UiT The Arctic University of Norway

Faculty of Engineering Science

Automated multi-rotor draft survey of large vessels

Muhammad Usman

Master's thesis in Aerospace Control Engineering STE-3900 May 2022

Project description

The use of multi-rotors for various inspection tasks is a growing field of operations, as these are viable to operate unmanned under rather harsh conditions and decreases both operation time and cost for tasks traditionally performed manually. A particular inspection task suitable for assignment to multi-rotor is visual inspection of large vessels, and especially performing draft measurements of iron ore vessels at the port during loading and unloading. This is an operation that is traditionally performed manually, which requires reading of draft meter indicators located on the exterior of the ship hull (aft and stern) as shown in the picture below. These indicators are located both at the port and seaside of the ship, and access to the seaside indicator may be difficult. This is especially a problem under harsh weather conditions, darkness, and troubled waters.

This project focus on developing an autonomous multi-rotor solution for performing depth measurements of the vessel. From a docked position, the multi-rotor should be able to fly close enough to the draft indicators to perform a quality reading by a camera. The level of autonomy will be specified during the project, but as a minimum the multi-rotor should be able to capture video of the indicators for manual interpretation on land.

Subtasks

- Perform a literature review on unmanned multi-rotor inspection in general, and its use for maritime inspection in particular.

- Develop specifications for a multi-rotor that can perform the inspection and suggest necessary system components. This must include requirements for the multi-rotor itself, as well as additional equipment such as sensors, lights and camera, and necessities for onboard data handling.
- Develop a solution for path planning to perform the flight operation.
- Develop guidance and control algorithms for the flight operation to enable the multi-rotor to follow the designated path and perform the inspection.

  All results must be supported by mathematical derivations as well as performance simulations in MATLAB/Simulink.

Summary

In maritime sector draft survey has a significant importance as it is used to determine many important factors used in maritime transportation. Draft is the vertical displacement from the bottom of the keel (the bottom-most element of a vessel) to the water line (the line of meeting point of hull and the water surface). It is used to measure the minimum water depth for safe navigation of vessel and to evaluate mass of cargo in the vessel by the change in displacement on the draft scale after loading of the cargo in the vessel. Draft measurement of a vessel has a vital role in maritime sector to ensure a safe equilibrium between maximum and minimum cargo that can be loaded in the vessel. Draft survey performed at the time of loading and unloading of cargo (Iron Ore) at the Narvik port to read out draft markings traditionally involved a round trip around the vessel in a small crew boat and it is a time consuming and challenging task specially in darkness (during night), shadows and when difficult to safely reach the crew boat close enough due to anchors and buoys. The goal of this study is to develop an autonomous multi-rotor system that can survey the large vessel to capture all the necessary draft measurements by reaching close enough even in challenging environments like nighttime and in presence of obstacles. This involves developing the solution for path planning to perform flight operation autonomously, developing guidance and control algorithm for the flight operation to enable the multi-rotor to follow the designated path and perform the inspection while avoiding all the hurdles using collision avoidance system. Along with developing the specifications for a multi-rotor that can perform the inspection and suggest necessary system components including multi-rotor itself and additional components such as sensors, lights and camera, and necessities for on-board data handling.

Preface

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

It is written to provide the realization about utilizing a drone (quadcopter) for an autonomous draft survey of large vessels. Idea is originated based on the need of automating the draft survey procedure for LKAB's iron ore transportation vessels at Narvik harbour.

I would like to thank my supervisor, Prof.Raymond Kristiansen for his continuous help, support and for providing valuable suggestions from his vast knowledge and expertise in control systems. I would also thank head Lecturer Tom Stian Andersen for his valuable help in simulation design and mathematical modelling whenever needed. Further more thanks to Associate Professor Jose Juan Corona Sanchez for his suggestions and help in this thesis from his expertise in hardware and firmware design.

Last, I want to thank my family and friends for their support and motivation for completing this thesis and also through out master's studies.

Muhammad Usman
Narvik, May 2022

Contents

List of Figures

List of Tables

1 Introduction

Unmanned aerial vehicles (UAVs) that are commonly known as drones have gained momentous importance in maritime sector. They turn out to be successful at performing complex tasks with autonomy. Drones are used for missions that are risky for humans, costly and time consuming with traditional methods and remote operations. Drones can safely reach to places where humans cannot or should not go due to health and safety restrictions like water ballast tanks, oil containers, high points, edges, and remote locations. Drones' applications have capability to replace human involvement in an existing solution for example power line inspection using a helicopter and wind turbine inspection using ropes can now be performed using drones [1]. Technological advancements in autonomous drone technology allow real-time inspections which reduces risk, cost, time and need of human operator. Moreover, drones equipped with cameras are used to inspect difficult places to reach and transfer videos or pictures in real-time to concerned inspector or the whole inspection can be recorded and used later for detailed analysis. Drone use is not only limited to inspection, but they are also glowingly used for delivering goods, search, and rescue operations and real-time monitoring. This study is focused on utilizing a drone for automating a draft survey for large vessels that transport iron ore of LKAB from Narvik harbour.

1.1 Background and motivation

Narvik harbour is in the center of the city Narvik, Nordland, Norway. It consists of four sections: LKAB's bulk harbor, central harbor, deep-water quay and Skarveneskaien [2]. This harbour has its importance because of it's ice-free ports. Every year around 18–20 million tonnes of cargo is transported from Narvik harbour, most of this cargo include LKAB's iron ore [2]. LKAB is a Swedish government owned company, and it's a mining company that mines iron ore at Kiruna and at Malmberget in northern Sweden [3]. Iron ore generated from these mines is transported to the Narvik harbour and the port of Luleå and from there it is shipped to customers all over the world. Narvik port is the largest harbour of LKAB and it has annual capacity of almost 30 million tons of products related to iron ore [4]. In 2018 notified ship arrival for LKAB at the Narvik harbour was 250 [5].

Whenever something is sold to any customer it is measured or weighed to calculate the price of thing that is sold in total. In normal cases when someone buy fruit from a shop, the quantity is the fruit is weighed and the price is calculated based on the weight. Similarly in maritime sector the cargo being transported is always weighed not only to check the

price but also to make sure the right amount of cargo is loaded and then delivered to the customer. In maritime industry the cargo is divided into two main categories as general cargo (packed cargo such as containers, boxes, drums, or bags) and bulk cargo (unpacked cargo such as grain, coal, iron ore and gravel) [6]. Usually in case of general cargo the weight of single package is known so the total weight cargo loaded on the ship can be calculated. Whereas in case of bulk cargo like iron ore the total of weight of loaded cargo cannot be calculated in this way. Instead, it is calculated indirectly by taking depth measurements of the ship, that is vertical displacement of ship in the water (Draft of ship) before and after loading cargo. This is illustrated in Fig. 1

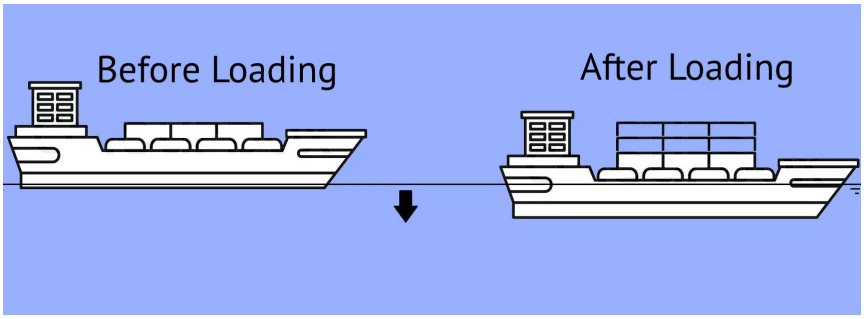

Fig. 1: Draft before and after loading [7]

Draft is the vertical displacement from the bottom of the keel (the bottom-most element of a vessel) to the water line (the line of meeting point of hull and the water surface). It is also used to measure the minimum water depth for safe navigation of vessel and to evaluate mass of cargo in the vessel by the change in displacement on the draft scale after loading of the cargo in the vessel. As shown in Fig. 1 if a vessel is loaded more heavily then it will sink deeper into the water and resulting in greater [7]. Draft measurement of a vessel has a vital role in maritime sector to ensure a safe equilibrium between maximum and minimum cargo that can be loaded in the vessel. It is also helpful in avoiding the point (in terms of load) at which the vessel will not be stable for navigation. If a vessel sinks extremely low in the water, then it became susceptible to flooding in slight waves as well. On the other hand, in the absence of cargo in the large vessels, they have tendency to raise in the water due to less draft. These vessels are lacking essential stability to normal natural phenomena like waves or winds, and under extreme condition this might results into a disaster [7]. A significant amount of international trade is being carried out by the use bulk cargo. That is also a reason that draft measurement plays an important role as these measurements are used to calculate weight of the cargo and these wight measurements are used for cargo claims [8]. Another

measurement that is known as air draft of the vessel is the distance from the top of the vessel to the water line. Air draft also have importance as it is needed when the vessel has to pass under a bridge, and the air draft measurement will ensure the clearance to pass under the bridges safely without any collision and damage to both bridge and the vessel [9]. Also, the revenue of shipping companies depends on increasing the amount of cargo shipped in the least amount of time with least cost. So, draft plays an important role to maintain safety for ship passage in the sea water. After loading the vessel with cargo sometimes it become tilted and a tilted vessel can be dangerous, to counter this the corrections in loading are made to make sure that draft on both side of the vessel matches to have zero tilt. The IMO (International Maritime Organization) specify the minimum and maximum drafts depending on the category whether it is oil, coal, high density material, passengers, or other material transportation.

