UiT The Arctic University of Norway

Faculty of Engineering Science

Trajectory Tracking with Collision Avoidance for Large Vessel Draft Surveys

A B M Saleheen

Master's thesis in Aerospace Control Engineering STE-3900 May 2024

Project Description

Unmanned aerial vehicles (UAVs) have emerged as invaluable tools in various industries, including marine surveying. This thesis project seeks to adapt the concepts of drone control, trajectory tracking, and collision avoidance, to enhance the draft surveying of large vessels. Draft surveys are critical for assessing vessel stability and cargo loading, and the application of drone technology can lead to more efficient and environmentally friendly methods.

The goal of this thesis project is to design a trajectory-tracking control system for a UAV that will be used for draft surveys of large vessels, integrating collision avoidance mechanisms for enhanced safety, and assessing the system's performance through simulations using available software and experimental tests by hardware implementation.

Subtasks

The main task can be divided into the following subtasks:

- Perform a literature review on the use of UAVs in the context of maritime applications, and surveys and inspections with a focus on relevant research and developments in trajectory tracking and collision avoidance.
- Develop a trajectory tracking guidance and control algorithm that allows the drone to accurately follow predefined survey paths around large vessels.
- Incorporate collision avoidance mechanisms to enhance the safety of drone operations, especially in close proximity to large vessels and potential obstacles.
- Integrate a nonlinear mathematical model of the quadrotor to evaluate the proposed solution using Matlab/Simulink simulations.
- Implement the proposed solution in hardware and perform experimental tests.

Summary

This thesis explores the innovative use of unmanned aerial vehicles to conduct draft surveys of large maritime vessels. This process is critical for determining vessel load through water displacement measurements. At the port of Narvik, a crew typically performs this task manually, where LKAB's iron ore cargo vessels are surveyed while docked. The usual method requires going around the ship in a small boat to check draft markings, which can be difficult, especially in bad weather or at dark times. The close quarters and dangerous conditions often create safety risks and operational difficulties.

The primary aim of this research is to automate the draft survey process by using a self-flying quadrotor that is equipped with an autonomous guidance and control system. This system utilizes Nonlinear Model Predictive Control for precise position control, allowing for optimal trajectory tracking and effective collision avoidance, along with a reduced attitude controller. The goal is for the UAV to autonomously follow a predetermined path around the vessel, and systematically capture images or videos of the draft markings for analysis.

This research presents a theoretical model and simulation framework for implementing the developed system. The find-ings contribute to the field of UAV applications in the maritime sector by proposing an integration of control theory, UAV technology, and maritime operation needs. This study not only paves the way for further techno-logical advancements in autonomous UAV systems but also enhances our understanding of their practical implementations in industry-specific scenarios.

Preface

This master's thesis is submitted in partial fulfillment of the requirements for the Master of Science degree in Aerospace Control Engineering at the Department of Electrical Engineering, UiT - The Arctic University of Norway.

This thesis aims to investigate the practicality of using autonomous quadrotors for performing draft surveys on large maritime vessels. The inspiration for this research stemmed from the requirement to automate draft survey procedures for LKAB's iron ore transport ships in Narvik harbor.

I extend my deepest gratitude to my supervisor, Professor Raymond Kristiansen whose continuous support and insightful guidance have been invaluable. His vast experience in theoretical control systems and academic writing has enriched this research in countless ways. I am also profoundly thankful to my advisor, Lecturer Tom Stian Andersen whose expertise in quadrotor design and control has been crucial to the success of this project. I particularly appreciate his prompt and insightful discussions on the small details, offering valuable suggestions on any project or theoretical topics. Moreover, I am grateful to all the teachers and well-wishers who have guided me in learning new things throughout the course duration.

Finally, I would like to express my respect and gratitude towards my family and good friends who have supported me throughout my academic journey, serving as a source of motivation.

A B M Saleheen
Narvik, May 14, 2024

Contents

List of Figures

1 Introduction

Over the past decade, unmanned aerial vehicles (UAVs), commonly known as drones, have evolved from niche military technology to become essential tools across various industries and sectors of society. Their capacity to operate autonomously, access hard-to-reach areas, and carry out tasks without direct human intervention has resulted in their widespread use for a wide range of applications. Drones have not only improved safety by minimizing human exposure to hazardous environments but also boosted efficiency and accuracy in data collection and operations management. Drones are increasingly used in various fields such as agriculture, for precision farming and pest management; in infrastructure, for inspecting buildings, bridges, and historical monuments; in energy, for monitoring power lines and wind turbines; and in public safety, for search and rescue operations and disaster management. Each application takes advantage of the drone's capacity to collect high-resolution imagery and real-time data. These are essential for making informed decisions and automating complex processes.

Particularly in the maritime industry, drones are revolutionizing traditional practices by providing safer, faster, and more cost-effective alternatives to manual inspections and operations. Conducting draft surveys of large vessels, which traditionally involve significant risks and logistical challenges, can be improved through the use of drone technology. This thesis focuses specifically on the development and application of drone technology for conducting draft surveys on large vessels. A draft survey is a crucial process used to calculate the weight of the cargo carried by a ship by measuring its draft, which is the vertical distance between the waterline and the ship's keel. Traditionally, this has been a labor-intensive task requiring surveyors to take measurements at various points around the ship, often under challenging and risky conditions. The primary objective of this thesis is to explore and enhance the capabilities of drones in automating the draft survey process. This aims to reduce risks, while also improving measurement accuracy and efficiency. The research on the draft survey was initially explored in Muhammad Usman's (2022) thesis [1], which served as a foundation for further developing the project and bringing it one step closer to reality.

1.1 Background and motivation

The port of Narvik, situated in the northern region of Norway, plays a vital role as a passage for the international transportation of iron ore. Its ice-free harbor operates without depth restrictions, enabling year-round, round-the-clock operations. Despite strong local winds during winter, Narvik offers a relatively sheltered environment that is protected from adverse weather conditions. Narvik's harbour infrastructure consists of three piers. The first

two piers are equipped with bollards that can handle a maximum strength of 20 tons, while the third pier supports up to 150 tons. The expansive harbor basin and surrounding fjord can accommodate vessels of various sizes. Typically, visiting ships carrying iron ore has a tonnage of around 75,000 and measure approximately 200 meters in length [2].

Iron ore from LKAB, a high-tech, international mining company and a leading producer of iron ore products for the steel industry, is transported to Narvik via the Ofotbanen railway from mining sites in Kiruna, Malmberget, and Svappavaara in Northern Sweden. It then continues its journey to other global destinations by sea [3].

Figure 1: Narvik harbour[2]

1.1.1 Importance of Cargo Weight Measurement

Accurately determining the weight of the cargo is crucial in the maritime sector, not only for maintaining the ship's equilibrium and safety but also for meeting commercial and regulatory requirements. Typically, a vessel's cargo weight is estimated from its draft - which refers to the vertical distance between the waterline and the keel of the ship.

Figure 2: Draft scale painted on a ship [4]

Marine surveyors can gauge this by measuring how deeply a ship sits in water, enabling them to calculate cargo weight based on known water displacement volume and density. As more cargo is loaded onto the vessel, it sinks deeper into the water as illustrated in Figure 3.

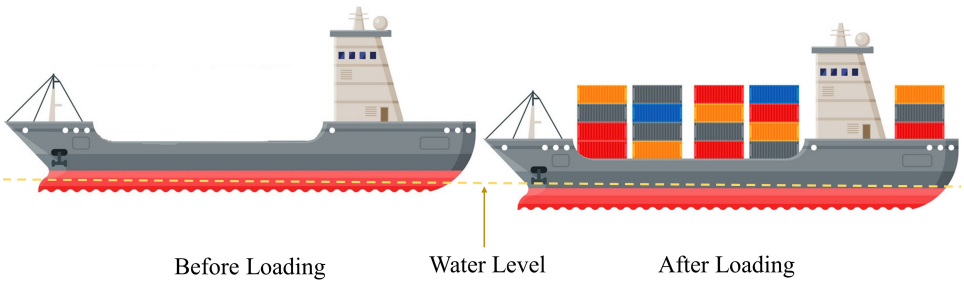


Figure 3: Waterline rising after loading cargo on a vessel

1.1.2 Traditional Methods of Measuring Draft

Traditionally, the draft of a ship is measured manually. This process involves marine surveyors or ship officers who physically measure the draft marks located at six points around the ship's hull: both sides of the bow, midship, and stern. The survey on the port side can be done standing from the dock area whereas accessing the marks on the starboard side often requires using a small boat, particularly for large vessels, which poses significant safety risks - especially in adverse weather conditions or busy shipping lanes.

Figure 4: Draft survey of a large vessel [5]

1.1.3 Challenges with Manual Draft Measurement

Several challenges impair the efficiency and safety of manual draft measurements:

1. **Manual Labour and Associated Risks:** Surveyors performing manual draft measurement must physically approach different parts of a ship's hull, often using small boats. This exposes them to various risks such as falling into the water, collisions with the ship or other objects in the water, and injuries from operating in confined and potentially unstable conditions. Moreover, this process is time-consuming and inefficient. The repetitive nature of these tasks can lead to physical strain, fatigue, increased risk of accidents, and impact the accuracy of measurements.

2. **Adverse Conditions:** In Narvik, being one of the northernmost parts of the world, the dark periods in a day are significantly longer during winter. Measuring draft scales in the darkness increases the risk of errors and accidents, as surveyors must rely on artificial lighting to view and record the draft marks, especially when conducting measurements under less-than-ideal circumstances like bad weather or during night time operations while the ship is still partially loaded or being unloaded.

3. **Visibility of Draft Marks:** Factors such as rust, wear-and-tear of paint, and the accumulation of marine growth or faded layers can compromise the visibility of draft marks on the ship's hull over time. This might require the crew to get closer to the draft or take a zoomed view of it in order to collect data accurately.

Figure 5: Faded paint and rusty draft scale [6]

1.1.4 Motivation for Drone-Assisted Draft Measurement

The risks and challenges of traditional draft surveys make a strong case for developing safer and more accurate technologies. Using drones with high-resolution cameras and automated measurement software offers a transformative solution. Drones can capture images of a ship's waterline safely and efficiently from multiple angles, regardless of environmental conditions. Automating the measurement process not only reduces human error but also minimizes physical risks to surveyors. Moreover, drones can operate in low-light conditions, making draft surveys more flexible and safe at any time of day or night.

1.2 Thesis Objective

The main goal of this thesis is to further develop a guidance and control algorithm designed for a quadrotor, with the ability to perform collision avoidance while tracking a trajectory around a vessel for the photographic documentation of its draft scales. This UAV will be programmed to autonomously follow a predetermined path around maritime vessels docked at the port. The key objective during these missions is to capture high-resolution photographs or videos of the draft scale on the ships.

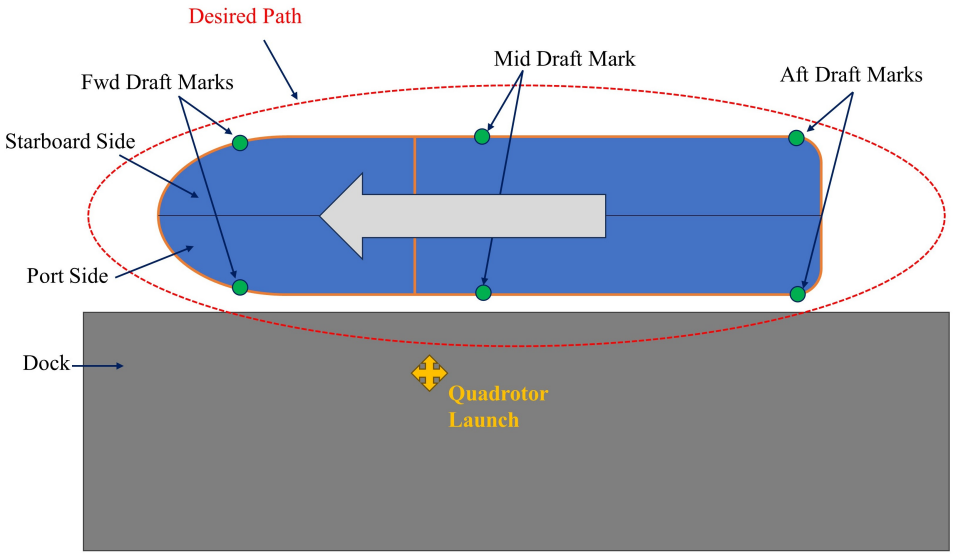


Figure 6: Potential plan for the trajectory tracking quadrotor around a vessel

Each mission segment is planned to ensure the quadrotor navigates safely around the vessel, effectively avoids any obstacles, takes photos or videos of the draft scales, and autonomously returns to its launch point upon completion. The successful implementation of this project is expected to significantly enhance the efficiency and safety of maritime operations by providing accurate and reliable visual documentation of a ship's waterline level without requiring direct human intervention at close quarters.

1.3 Literature review

1.3.1 UAVs in the Maritime Sector

Within the last decade, UAVs have been studied for implementation in the maritime industry. From a study in [7], the maritime industry is one of the expansions of drone applications. In Europe, the sea is used to transport 40% of internal freight and over 90% of foreign freight. Currently, in the maritime sector drones are used for real-time inspections which reduces risks, costs, and time on a number of applications which include hull surveys, inspecting cargo and confined spaces, damage surveys after some incident harbor monitoring, and more.

Entering and carrying out tasks for humans in confined spaces such as ship holds, fuel tanks and ballast tanks is a high-risk operation and according to the International Labour Organization (ILO), it requires a number of precautions and safety measures to be taken.

Figure 7: Manual inspection of a cargo hold in a ship [8]

Access to this kind of site by using drones has been made possible by the Polish Register of Shipping in 2017. There have also been field inspections carried out in the shipyards using UAVs. [7].

According to Det Norske Veritas (DNV) which is a well-known maritime company in Norway, the estimated cost of a single ship survey goes up to $1,000,000 while the ship has to be docked at the same time or 1-2 days which may be the reason to lose another few hundred thousand USD [9]. Det Norske Veritas (Norway) and Germanischer Lloyd (Germany) (DNV-GL) recently started testing autonomous drones with hyperspectral cameras for ship tank inspections. They also launched a project that is known as, autonomous drone-based surveys of ships in operation (ADRASSSO) which is planned to detect cracks using computer vision.

The use of Micro-Aerial Vehicles (MAV) in inspecting ships is a major development in maritime operations, mitigating the high expenses and safety hazards linked to conventional approaches. Bonnin-Pascual et al.'s research [10] presents an adaptable framework that enhances MAVs' abilities for ship inspections. This framework enables MAVs to function as mobile cameras, giving surveyors a virtual view of different areas of a vessel's hull, thereby enhancing inspection efficiency and safety.

The described MAV system utilizes a flexible design that can be adjusted to different sensor setups, and customized for specific operational situations and payload capacities. It features a supervised autonomy (SA) control structure responsible for safety-critical functions like collision avoidance while enabling the human operator to concentrate on inspection tasks through basic motion commands. This dual function lowers operational complexity and enhances the efficiency of the inspection process.

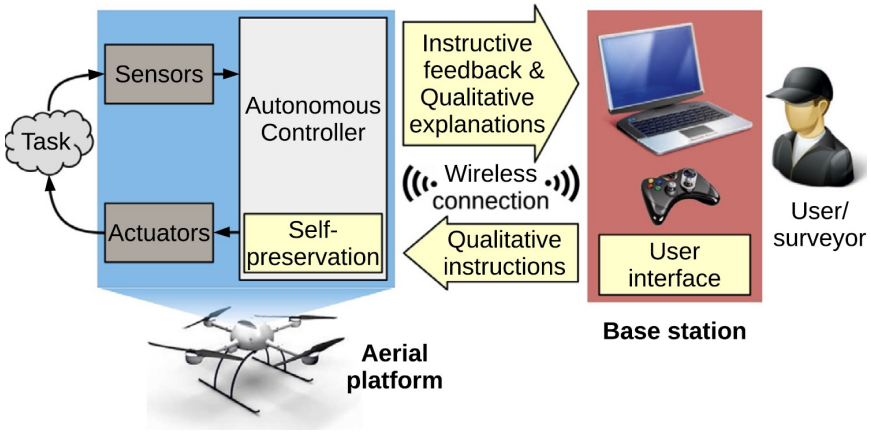


Figure 8: An outline of the inspection system structured based on the concept of Supervised Autonomy[10]

Unlike typical quadrotor structures, their control architecture includes an additional high-level layer, shown in Figure 9. The high-level layer oversees the execution of the MAV behaviors module, which encompasses various robot behaviors. These behaviors work together to send appropriate velocity commands to the middle layer.

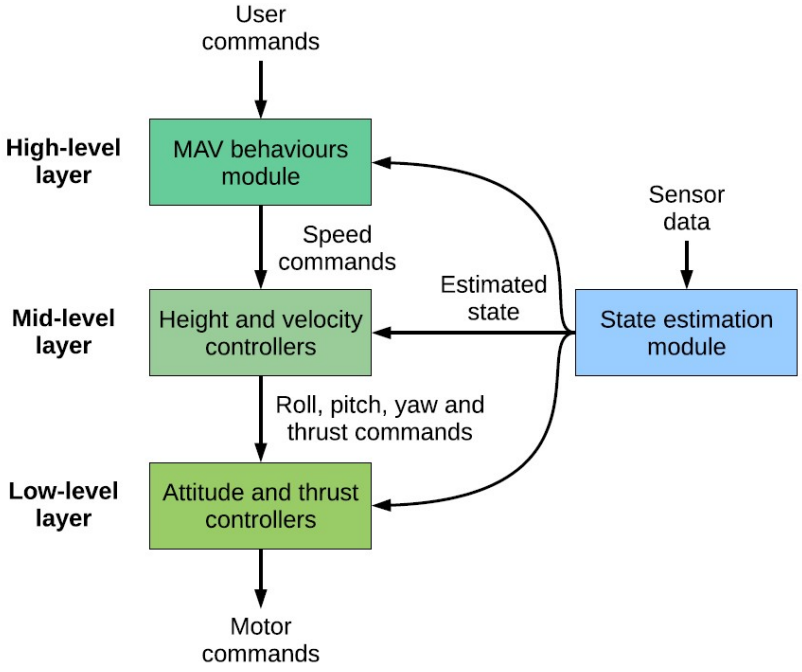


Figure 9: Multi-layered control architecture in conjunction with user commands [10]

Assessments conducted in controlled laboratory settings and actual field examinations confirmed the MAV's effectiveness in navigating the intricate surroundings of ship hulls. The system's flexibility stood out, especially in its capacity to function within confined and poorly lit areas such as ballast tanks and cargo holds, which are usually difficult and unsafe for human inspectors.

This research outlines a significant change in the way vessel inspections are carried out, utilizing advanced robotics to lower risks and expenses while improving the comprehensiveness of inspections. Incorporating MAVs into regular maritime maintenance procedures shows potential for enhancing both maritime safety and operational efficiency.

1.3.2 Application of UAVs in Inspections and Surveys

Danial Waleed et al. worked on drone-based condition monitoring of ceramic insulators on overhead power lines (OHPL) in [11]. It utilizes onboard cameras and a Raspberry Pi for real-time image processing using computer vision, either onboard the drone or at an onshore station. On the onshore computer vision, a regional convolution neural network (R-CNN) is used for object detection whereas the Single-Shot Multibox Detector (SSD) technique is used since this method is computationally less heavy. The system effectively classifies insulators into healthy, dirty, or broken categories using computer vision techniques.

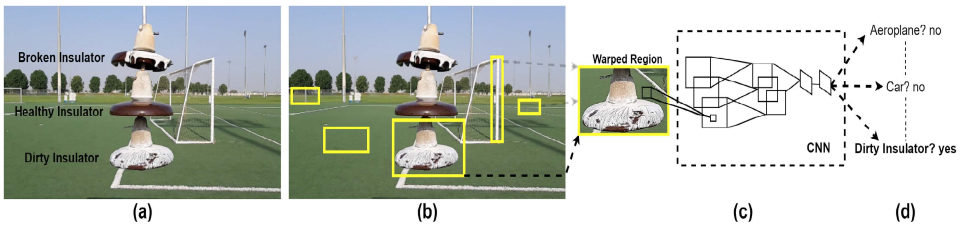


Figure 10: (a) Provide the input picture to the R-CNN algorithm. (b) R-CNN extracting around 2000 regions from the input picture. (c) Classified extraction algorithm (d) Final result. [11]

The research demonstrates that drone-based inspection offers a safer, more efficient alternative to traditional methods, with successful real-life testing that highlights the potential for autonomous operation and real-time detection in power line maintenance.

