R	Orientation error.
e_ω	Angular velocity error.
$F(\alpha)$	Allocation matrix mapping control inputs to forces and torques.
J_R	Matrix mapping control inputs to body-fixed frame.
$K_{\omega 1}, K_{\omega 2}, K_{\omega 3}$	Gain matrices for the PID controller (rotational motion).
K_{p1}, K_{p2}, K_{p3}	Gain matrices for the PID controller (translational motion).
M	Inertia matrix of the system.
p	The position of O_B in F_W.
R	Attitude of O_B in $F_W := {}^W R_B$.
u_w	Vector of control inputs (rotor speeds).
u_w^*	Optimal rotor speeds at the current time step.
$W1, W2, W3$	Weight matrices for the cost function.

INTRODUCTION

1 Introduction

PRESENCE of drones in our daily lives remains more an expectation than a reality, despite the rapid advancement of unmanned aerial vehicles (UAVs) technology across various sectors. Other than the fact that regulatory constraints and societal challenges primarily drive this paradox (Drone Industry Insights, 2023), there are also technological challenges that hinder the widespread application of multi-rotor drones. The most significant is energy efficiency, which remains a critical hurdle in extending drone operational capacities (Townsend et al., 2020). The main reason behind this energy efficiency challenge lies in the inherent trade-off between energy capacity and the weight of the battery. Addressing this trade-off dilemma, researchers and engineers are exploring innovative design solutions to mitigate the adverse impacts of decreased efficiency. A significant limiting factor is the endurance of standard underactuated UAVs—platforms that have fewer control inputs than degrees of freedom, meaning they cannot independently control all axes of motion—where energy efficiency directly impacts operational capacity. Novel multi-directional thrust platforms introduce an even larger endurance problem. These novel multi-rotor UAVs face greater challenges due to their complex thrust mechanisms, which demand more energy.

In pursuit of maximizing energy efficiency, it is essential to consider multi-rotor UAV platforms that can operate across a spectrum of configurations, from underactuated modes, like quadrotors that conserve energy by having all thrust vectors against gravity, to omnidirectional modes that offer exceptional performance by being able to sustain the platform's weight in any orientation. Building on this understanding, **the authors of Aboudorra et al. (2024) developed the Omnimorph platform, shown in Fig.1.1**, which dynamically adjusts its configuration along this spectrum to meet the specific demands of each mission. This design allows the Omnimorph to optimize energy efficiency while maintaining the versatility needed for a wide range of tasks, as detailed in Aboudorra et al. (2024).

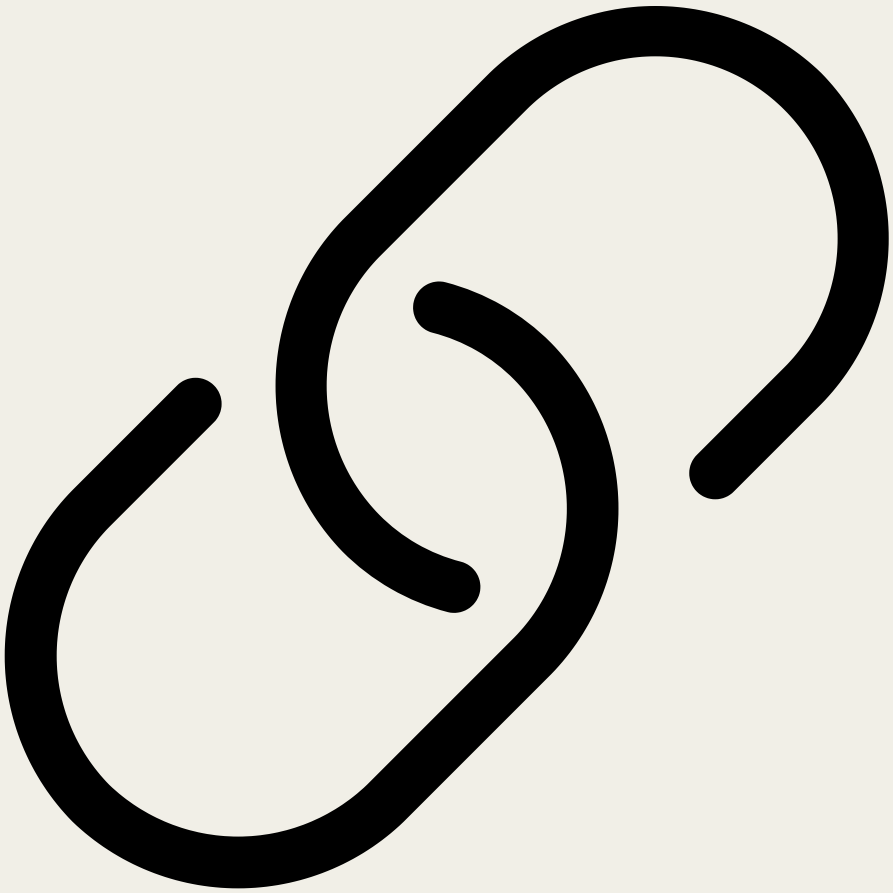
In its underactuated mode, the Omnimorph maximizes energy efficiency, making it ideal for tasks with minimal directional changes and extended operational time. When complex maneuvers require full omnidirectional capabilities, the platform adjusts to provide the necessary agility, at a relatively higher energy cost. This flexibility across the spectrum gives Omnimorph a significant advantage in energy efficiency compared to platforms limited to fully omnidirectional configurations, making it a versatile and efficient UAV solution.

This study primarily aims to bridge the gap between simulated and real-life experiments of the Omnimorph platform. Through rigorous testing and analysis, **it contributes to the broader goal of achieving stable flight for the Omnimorph—a milestone not yet realized.** This thesis represents a significant step forward in enabling Omnimorph to operate effectively in real-world conditions for the first time.

Figure 1.1: The OmniMorph platform. Left: OmniMorph in its most energy-efficient mode with all rotors pointing upwards. Right: OmniMorph in its omnidirectional mode, allowing it to hold its position in any orientation. Source: Aboudorra et al. (2024)

This thesis is structured to address the trajectory control performance of the Omnimorph platform, beginning with the related work (Chapter 2), which reviews adaptable and omnidirectional UAV designs, followed by a problem formulation (Chapter 3) outlining the research questions. The background (Chapter 4) covers the hardware and software design of the Omnimorph based on Aboudorra et al. (2024), while the contributions (Chapter 5) details key contributions of this thesis, including set of attainable forces and torques evaluation, an analysis of design trade-offs compared with the state of the art, a linear thrust mismatch model, and a dynamic weighting algorithm mitigating the effect of propeller performance degradation in the closed-Loop system. Simulations (Chapter 6) and experiments (Chapter 7) compare the Omnimorph performance in both simulated and real-world environments, and the discussion (Chapter 8) analyses these results. The conclusion (Chapter 9) provides insights into the findings and offers recommendations for future research.

RELATED WORK



2 Related Work

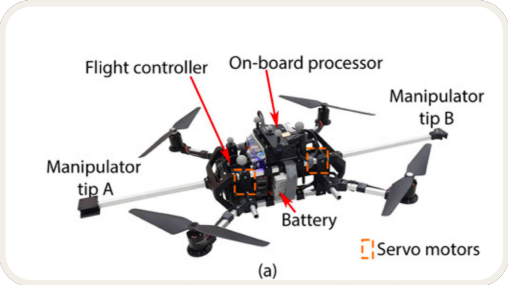
In the realm of high-performance multi-rotors UAVs, we will mainly define two primary categories of platforms based on their design capabilities. These categories can be classified by their unique design choices and functional attributes.

The first category comprises morphing and adaptable platforms. These UAVs are engineered to dynamically change or adapt their structural configuration while in flight. Such versatility allows them to perform a variety of tasks by adjusting their physical form to meet specific operational demands. Examples of these platforms are illustrated in Fig.2.1.

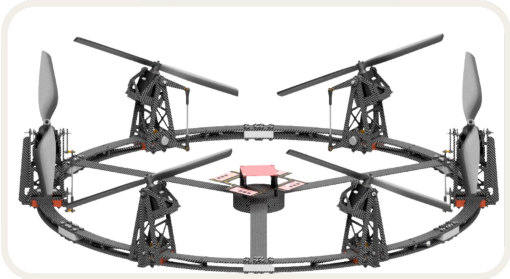
The second category includes omnidirectional platforms with fixed designs, which are shown in Fig.2.2. These UAVs are optimized for certain specific uses, maintaining their structural integrity without the need for in-flight morphological changes. An omnidirectional UAV is characterized by its ability to sustain its weight and maneuver in any direction and orientation, providing stable and robust performance.

By categorizing high-performance UAVs into these two classes—morphing/adaptable platforms and omnidirectional platforms with fixed designs—we can better understand their respective advantages and applications. The following sections will further explore examples of each category, highlighting their design choices and operational benefits.

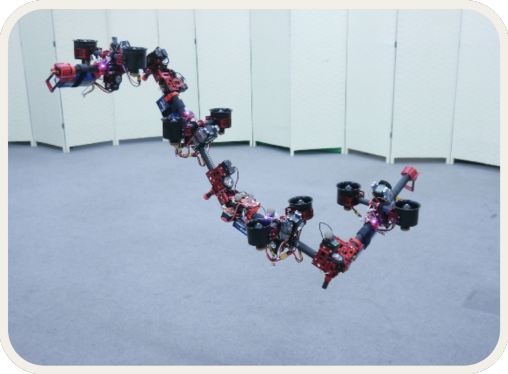
2.0.1 Morphing and Adaptable Platforms



(a) Tiltable Airframe Multirotor from Paul et al. (2023), ©2024 IEEE.



(b) FAST-Hex: actively tilting hexarotor from Ryll et al. (2022), ©2024 IEEE.



(c) DRAGON: Have the Ability of Multi-Degree-of-Freedom Aerial Transformation from Zhao et al. (2018), ©2024 IEEE.

(d) Prototype of Voliro from Kamel et al. (2018), ©2024 IEEE.

Figure 2.1: Types of Adaptable/Morphing platforms that can actively change structural configurations during flight.

Tiltable Airframe Multirotor

Paul et al. (2023) introduce an innovative tiltable airframe multirotor UAV, Fig.2.1a, designed to enhance aerial manipulation capabilities for tasks like inspection and maintenance in difficult-to-reach areas. Traditional underactuated platforms have mechanical limitations that restrict the manipulator's reach some points. Their proposed system reframes this problem by employing a tiltable airframe,instead of having tilting rotors, with auxiliary actuators, ensuring the rotors' axes remain upright, allowing stable hovering even when tilted. This design maintains vertical thrust direction, maximizing payload capacity and control consistency.

Fast-Hex

In their paper, Ryll et al. (2022) introduce FAST-Hex, a morphing hexarotor UAV shown in Fig.2.1b, that transitions seamlessly between under-actuated and fully-actuated configurations using a single additional motor to synchronously tilt all propellers. This design merges the efficiency of under-actuated configurations with the agility of fully-actuated systems. The FAST-Hex allows for independent control of 3D position and orientation, making it suitable for complex aerial tasks such as precise manipulation and navigation in cluttered environments. The control system is capable of receiving any reference pose in $R^3 \times SO(3)$ and prioritizes position tracking over orientation when constraints arise. Experimental results showcase the platform's ability to perform complex maneuvers, such as static hovering and dynamic trajectory tracking.

DRAGON

In their study, Zhao et al. (2018) introduce the DRAGON shown in Fig.2.1c, an articulated aerial robot designed for advanced aerial manipulation and grasping tasks. Unlike traditional aerial robots that attach manipulators to their frames, DRAGON integrates vectorable rotor units into each of its links, providing two degrees-of-freedom in thrust vectoring. This design enables stable aerial manipulation and precise grasping by leveraging the vectorable thrust for control rather than relying solely on joint torques, which are often limited in force. Experimental results demonstrate DRAGON's ability to perform complex manipulations and stable grasping in various scenarios, marking a significant advancement in the field of aerial robotics.

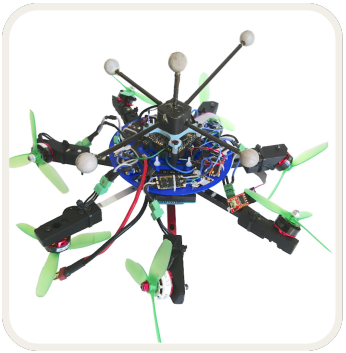
Voliro

In their paper, Kamel et al. (2018) introduce Voliro, an innovative omnidirectional hexacopter combining traditional multi-rotor benefits with enhanced maneuverability via tiltable rotors, displayed in Fig.2.1d. This design decouples position and orientation control, enabling complex tasks like uninterrupted camera movements and intricate inspections. Voliro,is currently a Swiss startup, features six tilting rotors to reduce counteracting forces and improve efficiency. Experimental results show Voliro's agility, such as transitions from horizontal to upside-down flight and stable vertical surface interactions. This research underscores the potential for UAVs with tiltable rotors in advanced maneuvers and physical interactions, expanding their application scope.

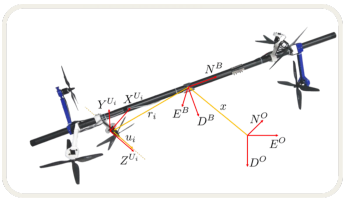
2.0.2 Omnidirectional Platforms

Omni-Plus-Seven

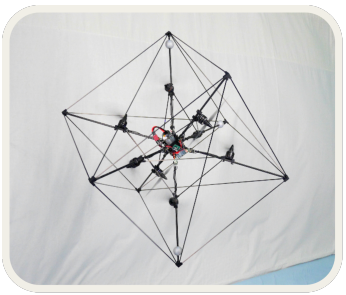
Hamandi et al. (2020) present the Omni-Plus-Seven (O7+) UAV, a novel omnidirectional aerial platform featuring seven uni-directional thrusters. This design aims to optimize UAV capabilities by balancing the placement and orientation of the propellers to achieve omnidirectional flight with a minimal number of thrusters. The O7+ employs a carefully optimized configuration where the propellers are strategically tilted to provide consistent thrust in all directions, allowing the UAV to maintain stable flight in various orientations, including horizontal, upside



(a) Omnidirectional platform with 7 uni-directional propellers from Hamandi et al. (2020), ©2024 IEEE.

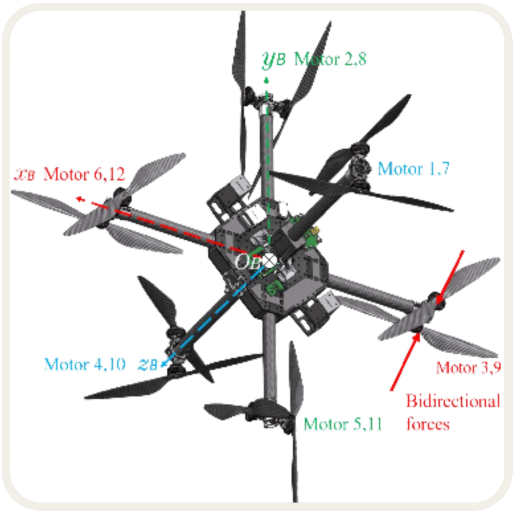


(b) First version of the ODAR hex-arotor from Park et al. (2016), ©2024 IEEE.



(c) Omni-directional octorotor from Brescianini and D'Andrea (2016), ©2024 Elsevier.

(d) CAD model of the Lynchpin prototype from Howard et al. (2023).



(e) Omni-Directional Aerial Vehicle with 6 co-axial rotors from Zhu et al. (2024), ©2024 IEEE.

Figure 2.2: Types of Omnidirectional Platforms that is design-optmised and cannot change structural configurations during flight.

down, and tilted angles. The platform's design minimizes weight while maximizing thrust efficiency, demonstrating significant improvements in both stability and control. Through simulations and real-world tests, the prototype showcased its ability to perform complex maneuvers, validating the design's effectiveness and potential applications in diverse aerial tasks.

ODAR

Park et al. (2016) introduce the ODAR (omni-directional aerial robot), a fully-actuated multirotor flying platform capable of generating arbitrary control wrenches in SE(3) using six bidirectional rotors. This design enables the ODAR to perform complex behaviors such as 360° photo/video shooting and resisting sideway gusts while maintaining its attitude and exerting downward pushing forces larger than its own weight. The results show that ODAR can achieve behaviors such as vertical flipping and exerting significant downward force, which are essential for applications like VR scene generation and aerial manipulation.

Omni-directional Octorotor

In their paper, Brescianini and D'Andrea (2016) present an omni-directional multirotor vehicle capable of independently controlling its thrust and torque in all six dimensions, thus allowing it to hover at any attitude and accelerate in any direction. The design leverages an octorotor configuration with fixed-pitch rotors capable of generating both positive and negative thrust, which are arranged to maximize the vehicle's agility and rotational invariance. Experimental results demonstrate the vehicle's capability to perform complex maneuvers, validating the effectiveness of the design and control scheme.

Lynchpin Prototype

Howard et al. (2023) present an innovative design for a six degrees of freedom (6DOF) unmanned aerial vehicle (UAV) named "The Lynchpin," which utilizes a novel geometry inspired by particle physics. The Lynchpin's design is based on a collapsed dodecahedron, providing unique maneuvering capabilities. The UAV features six bidirectional motors and propellers, allowing for omnidirectional flight and the ability to hover at arbitrary attitudes. A comprehensive flight mechanical analysis and control schemes were developed to manage the aircraft's thrust vectors. Initial flight tests demonstrated the aircraft's capabilities, although latency from thrust reversal remains a challenge. The symmetrical design of the Lynchpin enables it to connect with other units midair, forming complex structures for collaborative missions.