Draft measurements are taken by reading the draft scale as shown in Fig. 2 painted on the sides of the vessel generally on the bow and stern side of the vessel. The draft scale might be painted on several locations depending on the size of vessel. Mostly the bulk carrier vessels have draft markings painted on six points. A survey is carried out to by

Fig. 2: Draft scale [8]

reading the draft on the draft markings at six standard points on the ship's hull. These six standard points are forward, mid ship and stern on both port and starboard sides. Port and starboard sides are left and right sides of the vessel when a vessel is viewed in the direction of its forward movement. Basic ship terms are illustrated in Fig. 3.

Draft survey is in general carried out by safety and inspection teams that belong to the vessel itself or to the port where vessel is docked. These teams have the task to inspect the cargo, inspection of the exter-

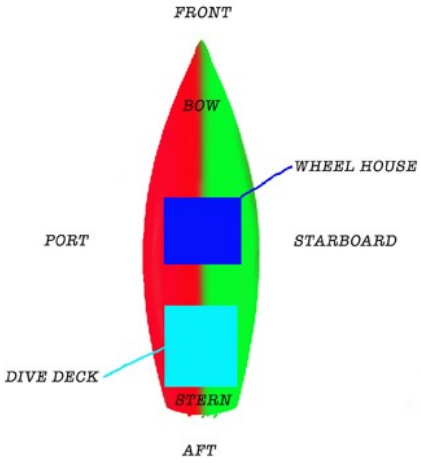

Fig. 3: Basic ship terms [10]

nal hull for any damage or wear and tear, inspection of propeller rudder and to measurement and verification of the draft of the vessel. Usually, the responsible officials use small crew boats or tugboats used for inspecting the vessel for any damage to make a round trip around the vessel to read the draft markings clearly by approaching the draft scale as close as possible as shown in Fig. 4. The concerned official then note

Fig. 4: Draft survey team reading draft scale [11]

down the draft measurements on a form and these measurements are then further used to calculate other parameters with the help of a computing mechanism build that involves certain charts and formulas or including other environmental factors such as water density [12].

It is significantly important to make sure that draft marks are read as accurate as possible. Reading the draft marks on the vessel is challenging and time consuming, However, some factors like marine growth on the marks, rust and poorly painted marks as shown in Fig. 5 make it more challenging to read it and might result in inaccuracy. Other fac-

Fig. 5: Poor paint and rust on draft scale [8]

tors like darkness (during night) shown in Fig. 6 and shadows create difficulty in carrying out these surveys.

Fig. 6: Draft scale captured during night [8]

Sometimes reading the draft marks during the night from a crew boat and from a distance is difficult and even not readable. As the Narvik city (Latitude 68.4385°North) lies inside the Arctic Circle (Lati-

tude 66.8° North) making nights longer and darker for almost 5 months as shown in Fig. 7. Also, weather conditions make visibility low in the

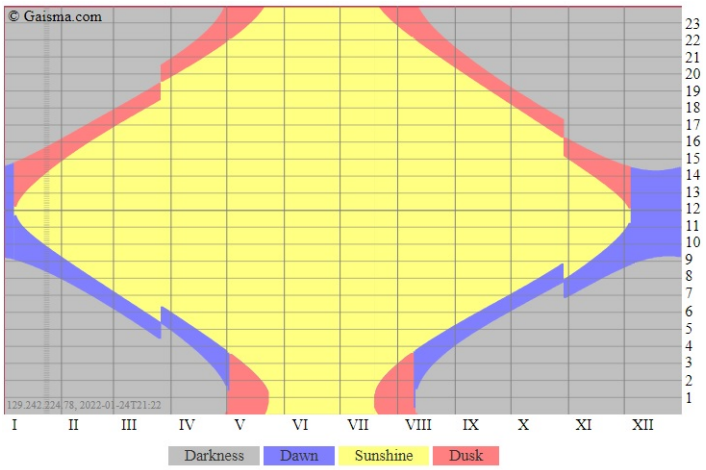

Fig. 7: Narvik Daylight Chart[13]

Narvik city. These factors cause it difficult to read the draft markings. Another challenge is that sometimes the crew boat cannot go safely close enough due to anchors and buoys. These factors make this task challenging and time consuming.

1.1.1 Proposed solution

To overcome the challenges of reading draft marks manually an automated and easy solution is to use a multi-rotor (Drone) that can fly around the vessel safely and read out the draft marks with the help of camera and other supporting equipment such as lights and sensors mounted on the drone. This is achieved by making the drone to follow a pre-planned trajectory around the vessel, that covers all the draft markings. This trajectory following is accomplished by a flight controller. A clear and stable video of all the draft scale will be recorded in the camera attached to the drone and will be later used by the operator to find out the draft scale readings.

1.2 Literature review

Although maritime sector has gone through a lot of technological advancements such as autonomous ships and smart propulsion systems, but still most of the draft surveys are carried out by traditional ways that is by using crew boat to visually read the draft marks. In recent years new methods are proposed for the automatic draft reading that

includes sensors, such as laser ranging sensor, ultrasonic ranging sensor, and liquid level sensors. But to use these technologies these sensors must be installed on the ship in advance [14]. And these solutions are expensive and may require modifications in the design of vessel.

Other measurements devices such as wave damping tubes that are very helpful to damp the wave in rough sea conditions. This tube must be attached to the vessels with the help of magnet or other means right over the draft marks. The water level in the tube needs to be viewed as close as possible and sometimes a rope ladder is required to do this when a boat is not available to perform this task [15].

To reduce human effort and to introduce autonomy several other measurement devices and methods have been proposed such as David Ray designed a portable draft measurement device that measures the distance from top of the vessel deck to the water level in which the ship is floating to measure air draft [16]. Huayao Zheng proposed a new system for level sensor for ship stability analysis and monitoring, with a pressure sensor installed below the level of water line of the ship [17].

Despite of these proposed methods most of the draft readings are taken manually based on visual observation as the first and most important step. Because of inaccuracies in sensor measurements these methods are not widely used. Other draft reading methods are proposed based on image processing such as Automatic draft reading based on image processing [14] and Ship Draft Detection Based on Machine Vision [18]. These methods can reduce the human error that can occur while reading the draft during visual observation. But image acquisition is done manually with conventional methods. Hence there is a need to automate the very first step in the draft survey that is to read the draft marks as accurate as possible.

The proposed solution in this study as mentioned in previous section is to use drone for draft survey. Drones are already glowingly used for delivering goods, search, and rescue operations and real-time monitoring. Some of the uses of drones in maritime are as follows:

- **Inspections:** This includes inspection of ship structure for wear and tear, hull inspection, and inspection of storage tanks (oil). Traditional methods of vessel inspection can be costly and vessel companies need a cost-effective solution. DNV GL has successfully performed inspection of a 19.4 meter high oil tank on-board a Floating Production, Storage and Offloading vessel [19].

- **Port traffic monitoring and control:** Drones deployed beyond shoreline to monitor vessel traffic to control the traffic efficiently [20].

- **Maritime Search and Rescue:** Drones equipped with optical radar VIDAR (Visual Detection and Ranging) are used to detect small ob-

jects on the sea surface. Recently two drowning swimmers were rescued with the help of a drone in Australia [21].

- **Security and Surveillance:** Drones are used for security and surveillance of ships and port installations with wide angle video instead of network of cameras mounted to cover a location. Nordic Unmanned and Textron are companies that won a contract with European Maritime Safety Agency (EMSA) regarding long-range general maritime surveillance [22].

- **Border coastal/offshore control:** Drones are used to keep an eye to prevent any illegal movement on coastline. Drone equipped with thermographic camera is used to detect anyone hiding in the ships [1].

- **Delivering goods:** Delivering of emergency supplies(medicines) to vessel at sea, delivering spare parts, documents, and other goods [23]. Recently Equinor achieved world's first logistic operation with the help of a drone [24]. Drone delivered a 3-D printed part for a lifeboat system from base to a platform in the North Sea.

- **Communication:** Drones equipped with antennas are used to provides 5G network coverage to the ships in areas where coverage is not available by communication towers and maritime satellites [25].

- **Emissions Monitoring:** Drones equipped with sensor that measure the level of CO2, SO2 and NO2 from ship exhaust are used for evaluation of ship fuel consumption remotely [23]. To reduce air pollution from shipping the international maritime organization has introduced standards and requirements for monitoring and documenting sulfur emissions from ships [26].

- **Oil spills detection:** Drones are used to track the oil spill to get an aerial overview of affected sea area that is needed by response team [27]. A company Nordic unmanned is providing emergency response for European Maritime Safety Agency (EMSA) [28].