Multi-sensor aerial trike has been used for the analysis and classification of roof surfaces in [12]. It uses visible images to generate 3D point clouds of roofs using photogrammetric and computer vision algorithms. The thermographic pictures can help determine the best places inside each roof

surface for solar panel installation by estimating areas, tilts, orientations, and obstructions. Additionally, it incorporates solar irradiation estimation, enhancing the potential of UAV-based surveys for energy efficiency in urban planning.

In [13] the potential of UAVs for remote sensing applications has been discussed. Although there is access to multisensor, multitemporal, and multifrequency image data from earth observation satellites, there are limitations like the cost and image quality. Because of their high frequency and high-resolution image-taking capabilities, UAVs have been a topic of interest among researchers. Based on numerous applications of UAVs, including environmental monitoring and large-scale mapping, and the efficiency of UAVs in diverse remote sensing contexts, they represent a framework as shown in Figure 11.

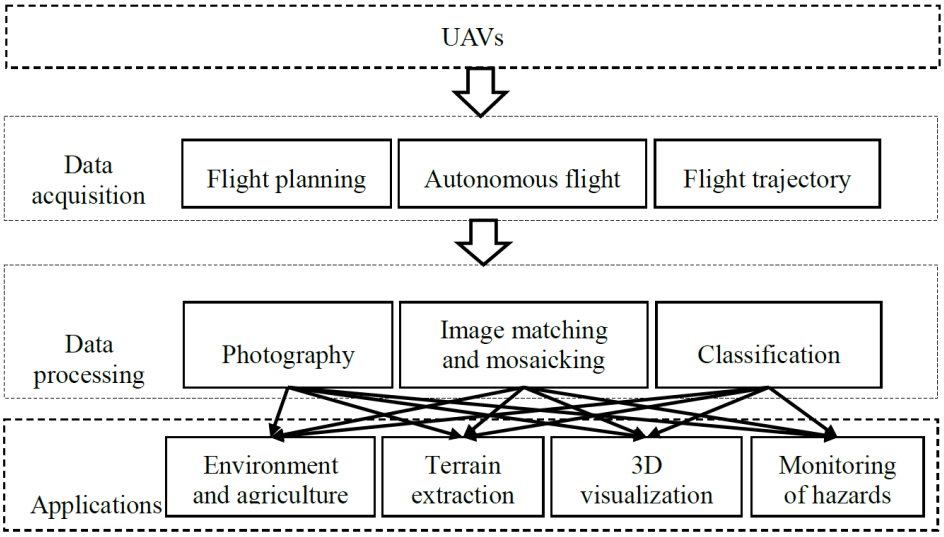


Figure 11: Applications of UAVs in remote sensing [13]

The authors in [14] propose a path-planning method for UAVs equipped with LiDAR for bridge inspection since traditional bridge inspection techniques take a lot of time and could be a dangerous task for humans. It integrates a Genetic Algorithm and A* algorithm to address the Traveling Salesman Problem, focusing on potential defect locations on bridge surfaces. The approach aims to minimize flight time while maximizing defect visibility. This method enhances data collection accuracy and efficiency, underscoring the advantages of UAVs in infrastructure inspection, especially in terms of safety and operational efficacy.

The expansion of power transmission networks has led to the need for

more efficient and automated inspection techniques. Tong He, Yihui Zeng, and Zhuangli Hu [15] addressed this demand by using multi-rotor UAVs to automate detailed inspections of transmission lines. Their study integrates advanced route planning algorithms with UAV technology to improve the safety and efficiency of power line inspections.
The authors have developed a theoretical model for UAV inspections to determine the optimal camera imaging range and establish safety standards for flight operations near transmission lines. The model enables the automatic generation of detailed inspection routes by incorporating critical waypoints essential for assessing the condition of transmission infrastructure. The paper's key contribution lies in the design of an autonomous navigation system capable of conducting detailed inspections by targeting specific components of transmission lines, such as insulators and fittings. Equipped with high-resolution cameras, the UAVs employ waypoint planning to capture precise images from multiple angles, ensuring comprehensive coverage of the transmission towers.

The proposed method's effectiveness was validated through comparative analysis against traditional manual inspections. Findings indicate that the use of UAVs can increase inspection efficiency by approximately 57.98% to 62.88%. Additionally, the positional accuracy of the UAVs during automatic inspections was reported to be within 10 cm, representing a substantial improvement over manual methods. This study not only highlights the potential of UAVs to transform power transmission line inspection but also lays a foundation for further research into autonomous robotic systems in industrial applications.

1.3.3 Trajectory Tracking and Collision Avoidance of UAVs in Experiments and Simulations

This part of the literature study mainly focuses on model predictive controller-based UAV trajectory tracking and collision avoidance systems.

The study in [16] presents strategies for designing and implementing Model Predictive Control (MPC), particularly for multi-rotor systems and fixed-wing UAVs. It emphasizes the use of Robot Operating System (ROS) for the integration and testing of these control strategies. The article also discusses the application of both linear and nonlinear MPC approaches, highlighting their effectiveness in achieving accurate trajectory tracking in UAVs. The Linear MPC was solved by generating a C-code solver using the CVXGEN framework, whereas the Non-linear MPC was solved using ACADO. Both approaches were tested and validated through simulation and experiments.

A real-time model predictive position control system with collision avoidance for commercial low-cost quadrotors was studied in [17]. The paper

introduces a novel MPC approach using a semilinear analytical quadrotor model for position control, integrated with a condensed multiple-shooting continuation generalized minimal residual method (CMSCGMRES). This method is optimized for real-time UAV applications and includes a collision avoidance mechanism based on a sigmoid function, providing an efficient solution for the dynamic control of quadrotors in potentially obstructive environments. The model was validated by applying it to an AR. Drone in two experimental environments with and without obstacles.

The development of a real-time algorithm for Flatness-Based Model Predictive Control (FMPC) for quadrotor trajectory tracking is studied in [18]. The research introduces a unique approach by coupling feedback model predictive control with feedforward linearization, which optimizes both trajectory tracking performance and real-time computational efficiency. They applied their FMPC on the inner loop of a UAV controller that rectified input time delays while the implementation to the outer loop improved trajectory tracking performance. The study demonstrates the algorithm's effectiveness through simulations and real quadrotor experiments, highlighting its potential for precise and efficient UAV trajectory tracking. Overall, the FMPC was found to be more performant than Linear Model Predictive Control (LMPC) and Non-linear Model Predictive Control (NMPC). It also showed more robustness than an MPC+Feedback Linearization (FBL) based controller.

Mohamed Owis and his team developed an NMPC for quadrotor trajectory tracking[19]. The NMPC model is able to incorporate constraints on the states, inputs, and outputs for the desired performance of a trajectory-tracking flight. The research implements this approach using the ACADO toolkit in MATLAB/Simulink for a mathematical simulation. Simulink was interfaced with ROS to simulate a UAV on the RotorS Gazebo simulator. The simulations showed agreeable results.

The article in [20] focuses on a novel NMPC for UAV navigation and obstacle avoidance. The authors present an advanced NMPC scheme that allows real-time solutions for dynamic obstacle avoidance, integrating a classification scheme to predict obstacle shape and trajectory. The Proximal Averaged Newton for Optimal Control (PANOC) which is a nonlinear, non-convex solver, and its associated software OpEn (Optimization Engine) are used for the penalty method so that the obstacles and other constraints are considered properly. The approach is validated through multiple laboratory experiments, demonstrating its effectiveness in dynamic environments and its potential for real-time, agile collision avoidance in UAV operations. The method is limited by its dependence on the classification of obstacles. If the predictive fails or entails a large error, the UAV might get into a collision.

1.4 Contributions and Delimitations of the Thesis

The previous discussion highlighted the laborious, challenging, and sometimes dangerous nature of manual draft surveys. This thesis proposes a developed solution that utilizes automated UAVs for this purpose. The contributions of this thesis are outlined in this section.

Automating Manual Draft Survey: In contemporary research, quadrotors are widely used in survey and inspection missions. However, their specific use for automated draft measurement of vessels represents a relatively new area of exploration within the field. This thesis aims to expand our understanding of UAV capabilities by focusing on precision navigation and automated operations in close proximity to large ships.

Optimal Control Algorithm and Collision Avoidance: In this study, a Nonlinear Model Predictive Control (NMPC) into the quadrotor's position control system has been integrated. This advanced strategy is designed to optimize trajectory tracking and enable effective collision avoidance, essential for navigating dynamic environments near maritime vessels. Additionally, it introduces a reduced attitude controller focused on orienting the quadrotor toward the desired force vector direction. This streamlined approach enhances stability during critical navigation tasks in survey operations. The developed NMPC and reduced attitude controllers hold potential for applications beyond maritime settings.

Building a Prototype: This thesis makes a pivotal contribution by developing a physical prototype to validate the theoretical models and control strategies discussed earlier. The design specifications are refined from a previous study [1], demonstrating a direct lineage and evolution of research within the field. Although the prototype is currently running manually due to a lack of implementation of the developed system, it could serve as an excellent starting point for future researchers in this field.

Delimitations of the Study

This thesis contains specific boundaries and delimitations related to environmental modeling and physical implementation of the UAV system:

Environmental Factors: While using a quadrotor for draft surveys around vessels presents unique challenges, especially with environmental factors like wind and water spray, this study did not integrate these elements into the simulation environment. Therefore, the control system's robustness and adaptability were assessed without directly considering these maritime environmental factors, which could affect the UAV's performance in real-

world scenarios.

Simulation-Based Assessment: The evaluation of the UAV system's capabilities primarily involved using simulations that employed a nonlinear model of the quadrotor. These simulations provided a controlled environment for testing theoretical constructs and control algorithms, without the unpredictable variables found in physical environments. While the physical construction of the prototype was complete, due to the project's scope and available resources, testing of the UAV was not feasible. Consequently, we were unable to practically apply or validate the system's performance in actual maritime conditions within this study. These aspects are recommended for future research.

1.5 Methodology

This thesis presents a systematic approach to designing and implementing a quadrotor UAV for performing a draft survey around large vessels. The methodology integrates the development of an advanced control architecture, simulation testing, and prototype building, aiming to evaluate the control algorithms.

A literature review was conducted, specifically focusing on the use of drones for inspections and surveys in various industries. A particular emphasis was placed on different control strategies, with a specific focus on evaluating their effectiveness in trajectory tracking and collision avoidance scenarios. It was found from multiple studies that Model Predictive Control, particularly its nonlinear variant, provides significant advantages when navigating complex operational environments like maritime surveys.

A simulation environment was created using MATLAB Simulink, utilizing well-established mathematical models of quadrotor dynamics. This environment encompasses important factors including aerodynamic forces, moments, rotor dynamics, and models for essential quadrotor sensors. The simulation results have highlighted the potential use of drones for ship draft surveys. Additionally, to improve the clarity of the simulation findings, an animation environment was developed in Blender to visualize the operation of the quadrotor based on the simulation data.

The practical aspect of this research involved building a prototype quadrotor equipped with various navigation and sensory tools, such as GPS, sonar, Lidar sensors, and a camera. The UAV also includes a reliable communication system for real-time data transmission and a dedicated power supply to ensure extended operational capabilities. At present, the prototype is operated manually using a remote controller; however, there are plans to implement autonomous control based on the tested algorithms.

The initial tests have shown promise, but it's crucial to fully implement the NMPC and reduced attitude controllers into the quadrotor's flight

system in order to realize the physical performance. Further testing and iterative refinements of the control algorithms should be conducted to ensure their reliability and effectiveness in real-world maritime survey operations.

The methodologies used in this thesis not only offer a blueprint for future advancements in UAV technology for maritime use but also make a significant contribution to the field of robotic navigation and control systems. Integrating theoretical models with practical implementations provides a comprehensive understanding of the challenges and solutions involved in deploying UAVs for complex industrial activities

2 Preliminaries

This section of the thesis lays the groundwork for the subsequent sections by defining the key notations and technical terms. Understanding these elements is crucial to comprehend the mathematical models and control strategies discussed later. This chapter serves as a reference point for the various symbols and conventions adopted, ensuring consistency and clarity in the presentation of the thesis.

2.1 Theoretical/Technological preliminaries

In this part, we define some fundamental concepts of classical mechanics, including frames of reference, Euler angles, and quaternions.

2.1.1 Reference Frames

When dealing with rigid body mechanics, it is essential to establish reference frames. This section presents and describes the two main coordinate systems used in designing a quadrotor system: the inertial frame, which is expressed as North-East-Down (NED), and the body frame, which is expressed as Forward-Right-Down (FRD).

Let **N** denote a right-hand inertial frame referred to as the North-East-Down frame. This frame is defined by its set of unit vectors $\{\vec{n_1}, \vec{n_2}, \vec{n_3}\}$ oriented towards north, east, and downward directions respectively.

If the quadrotor's **B**-frame is placed at the origin of the **N**- frame, the three axes (x,y,z) of both frames can be aligned. The body frame can be seen as a moving coordinate system, in reference to the static inertial coordinate system or the NED- frame.

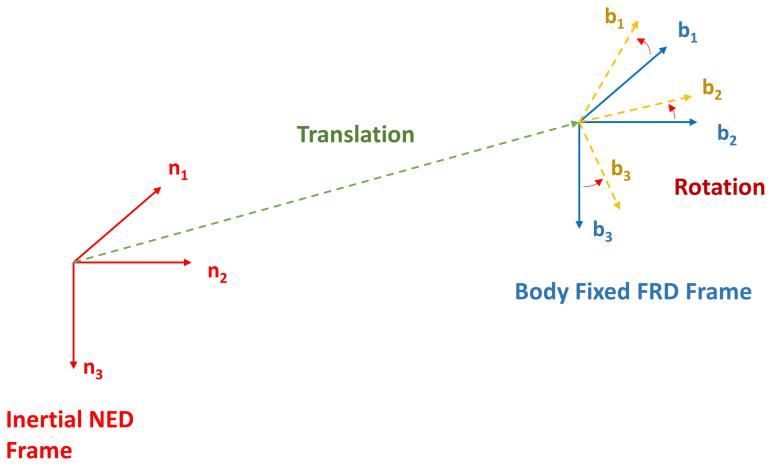


Figure 12: Inertial and Body frame of reference

2.1.2 Euler Angles

Euler angles are a set of three angles to describe the orientation of a rigid body about a fixed coordinate system $(\hat{x}, \hat{y}, \hat{z})$. Usually represented as ϕ, θ, and ψ (roll, pitch, and yaw), these angles denote rotations around the X, Y, and Z axes of a fixed coordinate system. Euler angles offer an intuitive way to understand and visualize the orientation of an object in 3D space. Each

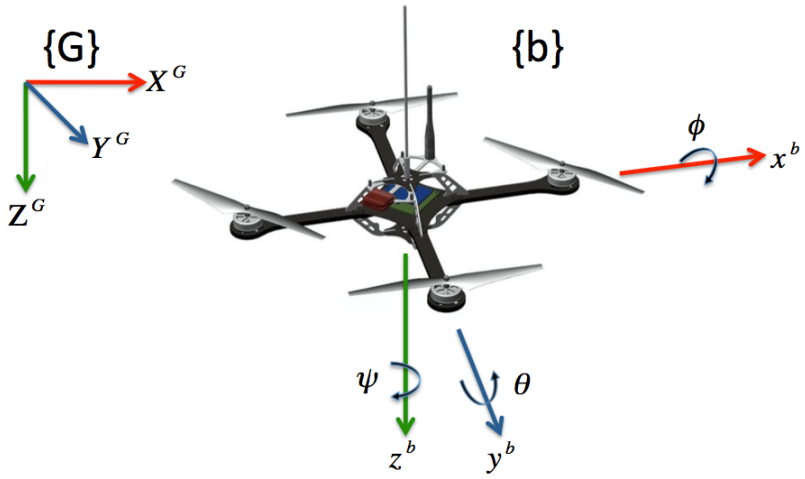


Figure 13: Euler Angles on a quadrotor [21]

angle has a clear physical meaning that corresponds to rotations around principal axes. Despite their advantages, Euler angles have significant limitations. The most well-known issue is gimbal lock, which happens when the axes of two of the three gimbals align, resulting in a loss of one degree of freedom in the angle calculations and making it impossible to determine one of the angles. Given the angular velocity vector,

$$\omega_{nb}^{b} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.1}$$

where p, q, and r are the roll, pitch, and yaw rates, respectively expressed in the body frame. The rate of change of the Euler angles expressed in the inertial frame can then be written as

$$\dot{E} = W(\phi, \theta)\omega_{nb}^{b} \tag{2.2}$$

where $\dot{E}$ represents the derivative of the Euler angles, and W is the transformation matrix defined as

$$\dot{E} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad W(\phi, \theta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}. \tag{2.3}$$

Matrix, W allows the conversion of angular velocities in the body frame to the rate of change of the Euler angles in the inertial frame, facilitating the integration of these rates into the dynamic model of the vehicle or system.

In this thesis, quaternions have been used for mathematical modeling, while Euler angles have been used for state estimation and demonstration.

2.1.3 Quaternions

Quaternions provide a compact and non-singular representation of orientations and rotations in three-dimensional space, making them particularly suited for the computational requirements of aerial robotics. A quaternion is typically expressed as $q = w + xi + yj + zk$, where w, x, y, and z are real numbers, and i, j, and k are the fundamental quaternion units. $\{w\}$ is also called the scalar term whereas $\{x, y, z\}$ represents a vector term. However, to understand how the quaternion is established for a specific rotation we will write it as

$$q = \begin{bmatrix} \eta & \epsilon^\top \end{bmatrix}^\top = \begin{bmatrix} \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^\top \tag{2.4}$$

where,

$$\begin{aligned} \eta &= \cos\frac{\theta}{2} \\ \epsilon &= \boldsymbol{n}\sin\frac{\theta}{2} \end{aligned} \tag{2.5}$$

where θ is the angle of rotation and $\boldsymbol{n}$ is a vector representing the axis of the rotation.

- **Formulation of Quaternion** : To establish the quaternion needed to rotate a vector $\boldsymbol{u}$ to another vector $\boldsymbol{v}$ using quaternions, we must first determine the axis of rotation and the angle through which to rotate. Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be unit vectors along the x-axis and y-axis, respectively as shown in Figure 14.

$$\begin{aligned} \boldsymbol{u} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top \\ \boldsymbol{v} &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top. \end{aligned}$$

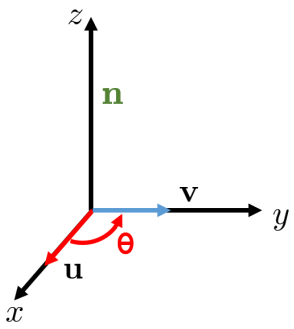


Figure 14: Rotating vector $\boldsymbol{u}$ to align with vector $\boldsymbol{v}$

The steps to find the quaternion representing this rotation are given in a sequence below.

1. **Finding Axis of Rotation**: The axis $\boldsymbol{n}$ around which to rotate $\boldsymbol{u}$ to align it with $\boldsymbol{v}$ can be found using the cross product as

$$\boldsymbol{n} = \boldsymbol{u} \times \boldsymbol{v}$$

   Since in this particular case, $\boldsymbol{u}$ and $\boldsymbol{v}$ are perpendicular and located on the x and y axes, respectively, $\boldsymbol{n}$ will be along the z-axis.

$$\boldsymbol{n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$$

2. **Finding Angle of Rotation**: The angle θ to rotate can be determined using the dot product formula,

$$\boldsymbol{u} \cdot \boldsymbol{v} = |\boldsymbol{u}||\boldsymbol{v}| \cos(\theta).$$

   Given that in our case, $\boldsymbol{u}$ and $\boldsymbol{v}$ are orthogonal, $\theta = \frac{\pi}{2}$ or 90 degrees.