Omni-Directional Aerial Vehicle

In their paper, Zhu et al. (2024) introduce a novel design and control approach for a spatial symmetric fully actuated multirotor (FAM) capable of omnidirectional motion and high-frequency 6D interactions. The FAM features fixed-tilt bidirectional coaxial rotors. Extensive experiments validate the FAM's capability for decoupled 6D motion control and precise interaction tracking, demonstrating superior performance compared to other robust control methods.

THE WHAT?

3 Problem Formulation

The development of advanced aerial robotic systems necessitates a thorough understanding of the differences between theoretical models and real-world implementations. This thesis investigates the discrepancies between simulated and real-world performances of the Omnimorph. The Omnimorph design is inspired by the optimized design proposed in a previous study by Brescianini and D'Andrea (2016). However, to facilitate easier mechanical realization, several design modifications were made, which will be discussed more in the design section 5. **This leads to three primary research questions:**

1. **Effect of Design Changes on Feasible Force and Torque Sets:**
   The original design in Brescianini and D'Andrea (2016) presents an optimized configuration for the aerial platform, focusing on maximizing the feasible sets of forces and torques. Omnimorph, while inspired by this design, incorporates changes aimed at simplifying its mechanical construction. The first research question explores **how these modifications impact the platform's performance, specifically in terms of the sets of forces and torques it can feasibly produce.** This comparison will provide insights into the trade-offs between mechanical simplicity and functional capability.
2. **Mitigating the Effect of Propeller Performance Degradation in the Closed-Loop System:**
   In Aboudorra et al. (2024), a novel, optimization-based control law has been proposed for Omnimorph and tested successfully in simulations. However, it was discovered that in the real prototype, the propeller performance is highly dependent on the propellers' tilting angle, leading to uncertainties in the closed-loop trajectory-controller system. The third research question addresses **how to mitigate the effect of this uncertainty in the closed-loop trajectory-controller system.** Developing an accurate model of propeller performance degradation is complex, and this research seeks to explore strategies to compensate for or mitigate these uncertainties to ensure stable and accurate control of the real platform.
3. **Behavioral Differences Between Simulated and Real Platforms:**
   Although a novel, optimization-based control law for Omnimorph has been successfully tested in simulations in Aboudorra et al. (2024), real-world applications often reveal discrepancies due to unmodeled complexities. These include aerodynamic effects among propellers, shifts in the center of mass due to the addition of electronics and batteries, and other unaccounted-for physical factors. The third research question aims to quantify these discrepancies by **comparing the position and attitude errors between the controlled real platform and its simulated counterpart.** This investigation involves:
   - **Testing** the control on the real platform with **fixedly tilted propellers** in an omnidirectional configuration.

Addressing these research questions involves a systematic approach to testing and analyzing both the control strategies and the mechanical design of Omnimorph. The outcomes will not only highlight the practical challenges in transitioning from simulation to real-world deployment but also guide future improvements in the design and control of advanced aerial platforms.

BACKGROUND

4 Background

In this chapter, we will delve into the design of the OmniMorph platform that was designed by the authors of Aboudorra et al. (2024), a novel omnidirectional morphing multi-rotor UAV. While the current tested prototype of the platform has made significant strides, it still lacks an active tilting mechanism. We will discuss both the hardware and software aspects of the OmniMorph design. The hardware design focuses on the structural components, propellers, and the integration of a single servomotor for synchronized tilting of all propellers. The software design encompasses the control algorithms and simulation environments used to optimize the performance and stability of the Omnimorph.

Additionally, in this thesis, a user and developer instructions document was developed. This guide will be helpful for future developers and users, providing detailed instructions and best practices to ensure the efficient use and further development of the OmniMorph. This document will serve as a foundational tool for anyone looking to contribute to or utilize the Omnimorph.

In the sections that follow, we will first discuss the design aspects of the OmniMorph, then the actual hardware prototype, and finally the software including the controller.

4.1 Design Aspects

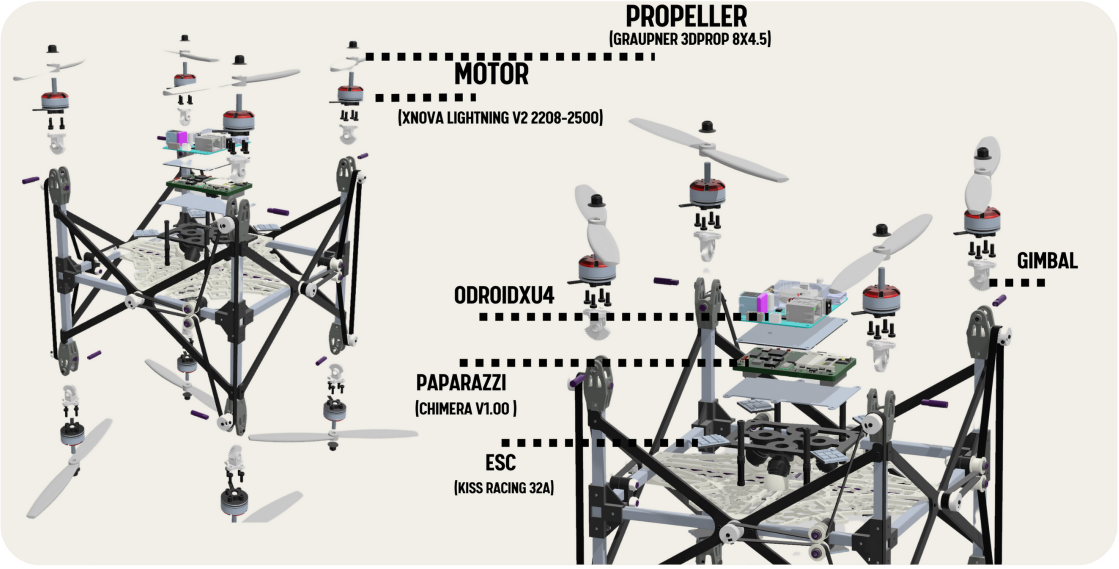


Figure 4.1: Exploded view of the omnimorph design

4.2 Hardware

Various components and key constants relevant to the OmniMorph are outlined in Table 4.1. It is important to note that the current prototype is not yet equipped with the active tilting mechanism described earlier. However, the rotors of the prototype can be manually adjusted to different angles using interchangeable 3D-printed parts, which are designed to simulate a fixed tilt angle.

Parameter		Value
Omnimorph Total mass (m_{om})		1.595kg
	Actuators:	0.367kg
	Battery:	0.368kg
	Frame:	0.320kg
	Avionics:	0.308kg
	Others (screws, landers, etc.):	0.232kg
Distance of i^{th} **rotor from center** ($||p_i||$)		0.2165m
Minimum (absolute) rotor thrust (f_{min})		0.15N
Maximum (absolute) rotor thrust (f_{max})		6.25N
smallest tilting angle (α_{min})		0°
Largest tilting angle (α_{max})		60°

Table 4.1: Parameters of modelled aerial manipulator

4.2.1 Components

As illustrated in Fig.4.1, The main Components of the Omnimorph are going to be listed, except that in the testing platform shown in Fig.7.1a there is no a tilting mechanism and teensy microcontrollers are used as a low level controllers to control the motors.

Frame

The frame is made of square aluminum hollow bars with 3D-printed corners. At the center, it has layered compartments housing the Odroid on top, the Paparazzi in the middle, the power distributor below, and the battery at the bottom.

Propellers

The two blade Graupner 3Dprop 8x4.5 bidirectional nylon propellers[1] are used for each of the eight motors.

Motors

Xnova Lightning V2 2208-2500 [2] which is has a KV of 2500KV are used for the eight motors.

Electronic Speed Controllers

This project uses Kiss Racing 32A ESCs[3]. Each of the 8 ESCs, corresponding to the octo-rotor configuration, translates control inputs from the motor controller into three-phase current outputs for the motors.

Motor Controllers

Two Teensy microcontrollers are used to interface with the 8 motors. They send the commanded velocities to the ESCs and read back the motor data, ensuring proper communication and control of the motors

Flight Controller

The prototype uses the Paparazzi Chimera V1.00 flight controller. It gathers data from the IMU such as the acceleration, angular velocity,etc. relaying important information back to the higher level controller.

[1] https://www.getfpv.com
[2] https://www.xnovamotors.biz
[3] https://droneshop.nl

High-level Computer

The single-board computer used for this platform is an Odroid XU4 running Ubuntu 20.04 for Odroid. The Odroid acts as the higher-level controller, handling communication with the PC, relaying input information to the flight controller, saving data, and potentially implementing more complex control algorithms in the future. However, during most of the testing phase, commands were instead sent from a stationary computer using Simulink via a 5 m cable. Once the controller is proven to be stable, it should be implemented on the Odroid.

4.3 Software

4.3.1 System Architecture

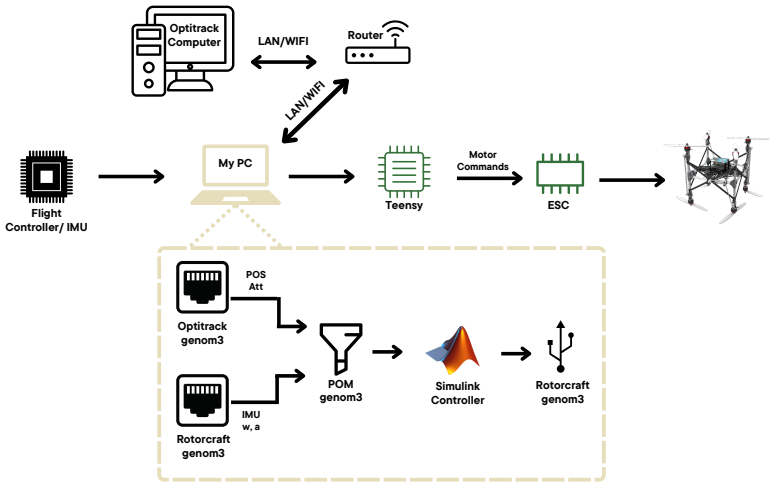


Figure 4.2: System architecture for controlling the Omnimorph.

The system architecture for controlling the Omnimorph integrates various hardware and software components. First, the Optitrack Computer, which captures position and attitude (p, R) data of the UAV using the Optitrack system. This data is then shared with other components in the system via a local network (LAN) or wireless network (WIFI) facilitated by a router.

The flight controller and inertial measurement unit (IMU) are onboard the UAV. The flight controller provides essential IMU data, including angular velocity (ω) and acceleration (a) in the body frame. This data is then communicated to a central PC Whether onboard or offboard, which coordinates data from various sources and runs the main control algorithms. The PC communicates with both the Optitrack system and the flight controller.

The Simulink Controller in this report, processes data from the IMU and Optitrack systems using genom3 state estimator, executes control algorithms, and sends commands to the UAV via the Rotorcraft genom3 interface. The Rotorcraft genom3 interface manages communication between the PC (running the Simulink Controller) and the UAV's teensys, ensuring that the control commands are correctly transmitted to the UAV.

4.3.2 Controller

This section summarises the control law done in Aboudorra et al. (2024), that allows Omni-Morph to follow a desired 6D trajectory switching between the underactuated and the fully actuated configurations to account for the minimization of trajectory tracking error and the minimization of the input u_w.

Given a Desired Trajectory = $(p^d, \dot{p}^d, \ddot{p}^d, R^d, \omega^d, \dot{\omega}^d)$ for the UAV, indicated through the superscript $(\cdot)^d$, then compute the reference acceleration of the robot, call it $\ddot{q}_r$, using a PD feedback control plus a feedforward term. The following desired input acceleration is obtained:

$$\ddot{q}_r = \begin{bmatrix} \ddot{p}^d + K_{p1}\dot{e}_p + K_{p2}e_p \\ \dot{\omega}^d + K_{\omega 1}e_\omega + K_{\omega 2}e_R \end{bmatrix} \tag{4.1}$$

where $e_R = \frac{1}{2}(R^T R^d - (R^d)^T R)^\vee$, with $\vee$ from $\mathfrak{so}(3)$ to $\mathbb{R}^3$ being the inverse of the hat map, $e_p = p^d - p$, and $e_\omega = \omega^d - \omega$.

Then an inner control loop was designed to track $\ddot{q}_r$ thanks to suitable inputs α^* and u_w^*, chosen as the solution to:

$$\min_{\alpha, u_w, \ddot{q}} J_1 + J_2 + J_3 \quad \text{subject to:} \begin{cases} M\ddot{q} = \mu + J_R F(\alpha) u_w, \\ Cu_w < b, \\ -\epsilon_\alpha \le \alpha - \alpha^*_{k-1} \le \epsilon_\alpha \end{cases} \tag{4.2}$$

where the cost function is composed of three terms: $J_1 = \|u_w\|^2_{W1}$ to minimize the norm of the input, $J_2 = \|\ddot{q}_r - \ddot{q}\|^2_{W2}$ to ensure tracking of the desired trajectory, and $J_3 = \|u_w - u^*_{w,k-1}\|^2_{W3}$ to minimize the propeller spinning accelerations. The quantities α^*_{k-1} and $u^*_{w,k-1}$ are the solution of the optimization problem at the previous time step. $\|\cdot\|_{Wi}$ is the 2-norm weighted by the positive definite weight matrix Wi.

The equality constraint is the system's dynamics, where $M = \begin{bmatrix} mI_3 & 0_3 \\ 0_3 & J \end{bmatrix}$, $\mu = \begin{bmatrix} -mge_3 \\ -\omega \times J\omega \end{bmatrix}$, and $J_R = \begin{bmatrix} R & 0_3 \\ 0_3 & I_3 \end{bmatrix}$. The second constraint is the input constraint, where C and b are properly defined constant quantities. The last constraint is on the rate of change of α. The maximum rate of change ϵ_α is here defined as symmetric but in general they may also be non-symmetric.

In order to solve this problem, which is not a QP problem because the first constraint is nonlinear in the optimization variables, The paper of Aboudorra et al. (2024) proceeded as follows. Consider the following problem for a fixed value of α, indicated as $\bar{\alpha}$:

$$\min_{u_w, \ddot{q}} J_1 + J_2 + J_3 \text{ subject to: } \begin{cases} M\ddot{q} = \mu + J_R F(\bar{\alpha}) u_w, \\ Cu_w < b \end{cases} \tag{4.3}$$

At each time step, the solutions α^* and u_w^* that correspond to the lowest value of the cost function are selected. A schematic representation of the control scheme of Aboudorra et al. (2024) is shown in Fig.4.3

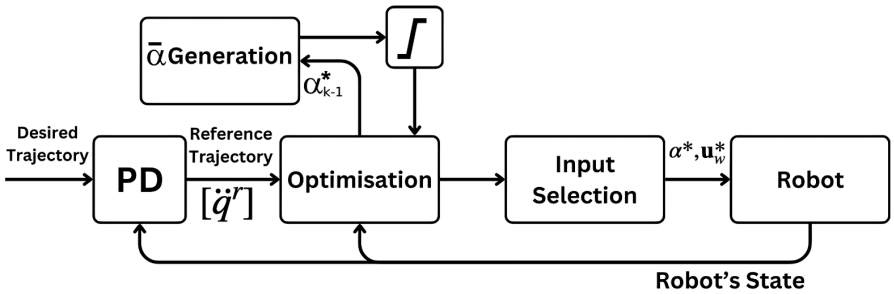


Figure 4.3: Block diagram of the proposed control method for the Omnimorph in Aboudorra et al. (2024).

In the real-life implementation for this thesis, the hardware for the actively tilting mechanism is not ready, hence the system will not require the α generation block, and the control focus will be solely on minimizing u^*. Additionally, a PID controller will be used instead of a PD controller, as it is more suitable for real experiments due to its ability to eliminate steady-state error and improve system stability in the presence of disturbances and noise. Moreover, the acceleration feedforward term was omitted to simplify the control implementation and because the feedforward term can be sensitive to modeling inaccuracies and disturbances, which are more prevalent in real-world scenarios.

The control law for the real-life implementation involves computing the reference acceleration of the robot, denoted as $\ddot{q}_r$, using a PID feedback control. The desired input acceleration is obtained as follows:

$$\ddot{q}_r = \begin{bmatrix} K_{p1}\dot{e}_p + K_{p2}\text{sat}(e_p) + K_{p3}\int_{-\infty}^{\infty}\text{sat}(e_p)\,dt \\ K_{\omega 1}e_\omega + K_{\omega 2}\text{sat}(e_R) + K_{\omega 3}\int_{-\infty}^{\infty}\text{sat}(e_R)\,dt \end{bmatrix} \tag{4.4}$$

where the terms $\text{sat}(e_p)$ and $\text{sat}(e_R)$ apply saturation functions to the errors, and the additional terms $K_{p3}\int_{-\infty}^{\infty}\text{sat}(e_p)\,dt$ and $K_{\omega 3}\int_{-\infty}^{\infty}\text{sat}(e_R)\,dt$ represent the integral components of the PID controller with saturation, accounting for cumulative errors over time while preventing integral windup. A schematic representation of the used control method in this report is shown in Fig.4.4.