During the COVID-19 pandemic social distancing and maintaining safety was strictly followed and the use of drones in this situation played significant role maritime sector. They were utilized for delivering goods to ships at sea while without any safety issue and maintaining distance [23]. They are also used in deliveries of medicines and other necessary items to offshore oil platform from the ships with no human contact for safety.

A general inspection of a structure is usually done by a drone equipped with camera and sensors required for guidance, navigation, and control

of drone. The drone fly to inspect the structure and the video or images are sent in real time to the ground station for processing or are stored on board and downloaded later for processing. Mostly the drones used for surveys and inspections are multi-rotors such as quad rotors, hexacopters and octocopters. These multi-rotors are generally categorized as vertical-takeoff-and-landing (VTOL) vehicles and capable to hover steadily in place. The goal for designing a drone is to complete task and carry out a mission human interaction as minimum as possible. However, achieving a fully autonomous solution is quite challenging and complex task. Whereas different level of autonomy can be considered for a specific task depending on complexity and existence. These levels of autonomy are described according to the National Institute of Standards and Technology (NIST) as follows [29].

- **Fully autonomous:** A fully autonomous drone complete a complex mission without any human guidance and all decision are made by on-board controller with available sensors measurements.
- **Semi-autonomous:** For complex missions involvement of a human operator is needed and the decision are taken by human operator when the drone is not capable of making a decision.
- **Teleoperated:** The human operator control the drone by sending commands based on the feedback data from on-board sensors such as based on surrounding obstacles data the operator can move the drone. And no direct visuals of drone are needed for this.
- **Remotely controlled:** The drone is completely manually controlled by a remote pilot with no data from the on-board sensor. And direct visual sight of drone is needed to control.

Depending on the level of autonomy the hardware and software design of the drone changes.

In recent years, drone-based inspection and survey have gained attention due to developments of micro-electromechanical systems (MEMS), micro inertial measurement unit (MIMU) and new flight control systems. Some of the methodologies and work done in inspection with drones are explained as follow

- **Bridge inspection using UAV [30]**

  In this research, design, and development of a UAV system for bridge inspection has been discussed. An open-source embedded flight controller (Pixhawk) is used for flight control. The on-board computer chosen was Raspberry Pi 2 with Robot Operating System (ROS) running on it. Integrated sensors like accelerometer, gyroscope, and magnetometer in Pixhawk are used to estimate attitude of UAV. For environment detection, collision avoidance and position estimation in the

absence of GPS they have used laser scanner, ultrasonic sensor, and a single point Light Detection and Ranging (LIDAR). In absence of GPS, measurements are bad (under the bridge), in that case fusion of laser scanner and accelerometer will done to estimate the position. A ground station platform has been designed as human-computer interaction to get images of the bridge in real-time and for mission planning that include setting a pre-planned map for inspection.

- **UAV-lidar supported intelligent power-line inspections [31]**

  In this research lidar sensor is used for power line inspection. Lidar can provide high-precision 3D measurements. As acquired lidar data lack the object properties so it needs more processing power. To overcome that issue this research proposed scene classification of raw lidar data. Lidar can provide accurate location for power line components. This concept is divided into two stages. First stage is generation of route based on lidar data. In this stage the position of power line component for guiding the route planning in the inspection is acquired using high precision of lidar. Then in next stage the UAV can achieve autonomous flight in inspection by following the pre-generated routes. For fault detection any kind of sensor (Such as camera) could be mounted on UAV. The standardized protocols for this inspections has four steps: point cloud classification, key point extraction, route generation, and fault detection as shown in Fig. 8. The Point cloud classification is used to segregate raw lidar data into subsets corresponding with certain objects. Purpose of key point extraction is to recognize feature points that represent the location of power line components that provide basis for guiding the route planning. Route generation is responsible to maintain a safe distance between the components of power line and UAV during the process of inspection. Path finding and search algorithms can be used to generate a safe route.

- **Research on multi-rotor for detailed autonomous inspection of transmission lines using route planning [32]**

  In this research flight route planning for autonomous inspection is presented. Initially flight inspection is manually controlled according to the pre-planned flight route. When UAV fly to target point the location information of waypoint, relative attitude, angle of photo and other related attributes will be acquired from the flight control via function of way-point planning, and the way-point information is transmitted to the software for ground station, which will be recorded this information as a waypoint. Then UAV will manually fly (with manual control) towards next target point and similarly the next way-point information will be recorded. Each way-point information

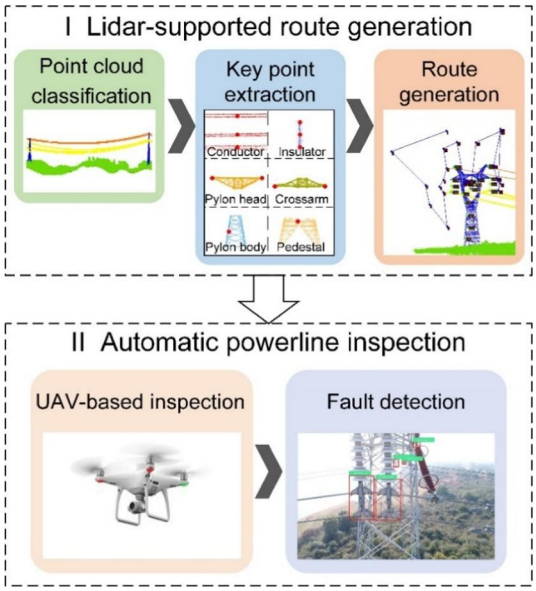

Fig. 8: Illustration of the proposed lidar-supported intelligent power line inspection concept [31]

will then be recorded and then way-points are connected to generate flight routes. The planned route will then be sent to flight control via ground station software and UAV can automatically perform the flight inspection. The way-point planning function is shown in Fig. 9.

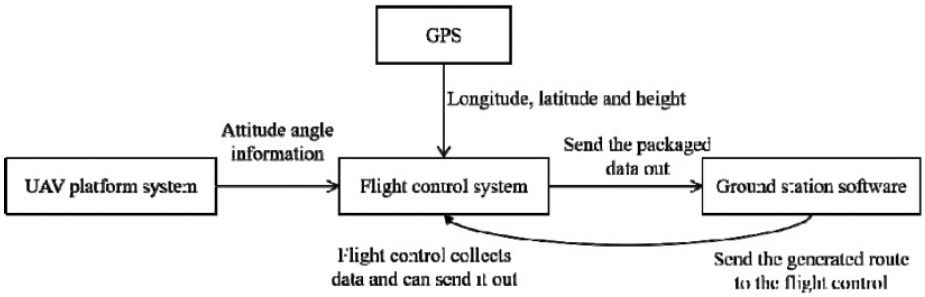

Fig. 9: Way-point planning function [32]

- **A re-configurable framework to turn a micro-aerial vehicle (MAV) into an effective tool for vessel inspection [33]**

  In this paper a framework is represented to change a Micro-Aerial Vehicle (MAV) into a flying camera that can allow human surveyor to virtually inspect different structures in the vessel hull. The architecture of system is developed to be re-configurable such that it is

compatible with different sensor suites for a proper state estimation. The control software is designed based on Supervised Autonomy (SA) paradigm, so that it can provide collision avoidance, while the human surveyor is responsible to supply motion commands while inspection. The SA framework is composed of different concepts such as Self-preservation: for preservation of platform against any hazard such as collision from obstacle and it is performed by MAV itself. Instructive feedback: This provides the operator with images of environment around MAV. Qualitative instructions: These are instructions by human operator to command MAV. Qualitative explanations: This is to tell human operator that what is going in the mission in a language that is easily understandable by operator such as obstacle detected, going up or forward. User interface: That allow human operator to get feedback from MAV and to send command to the MAV. This is illustrated in Fig. 10.

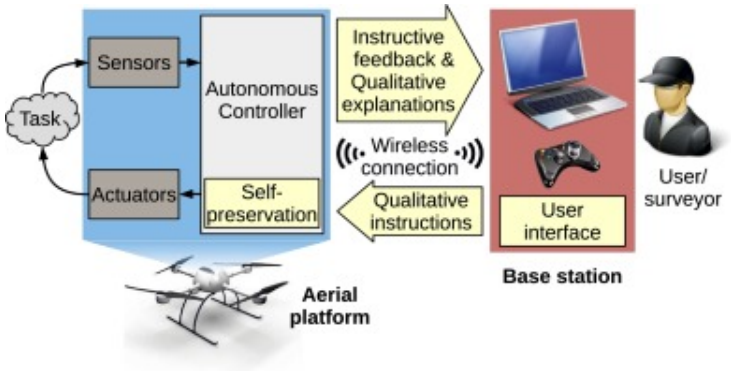

Fig. 10: Inspection system with supervised autonomy [33]

The proposed control architecture is designed based on layered structure, where each layer refers to a different control level. The architecture of this system is shown in Fig. 11. The high-level control layer accomplishes the MAV behaviors module. This contain a set of robotic behaviors which are responsible to full-fill the commands provided by operator such as obstacle detection and collision avoidance. In simple words, it combines commanded speed with the speed generated from sensor data through a reactive based control strategy to generate the desired velocity command. The sensors suites proposed are combinations of IMU, Ultrasonic range sensor, optical range sensor and optical flow sensors.