3. **Quaternion Formation**: The quaternion q representing the rotation can now be formed from Equations 2.4 and 2.5.

$$\eta = \cos \frac{\pi}{2 \cdot 2} = 0.707$$
$$\epsilon = \boldsymbol{n} \sin \frac{\pi}{2 \cdot 2} = \begin{bmatrix} 0 & 0 & 0.707 \end{bmatrix}^\top$$

   Thus,

$$q = \begin{bmatrix} \eta & \epsilon \end{bmatrix}^\top = \begin{bmatrix} 0.707 & 0 & 0 & 0.707 \end{bmatrix}^\top$$

Quaternion Operations

To effectively portray the rotational behavior of the quadrotor, it is advantageous to utilize quaternion algebra. Here, we will present a number of essential quaternion operations that enable the representation and control of the quadrotor's orientation and rotational movement. These operations have been used frequently in the simulation.

- ***v_to_q***$(\vec{v})$**:** This operator takes a vector $\vec{v} = \begin{bmatrix}\vec{v}_1 & \vec{v}_2 & \vec{v}_3\end{bmatrix}^T$ as input and outputs the following so that the vector can be used appropriately in quaternion mathematics.

$$\boldsymbol{v\_to\_q}(\vec{v}) = \begin{bmatrix}0 & \vec{v}_1 & \vec{v}_2 & \vec{v}_3\end{bmatrix}^T \tag{2.6}$$

- ***q_to_v***$(\boldsymbol{q})$**:** This operator extracts the vector part of a quaternion structure $\boldsymbol{q} = \begin{bmatrix}q_1 & q_2 & q_3 & q_4\end{bmatrix}^T$.

$$\boldsymbol{q\_to\_v}(\boldsymbol{q}) = \begin{bmatrix}q_2 & q_3 & q_4\end{bmatrix}^T \tag{2.7}$$

- ***q_conj***$(\boldsymbol{q})$**:** This operator outputs the quaternion conjugate of the given quaternion $q = \begin{bmatrix}q_1 & q_2 & q_3 & q_4\end{bmatrix}^T$.

$$\boldsymbol{q\_conj}(\boldsymbol{q}) = \begin{bmatrix}q_1 & -q_2 & -q_3 & -q_4\end{bmatrix}^T \tag{2.8}$$

- ***q_prod***$(\boldsymbol{q})$**:** This operator takes two quaternions as inputs and implements a quaternion product between them. Given two quaternions $\boldsymbol{q}_a = (a_0, \vec{a})$ and $\boldsymbol{q}_b = (b_0, \vec{b})$, where a_0 and b_0 are the scalar parts, and $\vec{a}$ and $\vec{b}$ are the vector parts of $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$ respectively, the quaternion product of $\boldsymbol{q}_a and \boldsymbol{q}_b$ is defined as:

$$\boldsymbol{q\_prod}(\boldsymbol{q}_a, \boldsymbol{q}_b) = \begin{bmatrix} a_0 b_0 - \vec{a} \cdot \vec{b} \\ a_0\vec{b} + b_0\vec{a} + \vec{a} \times \vec{b} \end{bmatrix} \tag{2.9}$$

- ***q_rot***$(\boldsymbol{q_{ab}}, \vec{v})$**:** Given a quaternion $\boldsymbol{q}_{ab}$ and a vector $\vec{v}^b$ expressed in frame **b** , the operator ***q_rot*** performs the rotation of the vector $\vec{v}^b$ from frame **b** to frame **a**.

$$\begin{gathered}\boldsymbol{q\_rot}(\boldsymbol{q}_{ab}, \vec{v}^b) = \vec{v}^a \\ \vec{v}^a = \boldsymbol{q\_to\_v}\left(\boldsymbol{q\_prod}\left(\boldsymbol{q}_{ab}, \boldsymbol{q\_prod}\left(\boldsymbol{v\_to\_q}(\vec{v}^a), \boldsymbol{q\_conj}(\boldsymbol{q}_{ab})\right)\right)\right)\end{gathered} \tag{2.10}$$

2.1.4 Moment of Inertia

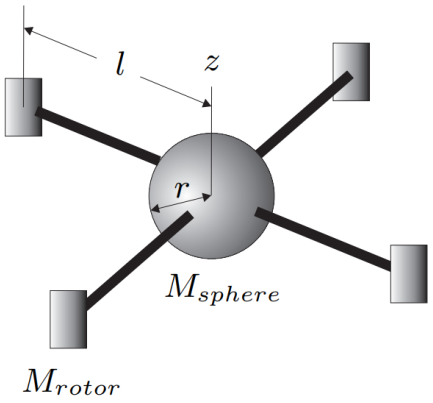


Figure 15: Moment of inertia of a quadrotor [22]

The moment of inertia is a parameter that determines the torque required to produce a specific angular acceleration in an object, or it indicates the extent of difficulty involved in changing the angular velocity of a rotating entity. In simpler terms, it represents how much resistance an object presents to changes in its rotational motion around a specific axis. This property depends on both the distribution of mass within the object and its axis of rotation. Since the quadrotor is a 3D object with three axes of rotation (x, y, z), we can express the moment of inertia matrix as

$$\mathbf{J} = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix} \tag{2.11}$$

The mechanical structure of the quadrotor, as shown in Figure 15, consists of a cross that supports the four motors at the ends of the cross beams and the electronics and battery at the intersection. The constant inertia matrix is essentially symmetric about all three axes; thus $J_{xy} = J_{xz} = J_{yz} = 0$, implying that

$$\mathbf{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \tag{2.12}$$

The moment of inertia can be estimated by making the assumption of a dense spherical core with mass M_{sphere} and radius r, along with point masses M_{rotor} situated at a distance l from the center of the quadrotor body. We

can write the inertia matrix in each of the axes as

$$\begin{aligned} J_{xx} &= \frac{2}{5} M_{sphere} r^2 + 2l^2 M_{rotor} \\ J_{yy} &= \frac{2}{5} M_{sphere} r^2 + 2l^2 M_{rotor} \\ J_{zz} &= \frac{2}{5} M_{sphere} r^2 + 4l^2 M_{rotor} \end{aligned} \tag{2.13}$$

2.2 Notations

This section explains the various types of notations used in the report.

- The axes of a three-dimensional coordinate system are expressed as

$$\begin{aligned} \boldsymbol{e}_1 &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top \\ \boldsymbol{e}_2 &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top \\ \boldsymbol{e}_3 &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top \end{aligned} \tag{2.14}$$

- $\boldsymbol{q}_{ab}^*$ represents the quaternion conjugate of a quaternion $\boldsymbol{q}^{ab}$. Given

$$\boldsymbol{q}_{ab} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^\top,$$

$$\boldsymbol{q}_{ab}^* = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \end{bmatrix}^\top. \tag{2.15}$$

- A quaternion rotation of any vector ($\boldsymbol{v}^b$) from frame **b** to frame **a** is expressed as

$$\boldsymbol{v}^a = (\boldsymbol{q}_{ab} \otimes \boldsymbol{v}^b \otimes \boldsymbol{q}_{ab}^*). \tag{2.16}$$

  This is also referred to as a quaternion sandwich product. A rotation in the reverse direction can be achieved by interchanging the positions of $\boldsymbol{q}_{ab}$ and $\boldsymbol{q}_{ab}^*$.

$$\boldsymbol{v}^b = (\boldsymbol{q}_{ab}^* \otimes \boldsymbol{v}^a \otimes \boldsymbol{q}_{ab}). \tag{2.17}$$

- **I** is used to represent an identity matrix with the diagonal elements being 1,

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.18}$$

- In some cases, $diag(a, b, c)$ expression is used to represent a diagonal matrix as

$$A = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}. \tag{2.19}$$

3 Mathematical Modelling

The first step in this research is to represent a mathematical model of the multi-rotor (quadrotor). Quadrotors have been extensively studied over the past two decades due to their unique capabilities in maneuverability and stability in aerial robotics. They are generally modeled as rigid bodies to simplify the dynamics involved in their flight. Establishing the mathematical model of the quadrotor requires considering the following assumptions:

- It is assumed that the quadcopter's structure is rigid, meaning that its shape and dimensions remain constant during flight.
- Similarly, the propellers are considered to be rigid. This implies that the propellers do not bend, flex, or change shape in response to aerodynamic forces or rotational speeds.
- The mathematical model assumes that the origin of the quadcopter's body-fixed frame coincides with its center of gravity. This simplifies the rotational dynamics by eliminating the need to account for moments generated by gravitational forces acting at a distance from the body frame origin.
- The quadcopter is assumed to be homogeneous in material distribution and symmetrical in geometry. This symmetry simplifies the inertia matrix and ensures uniform behavior across all axes.

3.1 Structural Overview

To model the dynamics of the quadrotor, it is necessary to understand the structure of a quadrotor as well as how it moves in the air creating forces. A quadrotor consists of four rotors, each powered by an individual motor, strategically mounted at the ends of a cross-shaped frame. This configuration enables the quadrotor to execute various flight maneuvers by individually adjusting the speed of each rotor to control thrust and torque, thus influencing the overall orientation and position of the aircraft.

3.1.1 Quadrotor Configurations

Usually, two popular configurations found on quadrotors are the "plus" (+) and the "X" configurations based on the distribution of the rotors along the body forward direction as shown in Figure 16. A quadcopter with an X-configuration is generally seen as more stable than a + configuration, which is known for its acrobatic abilities [23].

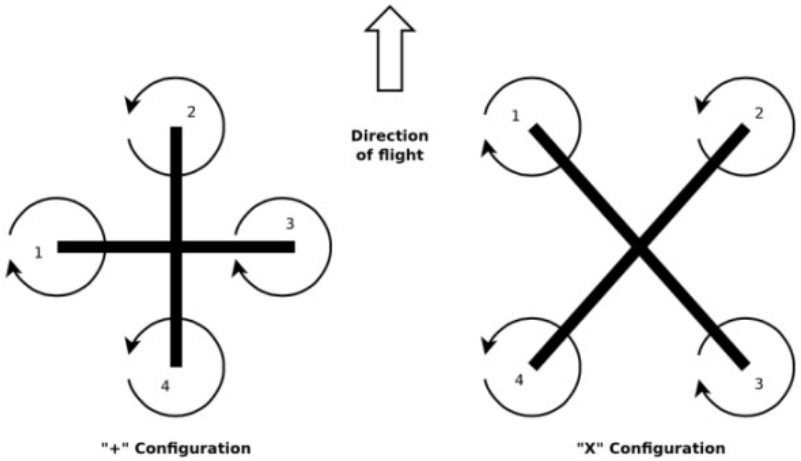


Figure 16: Quadrotor configurations: "+" and "X" [23]

3.1.2 Force Distribution

The body of a quadrotor is often associated with a body-fixed frame, typically centered at the geometric center of the quadrotor. This body frame, also referred to as the body-fixed coordinate system, defines the axes relative to the quadrotor's structure. The x-axis typically points forward from the center, the y-axis horizontally to the right, and the z-axis vertically downwards, adhering to the right-hand rule for coordinate systems.

Any movement of the vehicle is achieved by four desired inputs to the motors which decide their speed. The inputs are,

$\boldsymbol{T}$ - Thrust along the body z-axis.
$\boldsymbol{\tau}_x$ - x component of torque, τ. Responsible for Roll about body x-axis.
$\boldsymbol{\tau}_y$ - y component of torque, τ. Responsible for Pitch about body y-axis.
$\boldsymbol{\tau}_z$ - z component of torque, τ. Responsible for Yaw about body z-axis.

Thrust and torque about the center of the body are generated by the combination and coordination of all the motors. These forces act on the quadrotor to produce the desired motion and control. To explain this, we are going to consider the "X" configuration. As shown in Figure 16 motors, 1 and 3 rotate clockwise (CW) whereas motors, 2 and 4 rotate counterclockwise (CCW) thus creating an even force about the horizontal axes of the quadrotor.

The altitude, or thrust, adjustments are made by uniformly varying the speeds of all four motors.

Lateral movements (roll and pitch) are controlled by asymmetric adjustments. For instance, reducing the rotational speed of motors on the left side of the x-axis while increasing it on the opposite side about the axis of

rotation would roll the quadrotor counterclockwise.

Yaw control, which involves rotating the quadrotor about its vertical axis, is managed by differentially adjusting the speeds to create opposing torques. Specifically, a clockwise (CW) yaw movement is facilitated by decreasing the speeds of the motors rotating clockwise and increasing the speeds of those rotating counterclockwise, thereby generating a net clockwise torque and a corresponding rotational motion.

A quadrotor is considered an underactuated system because it has fewer actuators than degrees of freedom [24]. The system dynamics must control the remaining degrees of freedom corresponding to the translational velocity in the x-y plane. This method of controlling motion through differential motor speeds allows for precise maneuvers and stability in the flight dynamics of the quadrotor.

3.2 Quadrotor Kinematics

Starting with the kinematics of the quadrotor, we need to consider two frames of reference: the body frame **B** and the inertial frame (NED- frame) **N**.
Let $p^n \in \mathbb{R}^3$ be the position of the quadrotor in the inertial **N** frame and $v^n \in \mathbb{R}^3$ be the inertial linear velocity of the quadrotor. Then the translational kinematics of the quadrotor can be written as

$$\dot{p}^n = v^n \tag{3.1}$$

The attitude kinematics of the quadrotor can be expressed in terms of Direction Cosine Matrix (DCM), Euler Angles, or Quaternions. Each of these methods has its advantages and drawbacks. For instance, Euler angles are intuitive but suffer from gimbal lock, DCMs provide a clear geometric interpretation of rotation but are computationally expensive, and quaternions avoid gimbal lock and are more computationally efficient, making them widely preferred in practice. Thus, we are considering the quadrotor kinematics model using quaternion representation,

$$\dot{\boldsymbol{q}}_{nb} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_{1:3} \end{bmatrix} = \frac{1}{2}\boldsymbol{q}_{nb} \otimes p(\omega^b_{nb}) = \begin{bmatrix} -\frac{1}{2}q_{1:3}^\top \omega^b_{nb} \\ \frac{1}{2}(S(q_{1:3}) + q_0 I)\omega^b_{nb} \end{bmatrix}, \tag{3.2}$$

where $\boldsymbol{q}_{nb}$ is a quaternion that represents rotation from body frame **B** to inertial frame **N**, $\omega^b_{nb} \in \mathbb{R}^3$ is the angular velocity of the quadrotor's body frame with respect to the inertial frame, expressed in the body frame, "$\otimes$" represents the quaternion product and $S(q_{1:3})$ denotes the skew-matrix of $q_{1:3}$,

$$S(q_{1:3}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \in \mathbf{SO3} \quad \text{and} \quad p(\omega^b_{nb}) = \begin{bmatrix} 0 \\ \omega^b_{nb} \end{bmatrix}.$$

3.3 Quadrotor Dynamics

Given that the quadrotor is modeled as a rigid body with six degrees of freedom (6-DOF), its dynamics are governed by Newton-Euler equations of motion. Specifically, Newton's second law for translational motion states that the total force, F acting on the quadrotor is equal to the mass m of the quadrotor multiplied by the acceleration, $\dot{v}$ of its center of mass in the body frame, corrected for the inertial forces due to the angular velocity, ω_{nb}^b of the body. This relationship can be written mathematically as

$$F = m(\dot{v}^b + \omega_{nb}^b \times v^b) = f_G^b - f_T^b \tag{3.3}$$

where v^b is the linear velocity and $\dot{v}^b$ is the linear acceleration in the body frame, $m(\omega_{nb}^b \times v^b)$ represents the centrifugal force due to the body's rotational motion, and f_G^b is the gravitational force in the body frame which can be stated in terms of quaternions as

$$f_G^b = (\boldsymbol{q}_{nb}^* \otimes \boldsymbol{f}_G^n \otimes \boldsymbol{q}_{nb}) \quad and \quad f_T^b = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

where T is the thrust along the body's z-axis that the rotors generate, $\boldsymbol{q}_{nb}^*$ is the quaternion conjugate of $\boldsymbol{q}_{nb}$ as shown in (2.15).

$$f_G^n = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

In the inertial frame, the centrifugal force is ignored and the expression becomes,

$$m\dot{v}^n = f_G^n - F_u^n \tag{3.4}$$

$$\dot{v}^b = \frac{1}{m} f_G^n - \frac{1}{m} F_u^n \tag{3.5}$$

where $\boldsymbol{F}_u^n$ is the applied force on the quadrotor in the inertial N-frame,

$$\boldsymbol{F}_u^n = (\boldsymbol{q}_{nb} \otimes f_T^b \otimes \boldsymbol{q}_{nb}^*) \tag{3.6}$$

Similarly, the rotational dynamic of the quadrotor is based on Euler's equation of motion expressed in the body frame as

$$J\dot{\omega}_{nb}^b + (\omega_{nb}^b \times J\omega_{nb}^b) = \tau^b \tag{3.7}$$

where J is the inertial matrix of the quadrotor, $(\omega_{nb}^b \times J\omega_{nb}^b)$ represents the gyroscopic torque, which is the result of the body's rotation. τ^b is the vector of external torques acting on the quadrotor, including those generated by the motor thrust differentials. Finally, we can write the angular acceleration of the quadrotor in the body frame as

$$\dot{\omega}_{nb}^b = J^{-1}(\tau^b - \omega_{nb}^b \times J\omega_{nb}^b). \tag{3.8}$$

3.4 Quadrotor Aerodynamics

Rotor aerodynamics has been a subject of long-standing study. In designing a quadrotor model, it is essential to develop a basic simplified aerodynamics model for the rotor in order to predict the forces and moments generated by the rotors accurately[24].

The mathematical model described in this section follows the study in [24]. The thrust produced by a rotor in a steady hover—that is, without any horizontal or vertical motion—can be effectively modeled through the principles of momentum theory. The equation for the thrust T_i, for a given rotor i is defined as

$$T_i = C_T \rho A_{r_i} r_i^2 {\Omega_i}^2 \tag{3.9}$$

where A_{r_i} denotes the area of the rotor disk , r_i is the radius, ω_i is the angular velocity of the rotor, C_T is the thrust coefficient dependent on the rotor's geometry and airfoil profile, and ρ is the density of air. Instead of the detailed model, a simplified parameter model is often used:

$$T_i = c_T {\Omega_i}^2, \quad C_T = \frac{c_T}{\rho A_{r_i} r_i^2 {\Omega_i}^2} \tag{3.10}$$

where c_T is a positive constant determined experimentally through static thrust tests. This experimental approach to determining the thrust constant is particularly beneficial as it inherently includes the effects of aerodynamic drag induced by the rotor flow on the airframe.

The reaction torque, which results from rotor drag on the airframe during hover conditions in unobstructed air, can be analytically described as per the momentum theory references. The torque Q_i for rotor i is represented by,

$$Q_i = c_Q {\Omega_i}^2 \tag{3.11}$$

where c_Q is a coefficient that encapsulates the effects of rotor disk area A_{r_i}, rotor radius r_i, and air density ρ. This coefficient can be experimentally determined through static thrust tests that inherently account for both thrust and drag influences due to rotor airflow.

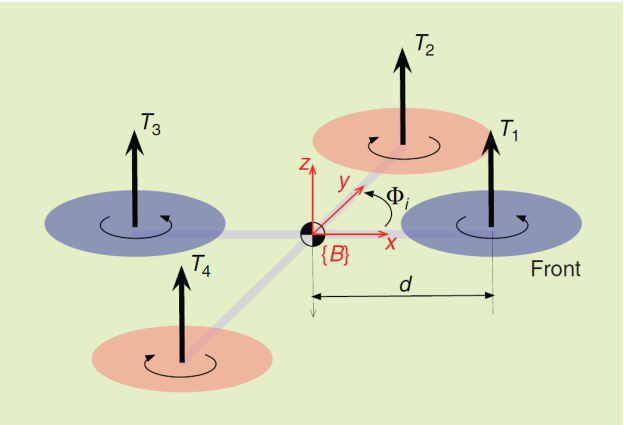


Figure 17: Notation for quadrotor equations of motion. N= 4; Φ_i is a multiple of $\pi/4$ [24]

For an airframe equipped with N rotors, the rotors are indexed as $i \in \{1, 2, \ldots, N\}$ and are arranged in a counterclockwise sequence around the airframe's center. The first rotor is positioned along the positive x-axis of the vehicle's body-fixed frame, indicating the frontal direction, as depicted in Figure 17. Each rotor is connected to the airframe via a support arm, which forms an angle Φ_i with the body-fixed frame's x-axis. The radial distance from the vehicle's central axis to each rotor is denoted as d. Additionally, the rotation direction of each rotor is indicated by σ_i, where $+1$ represents clockwise rotation and -1 represents counterclockwise rotation. The most common configuration involves an even number N of rotors placed symmetrically about the vehicle's axis, with adjacent rotors rotating in opposite directions.