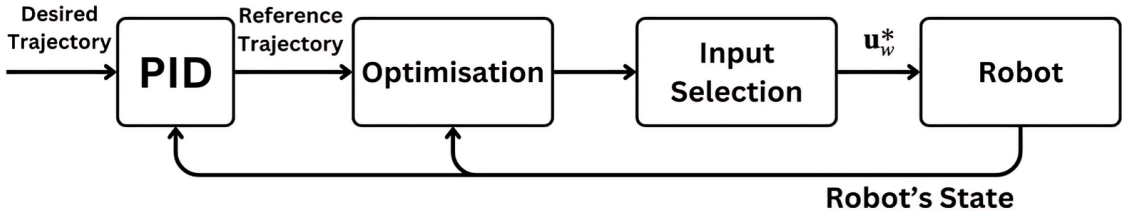


Figure 4.4: Block diagram of the used control method in this report for the Omnimorph for a fixed $\alpha(30°)$.

CONTRIBUTIONS

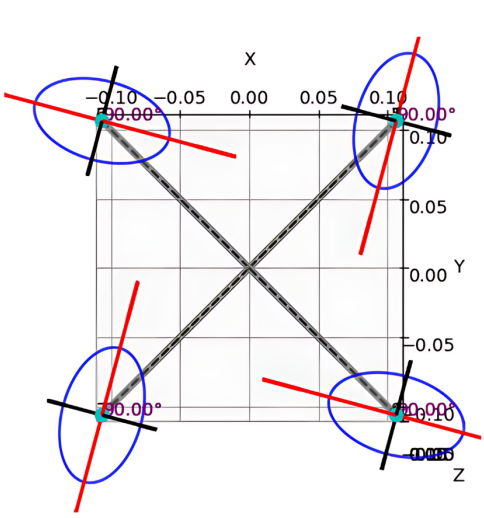
5 Contributions

5.1 A Designer Tool

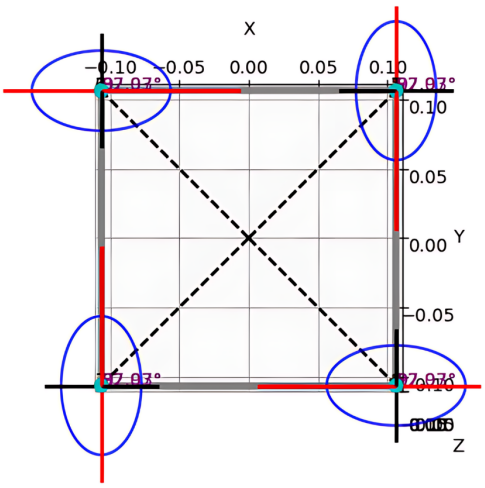
A tool was designed in python to help UAV designers and developers in the future. The tool begins by plotting the given drone design, including rotor positions and optional tilting axes, to visualize and verify the configuration. Next, it computes the force and torque allocations based on the provided positions, rotor normals, and optional uniform tilting angle (alpha) or individual tilting angles. It then evaluates the set of attainable forces and torques by generating all combinations of rotor inputs and calculating the convex hull of these points. Finally, the tool visualizes the convex hull, allowing users to understand the drone's performance envelope.

This tool can be adapted to any drone design by inputting the appropriate configuration parameters. We will use the designed tool for the upcoming evaluations of the Omnimorph and the relevant state-of-the-art omnicopter of Brescianini and D'Andrea (2016).

5.1.1 Mechanical Simplification



(a) Top view of the omnicopter of Brescianini and D'Andrea (2016), showing the direction of the rotor normals in black. The tilting axes in red designed by the authors of Aboudorra et al. (2024) **(Design Choice 1)**.



(b) Top view of the Omnimorph of Aboudorra et al. (2024), showing the direction of the rotor normals in black and the tilting axes in red **(Design Choice 2)**.

Figure 5.1: Topview of both platforms the omnicopter Brescianini and D'Andrea (2016), and the Omnimorph showing the normals of the rotor discs and their relative tilting axes to achieve morphability, which was calculated by the authors of Aboudorra et al. (2024).

Fig.5.1a presents a way that if the omnicopter of Brescianini and D'Andrea (2016) was morphable, which is a design choice done by the authors of Aboudorra et al. (2024) (Design Choice 1), these tilting axes shown in red arrows, would be used to transition it to an underactuated state using a single servo motor. On the other hand, the Omnimorph's propeller tilting axes sown in red arrows in Fig.5.1b lie along the edges of the cube formed by the propellers. This design is achieved by rotating Design Choice 1 tilting axes by $\frac{\pi}{12}$ radians (i.e., 15 degrees) around the Z_B axis. This rotation done by the authors of Aboudorra et al. (2024) results in a symmetric configuration that simplifies the mechanical realization of the tilting mechanism. By having all the tilting axes lie on the cube edges, a belt mechanism can be designed in a simpler manner,

facilitating easier and more reliable implementation. In addition, the same tilting angle (α) for the Omnimorph would be $\approx 54.73°$ in the configuration of the omnicopter platform.

5.2 Feasible Wrenches

The set of feasible wrenches $\boldsymbol{W}$ defines all wrenches that the structure can generate, considering the thrust limits of the rotors:

$$\boldsymbol{W} = \{\boldsymbol{A}\boldsymbol{u} \mid f_{\min} \preceq \boldsymbol{u} \preceq f_{\max}, \boldsymbol{u} \in \mathbb{R}^8\} \tag{5.1}$$

where $\boldsymbol{A}$ is the allocation matrix that maps the input thrust forces from the rotors to the resultant wrench (force and torque vectors), and $f_{\max}/f_{\min}$ is the maximum/minimum thrust force each rotor can produce in either direction. This set is bounded and forms a convex polytope in $\mathbb{R}^6$, Boyd and Vandenberghe (2004). Also, the operator " $\preceq$ " represents element-wise comparison.

The convex hull is defined by the vertices corresponding to the maximum and minimum attainable forces and torques. The uniformity of this set is determined by the ratio of the maximum distance $d_{\max}$ to the minimum distance $d_{\min}$ from the origin to the surface of the hull, as shown in Fig.A.4 in the appendix:

$$U = \frac{d_{\min}}{d_{\max}} \tag{5.2}$$

where:

- $d_{\max}$ is the maximum distance from the origin to any point on the convex hull.
- $d_{\min}$ is the minimum distance from the origin to any point on the convex hull.

This uniformity metric provides insight into the balance and distribution of the forces and torques that the UAV can exert. The closer to 100% the more uniform the convex.

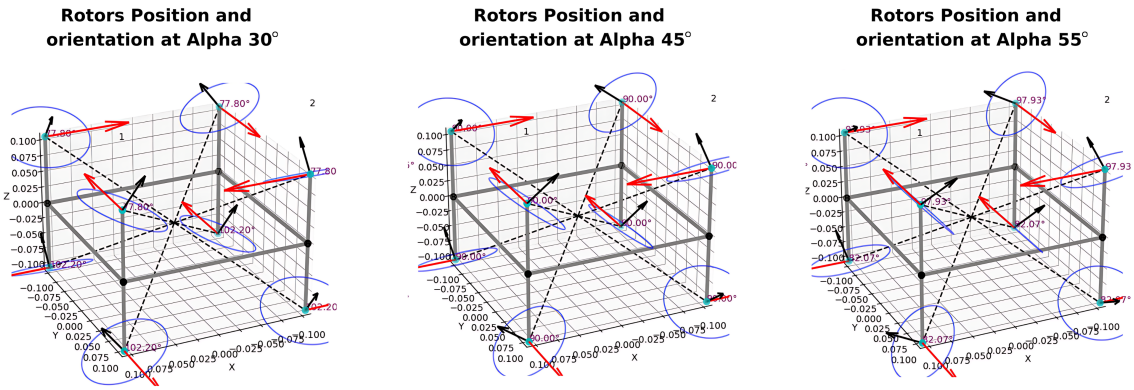


Figure 5.2: The three tilting angles of the Omnimorph that will be analyzed for their convex hull. Also, The angle between the z-axis of the propeller frame and the position vector from the body center of mass (COM) to the propeller is shown in purple.

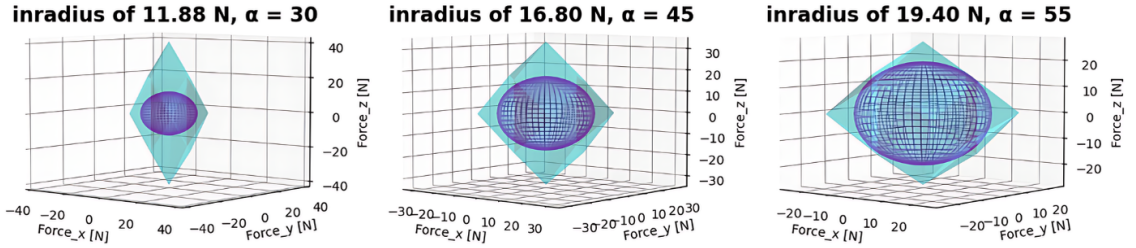


Figure 5.3: Set of attainable thrusts for zero torque, which is described by a convex Hull with different inradius (ODL) for different tilting angle α. With uniformity of **41%, 70%, and 87%** respectively.

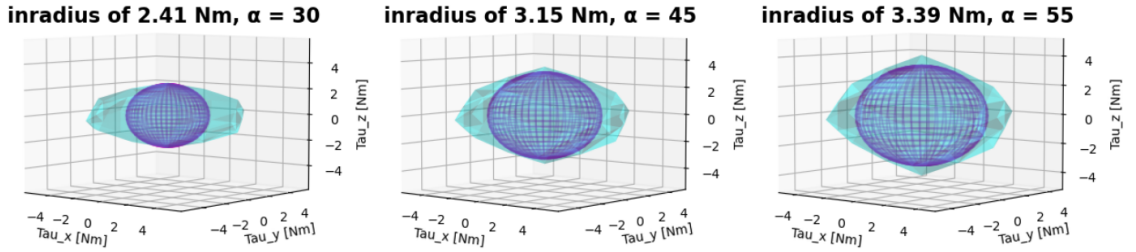


Figure 5.4: Set of attainable torques when generating a thrust of magnitude mg in any direction, where m denotes the vehicle's mass and g resembles the gravitational constant, i.e. $g = 9.81 ms^{-2}$. It is described by a convex Hull with different inradius (ODL) for different tilting angle α. With uniformity of **44%, 62%, and 65%** respectively.

According to Hamandi et al. (2021a), for a platform to be fully omnidirectional, the radius of the maximum inscribed sphere in a corresponding force convex, centered at the centroid of the convex, has to be greater than the weight of the platform. They refer to this as omnidirectional Lift (ODL). In other words, a platform is omnidirectional if it is fully actuated and can lift its weight in any direction about the origin. As can be seen from Fig.5.3, the Omnimorph is considered to be omnidirectional at $\alpha \approx 42°$. The Omnimorph shows optimal performance at an alpha of 55° because the convex hull is most uniform and has the largest inradius of 19.4 N. At an alpha of 45°, the rotor disk normals are orthogonal to the position vector of each propeller with respect to the body frame ($P^B_{p_i}$). In addition, the wrench convex at $\alpha = 30°$ is shown in Figs.5.4 and 5.3, this is the angle that will be used in real tests done on the platform in this report.

5.2.1 Wrench Space Comparison

The omnidirectional platform of Brescianini and D'Andrea (2018) has an inradius of 19.4 N and 3.4 Nm as seen in Fig.5.5, and they have a more uniform convex hull with the same inradius of the forces and torques attainable set as the Omnimorph. In the omnicopter of Brescianini and D'Andrea (2018), it can be seen that the rotor disks are aligned perpendicular to their position vector in order to maximize the torque output for a given propeller thrust. This alignment enhances the torque-generating capability, leading to a more uniform torque convex hull. However, in the Omnimorph, this alignment is not present as because of the mechanical simplification done in Fig.5.1 Aboudorra et al. (2024); discussed in the prior subsection, which results in **less uniformity** in the convex hull, as well as a resulting **reduction in the set of attainable torques inradius of about 6%.** The lack of perpendicular alignment in the Omnimorph means that while it can achieve omnidirectional lift, its torque generation is less optimal. However, the group considers the reduction of 6% in the torques to be acceptable as it offers a simpler and

more practical mechanical realization of the morphing platform, making the small sacrifice in torque well worth the benefits in terms of mechanical simplicity and feasibility.

This comparison directly addresses the first research question of this thesis, which investigates **how design changes in the Omnimorph, in comparison to the omnicopter of Brescianini and D'Andrea (2016), influence its feasible force and torque sets.** These changes, aimed at mechanical simplification, impact the platform's performance by altering the uniformity and the inradius of the torque convex hull. By analyzing these modifications, we can evaluate the trade-offs between achieving mechanical simplicity and maintaining more optimal torque generation.

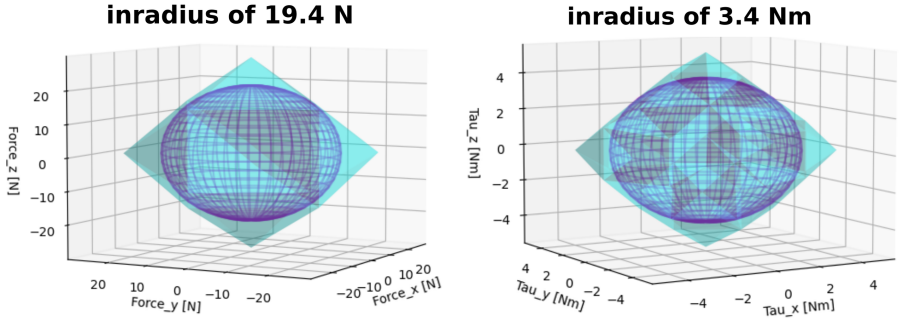


Figure 5.5: The set of attainable thrusts for zero torque, and the set of attainable torques when generating a thrust of magnitude mg in any direction of the omnicopter found in Brescianini and D'Andrea (2016), calculated based on the parameters found in Brescianini and D'Andrea (2018). Uniformity is **87% & 74% respectively.**

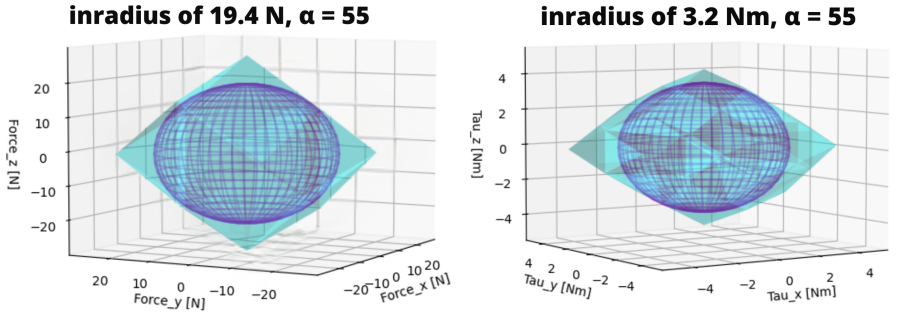


Figure 5.6: The set of attainable thrusts for zero torque, and the set of attainable torques when generating a thrust of magnitude mg in any direction of the Omnimorph, calculated based on the same parameters found in Brescianini and D'Andrea (2018) at the equivalent angle $\alpha = 54.73°$. Uniformity is **87% & 65% respectively.**

5.3 Thrust Mismatch

In the study conducted by Bazzana et al. (2024) on the Omnimorph platform, it was demonstrated that aerodynamic interference between propellers is significantly higher when the propellers are aligned ($\alpha = 0°$), as they are positioned directly above one another. This interference reduces as the angle α increases, indicating that the propeller configurations with non-zero tilting angles experience less aerodynamic influence from adjacent rotors. This observation aligns with the findings by Brescianini and D'Andrea (2018), who noted that rotors generate less thrust and torque during flight compared to isolated conditions due to the increased speed of the incoming airflow caused by aerodynamic interactions.

A critical observation in the study of Bazzana et al. (2024) was that when the platform was spinning all rotors at 120 Hz, the force lost in each of the bottom propellers amounted to 0.92 N,

representing a decrease of 34%. This significant loss underscores the impact of aerodynamic interference, where the downwash from the upper propellers adversely affects the thrust generation of the lower ones, leading to a thrust mismatch.

To address this issue in this thesis, a linear model was developed that accounts for variations in the thrust coefficient c_f based on the tilting angle α. This model, inspired by the findings in Barbara's paper and the constant model they implemented it in the paper of Aboudorra et al. (2024), aims to provide a more accurate prediction of thrust forces under different configurations. According to the paper of Bazzana et al. (2024), the maximum thrust coefficient mismatch multiplier was determined to be 35%. This value corresponds to the maximum reduction in thrust efficiency due to aerodynamic interference when the propellers are aligned ($\alpha = 0°$).

The model adjusts the thrust coefficient mismatch linearly from the maximum 35% ($\alpha = 0°$) to the minimum 70% ($\alpha = 60°$). This multiplier acts as follows: 100% indicates that there is no mismatch, meaning the thrust efficiency is at its nominal value with no aerodynamic interference. A 100% multiplier would correspond to conditions where the propellers are unaffected by the downwash from adjacent rotors. As the multiplier decreases to 35%, it signifies a greater degree of interference, leading to a maximum 35% loss in thrust efficiency at $\alpha = 0°$. Conversely, at $\alpha = 60°$, the multiplier reaches 70%, indicating that the thrust efficiency is reduced by only 30%, and the effects of aerodynamic interference are minimal.