- **A survey on vision-based UAV navigation [34]**

  In this article several vision-based methods that are used for UAV navigation are presented. Specifically for visual localization and map-

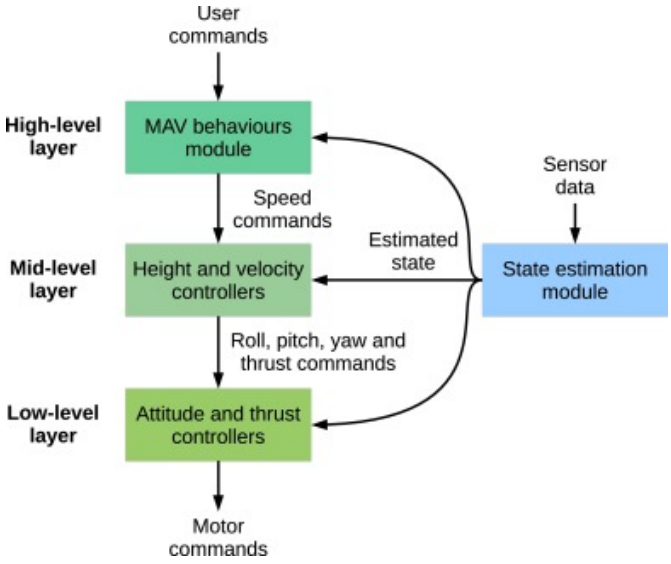

Fig. 11: Control Architecture [33]

ping, obstacle avoidance and path planning, which make the most important parts of visual navigation. Visual navigation uses visual sensors such as monocular cameras, stereo cameras, RGB-D cameras, and fish-eye cameras. A vision based navigation is illustrated in Fig.12. In this article Visual localization, and mapping systems are classified into three categories: map-less system, map-based system, and map-building system.

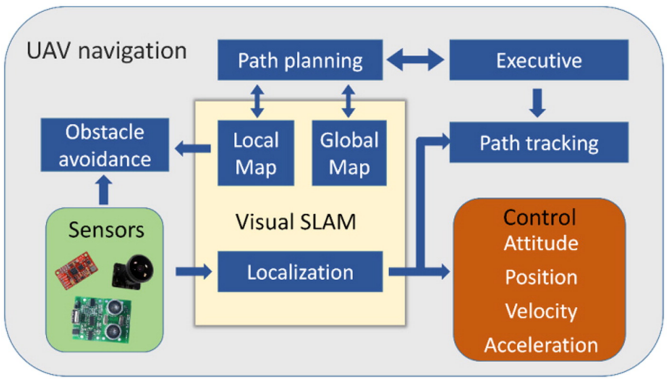

Fig. 12: Vision based navigation [34]

A map-less system navigates without know map and UAV navigation is done by acquiring distinct features in the observed environment. Most used method in map-less navigation is optical flow and feature tracking method. Optical flow method has two categories: local and

global. One method for optical flow that is discussed in this article is to calculate the optical velocity from two cameras (mounted on both side of robot) relative to wall. If both are same, then robot moves along the central line, otherwise it moves along the speed of small places forward. This method has poor performance in texture less environments. The feature tracking method tracks invariant features of moving objects, as well as lines, corners, and so on and discover the movement of an object by detecting the features. In map-based systems the layout of environment is predefined, allowing UAV to navigate. Map-building systems are used in both autonomous and semi-autonomous fields. They allow building maps at the time of flight in environments where it is difficult to navigate with preexisting map. SLAM based method discussed in this article for map building. Map and map-less navigation are shown in Fig. 13.

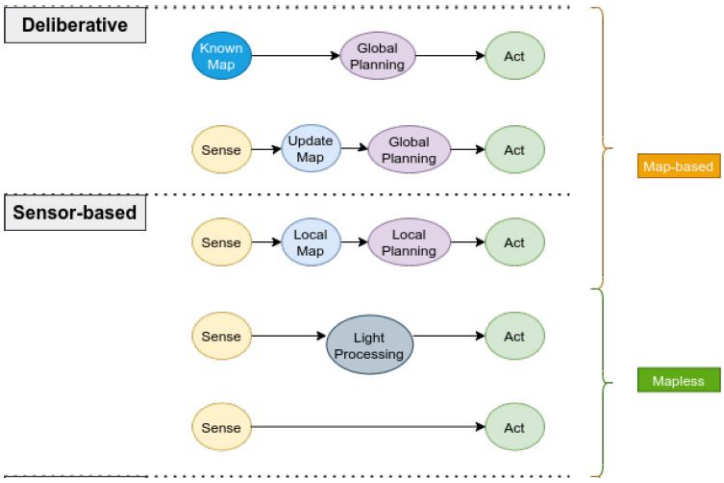

Fig. 13: Map and Map-less navigation [35]

- **The automatic flying security drone (AFSD) [36]**

  In this paper flying security drone is proposed. This AFSD is aimed to patrol petrol a route set by security personnel and stream live video data to the person wireless. The AFSD is capable of navigating via set GPS coordinates, and automatically updating its path in any deviation. The control of the UAV is done be human operator with the help of computer application. With simple click of the mouse the operator can path of the vehicle by defining new coordinates. The new route can be updated anytime during the flight. As well as the operator can see the actual coordinates of UAV on the application screen. The navigation of AFSD will be done by storing a series of GPS coordinates in its on-board memory and travelling to each location.

1.3 Objectives

The main objective of this study is to propose a solution for automating draft survey procedure for iron ore vessels. This includes a detailed literature review about existing technologies and methodologies for draft survey and research work for advancement in draft survey procedure. All the hardware specifications for proposed multi-rotor will be covered in this study. These specifications include type and design of multi-rotor itself and the additional components needed to make the multi-rotor capable of performing the designated task of automated draft survey. Additional components will be considered in this study based on the key sub objectives for a draft survey. Sub objectives are to record all the draft marking while flying around the vessels, and autonomous, safe, and stable flight operation. Additional components such as camera, lights, flight controller for data processing, sensors for navigation and a safe collision free flight will be addressed in this study.

Other than hardware specifications algorithms for smooth path generation for the flight operation, control algorithm that will make the multi-rotor to follow the desired path stably and an algorithm for avoiding any obstacle in the planned path. All these algorithms will be covered with mathematical details and will be implemented in a simulator such as MATLAB to analyze the performance, and results from these simulations will show the working of proposed algorithms.

1.4 Contributions and delimitations

As discussed in previous sections draft survey is a time-consuming task and require resources such as tug-boats and operators to perform this survey.

The contributions of this study to overcome this issue are as described as follow

Automating draft survey

To the best of the authors' knowledge, this thesis provided a first attempt in proposing the use of drone technology for an automated draft reading process. This study proposed a solution for automating this task and that will not only save time and effort but also the resource that is utilized in a traditional draft survey.

Trajectory planning

This thesis has provided an understanding of how a trajectory can be generated for surveying of large vessel or vessel like structures. The

generated trajectory and performance of drone on this path is demonstrated in this work.

Control

Control of drones (specifically quadcopters) is derived and modelled in this study. This control design also illustrated the mathematics of trajectory tracking control and its performance in the form of results for a trajectory around a vessel.

Collision avoidance

To achieve a safe autonomous flight operation, it is necessary to autonomously avoid obstacle in the planned trajectory. A detailed mathematical derivation and control law design for collision avoidance is demonstrated in this thesis. Performance of this avoidance controller is demonstrated using results and present different performance parameters of avoidance controller. This performance is demonstrated by considering different scenarios.

Design specifications

Design related requirements that include actual drone and other essential components that will help in completing the autonomous survey, are discussed in this study. A complete list of all the components depicting how the actual drone should look like is presented in this thesis.

Simulation design

All the mathematical modelling of system, controllers and collision avoidance algorithm elaborated in this study is also implemented on a simulator and detailed model of this simulation is discussed in this report. Results generated from this simulation are also explained.

There are some limitations of this work as this work is simulated by assuming general shape of vessel, therefore the flight path generated around the vessel is not applicable to vessels that have different shapes. Another assumption is that the vessel is docked at a known location, any change in the location of the vessel will need modifications in the flight path. Collision avoidance algorithm used in this study is limited to obstacles with already know dimensions. As this is a very early stage of the concept, so further research will be done on these results to make it generalized.

1.5 Report outline

This report is divided into eight chapters. **Chapter 1** covered the introductions about the problem itself, background about the concepts that will be used, a detailed literature review which covered existing solutions and challenges about particular problem and objectives of this report. In **Chapter 2** some theoretical and technological terms that are continuously being used later in this report are derived and elaborated here. **Chapter 3** is about expressing the system model in mathematical expression so that a model of the system can be created that can be used for designing of control systems and needed algorithms later in this chapter. In **Chapter 4** all needed peripherals and hardware components along with their specifications that are required to make the drone are explained. **Chapter 5** discussed the implementation of mathematical model that is created in Chapter 3 in a simulation. A detailed explanation of working of the simulation and all components of simulation are covered in this chapter. All the results and performance of the system is presented in the form of results generated from simulations in **Chapter 6** and these results are discussed and justified. **Chapter 7** is about challenges and problems faced during work on this thesis and also limitations of this work are discussed here. **Chapter 8** express the final conclusions about the work done and results achieved and also recommendations for the future work that can be done on this study.