The total thrust acting on the airframe during hover (T_Σ) is the combined result of the thrust generated by each individual rotor.

$$T_\Sigma = \sum_{i=1}^{N} |T_i| = c_T \left(\sum_{i=1}^{N} {\Omega_i}^2 \right) \tag{3.12}$$

The exogenous force in the body frame can be expressed as,

$$\boldsymbol{F} = T_\Sigma \vec{\boldsymbol{b}}_3 + \Delta \tag{3.13}$$

where the primary contribution is from the total thrust during hover (T_Σ), while Δ accounts for all secondary aerodynamic forces. These additional forces become significant when the rotor deviates from a pure hover state, due to dynamic adjustments or environmental interactions affecting stability and thrust efficiency. The unit vector $\vec{\boldsymbol{b}}_3$ represents the direction of primary thrust along the z-axis of the body frame.

The collective moment generated by the aerodynamic forces—which include the forces from the rotors and air resistance—acting on an N-rotor vehicle is represented by the vector $\tau = (\tau_1, \tau_2, \tau_3)$. The components of this moment are given by

$$\begin{aligned}
\tau_1 &= c_T \sum_{i=1}^{N} d_i \sin(\Phi_i) {\Omega_i}^2, \\
\tau_2 &= -c_T \sum_{i=1}^{N} d_i \cos(\Phi_i) {\Omega_i}^2, \\
\tau_3 &= c_Q \sum_{i=1}^{N} \sigma_i {\Omega_i}^2.
\end{aligned} \tag{3.14}$$

For the specified quadrotor configuration, we can succinctly express the total thrust and moment vectors as a matrix based on Equations (3.14).

$$\begin{bmatrix} T_\Sigma \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \underbrace{\begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & -dc_T & 0 & dc_T \\ dc_T & 0 & -dc_T & 0 \\ c_Q & -c_Q & c_Q & -c_Q \end{bmatrix}}_{\Gamma} \begin{bmatrix} {\Omega_1}^2 \\ {\Omega_2}^2 \\ {\Omega_3}^2 \\ {\Omega_4}^2 \end{bmatrix}$$

Considering we are provided with the desired thrust and moments, the rotor speeds can be solved using the inverse of the constant matrix Γ.

3.5 Associated Effects on Rotors

Many aerodynamic and gyroscopic phenomena related to rotors can affect the force model mentioned in the previous section. While most of these influences result in minor disturbances that are not relevant for a robotic system, blade flapping, and induced drag are fundamental effects important for understanding the stability of quadrotors. These effects are especially significant as they result in forces within the quadrotor's horizontal rotor plane (x-y plane), which present underactuated dynamics and are difficult to control using high-gain control methods.

Allowing the rotor to bend is crucial in quadrotor mechanical design. Using overly rigid rotors can lead to aerodynamic forces being directly transmitted to the rotor hub, possibly causing motor mounting or airframe failure. However, small vehicle rotors are more rigid compared to applied aerodynamic forces than full-scale copter rotors. The flexing of the rotors causes blade-flapping effects, while induced drag results from rotor rigidity primarily [24].

3.5.1 Induced Drag

The cumulative induced drag, d_{ind} is the sum of the individual thrust T_i, the induced drag coefficient D, and the ith rotor's velocity vector $\vec{v}_i^b$ considering that the vehicle is moving along the horizontal plane [24].

$$d_{ind} = \sum_{i=1}^{4} T_i D \vec{v}_i^b, \tag{3.15}$$

$$D = \begin{bmatrix} d_x & 0 & 0 \\ 0 & d_y & 0 \\ 0 & 0 & d_z \end{bmatrix}, \quad \vec{v}_i^b = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \boldsymbol{q}_{nb}^* \otimes \vec{v}^n \otimes \boldsymbol{q}_{nb} + \omega_{nb}^b \times \vec{p}_i^b \tag{3.16}$$

where $\vec{p}_i^b$ is the position of the ith propeller with respect to the center of gravity (CG) and T_i is the individual thrust of each propeller which can be calculated from (3.10). Position of all the rotors, $\vec{p}^b$ can be expressed as a matrix,

$$\vec{p}^b = \begin{bmatrix} d & 0 & -h \\ 0 & d & -h \\ -d & 0 & -h \\ 0 & -d & -h \end{bmatrix} \tag{3.17}$$

where h is the distance from CG to the rotor along the body z-axis in the FRD frame.

3.5.2 Blade Flapping

The force generated from the blade flapping by each propeller is found based on the study in [25].

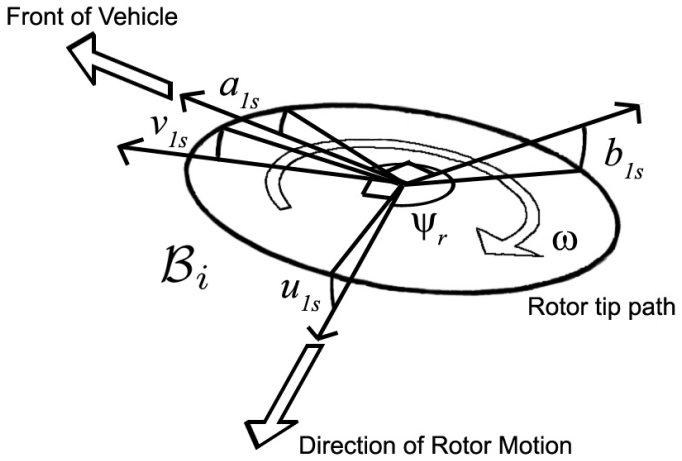


Figure 18: Blade flapping angle rotation [25]

The longitudinal and lateral flapping angles are calculated in the rotor frame (u_{1si} and v_{1si}) and then re-expressed in the body-fixed frame (a_{1si} and b_{1si}) using a rotation matrix realizable from Figure 18.

$$\vec{f}_{t_i}^{b} = T_i \begin{bmatrix} -\sin a_{1si} \\ \cos a_{1si} \sin b_{1si} \\ -\cos a_{1si} \cos b_{1si} \end{bmatrix} \tag{3.18}$$

$$\begin{aligned} a_{1si} &= u_{1si}\cos(\psi_i) - v_{1si}\sin(\psi_i) - \frac{16}{\gamma}\frac{q}{\Omega_i} + \frac{p}{\Omega_i} \\ b_{1si} &= u_{1si}\sin(\psi_i) + v_{1si}\cos(\psi_i) - \frac{16}{\gamma}\frac{p}{\Omega_i} + \frac{q}{\Omega_i} \end{aligned} \tag{3.19}$$

where the other parameters are defined depending on the shape and type of the propeller being used. The longitudinal and lateral flapping angle solutions of the ith rotor in the body frame **B**.

$$\begin{aligned} u_{1si} &= 2\mu_i(\frac{4}{3}\theta_t - \lambda) \\ v_{1si} &= \frac{4}{3}\mu_i\frac{\gamma}{8}(\theta_t - \frac{4}{3}\lambda) \end{aligned} \tag{3.20}$$

where λ is the non-dimensionalized inflow of the ith rotor, γ is the Lock Number.

$$\begin{aligned} \lambda = \sqrt{\frac{C_T}{2}}, \quad \gamma = \frac{\rho a_0 c_0 r^4}{I_b}, \quad \mu_i = \frac{\|\vec{v}_i^{b}\|}{\Omega_i r} \\ \psi_i = \arctan\frac{v_y}{v_x}, \quad \theta_t = \frac{2}{3}\theta_0 + \frac{1}{2}\theta_1 \end{aligned} \tag{3.21}$$

where a_0 and c_0 are the lift slope at the set point and chord of the propeller, and I_b is the propeller's inertia. θ_0 and θ_1 are root blade pitch and blade twist respectively.

3.5.3 Pitch and Roll Rotor Damping:

The pitch and roll damping effect on a rotor was also studied in [25]. A quadrotor usually has a horizontal displacement between its masts and CG. During the rolling or pitching of the quadrotor, the rotors experience a vertical velocity, which leads to a change in the inflow angle. The thrust coefficient C_T can be written in relation to the vertical velocity, V_c as

$$\frac{C_T}{\sigma} = \frac{a(\alpha)}{4}\left[\theta_{tip} - \frac{v_i + V_c}{\Omega_i r}\right] \tag{3.22}$$

where a represents the polar lift slope, θ_{tip} denotes the geometric blade angle at the tip of the rotor, v_i stands for the induced velocity through the rotor, and σ signifies the solidity of the disc, which is defined as ratio

between surface area of blades and rotor disc area. However, this relation can become highly nonlinear; hence it can be expressed as a variation around a setpoint denoted by C_{T0}.

$$C_{Ti} = C_{T0} + \Delta C_{Ti} \tag{3.23}$$

where ΔC_T is the change that is produced by altering inflow conditions. Based on (3.22) we can write,

$$\Delta C_{Ti} = \frac{-a_0}{4} \frac{\sigma}{\Omega_i r} \left(v^b + \omega_{nb}^b \times \bar{p}_i^b \right) \tag{3.24}$$

4 Main result

This section presents the proposed algorithms for the quadrotor's Guidance, Navigation, and Control (GNC) stack, specifically for trajectory tracking around a large vessel, with collision avoidance.

4.1 Control

Controlling aerial vehicles presents a significant challenge for several reasons. Firstly, most aerial vehicles have limited control capabilities, requiring control designs that utilize the interactions between dynamic states to navigate the vehicle. Secondly, these vehicles rely on aerodynamic effects for thrust and lift generation, which introduces unavoidable approximations in force regulation and leads to substantial modeling inaccuracies. Thirdly, external factors such as wind, turbulence, and vortex generation result in high levels of disturbance within the control systems. Finally, direct measurement of both the vehicle and aerodynamic state is often impractical or unfeasible necessitating the use of an observer or controller design based on fundamental principles instead of relying solely on explicit measurements. Given these challenges, a comprehensive control system is necessary to ensure stable and precise trajectory tracking of a quadrotor [26].

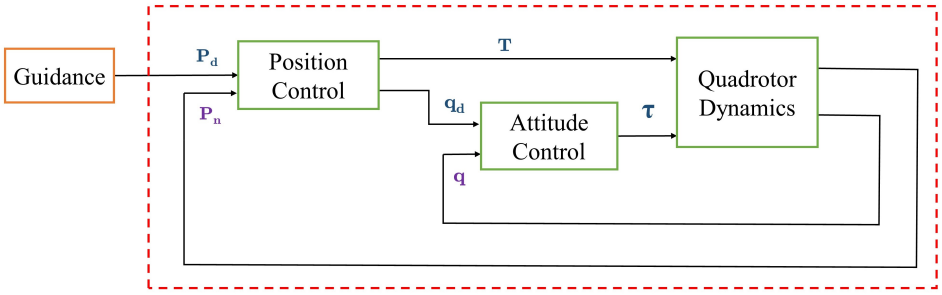


Figure 19: Typical quadrotor control architecture

In the context of quadrotor control, the typical configuration employed is a cascaded or hierarchical control structure as shown in Figure fig:controlFlow. This arrangement consists of two primary control loops: the outer loop, which is dedicated to position control, and the inner loop, which manages the attitude control. This methodology leverages the principle of time-scale separation, where the inner loop operates at a significantly higher speed than the outer loop, effectively optimizing the response and stability of the control system [27].
The positional dynamics of a quadcopter are defined along three axes: altitude (Z-axis), longitude (X-axis), and latitude (Y-axis). Given that a quadcopter is an under-actuated system, it is not feasible to control all six

degrees of freedom (DOF) directly. To effectively manage this limitation, the control architecture is divided into two loops. Control over the X and Y axes is collectively called horizontal control which is indirectly controlled by the outer loop by conveying the desired orientation (q) to the attitude controller. In contrast, the control over the Z axis is referred to as thrust control, which is managed within the outer loop of the control structure.

4.1.1 Position Control

In our case, the position controller is designed considering that the quadrotor's translational dynamics are fully actuated. This results in a fully actuated force F_u^n. The magnitude of this force $||\vec{F}_u^n||$ is then used as commanded thrust (T) for the quadrotor, and the normalized force vector $\frac{\vec{F}_u^n}{||\vec{F}_u^n||}$ is sent to the attitude controller after saturation.

Designing the position controller for a trajectory-tracking quadrotor around a large vessel comes with challenges such as dealing with surrounding wind disturbances, unpredictable movements of the ship, and various obstacles. Employing a Model Predictive Control (MPC) approach offers significant advantages, especially in scenarios that demand high precision and safety. MPC is chosen due to its ability to anticipate future states and make informed control decisions, which is critical for dynamic and potentially unpredictable environments. This control method is widely used in quadrotor control design due to its ability to handle constraints, making it adept at collision avoidance. This feature enables MPC to maintain the quadrotor's stability and track desired trajectories while also proactively adjusting its path to avoid obstacles. Such preemptive adjustment is essential for navigating through cluttered or dynamically changing environments, ensuring safer operation of the quadrotor. A non-linear MPC has been implemented in this thesis. This will be discussed in the following section.

Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) is an advanced approach to control, which involves solving real-time optimization problems in order to ascertain the most suitable control action. This method is especially advantageous for systems with nonlinear dynamics and constraints, as it builds upon the principles of linear MPC while addressing nonlinearities within the model, resulting in more precise and resilient control.

NMPC is based on predicting future system behavior over a limited time period using a nonlinear dynamic model and associated constraints. This methodology is often referred to as receding horizon control because, at each time step, it recalculates the system's behavior or states over an upcoming, finite interval.

At each sampling instant, an optimization problem is solved to minimize a cost function, which typically reflects the performance objectives and penalizes deviations from desired outcomes.

NMPC Problem Formulation

The formulation of the NMPC starts with the formulation of an Optimal Control Problem (OCP). The dynamic model of the system model is expressed in the nonlinear state space form as

$$\dot{x}(t) = f(x(t), u(t)) \tag{4.1}$$

Based on the Equations 3.1 and 3.5, the dynamic system model for the NMPC formulation in the position controller can be written as

$$\begin{bmatrix} \dot{p}^n \\ \dot{v}^n \end{bmatrix} = \begin{bmatrix} \dot{p}_x^n \\ \dot{p}_y^n \\ \dot{p}_z^n \\ \dot{v}_x^n \\ \dot{v}_y^n \\ \dot{v}_z^n \end{bmatrix} = \begin{bmatrix} v_x^n \\ v_y^n \\ v_z^n \\ a_x^n \\ a_y^n \\ a_z^n \end{bmatrix} \tag{4.2}$$

The system model can be represented in the Euler discretized form with a sampling time of ΔT,

$$\boldsymbol{x}(k+1) = \begin{bmatrix} p^n(k+1) \\ v^n(k+1) \end{bmatrix} = \begin{bmatrix} p_x^n(k) \\ p_y^n(k) \\ p_z^n(k) \\ v_x^n(k) \\ v_y^n(k) \\ v_z^n(k) \end{bmatrix} + \Delta T \begin{bmatrix} v_x^n(k) \\ v_y^n(k) \\ v_z^n(k) \\ a_x^n(k) \\ a_y^n(k) \\ a_z^n(k) \end{bmatrix} \tag{4.3}$$

where a^n represents the quadrotor's acceleration in the inertial $\boldsymbol{N}$ frame. Utilizing acceleration in the MPC formulation simplifies the problem by eliminating unnecessary rotation of the frame and excluding the mass m of the quadrotor, which is taken into account after solving each step by MPC to calculate the fully actuated force F^n. The running cost for the OCP is given by,

$$L(x(t), u(t)) = \|\boldsymbol{x_u} - \boldsymbol{x}\|_Q^2 + \|\boldsymbol{u} - \boldsymbol{u_r}\|_R^2 \tag{4.4}$$

where $\|\boldsymbol{x_u} - \boldsymbol{x}\|$ represents the difference between the predicted state $\boldsymbol{x_u}$ and the reference state $\boldsymbol{x_r}$, while $\|\boldsymbol{u} - \boldsymbol{u_r}\|$ denotes the variance between the control action, $\boldsymbol{u}$ and its reference value, $\boldsymbol{u_r}$. The second norm of both errors is then penalized by gains Q and R, which are also known as state and control weighting matrices respectively. In our scenario, we consider the model states and the control inputs as

$$\boldsymbol{x} = \begin{bmatrix} p^n \\ v^n \end{bmatrix} = \begin{bmatrix} p_x^n \\ p_y^n \\ p_z^n \\ v_x^n \\ v_y^n \\ v_z^n \end{bmatrix} \qquad \boldsymbol{u} = a^n = \begin{bmatrix} a_x^n \\ a_y^n \\ a_z^n \end{bmatrix}. \tag{4.5}$$

Evaluation of the running costs along the whole prediction horizon is calculated as,

$$J_N(x, u) = \sum_{k=0}^{N-1} L(x_u(k), u(k)) \tag{4.6}$$

where N is the prediction horizon and k is the prediction step. Finally, to find a minimizing control sequence the OCP can be interpreted as,

$$\begin{aligned}
\min_{u} \quad J_N(\boldsymbol{x_0}, \boldsymbol{u}) = \quad & \sum_{k=0}^{N-1} L(x_u(k), u(k)) \\
\text{subject to} \quad & \boldsymbol{x_u}(k+1) = f(\boldsymbol{x_u}(k), u(k)), \\
& \boldsymbol{x_u}(0) = x_0, \\
& \boldsymbol{u}(k) \in \mathcal{U}, \quad \forall k \in [0, N-1] \\
& \boldsymbol{x_u}(k) \in \mathcal{X}, \quad \forall k \in [0, N], \\
& (r_{obs_{scaled}} + r_{quad}) - d_{obs}(k) \leq 0, \\
& p_z^n(k) - 20 \quad \leq 0, \\
& \left(\sqrt{a_x^n(k)^2 + a_y^n(k)^2 + a_z^n(k)^2}\right) - 18 \leq 0, \\
& f(\boldsymbol{x}(k), \boldsymbol{u(k)}) - \boldsymbol{x_u(k+1)} = 0
\end{aligned} \tag{4.7}$$

The inequality constraints include obstacle avoidance and restrictions on the optimization variables, such as altitude, p_z^n, and acceleration, a^n. The equality constraint sets the error between the next model states, already known from the MPC calculation, $\mathbf{x_u}(k+1)$, and the predicted states calculated from the current states by integration, $f(\mathbf{x}(k), \mathbf{u}(\mathbf{k}))$, to 0.

Obstacle Avoidance

In (4.7), d_{obs} represents the distance between the quadrotor's center of mass and the center of the obstacle, which can be broken down into

$$d_{obs}(k) = \sqrt{(p_x^n(k) - x_{obs})^2 + (p_y^n(k) - y_{obs})^2 + (p_z^n(k) - z_{obs})^2}. \tag{4.8}$$

For this study, the NMPC considers each obstacle's center as known at the coordinates $(x_{obs}, y_{obs}, z_{obs})$. An imaginary spherical radius r_{obs} is set based on the obstacle's outmost edges or corners to ensure a safe proximity. In a physical environment, achieving this entails equipping the quadrotor with a lidar sensor to create a point cloud of the obstacle body. Furthermore, an imaginary spherical safety radius of r_{quad} is set around the quadrotor.