This linear adjustment ensures that the model accurately reflects the varying degrees of aerodynamic interference as the propeller angle α changes. Implementing this dynamic adjustment in the Simulink model (controller) allows for real-time modifications to the thrust coefficient, enhancing the overall model. This approach offers a more practical way to model the thrust mismatches caused by aerodynamic interference, as modelling the aerodynamic interference is a non-trivial approach.

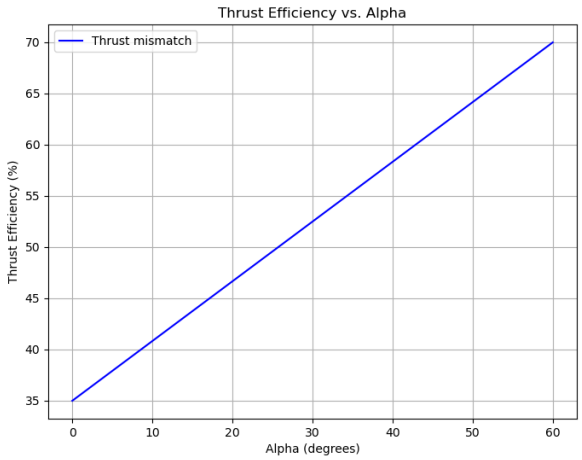


Figure 5.7: Thrust Efficiency vs. Alpha. This illustrative plot shows the assumed linear relationship between the tilting angle (α) of the propellers and the resulting thrust efficiency. The efficiency increases from 35% at $\alpha = 0°$ to 70% at $\alpha = 60°$, indicating a reduction in aerodynamic interference.

5.4 Dynamic Weights

To mitigate the thrust mismatch caused by the dependence of propeller performance on the tilting angle, this report presents an algorithm designed to balance energy conservation and performance in the closed-loop trajectory-controller system. The proposed solution dynamically adjusts the optimizer weights in Fig.4.3 on an input based on the position error relative to

a certain threshold. The logic flowchart is shown in Fig.5.8. If the position error is less than the threshold, the algorithm prioritizes energy conservation by increasing the weight on the input (W_1). This action reduces the energy consumption as the system moves towards a more accurate error so alpha is most likely to get smaller (closer to underactuation) if the tracking error is not compromised.

On the other hand, if the position error is greater than or equal to the threshold, the algorithm prioritizes performance by decreasing the weight on the input. This action enhances the system's responsiveness and performance.

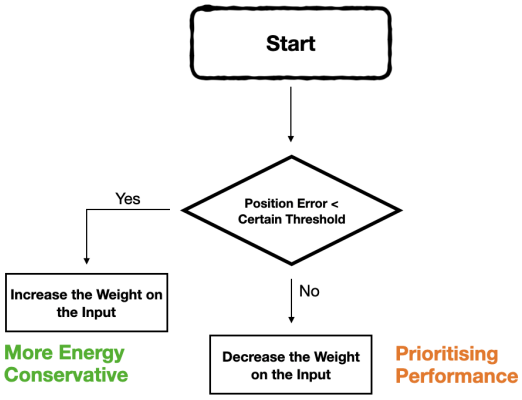


Figure 5.8: Illustration of the dynamic weight adjustment algorithm. The weight W is adjusted based on the position error e_p relative to a threshold $e_{p,\text{thresh}}$. When e_p is below the threshold, the weight increases exponentially. When e_p is above or equal to the threshold, the weight decreases exponentially.

The combined equation for updating the weight W can be expressed as:

$$W_{\text{k}} = W_{\text{k-1}} \times \left[r_{\text{inc}} \cdot H(e_{p,\text{thresh}} - e_p) + r_{\text{dec}} \cdot H(e_p - e_{p,\text{thresh}}) \right] \tag{5.3}$$

where:

- W_{k} is the new weight.
- $W_{\text{k-1}}$ is the current weight.
- r_{inc} is the increment rate, where $r_{\text{inc}} > 1$.
- r_{dec} is the decay rate, where $0 < r_{\text{dec}} < 1$.
- e_p is the position error.
- $e_{p,\text{thresh}}$ is the error threshold.
- $H(x)$ is the Heaviside step function defined as:

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

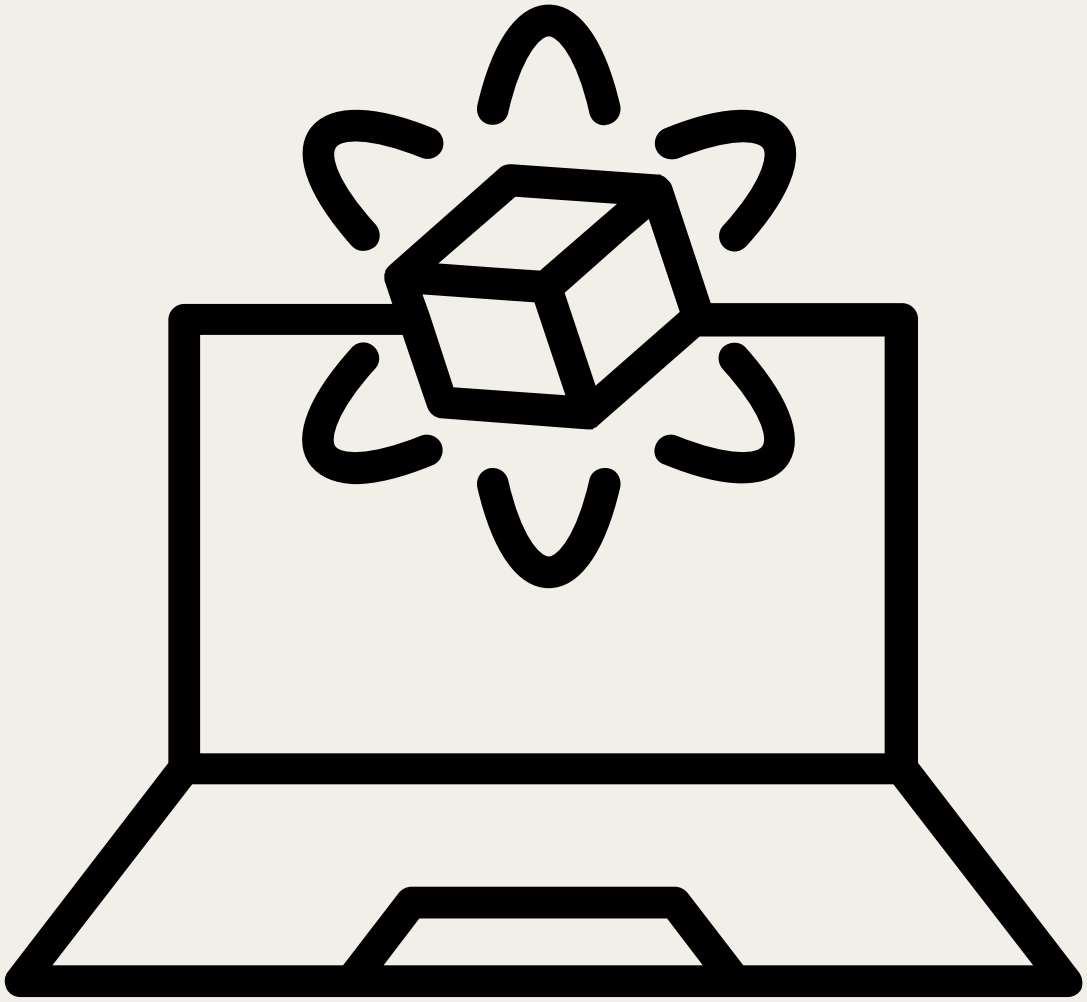
In this equation:

- When $e_p < e_{p,\text{thresh}}$, $H(e_p - e_{p,\text{thresh}}) = 0$ and $H(e_{p,\text{thresh}} - e_p) = 1$, so the weight increases exponentially.
- When $e_p \geq e_{p,\text{thresh}}$, $H(e_p - e_{p,\text{thresh}}) = 1$ and $H(e_{p,\text{thresh}} - e_p) = 0$, so the weight decreases exponentially.

This logic is preferred as it guarantees a positive definite increment or decrement, ensuring that W_1 (input weight) remains positive definite. Additionally, maximum and minimum limits on W_1 are necessary to prevent the weight from becoming excessively large or small, which could lead to instability or ineffectiveness in the system.

This dynamic adjustment mechanism addresses the research question of **"How to mitigate the effect of this uncertainty in the closed-loop trajectory-controller system?"** by providing a practical approach to mitigate the impact of propeller performance degradation due to tilting angles. It ensures that the closed-loop system can maintain stable and accurate trajectory tracking despite the inherent uncertainties, thereby enhancing the overall reliability and efficiency of the Omnimorph platform. The results of this dynamic adjustment mechanism is demonstrated in the simulations 6 section, where the results show that the closed-loop system can maintain stable and accurate trajectory tracking despite the inherent uncertainties.

SIMULATIONS



6 Simulations

In the paper of Aboudorra et al. (2024), numerical simulations have been conducted using a Unified Robot Description Format (URDF) model of the OmniMorph and an ordinary differential equation (ODE) physics engine in Gazebo. The control software has been implemented in Matlab-Simulink. The interface between Matlab and Gazebo is managed by a Gazebo-genom3 plugin[1]. A Simulink S-function embedding the qpOASES QP solver[2] has been utilized to solve the optimization problem.

6.1 Simulation Scenarios

In the following simulations, we aim to test the mitigation strategies and dynamic weight adjustments discussed earlier in section 5. To do so, we will replicate the simulation scenarios shown in the following Video, specifically cases A and B, as detailed in the study of Aboudorra et al. (2024). The simulations will use the same parameters and trajectory specifications found in Aboudorra et al. (2024) to ensure consistency in evaluation.

The mass and inertia of the robot are $m = 1.3150$ kg and $J = \text{diag}(1.16 \times 10^{-2}, 1.13 \times 10^{-2}, 1.13 \times 10^{-2})$ Nms^2. Other controller and optimiser parameters are:

- $\mathbf{K_{p1}} = \text{diag}(30, 30, 30)\,\text{s}^{-1}$
- $\mathbf{K_{p2}} = \text{diag}(300, 300, 300)\,\text{s}^{-2}$
- $\mathbf{K_{\omega 1}} = \text{diag}(40, 40, 40)\,\text{s}^{-1}$
- $\mathbf{K_{\omega 2}} = \text{diag}(100, 100, 100)\,\text{s}^{-2}$
- $\mathbf{W_3} = 10^{-5} I_8$

Then in Aboudorra et al. (2024), they defined two cases; Case A is characterized by a lower weight on the input, W_1, and a higher weight on the tracking error, W_2, compared to Case B. Specifically, the optimization weights for Case A are set as $W_1 = 10^{-8} I_8$ and $W_2 = \text{diag}(3 \cdot 10^6, 3 \cdot 10^6, 3 \cdot 10^6, 10^3, 10^3, 10^3)$. For Case B, the weights are $W_1 = 10^{-5} I_8$ and $W_2 = \text{diag}(3 \cdot 10^4, 3 \cdot 10^4, 3 \cdot 10^4, 10, 10, 10)$. The differences in these weight settings lead to distinct performance outcomes.

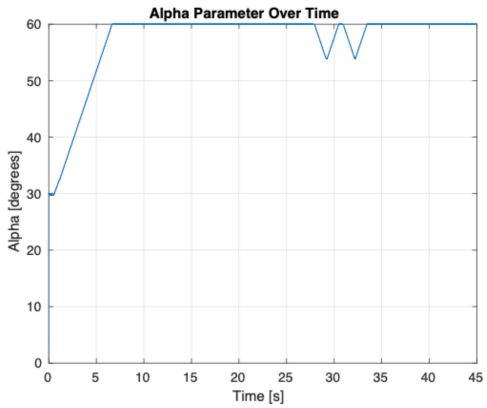
In Case A, the robot shows better tracking performance, with average position and attitude errors of $0.011m$ and $0.062°$, respectively. However, this comes at the expense of higher motor torque, indicating a trade-off between accuracy and energy efficiency.

In contrast, Case B demonstrated a failure in the referenced study due to the imposition of a constant thrust mismatch of 35%. This mismatch was introduced as a worst-case scenario to account for the observed reduction in propeller thrust performance due to aerodynamic interference. The constant thrust mismatch led to increased tracking errors, particularly in horizontal directions where the lower values of α (tilting angles) reduced the system's dexterity. Although the platform managed to complete the task in Case A, it failed in Case B under the same conditions, highlighting the need for more robust control strategies to handle such mismatches.

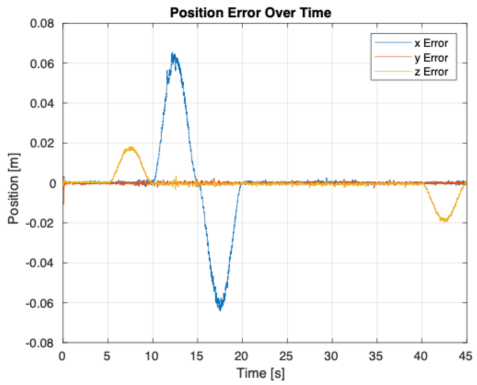
The proposed dynamic weight adjustment mechanism, which varies the control weights based on the position error, aims to mitigate these issues. By prioritizing tracking performance over energy efficiency when errors exceed a threshold, the system can better handle uncertainties like thrust mismatches. The simulation results presented nex twill demonstrate the effectiveness of this approach, showcasing how the proposed method improves system robustness and maintains trajectory accuracy despite significant variations in propeller performance.

[1] https://git.openrobots.org/projects/mrsim-gazebo
[2] https://github.com/coin-or/qpOASES

6.2 Simulation Results

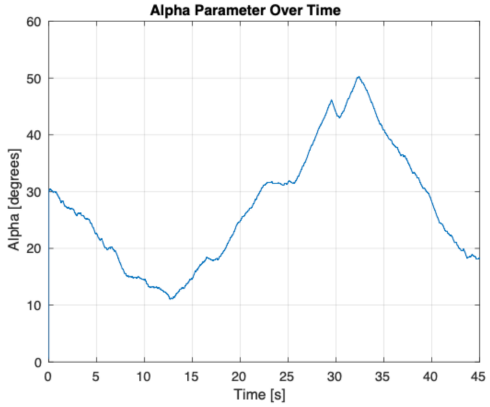


(a) Alpha Parameter Over Time

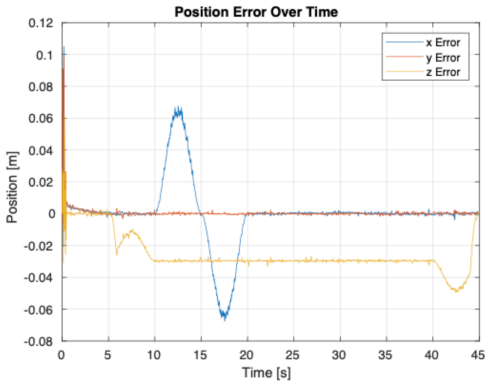


(b) Position Error Over Time

Figure 6.1: Replicated simulation results from **Case A** in Aboudorra et al. (2024). (a) shows the alpha parameter over time, and (b) shows the position error over time, **with and without the constant mismatch** (35% lower than nominal) shows the same performance.

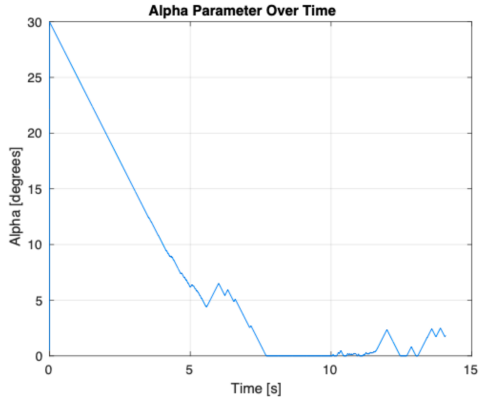


(a) Alpha Parameter Over Time (Without Mismatch)

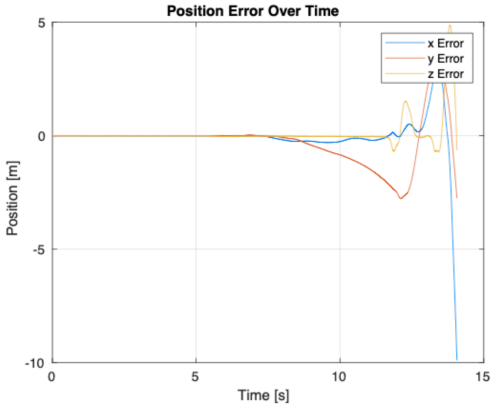


(b) Position Error Over Time (Without Mismatch)

Figure 6.2: Replicated simulation results from **Case B** in Aboudorra et al. (2024), **without mismatch**. (a) shows the alpha parameter over time, and (b) shows the position error over time.