2 Preliminaries

2.1 Theoretical/Technological preliminaries

2.1.1 Frame of reference

A frame of reference or reference frame is a system composed of set of coordinates and with respect to which motion or attitude of an object can be observed. There are several types of reference frames but in scope of this thesis two are considered that are body frame and inertial frame. If reference frame is fixed or moving with a constant velocity, it is called an inertial frame. Origin of inertial frame is fixed to the Earth, and in this study the inertial frame is fixed to the harbour where vessel is docked. Body frame is always attached to the body of vehicle, and it moves and rotate with the body of vehicle. Origin of body frame is usually placed at the center of gravity of the vehicle. A local reference frame (NED - North, East, and Down) that is fixed to the origin of inertial frame and that origin is fixed to the vessel docking location, and this is used in this thesis to represent position and velocity of vehicle in body frame and inertial frame. NED frame contains of three orthogonal axes in which X-axis points to true North, the Z-axis points downwards towards the Earth and Y-axis points towards the East. These two different reference frames are shown in Fig.14.

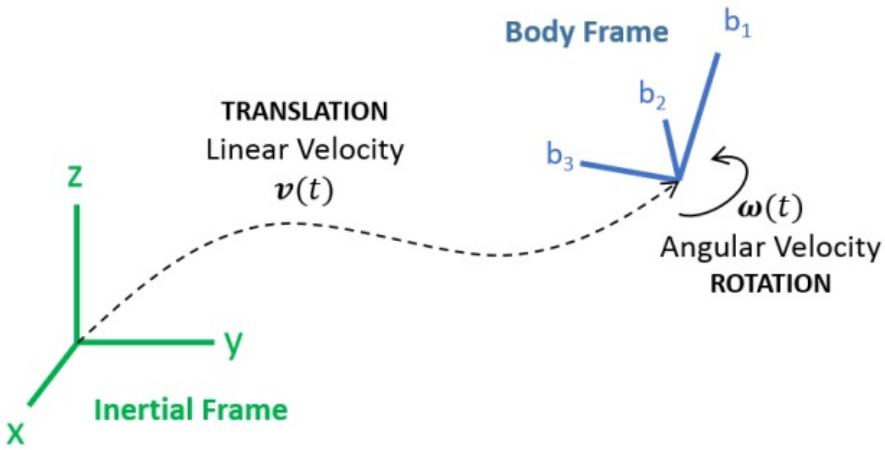

Fig. 14: Inertial and Body fame

2.1.2 Euler angles (Orientation)

Orientation of vehicle can be represented in different ways that usually involves angle representation. The most common representations are Euler angles and Quaternions or trigonometric functions. In this thesis Euler angles are used as they are simple, easy to understand and easy to calculate or performing different mathematical operations. Although Euler angles have singularity issue that cause gimbal lock in the rotating vehicle, whereas quaternions do not have this singularity issue, but they are difficult to understand and compute. Euler angles are represented as Roll (ϕ), Pitch (θ) and Yaw (ψ). These angles are illustrated in Fig.15.

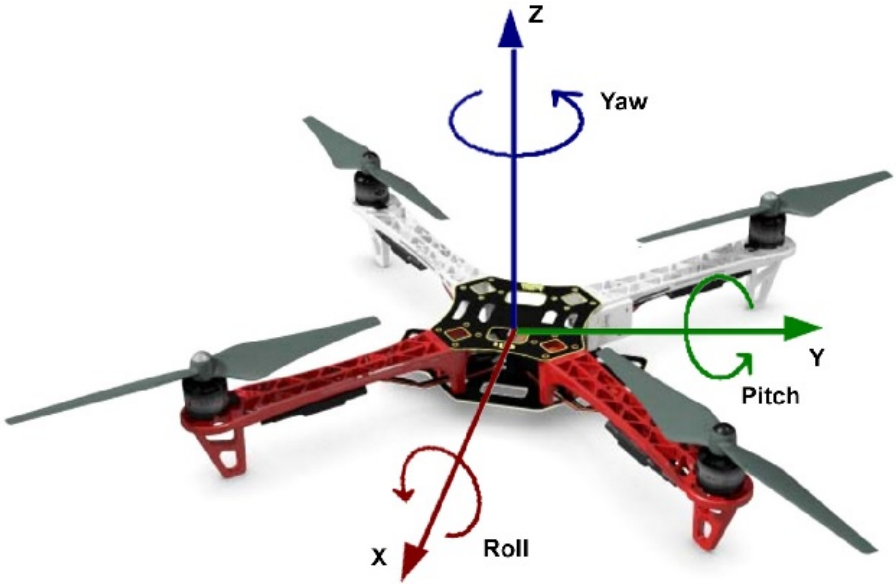

Fig. 15: Euler angles in Quadcopter [37]

2.1.3 Rotation and Transformation

In this section the rotation between inertial frame to body frame and vice versa will be described using Euler angles. Euler angle method is widely used and clearly understood and interpreted. The angular velocities are measured by gyroscope sensor that is attached to body of vehicle and rate of change of Euler angles is in inertial frame. Rate of change of Euler angles and angular velocities are not the same, because angular velocity is rate of change about the body axis whereas Euler angles represent rotations in its own reference frame. These entities are related to each other in kinematics and dynamics equations, and sometimes these entities are rotated from one frame to another. The

rotation about each principle axis is written in equations 2.1, 2.2 and 2.3.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.1}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2.2}$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

The matrix R_b^i that represents rotation from body frame to inertial frame is the product of above three rotations in zyx sequence:

$$R_b^i(\psi, \theta, \phi) = R_z(\psi) R_y(\theta) R_x(\phi) \tag{2.4}$$

$$R_b^i(\psi, \theta, \psi) = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\theta & c\psi s\theta c\phi + s\psi s\theta \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\theta & s\psi s\theta c\phi - c\psi s\theta \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.5}$$

where $c = cos$ and $s = sin$. Similarly the matrix that represents rotation from inertial to body frame is transpose of R_b^i:

$$R_i^b = {R_b^i}^T \tag{2.6}$$

The transformation matrix of angular rate transformation from body angular rates $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$ to Euler angular rates or angular velocity ($\omega = [pqr]$) is described as:

$$\omega_{ib}^b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_x^T(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_x^T(\phi) R_y^T(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{2.7}$$

$$\omega_{ib}^b = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.8}$$

$$W^{-1} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{2.9}$$

$$W = \begin{bmatrix} 1 & 0 & s\theta t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi & \\ 0 & s\phi/c\theta & c\phi/c\theta & \end{bmatrix} \tag{2.10}$$

where $c = cos$, $s = sin$ and $t = tan$.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W\omega_{ib}^b \tag{2.11}$$

2.1.4 Moment of inertia

Moment of inertia is a quantity that determines the required torque for a desired amount of angular acceleration in a body or it is a measure of how difficult it is to change angular velocity of rotating object. In easy way it can be described as how resistant an object is towards any changes in its rotational motion about a particular axis. Moment of inertia depends upon the distribution of mass within the object and the location of the axis around which the object is rotating. An object can have many different moments of inertia depending upon the position of its axis of rotation. Calculation of moment of inertia indicates the force (torque) an object will need to increase, decrease, or stop its rotation. Moment of inertia is sum of the product of mass of each particle with square of distance from its axis of rotation. As moment of inertia depends upon number of axes of rotation and in the case of drone there are three axis or rotation that are (XYZ), so the moment of inertia will be around three axes, and can be written in matrix form as:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{2.12}$$

In equation 2.12 the off-diagonal elements (I_{xy}, I_{xz}, I_{yz}) are products of inertia that describes an imbalance relative to coordinate axes. The diagonal elements (I_{xx}, I_{yy}, I_{zz}) are moments of inertia along the principal axes X, Y and Z respectively.

We assume that mass in the drone is distributed symmetrically which means the moment of inertia on the opposite sides of the axis of rotation will cancel each other. Resulting all the off-diagonal elements equal to zero $(I_{xy} = I_{xz} = I_{yz} = 0)$. Then inertia matrix will be simplified as:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2.13}$$

Moment of inertia calculation is very important factor for the control of quadcopter as it defines the amount of torque needed to produce angular acceleration around an axis. There are different methods to estimate moment of inertia. One method is to divide the body into several components and compute moment of inertia of each component separately and then add moment of inertia of all components around X, Y and Z axis of body. Based on assumption that mass is distributed symmetrically on each arm and the vehicle is rigid and this makes it easy to add the moment of inertia of each arm for x, y and z axis [38].

Moment of Inertia along (Yaw) Z-axis:

Yaw moment of inertia is along z-axis, and it will resist the yaw moment. It can be computed as the sum of moment of inertia of four arms of quadcopter:

$$I_{zz} = \frac{1}{3} \times ml^2 \tag{2.14}$$

$$I_{zz} = \frac{1}{3} \times ml^2 \times 4 \tag{2.15}$$

Where l indicate arm length of quadcopter.