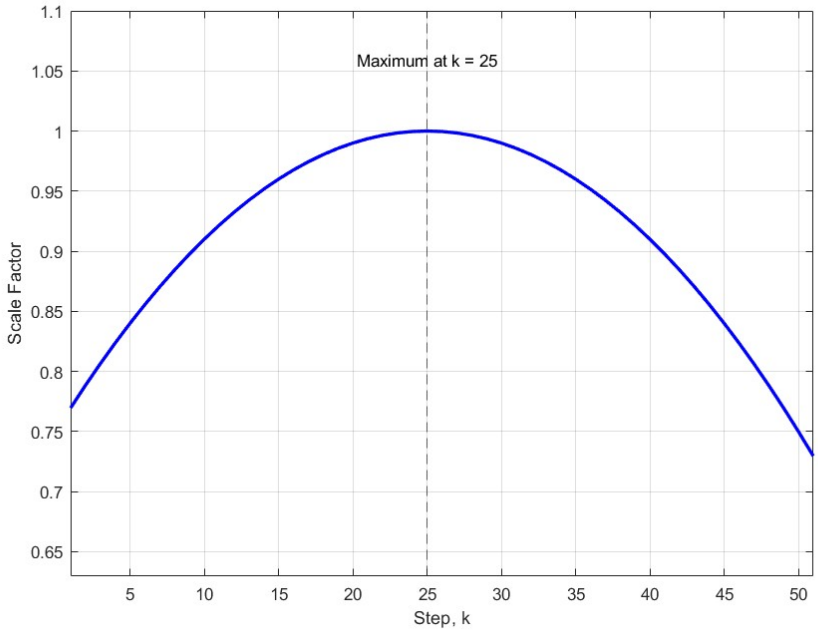


Figure 20: Parabloic scale factor along the prediction horizon

During the mission, the quadrotor's perception of the size of the imaginary sphere around the obstacle is dynamically adjusted based on its proximity. At each step, k in the prediction horizon, N, the algorithm recalculates the predicted obstacle radius for the entire prediction horizon. To this purpose, a parabolic scale factor is implemented that increases as k nears the midpoint of the prediction horizon, $N/2$, peaking at the midpoint, and then decreases as k moves away from the midpoint as shown in Figure 20. This modulation enhances the potential sensitivity to obstacles nearer to the midpoint, the maximum diameter of the imaginary sphere, visible by

the quadrotor when it is closest to the obstacle. Another way to say it is that the diameter of the spherical obstacle is where the quadrotor predicts that it might need increased maneuverability. The scale factor, S and the adjusted radius, $r_{obs_{scaled}}$ at prediction step, k is computed as

$$S = 1 - \left(\frac{1}{N^2}\right) \cdot (k - \frac{N}{2})^2 \tag{4.9}$$

$$r_{obs_{scaled}} = S \cdot r_{obs} + \frac{k}{100}. \tag{4.10}$$

Here r_{obs} represents the original radius of the imaginary sphere around the upcoming obstacle, and the term $\frac{k}{100}$ is used to incrementally increase the radius and simulate the heightened risk perception as the quadrotor progresses along its trajectory.

OCP to Nonlinear Programming (NLP) Conversion

CasADi, a versatile numerical optimization tool, has been used as the NPMC solver in this thesis. It excels in solving problems requiring gradient-based methods, especially in optimal control. It is not a solver that is explicitly tailored for addressing optimal control problems. Instead, its goal is to facilitate users with a set of resources that can be utilized to effectively construct solvers for OCPs, both general and specific, with minimal programming effort. Thus, an OCP problem is to be converted to an NLP problem for CasADi to solve our NMPC.

There are several methods for converting an OCP to an NLP. Some of these include:

- Single Shooting
- Multiple Shooting
- Collocation Method

In our case, we use the multiple shooting method, which involves using the system model as a state constraint at each optimization step, as shown by the declaration of the equality constraint in the OCP formulation. Multiple shooting is superior to the single shooting method because it elevates the problem to a higher dimension, which is known to improve convergence. Additionally, it allows the user to initialize with a known guess for the state trajectory. Although the size of the NLP problem often increases significantly, this is usually compensated by its sparsity [28].

First, we declare the problem decision variables,

$$\mathbf{w} = \begin{bmatrix} \mathbf{u}_0 & ... & \mathbf{u}_{N-1}, & \mathbf{x}_0 & ... & \mathbf{x}_N \end{bmatrix}. \tag{4.11}$$

$$\begin{aligned}
&\text{Objective Function:} \quad \min_{\mathbf{w}} \quad \Phi(\mathbf{w}) \\
&\text{subject to:} \quad \mathbf{g_1}(\mathbf{w}) = \begin{bmatrix} g_1(\mathbf{x}_0, \mathbf{u}_0) \\ \vdots \\ g_1(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) \\ g_1(\mathbf{x}_N) \end{bmatrix} \leq 0 \\
&\qquad\qquad\quad \mathbf{g_2}(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1 \\ \vdots \\ \mathbf{f}(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) - \mathbf{x}_N \end{bmatrix} = 0
\end{aligned} \tag{4.12}$$

The objective function is a direct function of the optimization variables $\mathbf{w}$ that should be minimized. $\mathbf{g_1}(\mathbf{w})$ is a vector that contains all the inequality constraints and $\mathbf{g_2}(\mathbf{w})$ is a vector that stores all the equality constraints. The constraints depend on the states, control inputs, and the known parameters relevant to the obstacles.

Position Control Input

The NMPC output is the optimized desired acceleration $\boldsymbol{a^n}$ for trajectory tracking, which is then used to compute the fully actuated control force $\boldsymbol{F}_u^n$ by,

$$\boldsymbol{F}_u^n = m\boldsymbol{a^n} - \boldsymbol{F_G^n} \tag{4.13}$$

The output of the position controller is the thrust T to quadrotor which is defined as,

$$T = \|\boldsymbol{F_u^n}\| \tag{4.14}$$

4.1.2 Attitude Control

The attitude controller in this thesis was based on a study from [29]. As discussed earlier, the control force $\boldsymbol{F_u^n}$ is divided into a desired direction and thrust magnitude. The attitude controller then aims to align the quadrotor's body z-axis with the direction of $\boldsymbol{F_u^b}$. Since the attitude controller tracks a desired attitude derived from $\boldsymbol{F_u^b}$, it faces challenges due to topological obstructions of the SO(3) manifold [30]. This problem is addressed by sending the saturated and normalized form of the control force, $\tilde{\boldsymbol{f}}_u^n$ into the attitude controller.

$$\tilde{\boldsymbol{f}}_u^n = \begin{cases} \boldsymbol{F}_u^n & \text{if } \|\boldsymbol{F}_u^n\| \leq 0 \\ \dfrac{\boldsymbol{F}_u^n}{\|\boldsymbol{F}_u^n\|} & \text{if } \|\boldsymbol{F}_u^n\| > 0 \end{cases} \tag{4.15}$$

Considering a quadrotor with the kinematics and dynamics described by (3.1),(3.2), (3.5), and (3.8) along with a force vector, $\tilde{\boldsymbol{f}}_u^n$ saturated by (4.15),

we define the input torque, $\boldsymbol{\tau}_u^b$ to the system as

$$\boldsymbol{\tau}_u^b = k_w(\boldsymbol{\omega}_{nb_{des}}^b - \boldsymbol{\omega}_{nb}^b) \tag{4.16}$$

here the desired angular velocity, $\boldsymbol{\omega}_{nb_{des}}^b$ is defined as

$$\boldsymbol{\omega}_{nb_{des}}^b = -(k_1 \boldsymbol{e}_3 \times \tilde{\boldsymbol{f}}_u^b) \tag{4.17}$$

$$\tilde{\boldsymbol{f}}_u^b = \boldsymbol{q}_{nb}^* \otimes \tilde{\boldsymbol{f}}_u^n \otimes \boldsymbol{q}_{nb} \tag{4.18}$$

To prove the boundedness of the angular velocity of the quadrotor, let a Lyapunov function candidate, V be defined as

$$V = \frac{1}{2}(\boldsymbol{\omega}_{nb}^b)^T \boldsymbol{J} \boldsymbol{\omega}_{nb}^b \tag{4.19}$$

which is positive definite and radially bounded. $\boldsymbol{J}$ is the inertia matrix of the quadrotor. We can write the derivative of it as

$$\dot{V} = (\boldsymbol{\omega}_{nb}^b)^T \boldsymbol{J} \dot{\boldsymbol{\omega}}_{nb}^b \tag{4.20}$$

now, inserting 3.8 and 4.16 into $\dot{V}$ we get

$$\dot{V} = (\boldsymbol{\omega}_{nb}^b)^T \left(k_w(\boldsymbol{\omega}_{nb_{des}}^b - \boldsymbol{\omega}_{nb}^b) - \boldsymbol{\omega}_{nb}^b \times J \boldsymbol{\omega}_{nb}^b \right). \tag{4.21}$$

Since $(\boldsymbol{\omega}_{nb}^b \times J \boldsymbol{\omega}_{nb}^b) = 0$, (4.21) yields

$$\dot{V} = (\boldsymbol{\omega}_{nb}^b)^T \left(k_w(\boldsymbol{\omega}_{nb_{des}}^b - \boldsymbol{\omega}_{nb}^b) \right) \tag{4.22}$$

inserting (4.17) we get,

$$\dot{V} = (\boldsymbol{\omega}_{nb}^b)^T \left(-k_1 k_w \boldsymbol{e}_3 \times \tilde{\boldsymbol{f}}_u^b - k_w \boldsymbol{\omega}_{nb}^b \right) \tag{4.23}$$

$$\dot{V} = -\frac{1}{2} k_w \|\boldsymbol{\omega}_{nb}^b\|^2 - k_1 k_w \|\boldsymbol{\omega}_{nb}^b\| \|\tilde{\boldsymbol{f}}_u^b\| \sin\theta_1 \cos\theta_2 - \frac{1}{2} k_w \|\boldsymbol{\omega}_{nb}^b\|^2 \tag{4.24}$$

where θ_1 is the angle between $\boldsymbol{e}_3$ and $\tilde{\boldsymbol{f}}_u^b$, and θ_2 is the angle between $(\boldsymbol{e}_3 \times \tilde{\boldsymbol{f}}_u^b)$ and $\boldsymbol{\omega}_{nb}^b$. Since, the inner angle between the two vectors is $\theta_1 \in [0, \pi]$ such that $\sin\theta_1 \in [0, 1]$ then,

$$\dot{V} = -\frac{1}{2} k_w \|\boldsymbol{\omega}_{nb}^b\|^2 - k_1 k_w \|\boldsymbol{\omega}_{nb}^b\| \|\tilde{\boldsymbol{f}}_u^b\| \cos\theta_2 - \frac{1}{2} k_w \|\boldsymbol{\omega}_{nb}^b\|^2 \tag{4.25}$$

By completing the squares in (4.25),

$$\begin{aligned} \dot{V} \leq & -\frac{1}{2} k_w \|\boldsymbol{\omega}_{nb}^b\|^2 - \left(\sqrt{\frac{k_w}{2}} \|\boldsymbol{\omega}_{nb}^b\| + k_1 \sqrt{\frac{k_w}{2}} \|\tilde{\boldsymbol{f}}_u^b\| \cos\theta_2 \right)^2 \\ & + \frac{{k_1}^2}{2} k_w \|\tilde{\boldsymbol{f}}_u^b\|^2 (\cos\theta_2)^2 \\ & \leq -\frac{1}{2} k_w \|\boldsymbol{\omega}_{nb}^b\|^2 + \frac{{k_1}^2}{2} k_w \|\tilde{\boldsymbol{f}}_u^b\|^2 \end{aligned} \tag{4.26}$$

From (4.15), we can say that, $\|\tilde{\boldsymbol{f}}_u^b\|$ is bounded by 1 such that

$$\dot{V} \leq -\frac{1}{2}k_w\|\boldsymbol{\omega}_{nb}^b\|^2 + \frac{{k_1}^2}{2}k_w. \tag{4.27}$$

This would be negative definite when

$$\begin{aligned} &-\frac{1}{2}k_w\|\boldsymbol{\omega}_{nb}^b\|^2 + \frac{{k_1}^2}{2}k_w < 0 \\ &\text{or,} \quad k_1 < \|\boldsymbol{\omega}_{nb}^b\| \end{aligned} \tag{4.28}$$

which implies that $\|\boldsymbol{\omega}_{nb}^b\|$ is bounded by k_1.

4.2 State Estimation

This study uses the Extended Kalman Filter (EKF) as the state estimator due to its effectiveness in handling the nonlinear dynamics of quadrotor flight. The EKF improves upon the classic Kalman filter by linearizing the nonlinear state equation around the current estimate to propagate the mean and covariance, making it suitable for systems with nonlinear state transition and observation models. The EKF operates in two phases: first, it predicts the estimation based on system dynamics, and then it corrects the estimation using sensor data and the predicted estimation.

Assuming that the system is expressed in nonlinear state space form as

$$\dot{x}(t) = f(x(t), u(t)) \tag{4.29}$$

$$\dot{x}(t) = \begin{bmatrix} \dot{p}^n \\ \dot{v}^n \\ \dot{E} \end{bmatrix} = \begin{bmatrix} v^n \\ \frac{1}{m}(f_G^n - F_u^n) \\ W(\phi,\theta)\omega_{nb}^b \end{bmatrix} \tag{4.30}$$

Since the angular velocity (ω_{nb}^b) is considered to be directly taken from the gyro, it is removed in the state space model from the EKF.

Prediction: The prediction phase of the EKF involves projecting the current state estimate forward to the next time step, which is essential for updating the system's state prior to receiving the next measurement. The equations for this phase are described by

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \tag{4.31}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k\mathbf{P}_{k-1|k-1}\mathbf{A}_k^\top + \mathbf{Q}_{k-1} \tag{4.32}$$

where:

- $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state estimate,

- $\mathbf{f}$ represents the state transition model which is a function of the previous state $\hat{\mathbf{x}}_{k-1|k-1}$ and the control input $\mathbf{u}_{k-1}$,
- $\mathbf{P}_{k|k-1}$ is the predicted state covariance,
- $\mathbf{A}_k$ is the Jacobian of the state transition model with respect to the state,

$$\mathbf{A}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}} \tag{4.33}$$

- $\mathbf{Q}_{k-1}$ is the process noise covariance matrix, which accounts for the uncertainty in the prediction.

Correction: After making a prediction, the EKF updates its estimates by incorporating the latest measurement data in the correction phase. This involves adjusting the predicted state based on new information from the measurements. The correction equations are then applied to make these adjustments.

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k)^{-1}, \tag{4.34}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})), \tag{4.35}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}, \tag{4.36}$$

where:

- $\mathbf{K}_k$ is the Kalman gain,
- $\mathbf{h}$ is the observation model,
- $\mathbf{H}_k$ is the Jacobian of the observation model with respect to the state,

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}} \tag{4.37}$$

- $\mathbf{R}_k$ is the measurement noise covariance matrix,
- $\mathbf{z}_k$ represents the new measurement,
- $\mathbf{I}$ is the identity matrix.

The EKF is an efficient and effective method for state estimation in nonlinear systems like quadrotors, enhancing the accuracy and reliability of navigation and control systems supporting autonomous flight operations.

4.3 Guidance

The motion planning for the quadrotor involves designing an elliptical trajectory around a large vessel to inspect its draft markings. To accommodate the substantial dimensions of these vessels (ranging from 300 m to 350 m in length and 30 m to 40 m m in width), a scaled down model is used for simulation, with the vessel's representation being reduced to 16 units in length, 4 units in width, at the height of 2 units above ground level. The quadrotor ascends to align with the desired altitude of 2 units and then begins tracking an elliptical trajectory based on parametric equations. Its goal is to capture images or videos focusing on the vessel's draft marking. The quadrotor is also meant to autonomously stop and capture images near the draft marking close to the vessel using a control system, but this functionality has been left out for future work. The trajectory of the motion over time $\boldsymbol{t}$ is defined by

$$\begin{aligned} x(\boldsymbol{t}) &= A\cos(w_x \cdot \boldsymbol{t} + \delta), \\ y(\boldsymbol{t}) &= B\cos(w_y \cdot \boldsymbol{t}), \\ z(\boldsymbol{t}) &= -h. \end{aligned} \tag{4.38}$$

The desired velocity can be described by the derivation of (4.38),

$$\begin{aligned} \dot{x}(\boldsymbol{t}) &= -Aw_x \sin(w_x \cdot \boldsymbol{t} + \delta), \\ \dot{y}(\boldsymbol{t}) &= -Bw_y \sin(w_y \cdot \boldsymbol{t}), \\ \dot{z}(\boldsymbol{t}) &= 0. \end{aligned} \tag{4.39}$$

The desired acceleration then can be derived from (4.39).

$$\begin{aligned} \ddot{x}(\boldsymbol{t}) &= -Aw_x^2 \cos(w_x \cdot \boldsymbol{t} + \delta), \\ \ddot{y}(\boldsymbol{t}) &= -Bw_y^2 \cos(w_y \cdot \boldsymbol{t}), \\ \ddot{z}(\boldsymbol{t}) &= 0. \end{aligned} \tag{4.40}$$

Here $\boldsymbol{t}$ represents the time vector ranging from 0 to t_{end} with increments defined by t_{step}. The parameters are set as follows:

- $A = 10.5$, the amplitude in the x-direction, reflecting the semi-major axis of the ellipse.
- $B = 3$, represents the amplitude in the y-direction and corresponds to the semi-minor axis.
- $\delta = \frac{\pi}{2}$, aligns the trajectory appropriately relative to the vessel by adjusting the phase shift in the x component.
- $w_x = 0.1$ and $w_y = 0.1$, the angular frequencies for the x and y components, respectively, ensuring a smooth and continuous elliptical motion.

- $h = 2$, the constant altitude, in negative to conform with the inertial NED frame.

This set of equations describes a 3D trajectory where $x(\boldsymbol{t})$ and $y(\boldsymbol{t})$) form a Lissajous curve, and $z(T)$ remains constant, implying motion along a flat plane at a height of h. The vessel outline and the desired trajectory is shown in Figure 21.

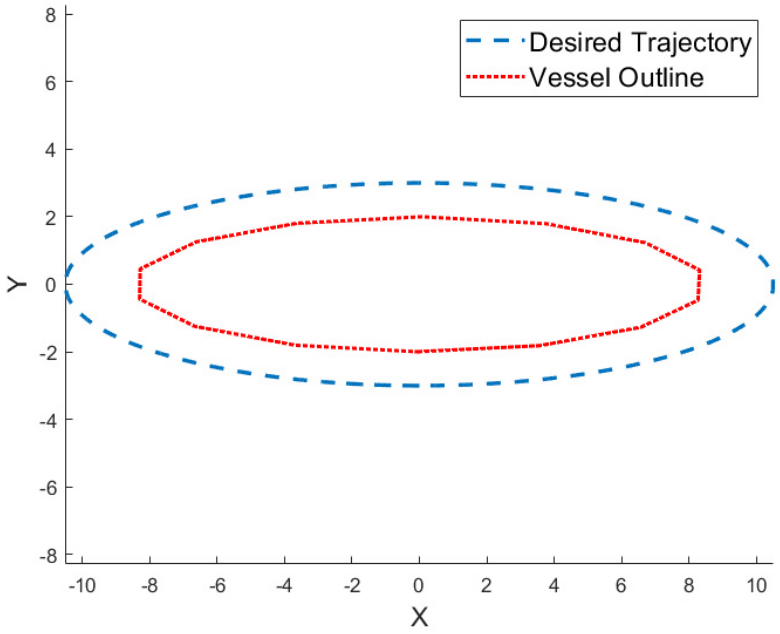


Figure 21: Ship outline and the quadrotor's desired trajectory around it

5 Simulation/Experimental results

Based on the mathematical modeling and control system development covered in the previous sections, a simulation was set up. This section presents and analyzes the results of the simulations, which are divided into three different scenarios based on the inclusion and exclusion of obstacles. Before delving into these results, let's first discuss the simulation setup in brief.

5.1 Simulation Setup

The simulation environment was created using MATLAB Simulink, which is an integrated platform that enables block diagram modeling of various systems. All the proposed control algorithms along with quadrotor equations were mathematically modeled for the implementation. The Simulink solver is configured as a fixed-point solver to ensure a constant sample time, which consistently captures coordinates of the desired trajectory at regular intervals without skipping any data points. An illustration of the complete Simulink setup can be found in Figure 22.