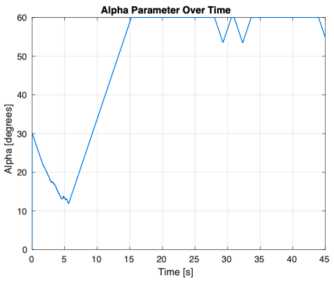


(a) Alpha Parameter Over Time (Constant Mismatch)

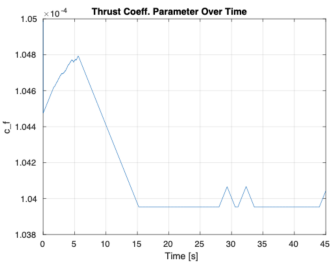


(b) Position Error Over Time (Constant Mismatch)

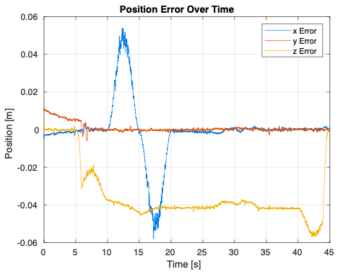
Figure 6.3: Replicated simulation results for **Case B** with a **constant mismatch** of 35% lower than nominal found in Aboudorra et al. (2024). (a) shows the alpha parameter over time, and (b) shows the position error over time. The results indicate that the Omnimorph fails and becomes unstable as the alpha parameter decreases to a point where recovery is not possible, highlighting the limitations in control authority and the critical impact of thrust mismatch.



(a) Alpha Parameter Over Time



(b) Thrust Coeff. Over Time



(c) Position Error Over Time

Figure 6.4: Replicated simulation results for **Case B** with introduced **variable mismatch** and **dynamic weighting**. (a) shows the alpha parameter over time, (b) shows the thrust coefficient parameter over time, and (c) shows the position error over time. The results demonstrate that the Omnimorph was able to recover from the disturbances, indicating improved stability and control with the dynamic weighing strategy.

6.3 Interpretation of Results

In the initial simulations conducted by Aboudorra et al. (2024), a constant thrust mismatch was applied as a worst-case scenario to assess the system's robustness. In this scenario, Case A successfully passed, maintaining stability and accurate tracking, as shown in Fig.6.1. However, Case B failed, as the system became unstable and was unable to recover from significant position errors, highlighted in Figs.6.3b and 6.3a.

Subsequently, a more accurate model was introduced in this thesis to account for the thrust mismatch based on the tilting angle α shown in Fig.6.4b and explained in Chapter 5. This model provided a variable mismatch that more closely represented the actual performance interference of the propellers. Despite this refinement, Case B still failed to maintain stability under these conditions, as seen in the replicated simulations. The platform was unable to recover its position, confirming the limitations of the fixed weighting strategy in the presence of variable aerodynamic effects.

To address these issues, dynamic weights were implemented in the control strategy in section 5. This approach adjusted the weighting based on the real-time system state and errors, allowing for more responsive and adaptive control. The results from these simulations demonstrated a significant improvement in stability and tracking accuracy. The platform successfully recovered its position and stabilized, as illustrated in Figs. 6.4a, and 6.4c.

The dynamic weighting strategy effectively mitigated the negative impact of thrust mismatch, particularly in scenarios with high initial weights on the input. This approach directly addresses the research question: **"How to mitigate the effect of this uncertainty in the closed-loop trajectory-controller system?"** By dynamically adjusting the control weights based on real-time system feedback, the method enhances the system's ability to adapt to uncertainties and maintain stability. The simulation results validate the proposed solution, demonstrating that the Omnimorph platform can robustly handle dynamic changes and disturbances.

6.4 Simulation Scenarios for Real Tests

Regarding the third research question to evaluate the performance of the proposed control strategy, two primary trajectories will be evaluated for both simulations and real-world tests. These trajectories aim to thoroughly assess the system's capabilities under different conditions and provide a comprehensive comparison of the control performance.

6.4.1 Trajectory 1: Hovering Test

The first trajectory is a simple hovering test designed to evaluate the platform's ability to maintain a stable position without tilting. This test involves:

1. **Initial Hover**: The platform ascends to a predetermined altitude and maintains a hover with zero attitude, meaning there is no intentional tilting of the platform.
2. **Lateral Movement**: After hovering stably for a specified duration, the platform makes a small movement along the y-axis. This movement is intended to test the system's responsiveness and precision in maintaining a stable hover while executing small position adjustments.

This trajectory focuses on the fundamental stability and control accuracy of the system in maintaining a hover and executing minor movements without inducing tilt.

6.4.2 Trajectory 2: Comprehensive Functionality Test

The second trajectory is designed to test a wide range of functionalities of the platform, ensuring that all critical aspects of its performance are evaluated. This trajectory involves:

1. **Ascent**: The platform ascends to a predetermined altitude, testing the vertical control capabilities.
2. **Hover**: The platform maintains a stable hover at the specified altitude, similar to the initial part of Trajectory 1.
3. **Lateral Movement**: The platform moves laterally without changing its orientation. This step tests the ability to move horizontally while maintaining a stable attitude.
4. **Tilt without Position Change**: The platform tilts to a specified angle without changing its horizontal position. This tests the control system's ability to manage attitude adjustments independently of positional control.

6.4.3 Evaluation Metrics

The performance of the control strategies will be evaluated using the following metrics:

1. **Root Mean Square Error (RMSE)**: RMSE will be used to quantify the accuracy of the platform in maintaining the desired trajectories. It provides a measure of the deviation between the desired and actual positions over time, with lower RMSE values indicating

higher accuracy. The RMSE is calculated using the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{6.1}$$

where x_i is the desired position, $\hat{x}_i$ is the actual position, and n is the number of samples.

2. **Total Power Consumption**: The total power consumed by the rotors will be calculated using the formula (Aboudorra et al. (2024))

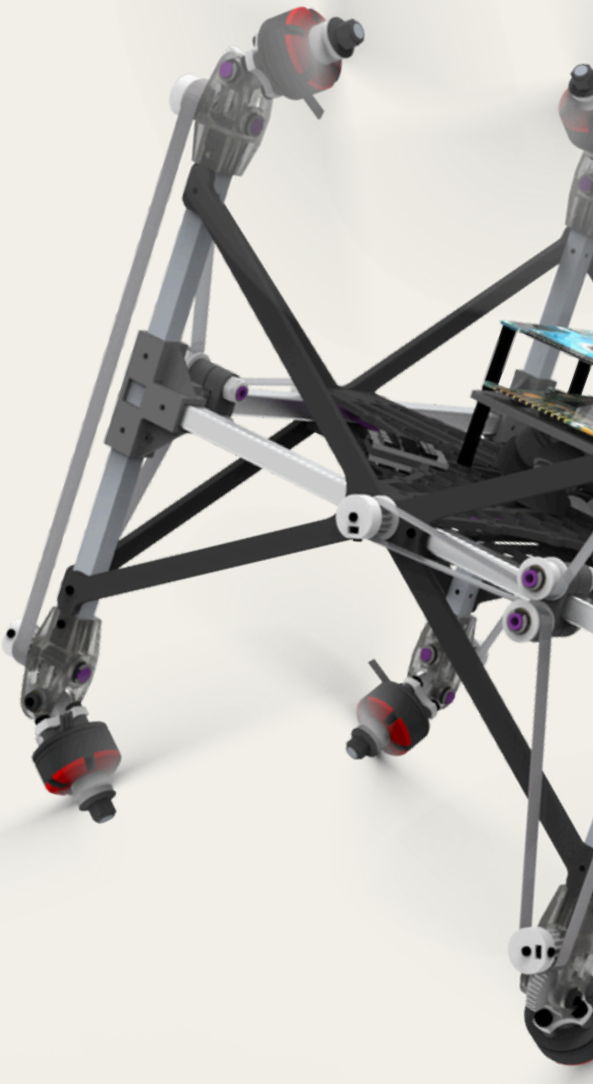
$$\sum_{i=1}^{8} c_\tau \cdot \omega_i^3 \tag{6.2}$$

where c_τ is the torque coefficient and ω_i is the angular velocity of the i-th rotor. This metric helps understand the behavior of the power usage in real vs. simulated tests to identify significant differences. Comparing the total power consumption across different control strategies provides insights into performnce of the real platform in comparison with the simulated platform.

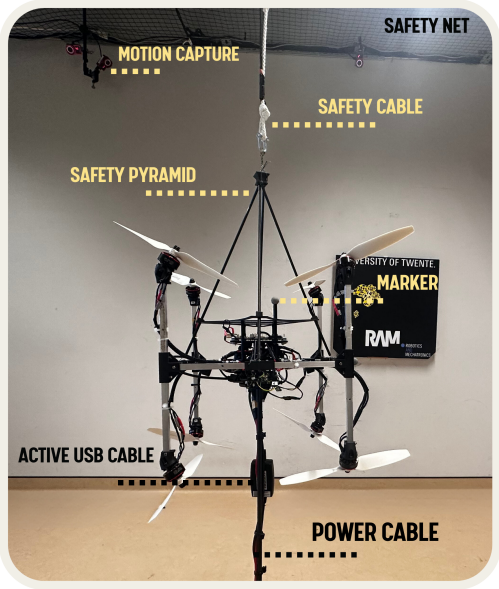
3. **Controller Signal Behavior**: The behavior of the controller signals will be analyzed to compare the responsiveness of the control strategies. This analysis will include examining the control inputs required to maintain the desired trajectories.

By designing these two trajectories and evaluating the performance using RMSE, power consumption, and controller signal behavior, we aim to answer the research question: **"How much does the behavior of the controlled real platform differ from the simulated one in terms of position and attitude errors?"** This approach will provide a comprehensive comparison between the simulations and real-world tests, helping to identify the main differences between real and simulated tests. The results of these comparisons will be presented in the following section of the experiments 7.2.

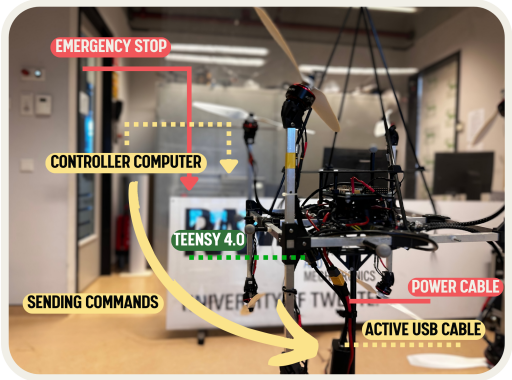
EXPERIMENTS



7 Experiments



(a) Experiment setup showing the current test bed.



(b) Experiment setup showing the current communicating protocol.

Figure 7.1: Experiment Setup of the Omnimoprh.

The experimental setup for testing the Omnimorph involves safety measures to ensure efficient and safe operation. The Omnimorph is suspended within a pyramid-like frame, which provides a safer environment for the Omnimorph to operate (see Fig.7.1). This structure helps in preventing it from colliding with the safety rope during tests. A rigid section of rope is used as a safety measure to avoid falling of the rope while the drone is increasing altitude. Additionally, the power and USB cables are attached to a rigid rod from the bottom, preventing the propellers from coming into contact with the cables.

The control commands are sent from a computer located behind the counter to a Teensy microcontroller via an active repeater USB cable. An emergency button is placed between the battery on the table behind the counter and the Omnimorph, providing an immediate way to cut power if necessary. We are using a battery instead of a power supply because the ESCs use regenerative braking, which can harm the power supply, also, the power supply can only provide a maximum of 60 A, which is insufficient for higher motor frequencies.

7.1 Motor Tuning

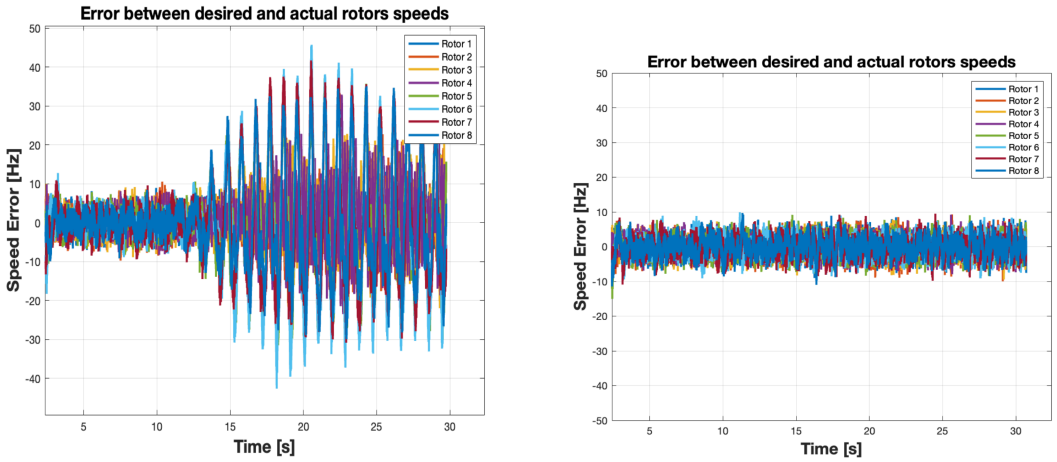
During the first flight tests, the outcome highlighted that the low-level controller was not properly tuned, causing errors between the desired rotor velocities and the actual ones to diverge as shown in Fig.7.2a. To address this, in this thesis we aimed to accurately model the Brushless DC (BLDC) motor and tune the PID gains based on the best model fit.

To evaluate the dynamic response of the BLDC motor model, a Pseudo-Random Binary Sequence (PRBS) input signal was applied, and the resulting speed output was measured in Hertz (Hz). The primary goal was to identify the plant, which is the motor in this case, using open-loop techniques and subsequently tune the PID gains based on the best model fit.

The initial test involved applying a PRBS input signal to the motor and recording the output speed. Fig.A.6a in the appendix shows the raw data collected from this experiment, illustrating the motor's response to the applied PRBS input signal. To improve the clarity of the data

and to provide a more accurate plant model, both the input and output data were detrended to remove any linear trends and offsets due to non-zero means. This detrending process is essential because all of the estimators assume that the data does not contain such linear trends or offsets, which can otherwise bias the model estimation. Fig.7.3 displays the detrended data.

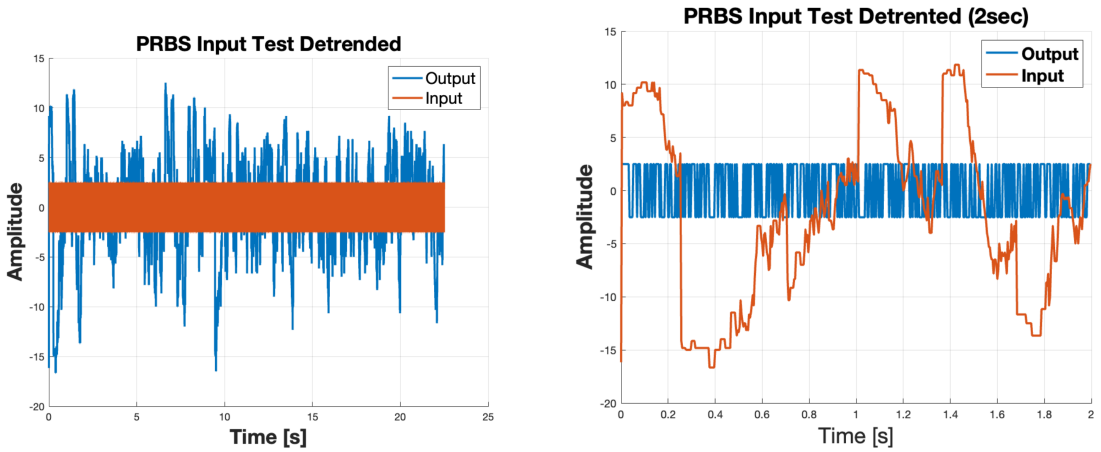
In addition, we focused on the first 2 seconds of the PRBS input signal, as it is periodic and exhibits white noise characteristics only within this period as can be shown from the autocorrelation Fig.A.6b in the appendix. This is done to ensure: uniform excitation across all frequencies, allowing the models to capture the true dynamics of the BLDC motor without being influenced by repetitive patterns or non-white noise.



(a) Out of tuned low level gains for the motor.

(b) Improved tuning low level gains for the motor.

Figure 7.2: Error between the commanded/desired rotor angular Velocities and the actual ones.



(a) Detrended real open-loop data.

(b) First 2 seconds of the detrended real open-loop data.

Figure 7.3: Detrended obtained real motor open-loop data using PRBS signal excitation.

We compared several model structures: ARX, ARMAX, Output-Error (OE), Box-Jenkins (BJ), and State-Space (SS). These models were validated against the collected experimental data, Fig.7.4 presents the simulated response comparison of the identified models with the validation data. Among these, the OE and BJ models showed the best matches. The OE model excels by directly minimizing the error between measured and model outputs, accurately capturing the input-output relationship crucial for BLDC motor dynamics. The BJ model explicitly models both system dynamics and noise, fitting the data better by separating noise from system dy-

namics, especially in the presence of common motor system noise. The model estimation did not provide the perfect PIDN values due to additional communication delays and the fact that implementing a closed-loop control outside the Teensy microcontroller often yields different results. Additionally, the control period significantly affects the gains since they are not time-independent. However, the model estimation did provide a starting point for the gains. We then used a heuristic approach to fine-tune the low-level PIDN controller.

Creating the function, **`compare_runs`**, evaluates the improvement of the current run over the previous one by comparing the mean and maximum errors of rotor velocities. It calculates and sums these errors across all rotors for both runs, determining if the current run's errors are lower. If both mean and maximum errors are reduced, the current run is considered better.

The function also logs the results and PIDN values into an Excel file as shown in the appendix Fig.A.5, facilitating iterative PIDN tuning by providing clear comparisons between runs and helping to identify optimal PIDN values.