Moment of Inertia along (Roll and Pitch) X and Y axes:

Moment along X and Y axis will be same because quadcopter is assumed to be symmetrical. So, it can be computed same as yaw moment of inertia just by changing the arm length by the perpendicular distance of motor from center of quadcopter.

$$I_{xx} = I_{yy} = \frac{1}{3} \times ml_s^2 \tag{2.16}$$

Where l_s indicate perpendicular distance of motor from center of quadcopter.

3 Mathematical Modelling

Mathematical modelling is the process of describing a real physical system in mathematical terms such as equations or graphs. This mathematical description helps to understand the insight working of the system and then also help in improving and making the system more efficient depending on our need. To begin with modelling of our multi-rotor (Quadcopter) we need to understand the motions and forces acting or generated by it.

A quadcopter has four propellers mounted on four motors and their rotation cause all the rotations and motions. There are two configurations of a quadcopter: Plus (+) and X configurations. There is no mechanical difference between these two configurations, but the only difference is where the front of quadcopter is. Fig. 16 show the difference between two configurations.

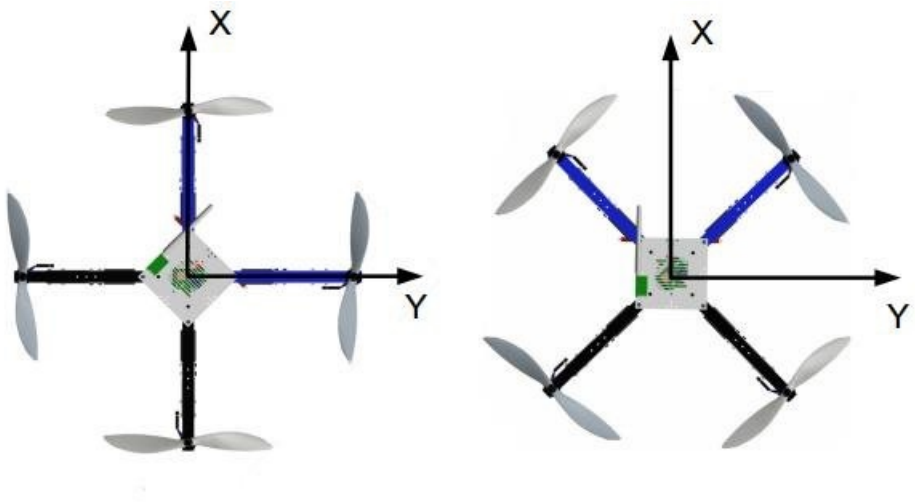

Fig. 16: Two configurations of the quadcopter[39]

Each motor generates both thrust and torque about the center of rotation of the quadcopter. Motion along vertical direction (Altitude or Throttle) is achieved by increasing or decreasing the speeds of all four motors at the same rate. Motion along any direction in horizontal axis (Roll and Pitch) is generated by decreasing the speed of motors along the desired direction and at the same time increasing the speed of motors on the opposite side of the desired direction. Whereas yaw is obtained by creating a rotation along the center of quadcopter, and this rotation is caused by introducing a torque by changing the speed of motors. A CW (clockwise) yaw is achieved by decreasing the speed of CW rotating

motors and simultaneously increasing the speed of CCW (Counterclockwise) rotating motor. It will induce a net CW torque and causing a CW yaw angle change. All these movements in X configuration are shown in Fig. 17. The blue colored circular arrow in Fig. 17 indicate the in-

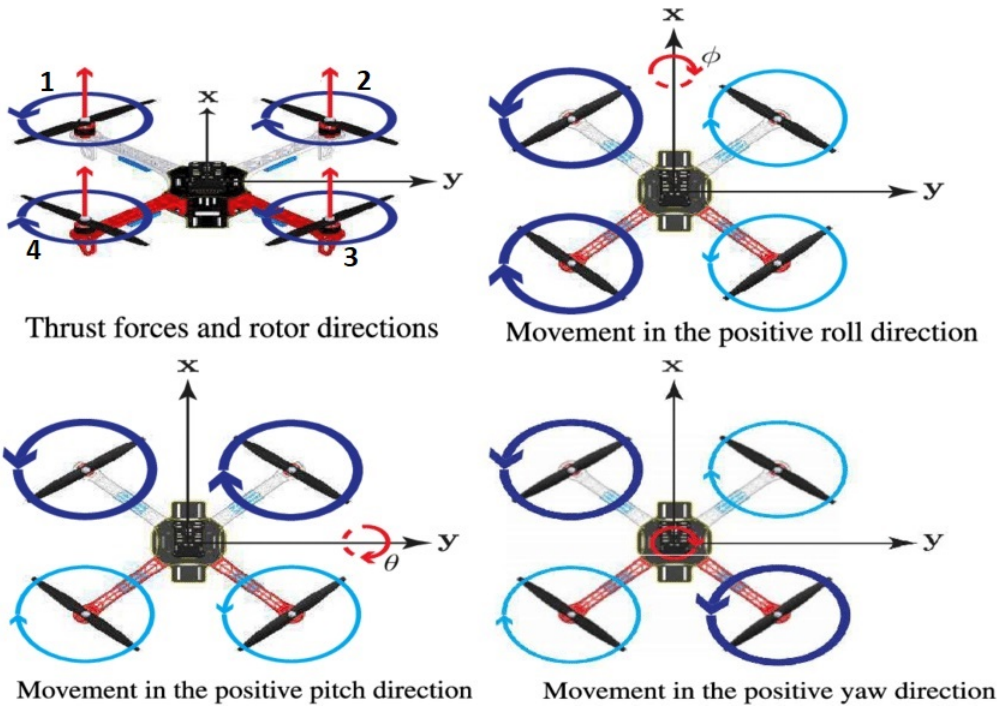

Fig. 17: Throttle, Roll, Pitch and Yaw movements in X configuration [40]

creased speed of rotation of motor as compared to cyan colored. Any certain movement in quadcopter is achieved by changing the input (Speed) of these four motors. These inputs are usually represented as follows:

- Throttle (u_1)
- Roll (u_2)
- Pitch (u_3)
- Yaw (u_4)

The mathematical model of quadcopter is developed using the dynamics of the quadcopter that is based on the following assumptions.

- quadcopter is a rigid body, means its shape and dimensions will not change.
- Propellers are also rigid.

- Origin of quadcopter in the body frame and center of gravity coincide.

3.1 Kinematics

The relationship between position derivative and velocity of vehicle in inertial frame is written as

$$\dot{p^i} = V^i \tag{3.1}$$

and relationship between position derivative in inertial frame and velocity in body frame is represented as

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_b^i \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.2}$$

For rotational kinematics the relation between Euler angles and the angular velocities in body frame is represented using transformation as

$$\dot{E} = W\omega_{ib}^b \tag{3.3}$$

Where $\dot{E}$ represent derivative of Euler angles, ω_{ib}^b angular velocities of body frame w.r.t inertial frame resolved in body frame.

3.2 Dynamics

As we already made assumption that quadcopter is a rigid body so Newton-Euler equations can be used to model dynamics. In body frame, the force that is needed for acceleration of mass $m\dot{V^b}$ and centrifugal force $\omega_{ib}^b \times (mV^b)$ are equal to the gravity gravitational force acting and thrust generated by rotors,

$$m\dot{V_b} + \omega_{ib}^b \times (mV_b) = R_i^b f_g - f_T^b \tag{3.4}$$

In equation 3.4 f_g is gravitational force and f_T^b is thrust force in the body frame, and these forces are written as

$$f_g = m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, f_T^b = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

Where T is total thrust generated by propellers. Whereas, in inertial frame centrifugal force is of no effect, only gravitational force and thrust force with both magnitude and directions are responsible for acceleration

$$\begin{aligned} m\dot{V^i} &= f_g - R_b^i f_T^b \\ \dot{V^i} &= \tfrac{1}{m} f_g - \tfrac{1}{m} R_b^i f_T^b \end{aligned} \tag{3.5}$$

Similarly, in body frame angular acceleration due to inertia $J_b\dot{\omega}_{ib}^b$ and centripetal forces $\omega_{ib}^b \times J_b\omega_{ib}^b$ are equal to external torques on the quadrotor

$$\begin{aligned} J_b\dot{\omega}_{ib}^b + \omega_{ib}^b \times J_b\omega_{ib}^b &= \tau^b \\ J_b\dot{\omega}_{ib}^b &= \tau^b - \omega_{ib}^b \times J^b\omega_{ib}^b \end{aligned} \tag{3.6}$$

Equations 3.3, 3.5 and 3.6 can be written in component form as:

$$\begin{aligned} \ddot{X} &= -\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right)\frac{F_T^b}{m} \\ \ddot{Y} &= -\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right)\frac{F_T^b}{m} \\ \ddot{Z} &= g - \cos\phi\cos\theta\frac{F_T^b}{m} \\ \dot{\omega_x} &= \frac{1}{I_x}\left(\tau_x - (I_z\omega_y\omega_z - I_z\omega_y\omega_z)\right) \\ \dot{\omega_y} &= \frac{1}{I_y}\left(\tau_y + (I_z\omega_x\omega_z - I_x\omega_x\omega_z)\right) \\ \dot{\omega_z} &= \frac{1}{I_z}\left(\tau_z - (I_y\omega_x\omega_y - I_x\omega_x\omega_y)\right) \\ \dot{\phi} &= \omega_x + \sin\phi\tan\theta\omega_y + \cos\phi\tan\theta\omega_z \\ \dot{\theta} &= \cos\phi\omega_y - \sin\phi\omega_z \\ \dot{\psi} &= \frac{\sin\phi}{\cos\theta}\omega_y + \frac{\cos\phi}{\cos\theta}\omega_z \end{aligned} \tag{3.7}$$