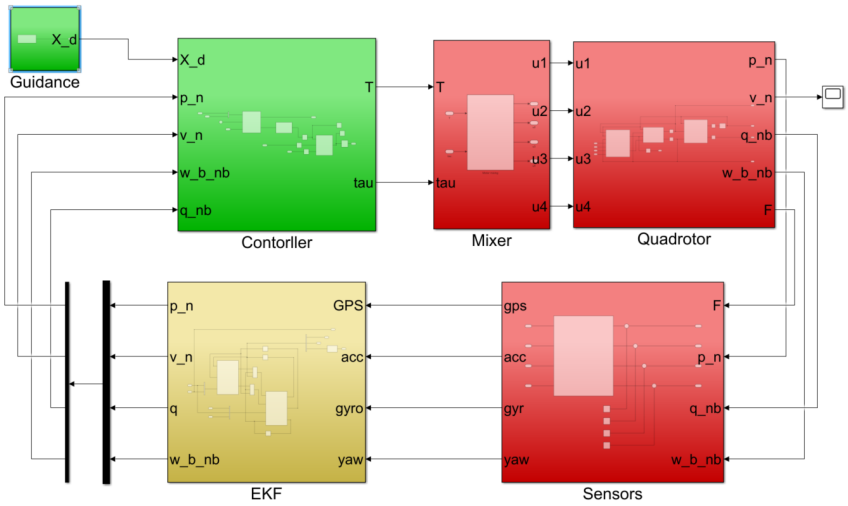


Figure 22: Simulation model structure in Matlab Simulink

General parameters used in the simulation are presented in Table 1.

Table 1: Simulation Parameters

Description	**Symbol**	**Parameter Value**
Simulation Period	t	70 s
Time Step	t_{step}	0.001 s
NMPC Frequency	-	20 Hz
Attitude Controller Frequency	-	500 Hz
Quadrotor Initial Position	p_0^n	$[-1.5, 5, 0]^\top$
Quadrotor Initial Velocity	v_0^n	$[0, 0, 0]^\top$
Quadrotor Initial Angular Velocity	$\boldsymbol{\omega}_{nb_0}^b$	$[0, 0, 0]^\top$
Quadrotor Initial Orientation	$\boldsymbol{q}_{nb_0}^b$	$[1, 0, 0, 0]^\top$

5.1.1 Implementation of the NMPC

The Position Control block is implemented using an NMPC approach, which operates at a frequency of 20 Hz. This control strategy computes the fully actuated force required for the quadrotor to follow the flight path by utilizing both the desired and current states of the quadrotor. We use a Matlab system block named "casadi_block" to integrate CasADi with Simulink as "interpreted Matlab code" since there is no integrated Simulink block for CasADi [31]. Transition rate blocks are employed to simulate changes in frequency. The position control block is depicted in Figure 23.

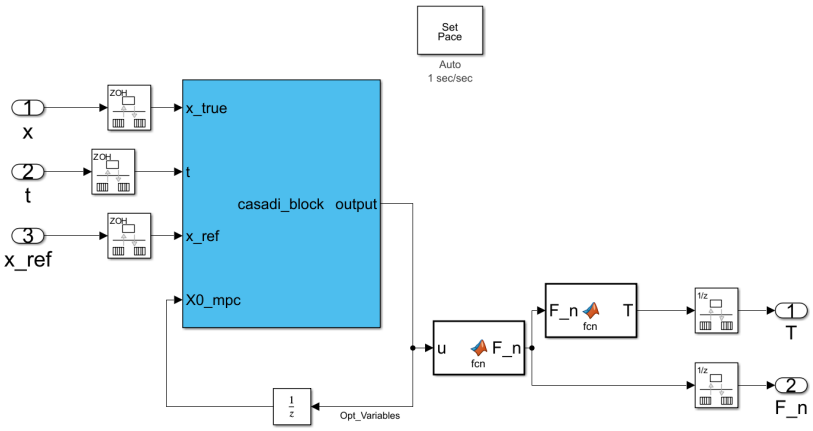


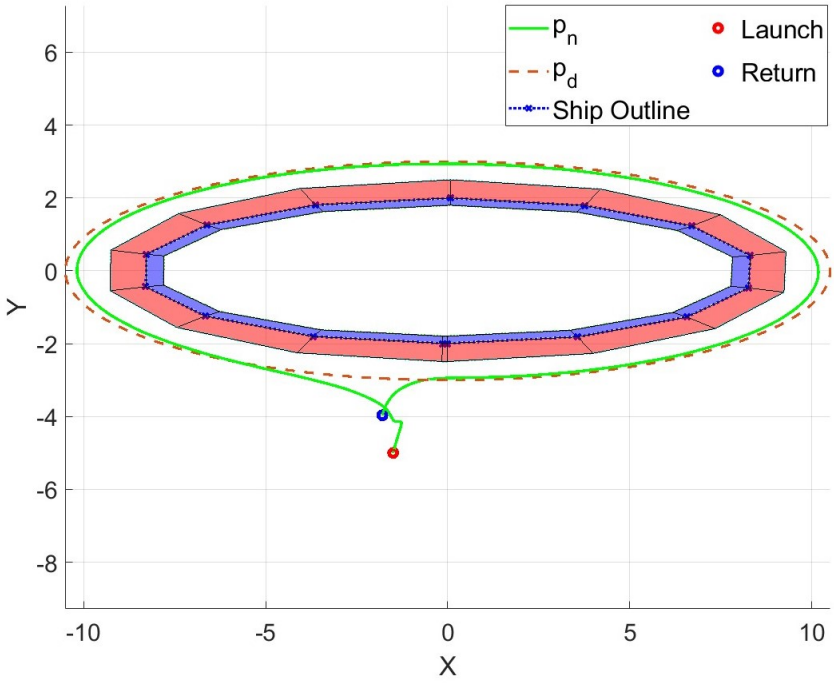
Figure 23: Position controller setup in Simulink

5.2 Simulation Results

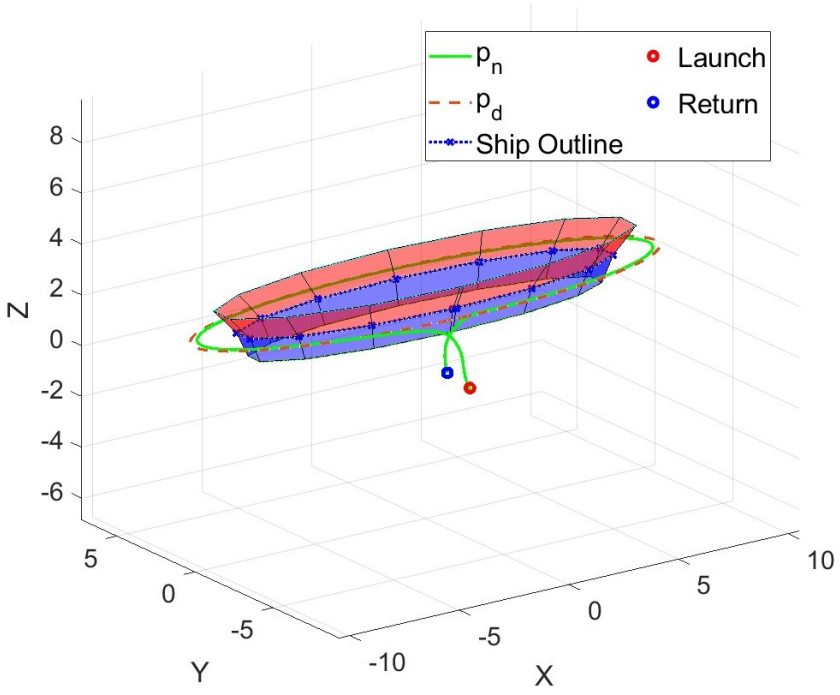
The simulation was conducted in three different cases. The first case involved no consideration of obstacles, while the second case took obstacles into account. In the latter case, two different scenarios were simulated to showcase the controller's effectiveness. For both scenarios, obstacle positions were randomly placed around the ship's body.

5.2.1 Trajectory Tracking without Obstacles

In the first case, just a simple trajectory tracking is conducted around the vessel on a pre-planned path shown in Figure 24.



(a) Top view



(b) Corner view

Figure 24: Trajectory tracking around a vessel without obstacles [3D View]

The controller is capable of maintaining precise trajectory tracking, as illustrated in Figure 25.

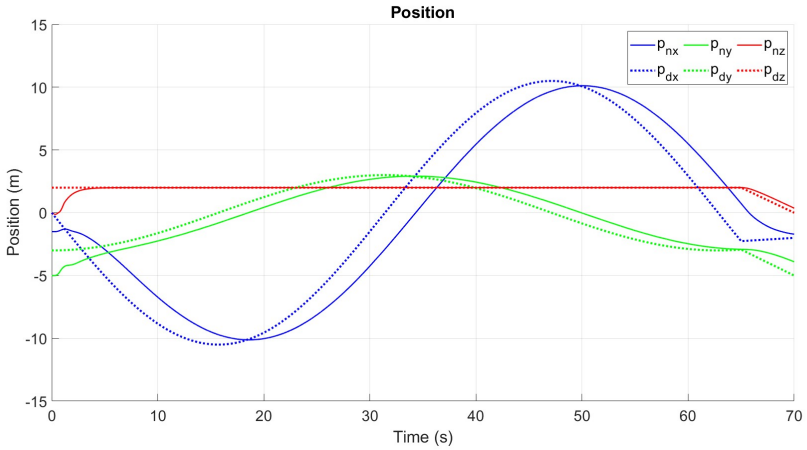


Figure 25: Position of the quadrotor during trajectory tracking

In Figure 26, we observe a consistent position error during the path following, particularly along the x and y-axis. The position error here relates

to several factors such as the sample rate of MPC, the prediction horizon, the feasibility of the trajectory, and the underactuation of the quadrotor. A prediction horizon of 100 along with an MPC sample time of 0.3 s was also tried along with some tuning on the Q and R. However, the error was not removed completely.

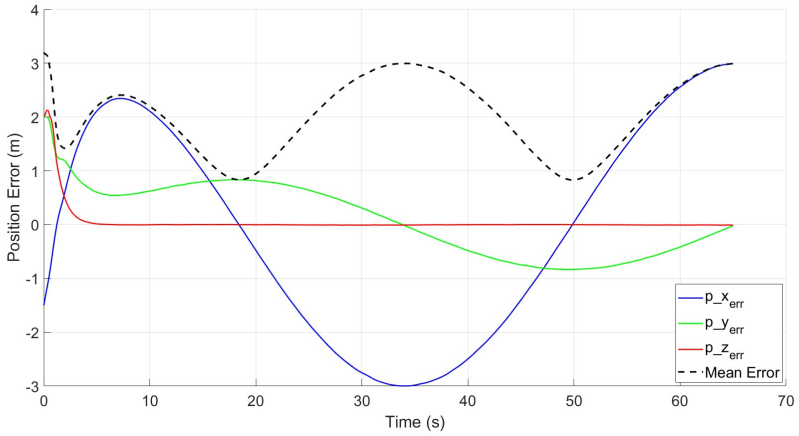


Figure 26: Position error during trajectory tracking

This could also happen due to the delay in the NMPC controller, which can be caused by its computational time, resulting in a lag when tracking the desired point. The position error does not noticeably deviate the quadrotor from its actual trajectory but rather makes it track the desired point a few seconds later. The 3D plot in Figure 24 shows a more accurate tracking of the position within the x-y plane, without considering the lag. In contrast, in Figure 25, the error becomes noticeable when compared against time. Therefore, despite appearing high, the position error does not significantly affect the accuracy of position tracking as depicted in Figure 24. The inertial velocity of the UAV is depicted in Figure 27. It tracks the derivative of the desired trajectory. Similar to how the position is traced, there's also a noticeable delay in velocity tracking. Additionally, during takeoff, a maximum lift velocity of about 2.25m/s is attained, while a steady speed is maintained for the rest of the path to achieve its objective.

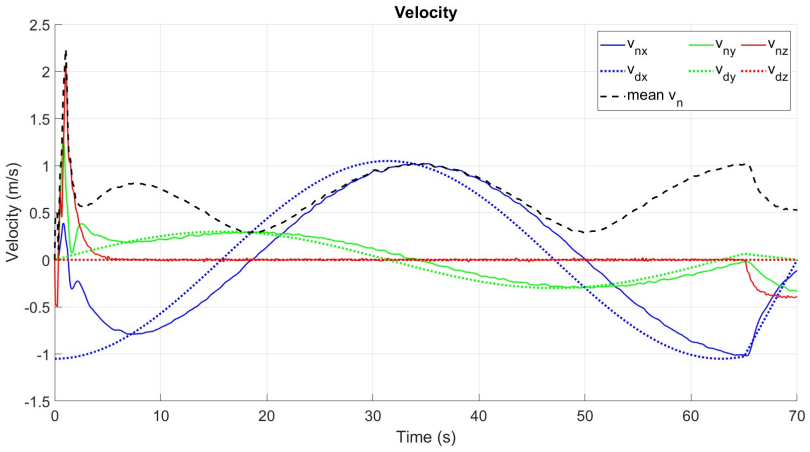


Figure 27: Velocity of the quadrotor during trajectory tracking

Figure 28 shows the optimized force vector, which is later broken down into thrust and desired orientation vectors. Initially, only a thrust force is visible for an obstacle-free trajectory. Subsequently, there are no significant fluctuations except for small stabilizing control forces.

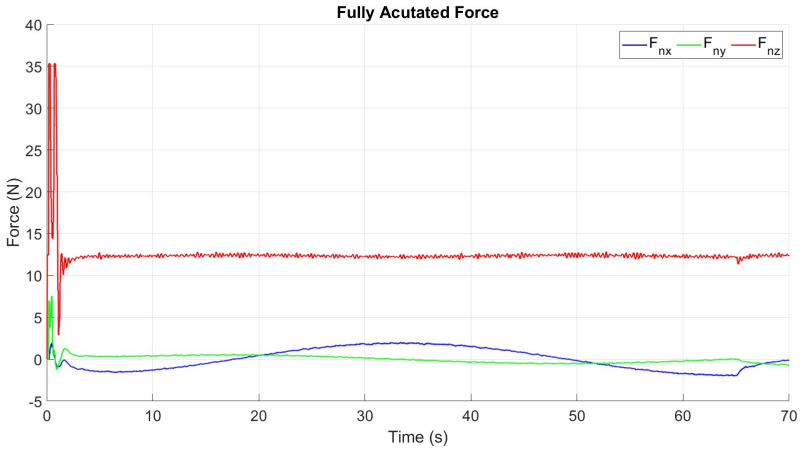


Figure 28: Optimized fully actuated force vector

The quadrotor in Figure 29 reaches its highest mean angular velocity during takeoff. The oscillation at this time is caused by the initial lifting velocity. Additionally, the angular velocities are within the bounded region.

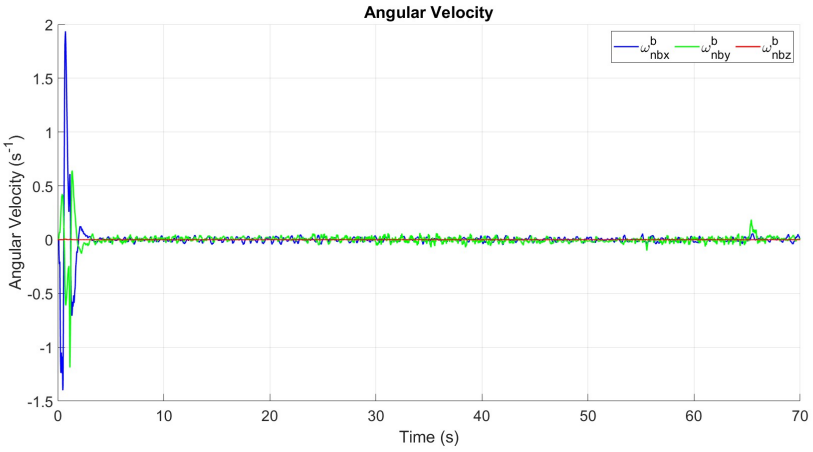


Figure 29: Angular velocity of the quadrotor during trajectory tracking

The angular velocity is determined by the applied torque. Thus, a similar pattern also appears in the curves showing the control torque in Figure 30.

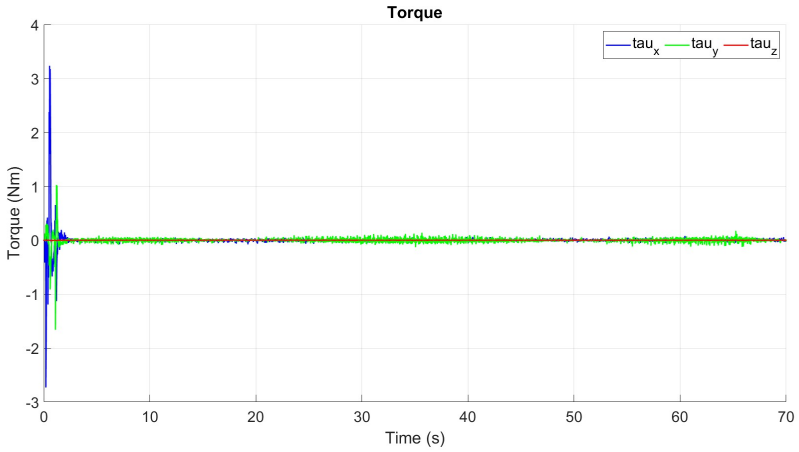
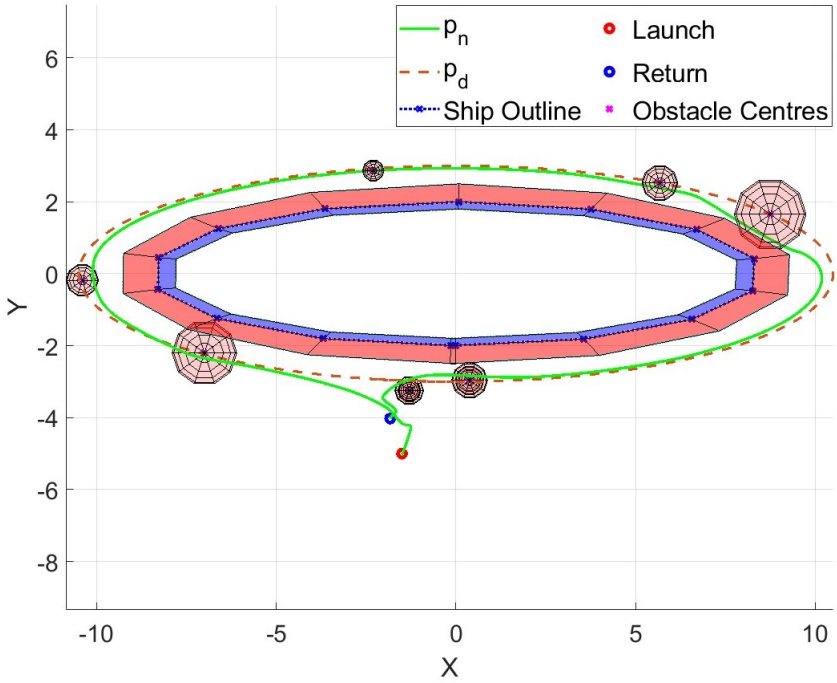


Figure 30: Control torque during trajectory tracking without obstacles

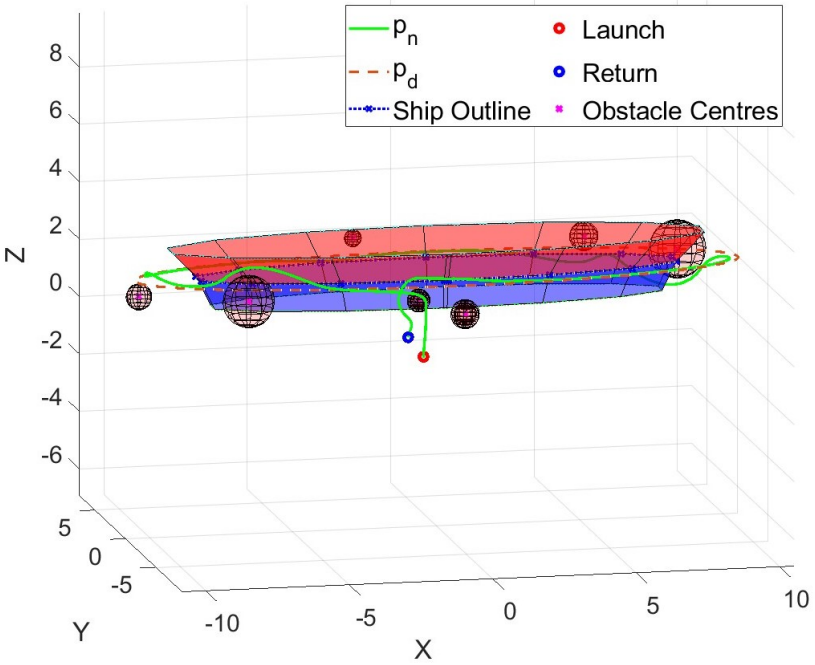
5.2.2 Trajectory Tracking with Obstacles

Scenario 1

In this simulation scenario, seven obstacles of various sizes were randomly placed around the ship within the environment, as depicted in Figure 31.