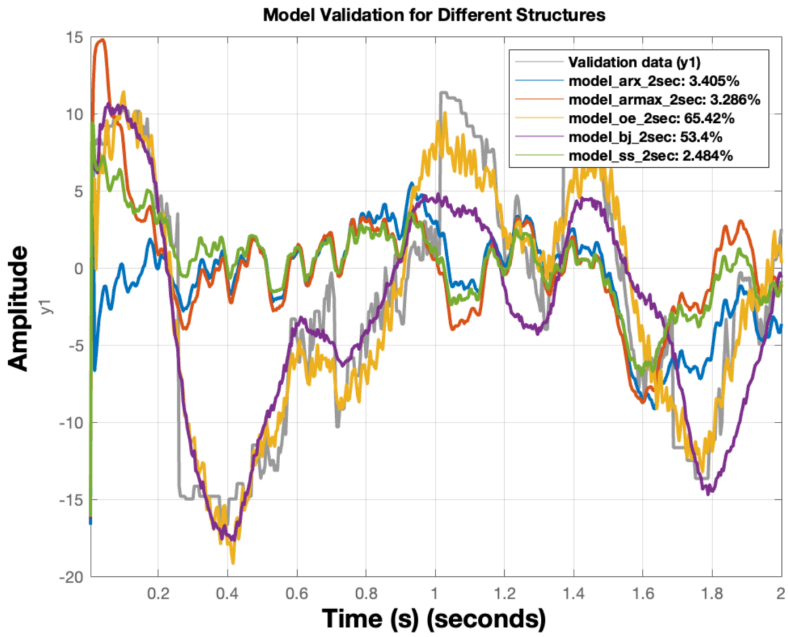


Figure 7.4: Simulated Response Comparison of Identified Models with Validation Data. The OE model showed the best fit with a performance of **65.42%**.

By tuning the PIDN gains based on the most accurate model obtained from these tests and using a heuristic approach, we aim to address the discrepancies observed between the simulated and real-world behaviour of the platform, results shown in Fig.7.2b prove that the low-level controller is now properly tuned for the operating frequencies. This process directly supports our research question on **how the behaviour of the controlled real platform differs from the simulated one in terms of position and attitude errors.**. Improving the low-level controller will reduce the differences between simulation and reality, as simulations often assume near-instantaneous command execution.

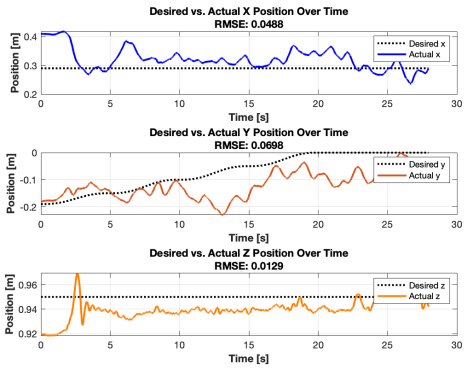
7.2 Flight Tests

In the subsequent section, we will present the results of our experiments, including both plots and quantitative data, in alignment with the trajectory scenarios defined earlier in Section 6.4. This section aims to provide a comprehensive comparison outline of the platform's performance by comparing the outcomes of real-world experiments with simulated results, thereby

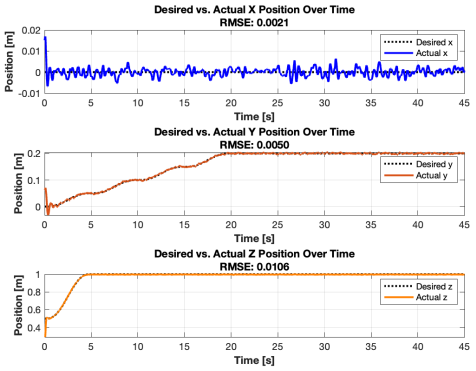
assessing the system's accuracy and behaviour. Additionally, we will examine the plots and results based on the three key metrics defined earlier: RMSE, total power consumption, and controller signal behaviour.

- **Trajectory 1:**
  - Involves hovering with no tilt and a small lateral movement in the y-axis.
  - Focuses on stability during simple maneuvers.
- **Trajectory 2:**
  - Includes ascending, hovering, lateral movements without tilt, and then descending.
  - Tests the platform's ability to handle more complex maneuvers.

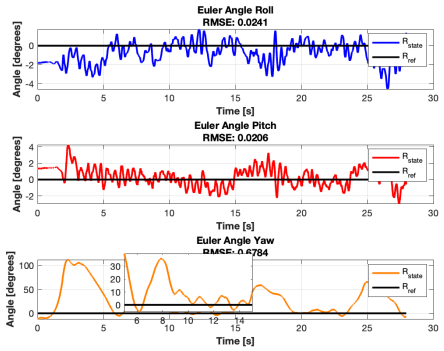
For more detailed information about these trajectories and the metrics used for assessment, please refer back to Section 6.4.



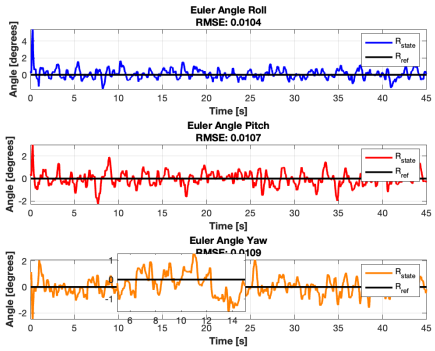
(a) The plots show the real x, y, and z positions with respective **RMSE values of 0.049, 0.070, and 0.013.**



(b) The plots show the simulated x, y, and z positions with respective **RMSE values of 0.002, 0.005, and 0.011.**

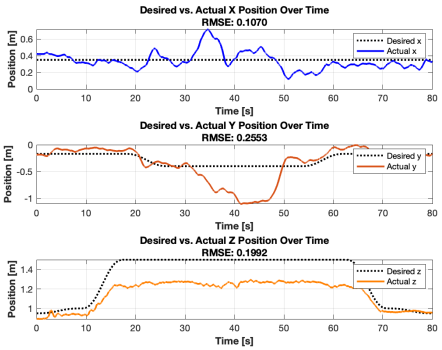


(c) The plots show the real roll, pitch, and yaw angles with respective **RMSE values of 0.024, 0.021, and 0.68.**

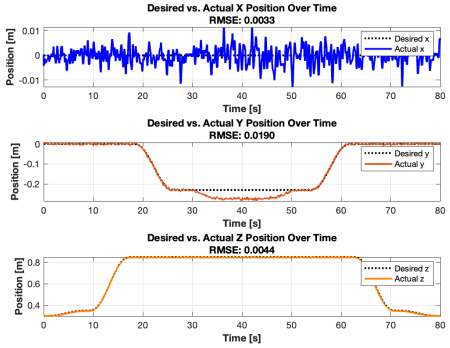


(d) The plots show the simulated roll, pitch, and yaw angles with respective **RMSE values of 0.010, 0.011, and 0.011.**

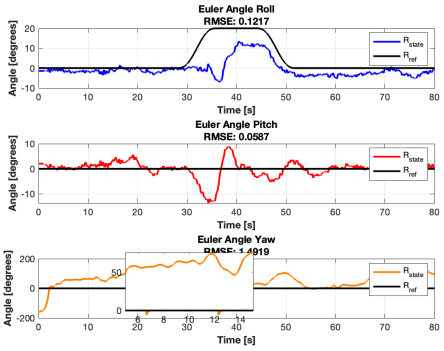
Figure 7.5: Trajectory 1; Comparison of Position and Attitude for Real vs. Simulated Data during Trajectory 1: Hovering Test with Lateral Movement. The subfigures illustrate the performance of the platform in terms of (a) position and (b) attitude for the real platform, and (c) position and (d) attitude for the simulated platform. These plots provide insights into the accuracy of the platform in achieving the specified trajectory.



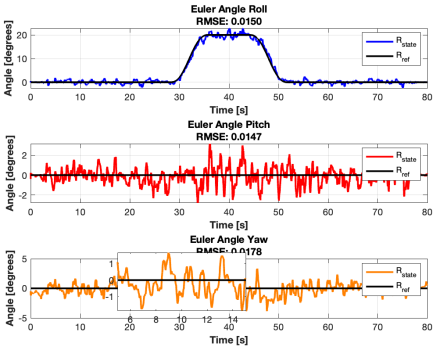
(a) The plots show the real x, y, and z positions with respective **RMSE values of 0.11, 0.26, and 0.20.**



(b) The plots show the simulated x, y, and z positions with respective **RMSE values of 0.003, 0.02, and 0.004.**



(c) The plots show the real roll, pitch, and yaw angles with respective **RMSE values of 0.12, 0.059, and 1.49.**



(d) The plots show the simulated roll, pitch, and yaw angles with respective **RMSE values of 0.015, 0.015, and 0.018.**

Figure 7.6: Trajectory 2; Comparison of Position and Attitude for Real vs. Simulated Data during Trajectory 2: Comprehensive Functionality Test. The subfigures illustrate the performance of the platform in terms of (a) position and (b) attitude for the real platform, and (c) position and (d) attitude for the simulated platform. These plots provide insights into the accuracy of the platform in achieving the specified trajectory.

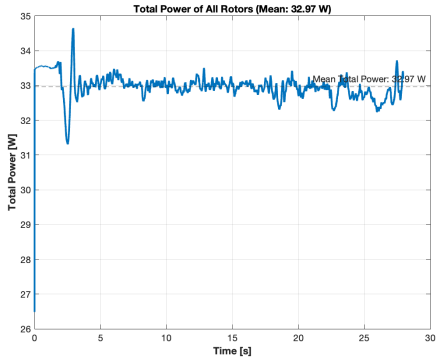
Table 7.1: RMSE Comparison between Simulation and Real Experiments for Trajectories 1 and 2

Trajectory 1: Hovering Test with Lateral Movement			
Metric	**RMSE Sim**	**RMSE Real**	**RMSE Ratio (Real/Sim)**
X Position (m)	0.002	0.049	24.5
Y Position (m)	0.005	0.070	14.0
Z Position (m)	0.011	0.013	1.18
Roll (deg)	0.010	0.024	2.40
Pitch (deg)	0.011	0.021	1.91
Yaw (deg)	0.011	0.68	61.8
Trajectory 2: Comprehensive Functionality Test			
X Position (m)	0.003	0.11	36.7
Y Position (m)	0.02	0.26	13.0
Z Position (m)	0.004	0.20	50.0
Roll (deg)	0.015	0.12	8.00
Pitch (deg)	0.015	0.059	3.93
Yaw (deg)	0.018	1.49	82.8

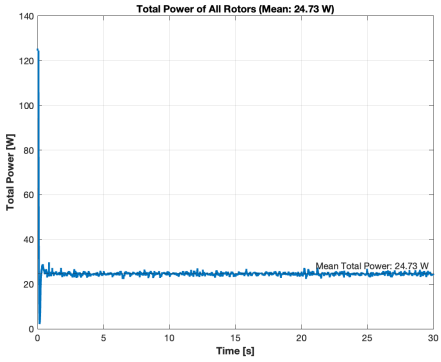
Note: The videos of the real experiments can be found in the following links Trajectory 1, and Trajectory 2.

The RMSE ratios between the real-world experiments and simulations show varying degrees of alignment across different metrics. A lower RMSE ratio indicates a closer match between simulation and real-world performance, which is desirable as it suggests that the simulator is accurately modelling the system dynamics. For instance, the Z Position in Trajectory 1 has a relatively low RMSE ratio of 1.18, reflecting good agreement between the simulation and real-world results.

While a lower RMSE ratio is generally preferred as it reflects closer adherence to the ideal simulated conditions, higher ratios can be beneficial from a different perspective. They reveal areas where the simulator's model may require refinement and highlight the complexity of real-world dynamics that the simulation may not fully capture. This insight can be used to drive further improvements in the simulator, ultimately enhancing its fidelity and making it a more robust tool for predicting real-world performance. More details about the analysis of the results will be presented in the next chapter of discussion 8.

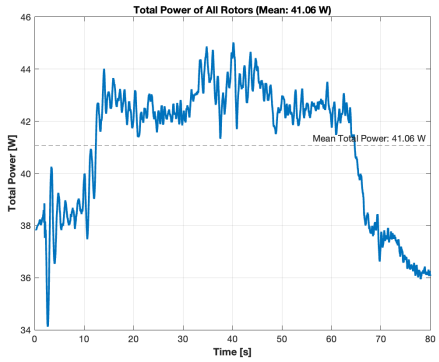


(a) Real experiment: Total power consumed by all rotors **(Mean: 32.97 W)**.



(b) Simulation: Total power consumed by all rotors **(Mean: 24.73 W)**.

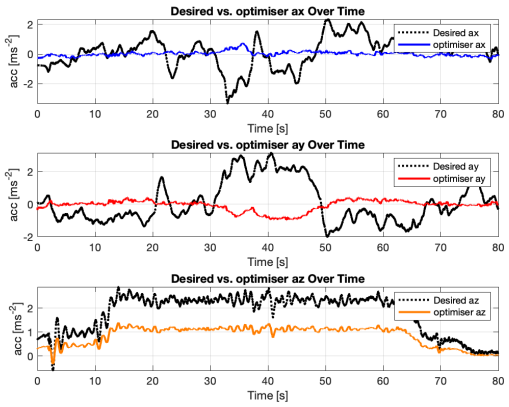
Figure 7.7: Trajectory 1; Comparison of total power consumed by all rotors in real experiments (left) vs. simulations (right).



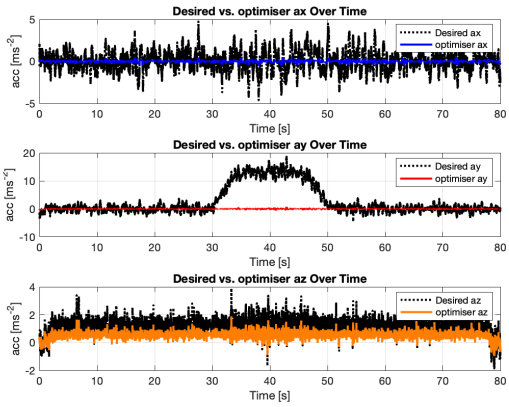
(a) Real experiment: Total power consumed by all rotors **(Mean: 41.06 W)**.

(b) Simulation: Total power consumed by all rotors **(Mean: 25.36 W)**.

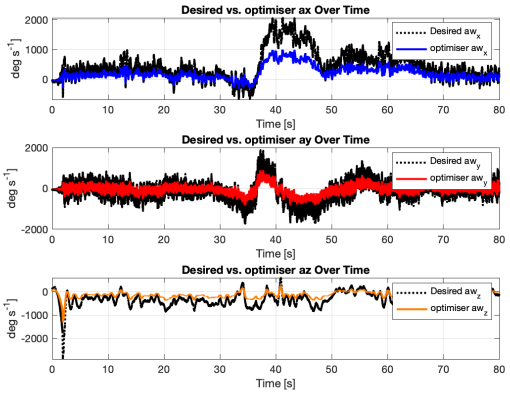
Figure 7.8: Trajectory 2; Comparison of total power consumed by all rotors in real experiments (left) vs. simulations (right).



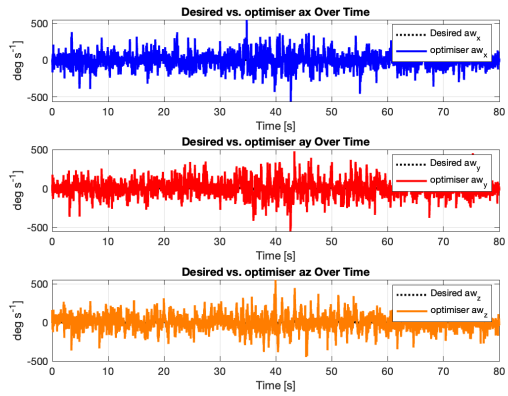
(a) Real experiment: Comparison between the acceleration output from the PID block and the virtual acceleration output from the optimiser.



(b) Simulation: Comparison between the acceleration output from the PID block and the virtual acceleration output from the optimiser.



(c) Real experiment: Comparison between the angular acceleration output from the PID block and the virtual angular acceleration output from the optimiser.



(d) Simulation: Comparison between the angular acceleration output from the PID block and the virtual angular acceleration output from the optimiser.

Figure 7.9: Trajectory 2; Comparison between real and simulated control signals. In each figure, the output acceleration and angular acceleration from the PID block are compared with the virtual acceleration and angular acceleration outputs from the optimiser.

Note: While we measure angular acceleration in our experiments, it is important to note that there is an error in the plots showing the unit as degs^{-1}, which should be correctly noted as degs/s^2.

DISCUSSION

“

8 Discussion of Results

8.1 Flight Tests Results Analysis

8.1.1 RMSE Analysis

Complexity and RMSE Comparison

The RMSE values for Trajectory 2 were observed in Tab.7.1 to be greater than those for Trajectory 1. This increase can be attributed to the higher complexity of Trajectory 2, which involves more demanding manoeuvres such as ascents, lateral movements, and tilting. These manoeuvres require higher operating frequencies for the motors' low-level controllers, which increases the system's sensitivity to unmodeled dynamics and external disturbances.

Additionally, both trajectories were conducted with a safety cable attached to the platform. The cable's unaccounted weight is added to the platform as soon as it hovers, introducing an element of uncertainty. However, this effect is more pronounced in Trajectory 2, where the platform ascends beyond the hovering height. During the ascent, additional unaccounted weight from the attached USB and power cables increases the overall load, further increasing the uncertainties and contributing to the higher RMSE.