3.3 Forces and moments

Linear acceleration on the quadcopter is generated due to the net forces acting. The thrust force and gravitational pull are the two forces that are acting on quadcopter. Thrust force generated by each motor can be modeled using momentum equations and it is expressed as

$$F_i = C_T\rho A_{r_i} r_i^2\omega_i^2 \tag{3.8}$$

Where ω_i is rotational speed of the number of propeller (1,2,3,4) in rad, ρ is air density in kg/m^3, r is radius of i_{th} propeller and C_T is aerodynamic thrust coefficient. We can equate all the constant in above equation as $c_T = C_T\rho A_r r^2$ and rewrite equation as:

$$F_i = c_T\omega_i^2 \tag{3.9}$$

As all four motors have physical properties and dimensions so the total thrust force will become sum of individual force generated by each rotor

and it is represented as

$$F_T^b = c_T \sum_{i=1}^{N} \omega_i^2 \tag{3.10}$$

And the gravitational force will pull the quadcopter downward relative to the inertial frame, and it is written as:

$$F_g = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{3.11}$$

Torques acting upon the quadcopter are main reason to cause angular acceleration. According to Newton's third law of motion the rotating propeller will produce a counter reaction torque due to rotor drag that will be acting on the airframe and it can be modeled as

$$\tau_i = C_Q \rho A_{r_i} r_i^2 \omega_i^2 \tag{3.12}$$

where C_Q is aerodynamic drag coefficient and after equating constant in above equation as $c_Q = C_Q \rho A_r r^2$ this can be rewritten as

$$\tau_i = c_Q \omega_i^2 \tag{3.13}$$

The net moment created from the aerodynamics that is combination of thrust forces generated from the motors and air drags can be represented as

$$\tau = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} Roll \\ Pitch \\ Yaw \end{bmatrix} \tag{3.14}$$

The rolling torque is generated by the difference between thrust force of motor pair 1–4 and 2–3 and this is written as

$$\tau_x = L[(F_1 + F_4) - (F_2 + F_3)] \tag{3.15}$$

Where L is moment arm length for each motor, that is distance from the center of quadcopter. We can insert expression for F from equation 3.6 and rewrite as

$$\tau_x = L[(c_T \omega_1^2 + c_T \omega_4^2) - (c_T \omega_2^2 + c_T \omega_3^2)] \tag{3.16}$$

Similarly, pitching torque is generated by the difference between thrust force of motor pair 2–4 and 1–3 and this is written as

$$\tau_y = L[(F_1 + F_2) - (F_3 + F_4)] \tag{3.17}$$

$$\tau_y = L[(c_T \omega_1^2 + c_T \omega_2^2) - (c_T \omega_3^2 + c_T \omega_3^2)] \tag{3.18}$$

Reaction torque from all motors will induce Yaw torque around the center of the quadrotor. Direction of this depends on the direction of motors spin. Clockwise rotating motors generate a positive moment and anti-clockwise rotating motor will induce negative moment. This is modeled as

$$\tau_z = \sigma_i \sum_{i=1}^{N} c_Q \omega_i^2 \tag{3.19}$$

where $\sigma_i \in \{-1, +1\}$ represents direction of rotation fori_{th} motor: +1 indicates clockwise and -1 indicates anti-clockwise rotation. All the above computed forces and torques on the quad-rotor can be written in matrix form as

$$\begin{bmatrix} F_T^b \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ lc_T & -lc_T & -lc_T & lc_T \\ lc_T & lc_T & lc_T & -lc_T \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{3.20}$$

3.4 Control

As discussed in previous section that control inputs are responsible for generating desired forces and torques in quadcopter. The purpose of a dedicated control system is to stabilize the quadcopter in terms of attitude and altitude at the required position and follow a desired reference trajectory. Although a lot of control strategies have been available to design a control scheme of quadcopter ranging from very basic to advanced and machine learning based control. But the type of control scheme is limited to implementation cost and requirements in the system operation. Based on the requirements I have opted for PID (Proportional Integral and Derivative) control for altitude, position, and attitude. For trajectory tracking and collision avoidance another control scheme from (**Ha Le Nhu Ngoc Thanh,Bui and Hong [41]**) has been selected. This paper has proposed a nonlinear Control for trajectory tracking along with collision avoidance system based on geometric relations.

A PID controller works on three main terms that are Proportional, Integral and Derivative. Error signal that is difference between a reference or desired point and an actual value is the input to a PID controller and this input is then multiplied by PID gains to achieve a controlled and stable process. Proportional term (P) simply scales the error signal by multiplying error by a constant gain. Integral term (I) integrates the error signal with respect to time until it the error is reduced to zero and the (I) term removes the steady state error. Derivative term (D) as name suggest that it produces an output that is proportional to the derivative of error signal with respect to time that is rate of change of the error signal. D term actually minimize the change of error so it act

as a damper and remove oscillations in the system.

A PID controller can be written in mathematical form with a reference/desired signal $r(t)$, actual output $y(t)$, error signal $e(t)$, and the controller output $u(t)$ as follows:

$$\begin{aligned} e(t) &= r - y(t) \\ u(t) &= K_p e(t) + K_I \int_0^t e(j)dj + K_d \frac{de(t)}{dt} \end{aligned} \tag{3.21}$$

A general structure of a PID controller is shown in Fig. 18. A quad-

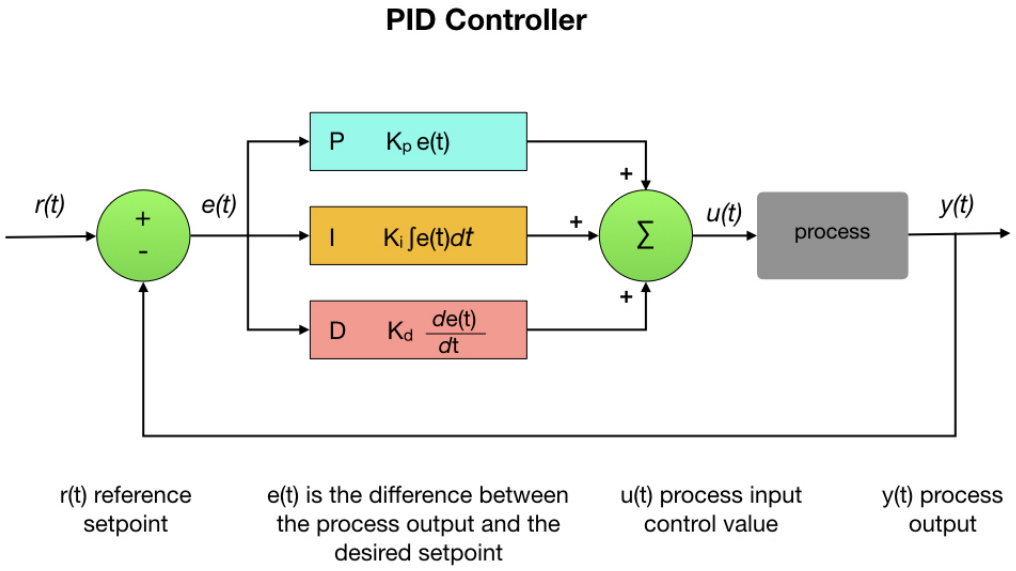

Fig. 18: General structure of PID controller[42]

copter has six main outputs$(X, Y, Z, \phi, \theta, \psi)$ where as it has only four control inputs $(F_T^b, \tau_x = \tau_\phi, \tau_y = \tau_\theta, \tau_z = \tau_\psi)$. This constraint is solved by decoupling the controller into two distinct loops, one inner loop that deals with the attitude variables and second outer variable dealing with the position variables. The rotational motion of the quadcopter is independent of translational variables, whereas the translational motion is dependent on the rotational motion that are Euler angles. The purpose of splitting the controller into two loops is to first control the attitude because it is independent and so that it can provide reference signal to the translational control loop that is dependent. For this purpose, the inner loop is always running at least 5 times faster than the outer loop.

Control design is simplified by considering quadcopter movement in a hovering state. In hovering state roll (ϕ) and pitch (θ) angle are very

small i.e (≈ 0). And we can use small-angle approximation as:

$$\begin{aligned} \sin(x) &\approx x \\ \cos(x) &\approx 1 \\ \tan(x) &\approx x \end{aligned} \tag{3.22}$$

Also angular velocities $(\omega_x, \omega_y, \omega_z)$ in hovering state are very small i.e (≈ 0)

3.4.1 Position Control

Position of quadcopter is constituted of altitude (height in Z direction), longitude (Position in X direction) and latitude (Position in Y direction). X and Y position control combined is called horizontal control. Position control is implemented in outer loop. Altitude control generate thrust force, whereas longitude and latitude position control generate reference roll and pitch angle (ϕ_r, θ_r) respectively, for the attitude controller. This is because the quadcopter is an under-actuated system and control of all 6-DOF of quadcopter DOF is not possible directly. As mentioned earlier this constraint is overcome by splitting control system into two loops. Indirect control of X-position and Y-position is achieved by outer loop. As its provides desired roll and pitch angles to the inner loop that will change the orientation accordingly to drive quadcopter to the desired X, Y positions.