(a) Top view



(b) Corner view

Figure 31: Trajectory tracking around a vessel without obstacles [3D View]

Figure 32 shows the quadrotor's position during its survey around the vessel, including the obstacle centers and its position in three axes. As the quadrotor approaches an obstacle along the desired path, it deviates from its path and finds a more suitable route to avoid the collision.

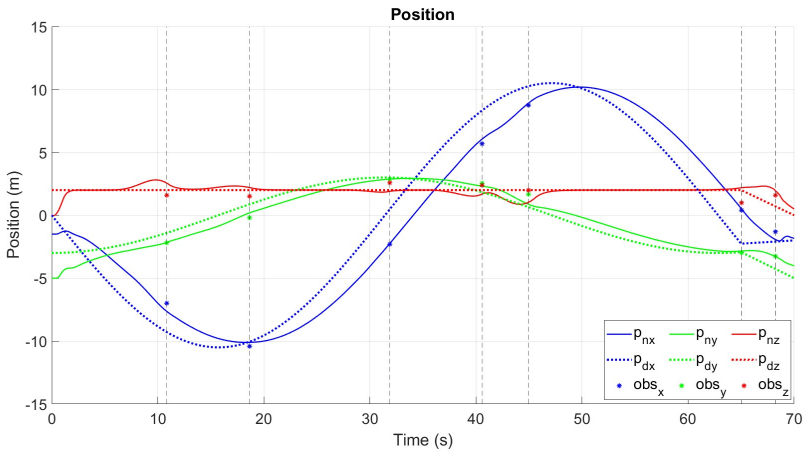


Figure 32: Position of the quadrotor during trajectory tracking with obstacles

In Figure 33, we observe the existing position error mentioned in the previous section. Additionally, we can see the errors increasing with a bump when the quadrotor is bypassing an obstacle.

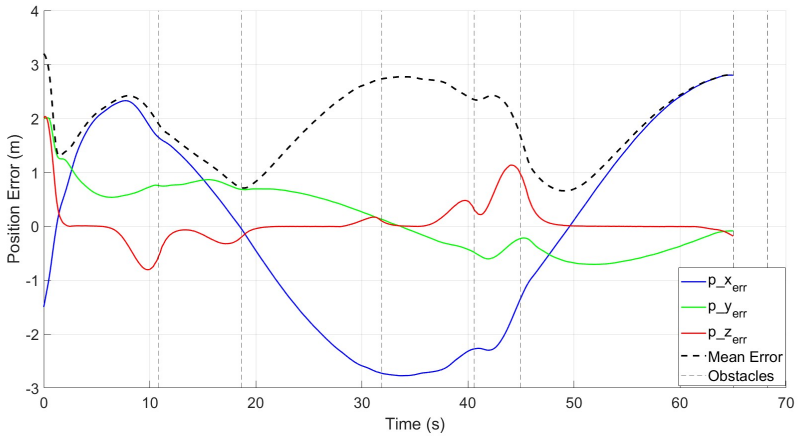


Figure 33: Position error during trajectory tracking with obstacles

Throughout the mission, the quadrotor maintains a stable velocity. However, there are some oscillations during takeoff and landing. Additionally, a subtle reduction in the mean velocity can be observed when approaching obstacles (Figure 34).

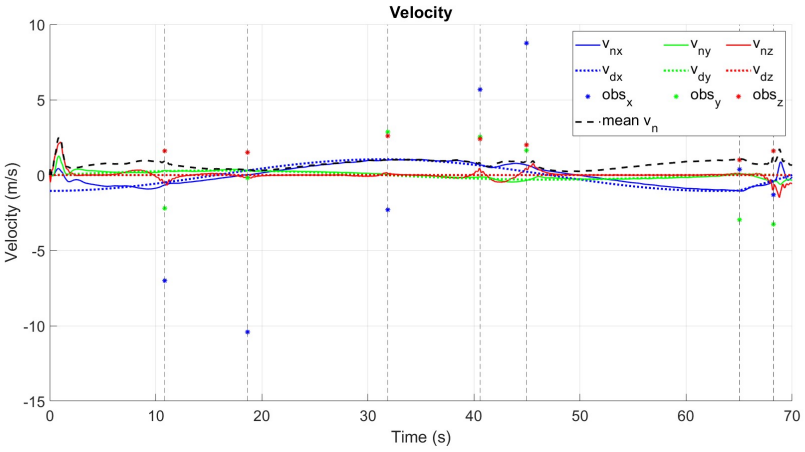


Figure 34: Velocity of the quadrotor during trajectory tracking with obstacles

From Figure 35, we observe that the lift force generates approximately 35 N of vertical inertial force, which is the takeoff thrust. When encountering an obstacle, we witness the force vector in action causing deviation from the intended trajectory. As the quadrotor corrects its course, the force vector returns to a stable state.

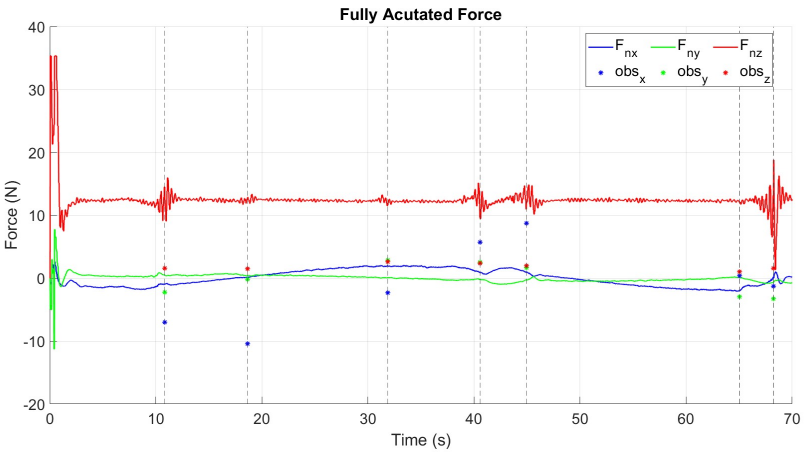


Figure 35: Optimized fully actuated force vector

The angular velocity shown in Figure 36 indicates a smooth flight for the quadrotor. Initially, the takeoff generates maximum angular velocity to position the quadrotor from launch into its trajectory. Nevertheless, it stays within the boundary. The angular velocity fluctuates within a very narrow range, except when approaching obstacles where it becomes more pronounced. This is due to the obstacles being spaced at a distance that

allows comfortable maneuvering for the quadrotor. The spike at the end corresponds to a critical maneuver around an obstacle just before landing.

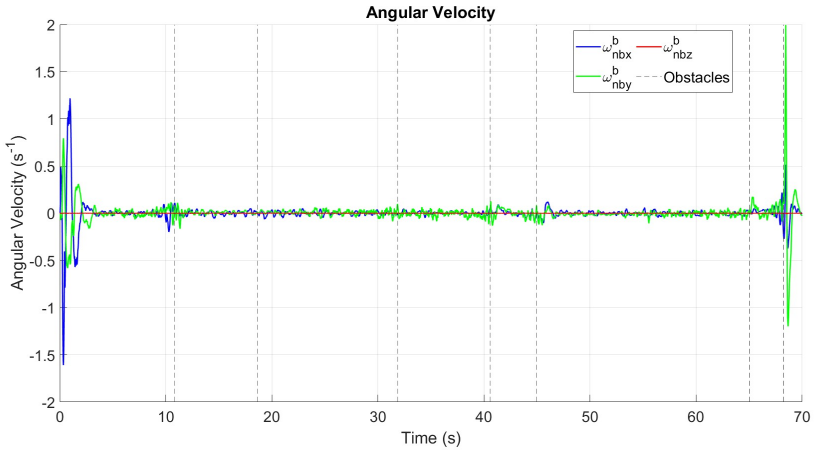


Figure 36: Angular velocity of the quadrotor with obstacles in the path

The attitude controller generates control torque to maneuver the quadrotor by adjusting its orientation, thereby utilizing the thrust force for horizontal acceleration. After the takeoff, the torque fluctuates within a very small range, indicating a steady flight. However, slightly larger spikes are noticeable near obstacles.

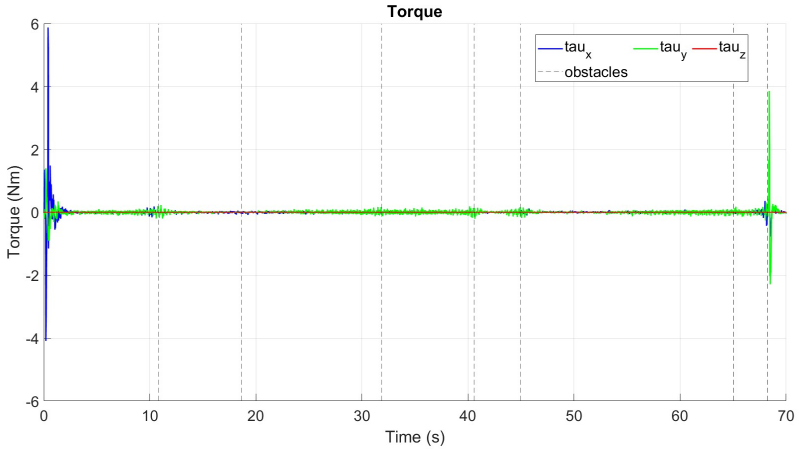
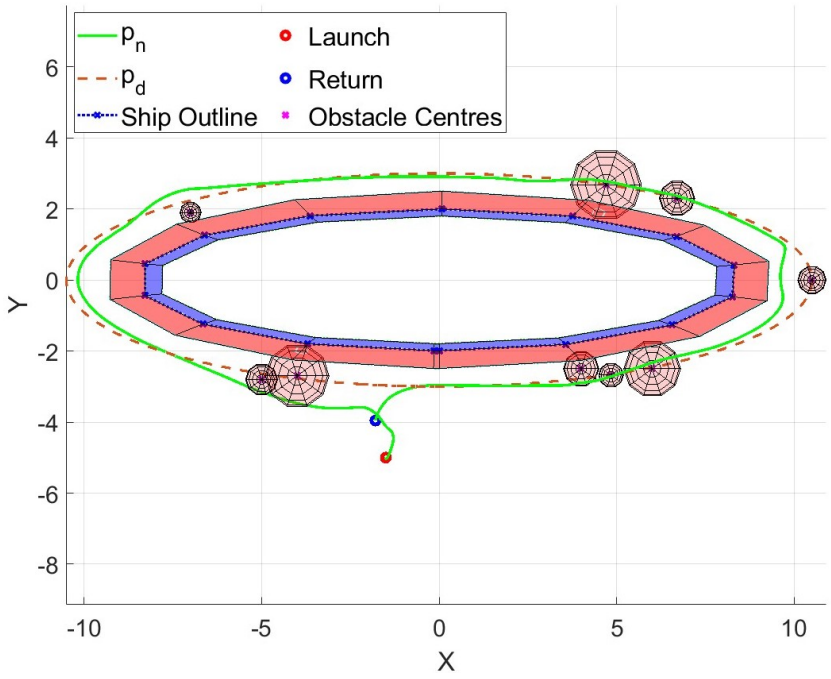


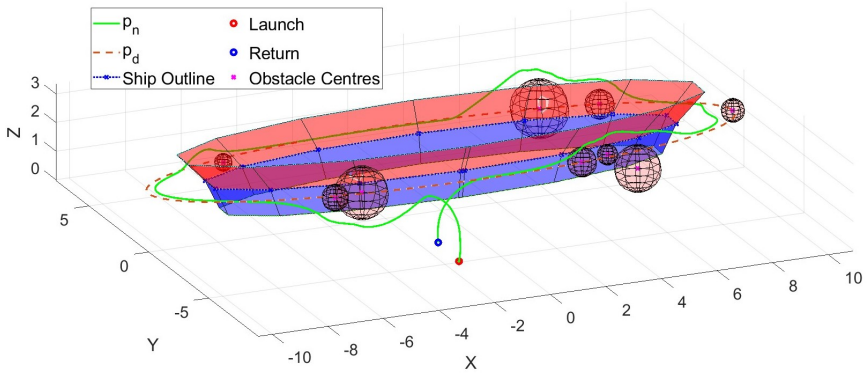
Figure 37: Control torque applied during trajectory tracking with obstacles

Scenario 2

In this simulation scenario, nine obstacles of various sizes were placed randomly around the ship within the environment, as shown in Figure 38.



(a) Top view



(b) Corner view

Figure 38: Trajectory tracking around a vessel without obstacles [3D View]

This time, multiple obstacles were placed adjacent to each other at different locations to test the controller's response to detecting multiple objects in close proximity.
The position curve in Figure 39 demonstrates a satisfactory outcome by successfully avoiding the obstacles while maintaining a safe distance. Once the quadrotor passes the obstacles, it returns to its original trajectory.

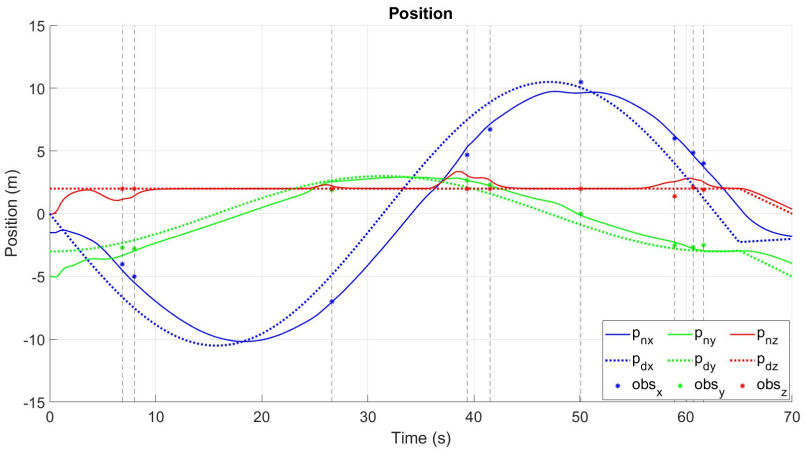


Figure 39: Position of the quadrotor during trajectory tracking with obstacles

In Figure 40, the position error shows the additional divergence from the intended path to avoid obstacles.

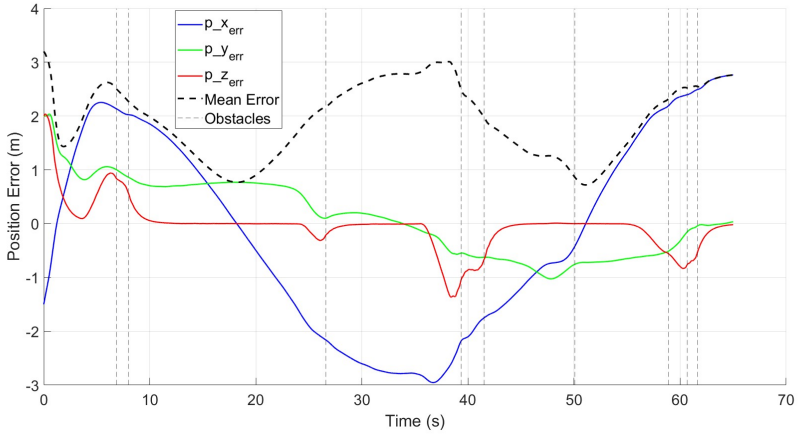


Figure 40: Position error during trajectory tracking with obstacles

During takeoff, the mean velocity reaches its peak, as depicted in Figure 41. Throughout the entire flight, the quadrotor maintains a low velocity to

ensure stability while capturing images or videos. Minor fluctuations are noticeable around obstacles.

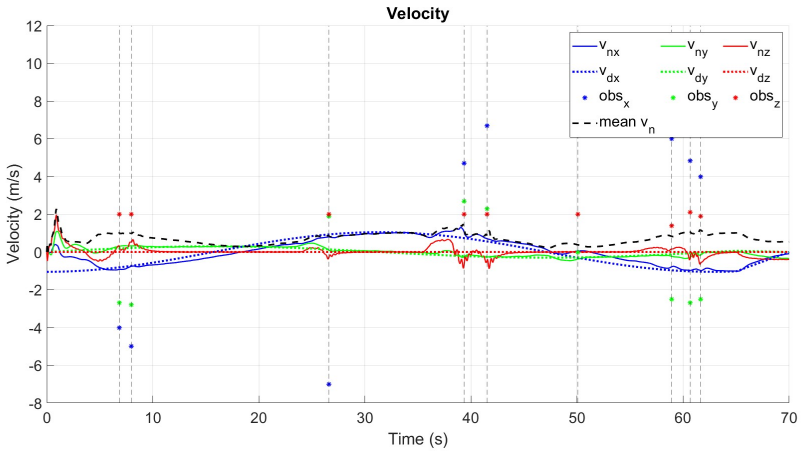


Figure 41: Velocity of the quadrotor during trajectory tracking

The force vector depicted in Figure 42 exhibits notable oscillations near the obstacles when compared to the first scenario. This is attributed to the larger size and closer proximity of multiple obstacles in places, necessitating more substantial maneuvering by the quadrotor.

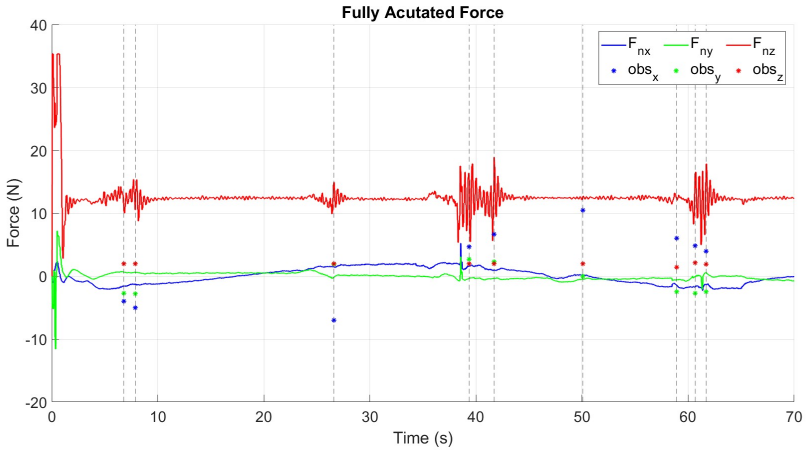


Figure 42: Optimized fully actuated force vector

The angular speed depicted in Figure 43 demonstrates the quadrotor's movement in a setting with numerous neighboring obstacles. This necessitates swift adjustments in orientation, leading to more prominent peaks in the graph compared to scenario 1.

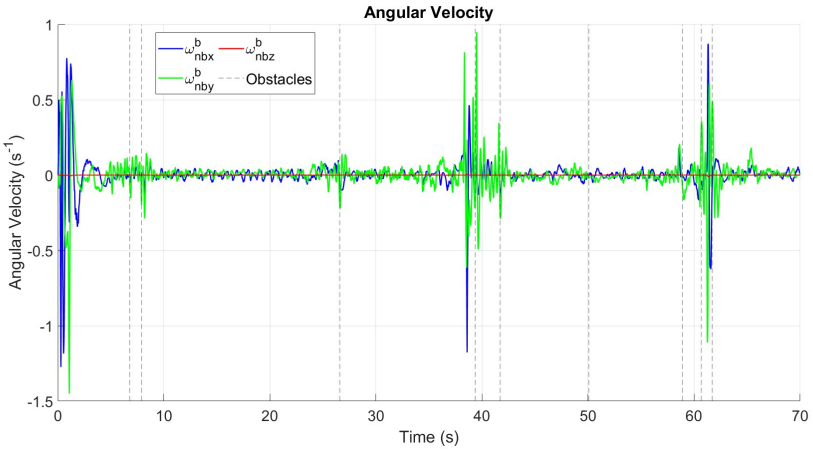


Figure 43: Angular velocity of the quadrotor during trajectory tracking

In contrast to the first scenario, the takeoff torque is lower in this case, as indicated in Figure 44. This results in a smoother transition from launch to trajectory. Very minimal control torque is generated over the planned path to maintain stable horizontal flight. Oscillations are noticeable near and around obstacles. A longer duration of control torques in action indicates a broader area of obstacles.