Impact of Unaccounted Weight and Omnidirectionality

Another significant factor affecting performance, particularly in Trajectory 2, is the loss of omnidirectionality due to the increased total weight. According to the convex set of attainable forces and torques, the platform maintains its omnidirectional capabilities as long as the total weight does not exceed 1213 grams. However, the added unaccounted weight from the safety, USB, and power cables, estimated at approximately 500 grams, compromises the platform's omnidirectional control, especially during tilting manoeuvres, leading to degraded performance and higher RMSE.

Yaw Angle Performance and Controller Limitations

The RMSE for the Yaw angle was consistently the highest across both trajectories. This can be attributed to the higher operating frequencies required by the yaw control, combined with the limitations of the low-level controllers used in the experiments. The platform's low-level controllers consisted of two Teensy microcontrollers, each connected to four motors as shown in Fig.A.7 in the appendix. These controllers introduced additional delays in command transmission between the upper and lower rotors, further impacting the yaw performance.

Moreover, the decision to lower the PID gains for the yaw control was made to prevent issues that arose from the higher operating frequencies, which were pushing the limits of the hardware used and introducing unstable behaviour due to the lagging in following the motor commands. This compromise, while necessary to ensure stability, resulted in less precise yaw control, as reflected in the higher RMSE values.

8.1.2 Power Consumption Analysis

General Observations

A clear observation from the power consumption plots shown in Fig.7.8 and Fig.7.7 is that the real-world experiments consistently consume more power compared to the simulations. This discrepancy is evident in both Trajectory 1 and Trajectory 2. Specifically, the real-world experiments for Trajectory 1 show an average power consumption of 32.97 W, while the simulation reports an average of 24.73 W, marking a difference of approximately **33.4%**. Similarly, in Trajectory 2, the real-world average power consumption is 41.06 W, in contrast to 25.36 W in the

simulation, reflecting a more substantial difference of about **61.9%**. These percentages highlight the challenges in accurately simulating real-world conditions, where additional factors such as cable drag, air resistance, and hardware inefficiencies contribute to higher power demands.

Explaining the Higher Power Consumption in Real-World Tests

The higher power consumption observed in the real-world experiments compared to simulations can largely be attributed to the added weight and uncertainties inherent in the physical setup. In reality, the platform has to contend with additional unaccounted weight from the safety cables, USB, and power cables, which are attached from below as shown in Fig.7.1a. This extra weight increases the overall load on the system, requiring the motors to operate at higher speeds to maintain stability and control, thereby consuming more power.

This effect is more pronounced in Trajectory 2 due to its complexity. The larger added weight from the cables during the ascent phase of Trajectory 2 introduces significant uncertainties. As the platform ascends, more cable is also added, which adds more weight that the system needs to lift. This added weight leads to a higher demand on the motors, causing them to operate at even higher speeds as shown in Fig.A.8 in the appendix, than in the simulated case.

8.1.3 Controller Signal Behavior Analysis

In the following analysis, we will primarily focus on the comparison between real and simulated control signals for Trajectory 2. This trajectory is more complex, which results in more significant differences in the control signals between the real and simulated cases.

In the plots shown in Figure 7.9, the colored lines represent the virtual accelerations output from the optimizer, while the black lines represent the acceleration references output from the PID controller. Ideally, these virtual accelerations, which correspond to the commanded rotor speeds, should closely follow the acceleration references provided by the PID controller. However, due to the weights applied to different costs in the optimization process and the constraints imposed on the system, the optimizer might find a different feasible solution that does not perfectly match the PID output.

A notable example can be observed in the a_y (lateral acceleration) behavior around $t = 30$ seconds in both the real and simulated cases. The optimizer does not attempt to follow the acceleration reference in the y-axis during this time. This behavior is consistent across both the real and simulated platforms and results in the same error in the y-axis as soon as the platform begins tilting at $t = 30$ seconds. However, the magnitude of this error is larger in the real platform, likely due to the uncertainties discussed earlier, such as unaccounted weight.

The reason the optimizer behaves this way could be attributed to the specific weights used in the cost function, which may not place enough emphasis on minimizing the position tracking error. While increasing the weight on position tracking could improve tracking performance, it would also push the system to a limit where more extensive low-level controller tuning would be required. This additional tuning was beyond the scope of this thesis, as the controllers were already tuned as part of this work. It is possible that the tuning achieved represents the best performance attainable with the tested hardware and firmware setup, though further experiments would be necessary to confirm this.

8.1.4 Improved Hardware for Enhanced Performance

Based on the observed limitations of the low-level controllers during the flight tests, the group has developed an improved hardware configuration. This new setup includes better-tuned controllers and a higher firmware operating frequency, increased by approximately 500 Hz. This enhancement is expected to reduce the delays and improve overall performance, par-

ticularly in the yaw control, by allowing for more precise command execution and better handling of the platform's dynamics.

8.2 Conclusion of Flight Tests Results

The analysis conducted in this chapter directly addresses the research question: "How much does the behavior of the controlled real platform differ from the simulated one in terms of position and attitude errors?" The findings illustrate that there is a notable discrepancy between the simulated and real-world performance of the platform, particularly evident in more complex trajectories such as Trajectory 2.

The RMSE values were higher in the real experiments due to mostly unmodeled dynamics, such as the additional unaccounted weight from cables, which degraded performance. The power consumption analysis further highlighted these discrepancies, with **real-world tests consuming substantially more power**, indicating that the real platform had to work harder to maintain stability and control. The controller signal behavior analysis also revealed that the optimizer, influenced by specific weights in the cost function, did not always match the acceleration references from the PID controller, especially in the lateral direction. This led to position tracking errors, which were more pronounced in the real-world tests due to the aforementioned uncertainties.

These findings underscore the challenges in replicating real-world conditions within simulations, particularly for complex maneuvers. While the simulations provided valuable insights, the real platform's performance was impacted by additional factors that the simulations could not fully capture. This can give future interesting insights on how to improve the simulator based on these uncertainties, which is mostly based on the test setup environment.

CONCLUSION

9 Conclusions and Recommendations

9.1 Conclusions

This thesis explored the *Trajectory Control Performance of the Omnimorph*, an omnidirectional morphing multi-rotor UAV, with a focus on addressing the gap between simulated performance and real-world implementation. By outlining the discrepancies between simulation and reality, the study successfully made the Omnimorph achieve stable flight for the first time in the literature. The conclusions drawn from the research questions are recapped below:

9.1.1 How do design changes impact the feasible sets of forces and torques?

The first research question examined how the mechanical simplifications of the Omnimorph, compared to the fixed propellers omnidirectional platform design proposed by Brescianini and D'Andrea (2016), affected the sets of feasible forces and torques. **The study found that the design simplifications resulted in a 6% reduction in torque generation and a 10% reduction in torque set uniformity.** However, these findings were deemed acceptable by the group, as the simplifications significantly facilitated easier and more reliable implementation of the tilting mechanism. Despite these reductions, the Omnimorph maintained its omnidirectional capabilities, striking a balance between mechanical simplicity and functional performance, which made it more feasible for real-world applications.

9.1.2 How can we mitigate the effect of propeller performance degradation in the closed-loop trajectory-controller system?

The second research question focused on the issue of propeller performance degradation due to aerodynamic interference studied in Bazzana et al. (2024), especially when the propellers are aligned. To address this, **a linear adjustment model was developed, dynamically adjusting the thrust coefficient based on the tilting angle α.** The linear thrust mismatch model improved the simulated platform by bringing it closer to real-world performance, as modelling aerodynamics is non-trivial. Instead of precisely modelling the aerodynamics, the model adjusted the thrust coefficient based on the tilting angle, α. Additionally, the proposed dynamic weighting algorithm played a crucial role in stabilizing the scenario where the optimizer weights were designed for greater energy efficiency (Case B). Initially, Case B failed when the linear model of the thrust mismatch was introduced. However, **the dynamic weighting algorithm helped stabilize the platform even under thrust mismatch conditions, ensuring the Omnimorph could maintain stable flight in simulation.**

9.1.3 What are the behavioral differences between simulated and real platforms?

The third research question investigated the differences between simulated and real-world performance. **The real-world results showed significantly larger errors, ranging from 1 to 2 orders of magnitude higher in terms of root mean square error (RMSE).** These discrepancies were due to unmodeled disturbances in the simulated environment, such as the dynamic weight of the cables and initial high-error conditions caused by the platform starting while hanging in mid-air.

On the real-world side, the primary challenge was tuning the motors of the initial prototype hardware, which were not optimized at the start. After tuning, the motors performed better at higher frequencies, but substantial tracking errors remained when fine-tuning the system across all degrees of freedom, leading to yaw instability. **According to these findings, the group updated the platform with newer hardware and faster firmware, improving the system's low-**

level motor tracking. These findings highlighted the limitations of the original design and the need for further refinement to bridge the gap between simulation and reality.

9.2 Recommendations

9.2.1 Recommendation 01

Conduct more extensive experiments with the improved hardware to systematically address and mitigate the challenges outlined in this study. These experiments should focus on fine-tuning the platform's performance, addressing the remaining discrepancies between the simulated and real-world behaviour, and testing the updated firmware and hardware under diverse operating conditions. Such tests will help further validate the control system's stability and adaptability in real-world environments.

9.2.2 Recommendation 02

Experiment with the dynamic weighting algorithm in real-world scenarios to enhance the platform's robustness and gain a deeper understanding of the optimizer's influence on its performance. Real-world testing of the dynamic weighting could reveal additional insights into how the system adapts to varying conditions, especially in cases of thrust mismatch, and further improve the control strategy by dynamically adjusting energy efficiency versus performance.

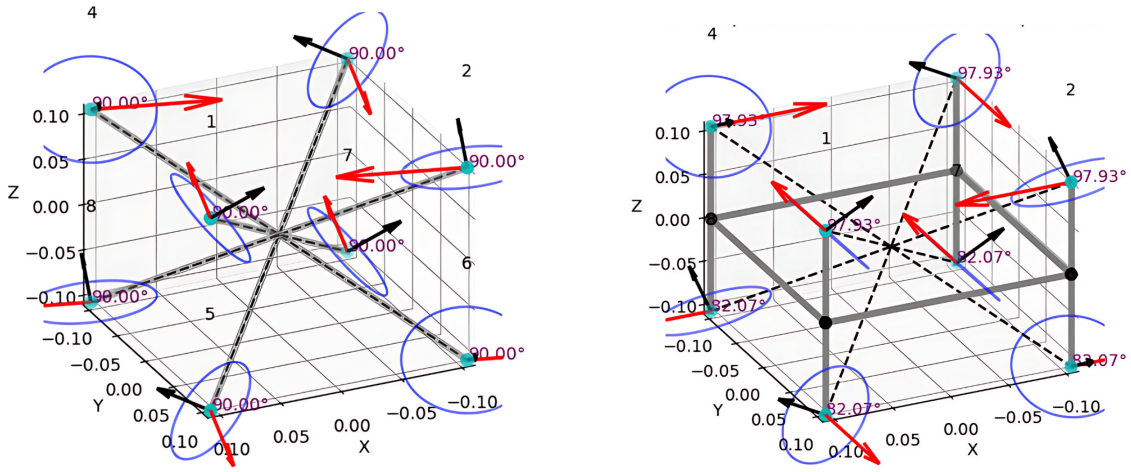
9.2.3 Recommendation 03

Due to the asymmetrical design resulting from the mechanical simplification, it is recommended to **investigate the energy and power consumption under specific configurations where only a subset of motors are operating to sustain the entire platform's weight.** For instance, when the platform is tilted 90° on the roll or pitch axis, only 4 motors will be active to provide upward thrust. Studying these scenarios will offer valuable insights into optimizing power usage in tilted configurations, which may lead to improved energy efficiency and prolonged flight times in real-world operations.

APPENDIX

A Appendix

A.1 Mechanical Simplification



(a) Isometric view of the omnicopter of Brescianini and D'Andrea (2016), showing the direction of the rotor normals and the tilting axes.

(b) Isometric view of the Omnimorph of Aboudorra et al. (2024), showing the direction of the rotor normals and the tilting axes.

Figure A.1: Isometric View of both platforms the omnicopter Brescianini and D'Andrea (2016), and the Omnimorph showing the normals of the rotor discs and their relative tilting axes to achieve morphability.

A.2 Wrench Map

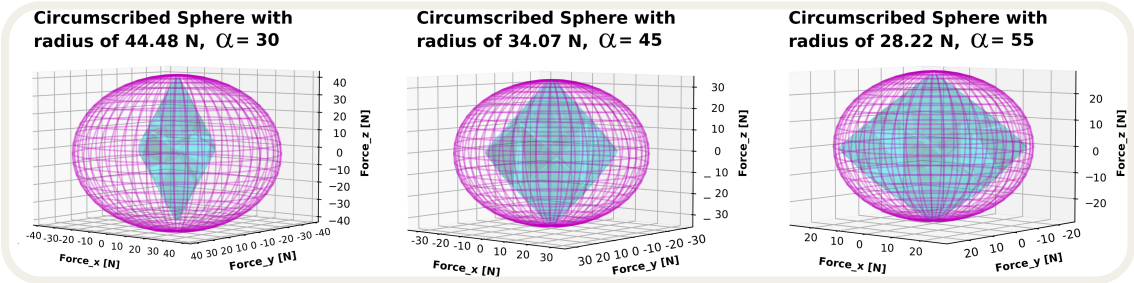


Figure A.2: Set of attainable thrusts for zero torque, which is described by a convex Hull with different circumradius (MDL) for different tilting angle α. With uniformity of **35%, 60%, and 75%** respectively.

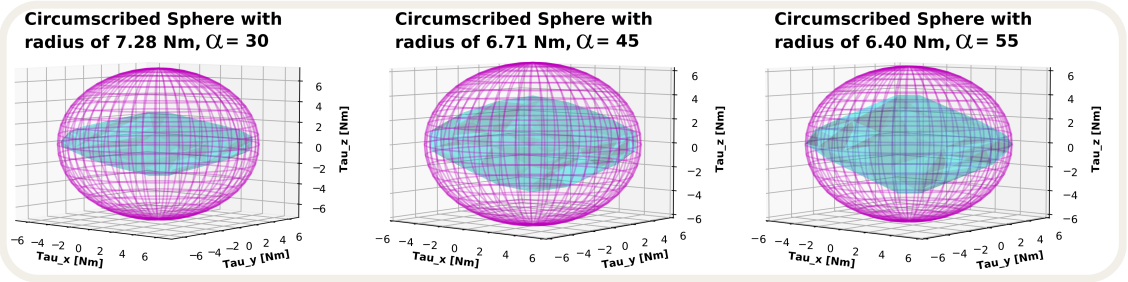


Figure A.3: Set of attainable torques when generating a thrust of magnitude mg in any direction, where m denotes the vehicle's mass and g resembles the gravitational constant, i.e. $g = 9.81 ms^{-2}$. It is described by a convex Hull with different circumradius (MDL) for different tilting angle α. With uniformity of **44%, 60%, and 58%** respectively.

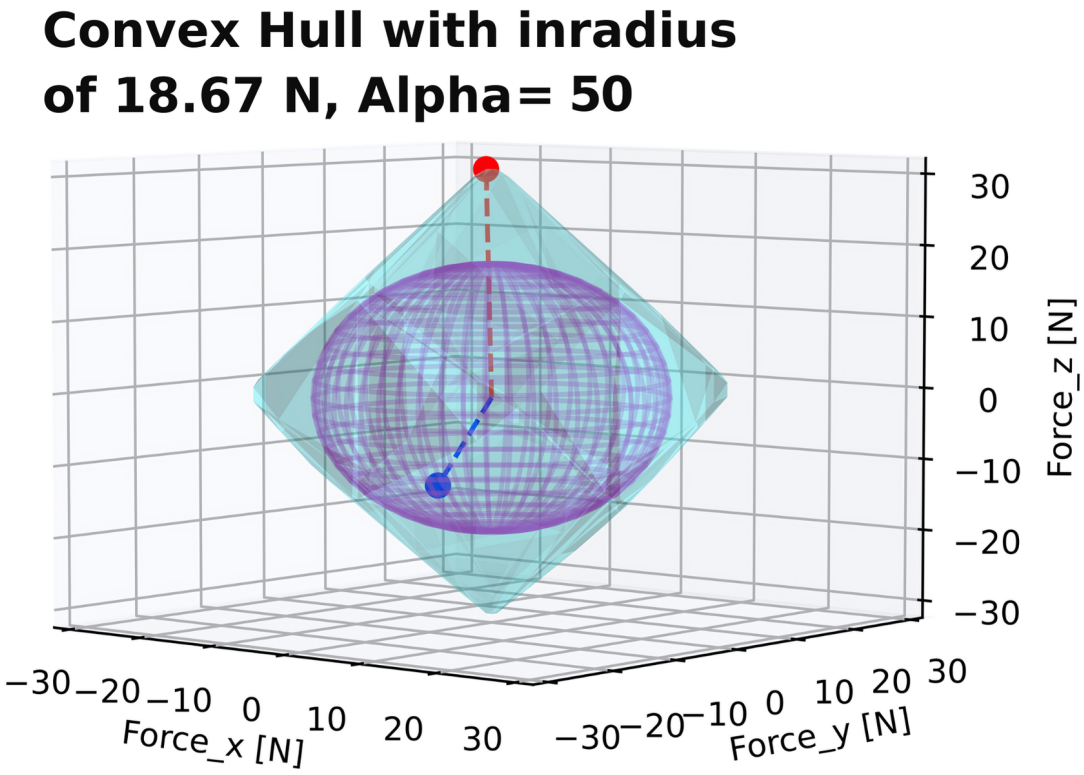
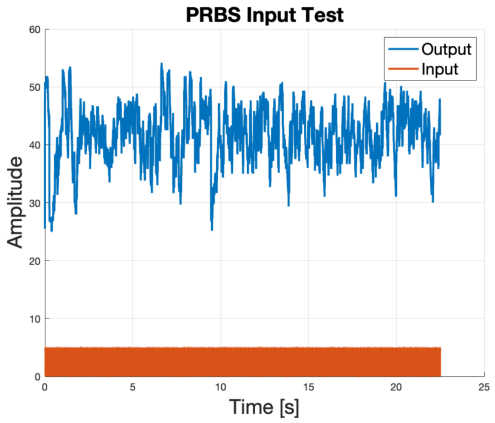


Figure A.4: Convex hull of the force vectors generated by the UAV's propellers. The uniformity of the convex hull is calculated based on the ratio between the furthest point (red point) and the shortest distance (blue point) from the origin to the surface of the hull.