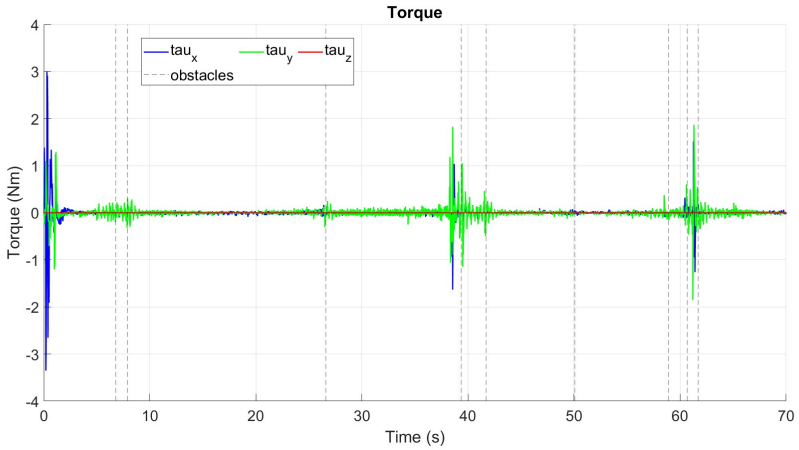


Figure 44: Control torque applied during trajectory tracking with obstacles

The simulation animations can be seen in videos uploaded to [32].

5.3 Experimental Results

Building the drone according to the planned configuration model was a part of the laboratory work during the thesis period. The complete assembly took

quite some time, and although we intended to implement the simulated GNC algorithm, this proved unfeasible due to our limited time frame. However, it's worth noting that the lab-built drone is fully functional when operated manually using a remote controller and existing autopilots such as Mission Planner or QGroundControl.

Figure 45: Drone assembled in lab

5.3.1 Design Specification

The design specification included the following components:

- Holybro X500 Quadrotor Frame
- Power Distribution:
  - Holybro XT60 Power Module (PM02 V3): This PM02 Power Module offers a straightforward method for delivering regulated 5.2V to the flight controller from the battery, along with current consumption and battery voltage monitoring capabilities.
  - Holybro PDB01: Power distribution board that distributes power from the power module to the motors.
- Flight Controller: PX4 Pixhawk 4 Mini flight controller where the GNC algorithms are stored. The flight management unit (FMU) processor in this autopilot is an STM32F765 microcontroller with 2MB flash memory and 512KB RAM. It also has an Inertial Measurement Unit (IMU) and barometer sensor integrated.
- Battery: Considering the payload and the structural mass of the drone a 4S 5000mAh -30C-Spektrum Lipo battery was adequate for the test.
- Remote Control:

  - Radiomaster TX12 Transmitter: For controlling the drone manually over a radio connection.
  - R81 Radio Receiver: It is plugged into the drone's flight controller to receive commands from the transmitter.

- Sensors:
  - Holybro Pixhawk 4 Neo-M8N GPS: GPS sensor to receive real-time location data.
  - TeraRanger Tower Evo (EVO 60m): For collision avoidance, a TeraRanger Tower was added which has 4 Lidar sensors at 90° apart.
  - MB1040 XL-MaxSonar-EZ4: Sonar Sensor to be used as an altitude measurement sensor.

- Camera:
  - RunCam 5 which is a good camera for quadrotors flying at high speed and added vibration. Currently, the camera is only for recording and no live feed functionalities have been implemented.
  - Quark Stabilizer 2: Single Axis Gimbal for Drone Camera.
  - Ulanzi LED Light

5.3.2 Design Limitations and Sugesstions

The development of the physical prototype revealed practical limitations in the previous design and highlighted the need for enhancements to meet the rigorous demands of maritime operations. The key potential improvements in the prototype's design are outlined as follows:

Sensor Integration and Environmental Perception: The drone's environmental awareness and operational reliability can be improved by upgrading the integration of sensors. Implementing a fusion of GPS and barometric sensors will provide more accurate altitude measurements compared to the previously specified sonar sensors, which were found to be less reliable, especially in complex airflow environments near large vessels.

Material Selection for Durability: The material choices need to be revised in order to enhance the drone's durability in maritime conditions. This requires selecting materials that can resist corrosion from exposure to saltwater and withstand the physical stress experienced during flights in high winds and wet conditions commonly found in marine environments.

Aerodynamic Features and Propeller Protection: Improvement in the aerodynamic features can enhance the stability and energy efficiency of the UAV in strong maritime winds. In addition, designing propeller protectors to safeguard the rotors against debris and incidental contact with ship structures during close-range operations ensures safety and operational continuity.

Inflatable Landing Gear: Given the high cost and critical nature of the mission-specific equipment onboard, having an inflatable landing gear system would be desirable to facilitate emergency water landings without causing damage to the drone. This feature greatly improves the ability to recover the UAV in case of a failure over water, reducing the risk of complete loss and enabling vehicle reusability.

These recommendations not only address the previously identified design limitations but also significantly enhance the operational capabilities and safety of the UAV for maritime draft survey missions.

The video of the quadrotor in manual flight is uploaded to [33].

6 Discussion

This thesis introduces an optimization-based guidance and control algorithm for a quadrotor, intended to conduct draft surveys of large vessels autonomously. The algorithm is designed to follow a predefined trajectory around the ship and capture high-resolution images. The guidance algorithm creates an elliptical trajectory around the vessel and sends the desired path data to the controller as a setpoint. An NMPC algorithm was employed as the outer loop position controller, representing the primary contribution of this project. In the inner loop, a faster-reduced attitude controller directs the quadrotor's body z-axis along the fully actuated force vector.

After analyzing the simulation results, it is evident that both control force and control torque yielded precise responses based on their objectives in multiple scenarios. Initially developed without a state estimator using true states from the quadrotor, the controller's performance remained strong even after implementing sensors and the EKF as a state estimator. The inclusion of constraints, such as collision avoidance in NMPC, has proven to be effective. Collision avoidance capabilities have demonstrated promise in enhancing safety and operational efficiency throughout the mission. The quadrotor also delivered satisfactory results when encountering obstacles of various sizes at different locations along its intended path. Additionally, the control algorithm's ability to adjust its perception of multiple nearby obstacles has been effective in the simulations.

Certain limitations were observed in terms of the NMPC solver load. Handling nonlinear dynamics and constraints, especially with NMPC algorithms, typically requires advanced mathematical libraries and significant processing power that may not be provided by standard flight controllers. Computing control actions in real-time within the timing requirements of UAV operations is essential for NMPC, which can be problematic to achieve on a standard micro-controller without real-time operating system support for high-level mathematical computations.

One of the challenges faced during this project was implementing the algorithm on a Gazebo simulation using the PX4 Firmware, which is a more realistic simulation environment. However, it proved to be a lengthy and complex endeavor to integrate NMPC with the flight controller within our limited time frame. Offboard control using ROS 2 (Robot Operating System 2) can also be a good way to start with the implementation of the controller on a Gazebo simulation. In the ROS2-PX4 interface, it is possible to send direct control commands to the quadrotor by bypassing the firmware's controllers.

These issues underline the importance of continued refinement of the control system and the development of a collision detection system to handle diverse settings for autonomous surveys around a vessel more effectively.

7 Conclusion

This thesis has addressed the challenge of manually performing draft surveys around large maritime vessels, proposing an automated solution to minimize human effort and risks. The main goal was to enhance safety and efficiency by automating the traditionally manual draft survey process. Through theoretical development, simulation testing, and analysis, this research has shown significant potential for UAV applications in a complex maritime environment.

This study demonstrated that the guidance algorithm can accurately send a desired path around the vessel. Implementing Nonlinear Model Predictive Control on the quadrotor's position controller effectively improves trajectory tracking and collision avoidance, affirming the robustness of the control system against environmental obstacles. The attitude controller has been shown to be adequate for maintaining UAV stability and orientation, enabling precise navigation in both trajectory tracking and obstacle avoidance. The complete simulations included sensor models and EKF for state estimation, which also tested the performance of the controller.

Given that this study advances an idea first introduced a few years ago, further research must continue to build upon the current foundation. Such sustained effort is crucial to realizing the full prospect of automated draft surveys within the maritime industry. This progression will not only refine the methodologies used but also pave the way for the adoption and implementation in real-world maritime operations.

7.1 Future Work

The results achieved from the initial simulations and theoretical evaluations indicate numerous potential areas for future research and advancement.

1. **Precise Yaw Tracking:** Integration of precise yaw tracking into the control system will enhance the UAV's photographic capabilities. This upgrade will maintain a steady focus on the vessel, ensuring consistent quality in the imagery captured throughout the survey.

2. **Controller Optimization:** The control algorithm needs further optimization to improve runtime and reduce system load. It is also important to enhance the UAV's adaptability to sudden environmental changes and more complex obstacle configurations by refining response mechanisms.

3. **Integration with a flight controller:** Implementing the optimized control algorithm into an open-source autopilot software such as PX4 will broaden its scope of application and make integration with existing

hardware easier. This will also streamline the addition of extra sensory inputs, like point cloud data from LiDAR systems, enhancing the precision of the UAV's navigation and its environmental perception.

4. **Simulation Testing Using ROS and Gazebo:** Before physically deploying the enhanced control system, it should be tested in a simulated environment using ROS and Gazebo. This approach will enable detailed assessments under controlled conditions, allowing for fine-tuning of the system before real-world implementation.

5. **Physical Drone Testing:** After successful simulation tests, the next phase will involve field trials using an actual drone in a controlled lab environment. This phase is crucial for validating the practical effectiveness of the control system and its components under realistic operational conditions.

Accomplishing these steps will improve the functionality and reliability of the UAV system. Integrating advanced control features and conducting vigorous testing will pave the way for deployment in other industrial settings, expanding the potential uses of UAVs.

References

[1] M. Usman, "Automated multi-rotor draft survey of large vessels," May 2022. Available at https://hdl.handle.net/10037/25775.

[2] N. Havn, "Narvik havn, maritime." https://www.narvikhavn.no/en/maritim/, 2024. Accessed: April 11, 2024.

[3] "Lkab, narvik." https://www.visitnarvik.com/lkab, 2024. Accessed: April 12, 2024.

[4] CruiseHive, "How ship draft affects a ship's performance." https://www.cruisehive.com/ship-draft/109970, 2024. Accessed: April 23, 2024.

[5] gdensmarine, "Draft survey." https://gdensmarine.com.my/draft-survey/, 2024. Accessed: April 24, 2024.

[6] http://tiny.cc/gv6zxz, 2024. Accessed: May 1, 2024.

[7] A. Krystosik-Gromadzińska, "The use of drones in the maritime sector–areas and benefits," *Zeszyty Naukowe Akademii Morskiej w Szczecinie*, 2021.

[8] "$2 million saved using elios 2 for oil tanker cargo inspection." https://www.flyability.com/casestudies/oil-tanker-drone-inspection, 2024. Accessed: April 29, 2024.

[9] "Autonomous drone-based surveys of ships in operation.." (Accessed 3-January-2024).

[10] F. Bonnin-Pascual, A. Ortiz, E. Garcia-Fidalgo, and J. Company-Corcoles, "A reconfigurable framework to turn a mav into an effective tool for vessel inspection," *Robotics and Computer-Integrated Manufacturing*, vol. 56, pp. 191–211, 04 2019.

[11] D. Waleed, S. Mukhopadhyay, U. Tariq, and A. H. El-Hag, "Drone-based ceramic insulators condition monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[12] L. López-Fernández, S. Lagüela, I. Picón, and D. González-Aguilera, "Large scale automatic analysis and classification of roof surfaces for the installation of solar panels using a multi-sensor aerial platform," *Remote Sensing*, vol. 7, no. 9, pp. 11226–11248, 2015.

[13] L. Ma, M. Li, L. Tong, Y. Wang, and L. Cheng, "Using unmanned aerial vehicle for remote sensing application," in *2013 21st International Conference on Geoinformatics*, pp. 1–5, 2013.

[14] N. Bolourian and A. Hammad, "Lidar-equipped uav path planning considering potential locations of defects for bridge inspection," *Automation in Construction*, vol. 117, p. 103250, 2020.

[15] T. He, Y. Zeng, and Z. Hu, "Research of multi-rotor uavs detailed autonomous inspection technology of transmission lines based on route planning," *IEEE Access*, vol. 7, pp. 114955–114965, 2019.

[16] Y. Shimizu, T. Ohtsuka, and M. Diehl, "A real-time algorithm for nonlinear receding horizon control using multiple shooting and continuation/krylov method," *International Journal of Robust and Nonlinear Control*, vol. 19, pp. 919 – 936, 05 2009.

[17] J. Dentler, S. Kannan, M. A. O. Mendez, and H. Voos, "A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors," in *2016 IEEE Conference on Control Applications (CCA)*, pp. 519–525, 2016.

[18] M. Greeff and A. P. Schoellig, "Flatness-based model predictive control for quadrotor trajectory tracking," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6740–6745, 2018.

[19] M. Owis, S. El-Bouhy, and A. El-Badawy, "Quadrotor trajectory tracking control using non-linear model predictive control with ros implementation," in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, pp. 243–247, 2019.

[20] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, 2020.

[21] , "System modeling." `https://wilselby.com/research/arducopter/modeling/`. Accessed: 2024-04-27.

[22] A. Sanca, P. Alsina, and J. Cerqueira, "Dynamic modelling of a quadrotor aerial vehicle with nonlinear inputs," *Latin American Robotics Symposium and Intelligent Robotics Meeting*, vol. 0, pp. 143–148, 10 2008.

[23] K. Thu and A. Gavrilov, "Designing and modeling of quadcopter control system using l1 adaptive control," *Procedia Computer Science*, vol. 103, pp. 528–535, 12 2017.

[24] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[25] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a quadrotor robot," 12 2006.

[26] R. Mahony, R. Beard, and V. Kumar, "Modeling and control of aerial robots," pp. 1307–1334, 01 2016.

[27] D. Brescianini, M. Hehn, and R. D'Andrea, "Nonlinear quadrocopter attitude control," 2013.

[28] M. W. Mehrez, "Optimization based Solutions for Control and State Estimation in Dynamical Systems (Implementation to Mobile Robots)." GitHub, 2024. Accessed: Apr. 26, 2024.

[29] T. S. Andersen and R. Kristiansen, "Reactive-based position control of an underactuated quadrotor," in *2020 European Control Conference (ECC)*, pp. 1656–1661, 2020.

[30] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems Control Letters*, vol. 39, no. 1, pp. 63–70, 2000.

[31] CasADi, "Driven mpc in simulink (part 1)," 2024. Accessed: 2024-05-01.

[32] , "Blender visulisation of simulation." `http://tiny.cc/yyw1yz`. Accessed: 2024-04-27.

[33] , "Prototype quadrotor manual flying video." `http://tiny.cc/cv6zxz`. Accessed: 2024-04-23.

A Digital Attachment

The digital attachment includes:

- Simulator: The simulator is located in the "Simulator" directory, containing a combination of Matlab script (.m) functions and Simulink (.slx) files. The Simulink model can be found in the "Simulation.slx" file. Running the simulation only requires pressing "Run" on the "Launcher.m" file, with or without opening the "Simulation.slx". However, additional scopes of important data have been added to "Simulation.slx," which can be kept open to view real-time plotting.

  The "Scenerio" needs to be changed in the "Launcher.m" file (line 23) as well as in "casadi_block.m" (line 114) to simulate each of the three scenarios mentioned in Section 5.2. Upon completion of the simulation, a 3D plot depicting trajectory tracking will appear automatically. After the simulation has been completed, 2D plots can be generated by running "Benchmark.m".

  After running the simulation in Simulink, we can visualize the results using a Blender animation. In the simulation folder, there is a file named "visual_quad.blend" with a .blend extension that needs to be opened using Blender software. Once opened, run the script "quadrotor_script" from the left panel and then press the "Play Animation" button located in the bottom panel to view the animation as shown in Figure 46.

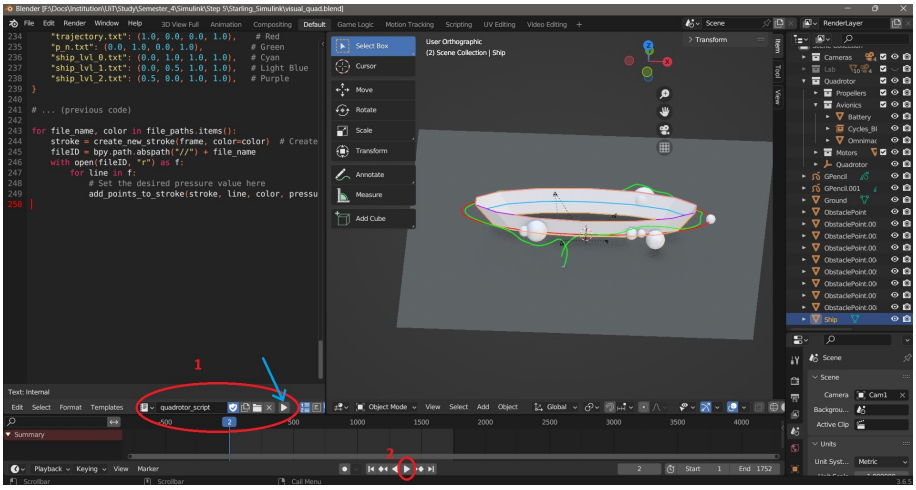


Figure 46: Steps to be followed to run the blender environment

It is to be noted that in the simulation animation, the ship's body surface only outlines the shape of the ship within the drone's flying

path and does not extend down to the ground (waterline).

- The "Videos" directory contains videos of manually flying the quadrotor and the screen recording of the blender animations.

B Parameters

Table 2: Quadrotor Model Parameters

Description	**Symbol**	**Parameter Value**
Quadrotor Mass	m	1.27 kg
Inertia x-axis	J_{xx}	0.0434 kgm^2
Inertia y-axis	J_{yy}	0.0434 kgm^2
Inertia x-axis	J_{zz}	0.0705 kgm^2
Thrust Constant	c_T	1.5×10^{-5}
Drag Coefficient	c_Q	1.89981×10^{-7}
Max Rotor Speed	Ω_{max}	600 rpm
Rotor Horizontal Distance from CG	d	0.149352 m
Rotor Vertical Distance from CG	h	0 m
Inertia of Propeller	I_b	4.0675×10^{-5}kgm^2
Root Pitch	θ_0	0.49 rad
Blade Twist	θ_1	-0.33 rad
Lift Slope at Set Point	a_0	5.7
Air Density	ρ	1.225
Propeller Radius	r	0.128016 m
Propeller Chord	c_0	0.027432 m
Induced Drag Coefficient	D	$diag\,(0.1,\, 0.1,\, 0)$

Table 3: Control Parameters

Description	**Symbol**	**Parameter Value**
NMPC Prediction Horizon	N	50
State Weighting Matrix (without obstacles)	Q	$diag\,(1.4, 1.4, 1.4, 1.0, 1.0, 1.0)$
Control Weighting Matrix (without obstacles)	R	$diag\,(0.8, 0.8, 0.001)$
State Weighting Matrix (with obstacles)	Q	$diag\,(1.4, 1.4, 1.4, 1.0, 1.0, 1.0)$
Control Weighting Matrix (with obstacles)	R	$diag\,(0.6, 0.6, 0.001)$
Attitude Control Gain	k_1	7
Attitude Control Gain	k_w	1.4

Table 4: EKF Parameters

Description	**Symbol**	**Parameter Value**
Process Noise Covariance Matrix	**Q**	$diag\,(7, 7, 7, 4, 4, 4, 5, 5, 5)$
Measurement Noise Covariance Matrix	**R**	$diag\,(1.4, 1.4, 1.4, 1.0, 1.0, 1.0)$