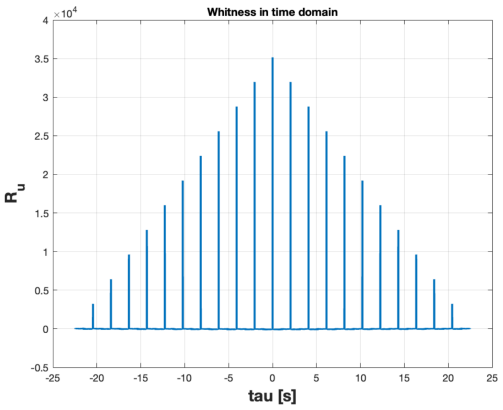
A.3 Motor Tuning

	Timestamp	SumPrevMeanError	SumPrevMaxError	SumCurrMeanError	SumCurrMaxError	IsBetter	PIDN_1	PIDN_2	PIDN_3	PIDN_4
2	13/06/2024 15:44	-0,107509971	81,32661433	-0,247424864	71,12433995	TRUE	0,065	0,234	0,509901951	0,99
3	13/06/2024 15:47	-0,107509971	81,32661433	-0,277483472	72,38644334	TRUE	0,075	0,27	0,547722558	0,99
4	13/06/2024 15:49	-0,107509971	81,32661433	-0,223933488	77,00669634	TRUE	0,1	0,36	0,632455532	0,99
5	13/06/2024 15:51	-0,107509971	81,32661433	-0,116119157	95,42127555	FALSE	0,1	0,36	0	0,99
6	13/06/2024 15:52	-0,107509971	81,32661433	-0,237870772	68,14414025	TRUE	0,1	0,36	1,264911064	0,99
7	13/06/2024 15:54	-0,107509971	81,32661433	-0,323142075	68,17645518	TRUE	0,1	0,36	1,897366596	0,99
8	13/06/2024 15:55	-0,107509971	81,32661433	-0,221839252	71,96805904	TRUE	0,12	0,432	1,385640646	0,99
9	13/06/2024 15:56	-0,107509971	81,32661433	-0,097436993	96,71658005	FALSE	0,1	0,36	1,264911064	0,1
10	13/06/2024 15:57	-0,107509971	81,32661433	-0,195448273	71,10209939	TRUE	0,1	0,4	1,264911064	0,99
11	13/06/2024 15:58	-0,107509971	81,32661433	-0,282442164	67,4066806	TRUE	0,1	0,3	1,264911064	0,99
12	13/06/2024 16:00	-0,107509971	81,32661433	-0,355923023	65,49201374	TRUE	0,1	0,2	1,264911064	0,99
13	13/06/2024 16:01	-0,107509971	81,32661433	-0,493756288	66,39007884	TRUE	0,1	0,1	1,264911064	0,99
14	13/06/2024 16:02	-0,107509971	81,32661433	-0,430035158	65,1832122	TRUE	0,09	0,18	1,2	0,99
15	13/06/2024 16:03	-0,107509971	81,32661433	-0,194987465	70,55212765	TRUE	0,15	0,3	1,549193338	0,99
16	13/06/2024 16:04	-0,107509971	81,32661433	-0,351299884	64,29721182	TRUE	0,15	0,15	1,549193338	0,99
17	13/06/2024 16:11	-0,107509971	81,32661433	-0,591364002	66,04601822	TRUE	0,15	0,075	1,549193338	0,99
18	13/06/2024 16:12	-0,107509971	81,32661433	-0,510888684	65,36574048	TRUE	0,15	0,12	1,549193338	0,99
19	13/06/2024 16:13	-0,107509971	81,32661433	-0,506815861	66,23676549	TRUE	0,2	0,1	1,788854382	0,99

Figure A.5: Generated table after each set of new tuning gains and their results along with the used tuning gains.



(a) Motor open-loop data using PRBS signal excitation.



(b) Autocovariance of Input Data.

Figure A.6: Raw open loop data of the input and outputs of a single motor.

A.4 Hardware

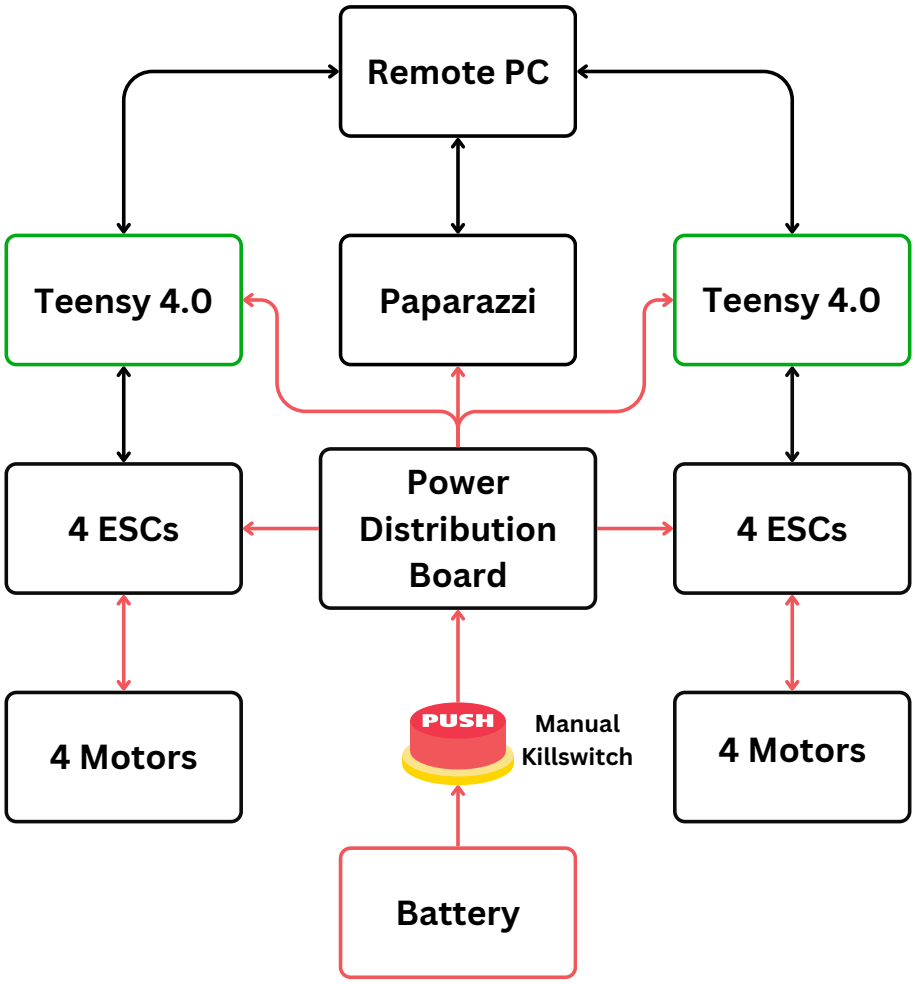
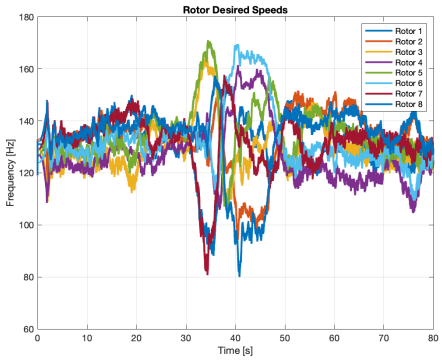
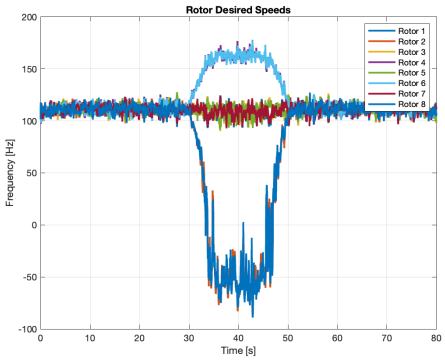


Figure A.7: The Omnimorph hardware's architecture used in this thesis with, data connections in black, and power connections in red

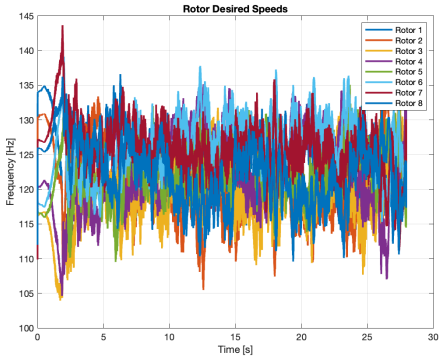
A.5 Flight Tests Results



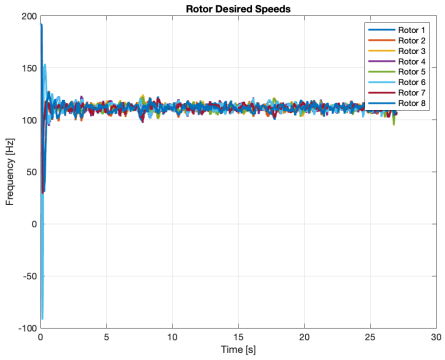
(a) Trajectory 2 (Real): Rotor desired speeds during the entire flight test.



(b) Trajectory 2 (Simulated): Rotor desired speeds during the entire flight test.



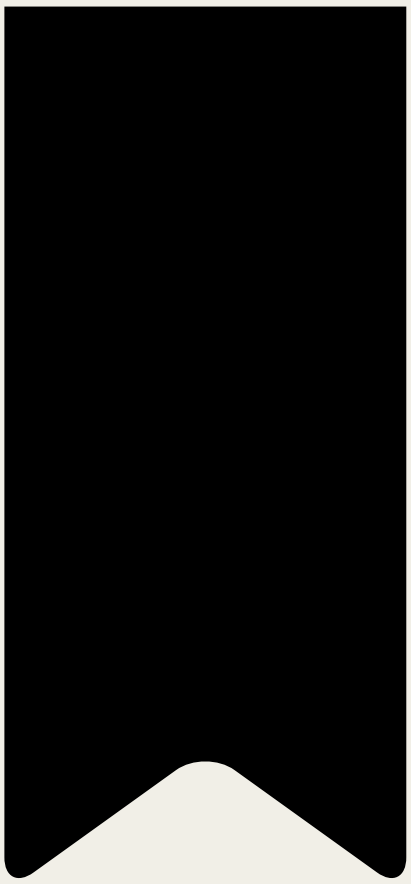
(c) Trajectory 1 (Real): Rotor desired speeds during the entire flight test.



(d) Trajectory 1 (Simulated): Rotor desired speeds during the entire flight test.

Figure A.8: Comparison of desired rotor speeds for both real and simulated experiments across the two trajectories. The plots illustrate the rotor speeds required by the control system to achieve the desired trajectories.

BIBLIOGRAPHY



Bibliography

Aboudorra, Y., C. Gabellieri, R. Brantjes, Q. Sablé and A. Franchi (2024), Modelling, Analysis, and Control of OmniMorph: an Omnidirectional Morphing Multi-rotor UAV, *Journal of Intelligent and Robotic Systems: Theory and Applications*, doi:10.1007/s10846-024-02054-x.

Bazzana, B., R. Brantjes, C. Gabellieri and A. Franchi (2024), An Experimentally Validated Model of the Propeller Force Accounting for Cross Influences on Multi-Rotor Aerial Systems, in *International Conference on Unmanned Aircraft Systems, ICUAS 2024*.

Boyd, S. and L. Vandenberghe (2004), *Convex optimization*, Cambridge University Press.

Brescianini, D. and R. D'Andrea (2016), Design, modeling and control of an omni-directional aerial vehicle, in *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2016-June, ISSN 10504729, doi:10.1109/ICRA.2016.7487497.

Brescianini, D. and R. D'Andrea (2018), An omni-directional multirotor vehicle, *Mechatronics*, **vol. 55**, ISSN 09574158, doi:10.1016/j.mechatronics.2018.08.005.

Drone Industry Insights (2023), Drone Industry Barometer.

Hamandi, M., Q. Sable, M. Tognon and A. Franchi (2021a), Understanding the omnidirectional capability of a generic multi-rotor aerial vehicle, in *AIRPHARO 2021 - 1st AIRPHARO Workshop on Aerial Robotic Systems Physically Interacting with the Environment*, doi:10.1109/AIRPHARO52252.2021.9571051.

Hamandi, M., Q. Sable, M. Tognon and A. Franchi (2021b), Understanding the omnidirectional capability of a generic multi-rotor aerial vehicle, in *2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*, doi:10.1109/AIRPHARO52252.2021.9571051.

Hamandi, M., K. Sawant, M. Tognon and A. Franchi (2020), Omni-Plus-Seven (O7+): An Omnidirectional Aerial Prototype with a Minimal Number of Unidirectional Thrusters, in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, doi:10.1109/ICUAS48674.2020.9214065.

Hamandi, M., F. Usai, Q. Sablé, N. Staub, M. Tognon and A. Franchi (2021c), Design of multirotor aerial vehicles: A taxonomy based on input allocation, *International Journal of Robotics Research*, ISSN 17413176, doi:10.1177/02783649211025998.

Howard, T. D., C. Molter, C. D. Seely and J. Yee (2023), The Lynchpin—A Novel Geometry for Modular, Tangential, Omnidirectional Flight, *SAE International Journal of Aerospace*, doi:10.4271/01-16-03-0018.

Kamel, M., S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart and I. Gilitschenski (2018), The Voliro Omniorientational Hexacopter: An Agile and Maneuverable Tiltable-Rotor Aerial Vehicle, *IEEE Robotics & Automation Magazine*.

Lee, T., M. Leok and N. H. McClamroch (2010), Geometric tracking control of a quadrotor UAV on SE(3), in *49th IEEE Conference on Decision and Control (CDC)*.

Mehendale, N. (2023), Investigating the Battery Life Issues in Unmanned Aerial Vehicles: An Analysis of Challenges and Proposed Solutions, *SSRN Electronic Journal*, doi:10.2139/ssrn.4324196.

Park, S., J. Her, J. Kim and D. Lee (2016), Design, modeling and control of omni-directional aerial robot, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, doi:10.1109/IROS.2016.7759254.

Paul, H., R. R. Martinez, B. Sumetheeprasit and K. Shimonomura (2023), A Tiltable Airframe Multirotor UAV Designed for Omnidirectional Aerial Manipulation, in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*.

Ryll, M., D. Bicego, M. Giurato, M. Lovera and A. Franchi (2022), FAST-Hex - A Morphing Hexarotor: Design, Mechanical Implementation, Control and Experimental Validation, *IEEE/ASME Transactions on Mechatronics*, doi:10.1109/TMECH.2021.3099197.

Ryll, M., H. H. Bülthoff and P. R. Giordano (2015), A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation, *IEEE Transactions on Control Systems Technology*, ISSN 10636536, doi:10.1109/TCST.2014.2330999.

Smeur, E. J., Q. Chu and G. C. De Croon (2016), Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles, *Journal of Guidance, Control, and Dynamics*, doi:10.2514/1.G001490.

Tognon, M. and A. Franchi (2018), Omnidirectional Aerial Vehicles with Unidirectional Thrusters : Analysis , Optimal Design, and Motion Control, *IEEE Robotics and Automation Letters*.

Townsend, A., I. N. Jiya, C. Martinson, D. Bessarabov and R. Gouws (2020), A comprehensive review of energy sources for unmanned aerial vehicles, their shortfalls and opportunities for improvements, doi:10.1016/j.heliyon.2020.e05285.

Zhao, M., T. Anzai, F. Shi, X. Chen, K. Okada and M. Inaba (2018), Design, Modeling, and Control of an Aerial Robot DRAGON: A Dual-Rotor-Embedded Multilink Robot With the Ability of Multi-Degree-of-Freedom Aerial Transformation, *IEEE Robotics and Automation Letters*, doi:10.1109/LRA.2018.2793344.

Zhu, Z., J. Yu, Y. Lin and Y. Zhang (2024), Design, Modeling and Adaptive Robust Control of a Spatial Symmetric Omni-Directional Aerial Vehicle, *IEEE Robotics and Automation Letters*, doi:10.1109/LRA.2024.3396405.