Altitude Control

Altitude controller is responsible for maintaining a desired height. This is implemented using PID controller, error is computed by taking difference of the desired height and actual height that is being measured by a sensor this error signal is fed into PID and the controller signal is then compensated for gravitational pull so that the controller does not have to do extra work for gravity. Output of this altitude controller is the thrust force. The required thrust force generated by the PID control is calculated using $\ddot{Z}$ equation from Equation 3.7 and then applying small-angle approximation as:

$$\ddot{Z} = g - \frac{F_T^b}{m} \tag{3.23}$$

$$e_z = z_r - z \tag{3.24}$$

$$\dot{e_z} = \dot{z_r} - \dot{z} \tag{3.25}$$

$$\ddot{e_z} = \ddot{z_r} - \ddot{z} \tag{3.26}$$

$$\ddot{e_z} = \ddot{z_r} - g + \frac{F_T^b}{m} \tag{3.27}$$

$$F_T^b = T = m\left[g - \ddot{z_r} - K_{p,z}e_z(t) - K_{I,z}\int_0^t e_z(j)dj - K_{d,z}\frac{de_z(t)}{dt}\right] \quad (3.28)$$

As thrust is calculated in the body frame so it will than multiplied by rotation matrix R_b^i to get thrust force in the inertial frame as per Equation 3.5.

Longitude-X and Latitude-Y control

As discussed previously horizontal control is achieved indirectly, so from Equation 3.7 X,Y-position variables are written as

$$\ddot{X} = -\left(\cos\phi_r \sin\theta_r \cos\psi + \sin\phi_r \sin\psi\right)\frac{F_T^b}{m} \quad (3.29)$$

$$\ddot{Y} = -\left(\cos\phi_r \sin\theta_r \sin\psi - \sin\phi_r \cos\psi\right)\frac{F_T^b}{m} \quad (3.30)$$

Equation 3.29 and 3.30 can be more simplified by small-angle approximation:

$$\ddot{X} = -\left(\theta_r \cos\psi_r + \phi_r \sin\psi\right)\frac{F_T^b}{m} \quad (3.31)$$

$$\ddot{Y} = -\left(\theta_r \sin\psi - \phi_r \cos\psi\right)\frac{F_T^b}{m} \quad (3.32)$$

Equations 3.31 and 3.32 can be written in matrix form as

$$\begin{bmatrix}\ddot{X}\\ \ddot{Y}\end{bmatrix} = \frac{F_T^b}{m}\begin{bmatrix}-\sin\psi & -\cos\psi\\ \cos\psi & -\sin\psi\end{bmatrix}\begin{bmatrix}\phi_r\\ \theta_r\end{bmatrix} \quad (3.33)$$

Based on our assumption that quadcopter is in hover state so $F_T^b = mg$

$$\begin{bmatrix}\ddot{X}\\ \ddot{Y}\end{bmatrix} = g\begin{bmatrix}-\sin\psi & -\cos\psi\\ \cos\psi & -\sin\psi\end{bmatrix}\begin{bmatrix}\phi_r\\ \theta_r\end{bmatrix} \quad (3.34)$$

To make controller design simple, equation 3.34 can be rearranged and error terms can be defined in matrix form

$$\begin{aligned}\ddot{P_n} = \begin{bmatrix}\ddot{X}\\ \ddot{Y}\end{bmatrix}, R_\psi = \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix},\\ K = \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}, E_n = \begin{bmatrix}\phi_r\\ \theta_r\end{bmatrix}, h_r = \begin{bmatrix}X_r\\ Y_r\end{bmatrix}\end{aligned} \quad (3.35)$$

Now equation 3.34 can be re-written in simplified algebraic form by using variables defined in equation3.35 as

$$\ddot{P_n} = -gR_\psi K E_n \quad (3.36)$$

$$e_h = h_r - P_n \tag{3.37}$$

$$\ddot{e_h} = \ddot{h_r} - \ddot{P_n}; \tag{3.38}$$

$$\ddot{e_h} = \ddot{h_r} + gR_\psi K E_n \tag{3.39}$$

$$E_n = \frac{1}{g} K^{-1} R_\psi^T \left(-\ddot{h_r} - K_{p,h} e_h - K_{I,h} \int_0^t e_h(j)dj - K_{d,h} \dot{e_h} \right) \tag{3.40}$$

3.4.2 Attitude Control

Attitude control is responsible for stabilizing the attitude of quadcopter as per desired attitude command, and it is part of inner control loop. This is also implemented using PID control. Based on error difference between the measured/actual quadcopter orientation and desired orientation the controller will generate actuating signal(torques) for actuators to re-orient quadcopter to desired orientation. Attitude controller will generate actuating signal for all three axis $(X/\phi, Y/\theta, Z/\psi)$. To write controller actuation generated by PID, I will re-write angular velocities and Euler angle from equation 4.7 using hover assumption $(\omega_x, \omega_y, \omega_z)$ are very small i.e (≈ 0) as:

$$\begin{aligned}
\dot{\omega_x} &= \tfrac{1}{I_x}\tau_x \\
\dot{\omega_y} &= \tfrac{1}{I_y}\tau_y \\
\dot{\omega_z} &= \tfrac{1}{I_z}\tau_z \\
\dot{\phi} &= \omega_x \\
\dot{\theta} &= \omega_y \\
\dot{\psi} &= \omega_z
\end{aligned} \tag{3.41}$$

Equation 3.41 can be re-written in matrix form as

$$I^b \ddot{E} = \tau^b \tag{3.42}$$

Where I^b is inertia matrix, $\ddot{E}$ angular acceleration, and τ^b torque in body frame. Error in attitude can be defined as

$$e_E = E_r - E \tag{3.43}$$

$$\ddot{e_E} = \ddot{E_r} - \ddot{E} \tag{3.44}$$

Multiplying equation 3.44 with Inertia matrix

$$I^b \ddot{e_E} = I^b \ddot{E_r} - I^b \ddot{E} \tag{3.45}$$

$$I^b \ddot{e_E} = I^b \ddot{E_r} - \tau^b \tag{3.46}$$

$$I^b \ddot{e_E} = I^b \ddot{E_r} - \tau^b \tag{3.47}$$

$$\tau^b = I^b \ddot{E_r} + K_{p_E} e_E + K_{I_E} \int_0^t e_E(j)dj + K_{d_E} \dot{e_E} \tag{3.48}$$

Where $E_r = \begin{bmatrix} \phi_r \\ \theta_r \\ \psi_r \end{bmatrix}$, ϕ_r and θ_r are generated in the outer-loop that is position control. Equation 43.48 can be written in component form as

$$\begin{gathered} \tau_{x,\phi} = I_{xx} \ddot{\phi_r} + K_{p,\phi} e_\phi + K_{I,\phi} \textstyle\int_0^t e_\phi(j)dj + K_{d,\phi} \dot{e_\phi} \\ e_\phi(t) = \phi_r(t) - \phi(t) \end{gathered} \tag{3.49}$$

$$\begin{gathered} \tau_{y,\theta} = I_{yy} \ddot{\theta_r} + K_{p,\theta} e_\theta(t) + K_{I,\theta} \textstyle\int_0^t e_\theta(j)dj + K_{d,\theta} \dot{e_\theta} \\ e_\theta(t) = \theta_r(t) - \theta(t) \end{gathered} \tag{3.50}$$

$$\begin{gathered} \tau_{z,\psi} = I_{zz} \ddot{\psi_r} + K_{p,\psi} e_\psi(t) + K_{I,\psi} \textstyle\int_0^t e_\psi(j)dj + K_{d,\psi} \dot{e_\psi} \\ e_\psi(t) = \psi_r(t) - \psi(t) \end{gathered} \tag{3.51}$$

3.4.3 Trajectory Tracking Control

Trajectory tracking control is responsible to converge the position error to zero. This error is calculated between the quadcopter's actual position and time-varying reference position of desired trajectory. This controller only controls position on two dimensions X and Y, as it considers that the vehicle is stably moving at a constant height. Based on this consideration velocity of quadcopter can be computed geometrically from Fig. 19. From Fig. 19 equation for velocity can be written as follows [41]

$$\begin{bmatrix} \dot{x_m} \\ \dot{y_m} \\ \dot{\psi_m} \end{bmatrix} = \begin{bmatrix} \cos\psi_m & 0 \\ \sin\psi_m & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_m \\ \omega_m \end{bmatrix} \tag{3.52}$$

Where (x_m, y_m, ψ_m) are actual positions and heading(Yaw), and (v_m, ω_m) are actual velocity and heading angular rate of the quadcopter respectively. To calculate the error difference same equation as equation 3.52 need to be defined for attributes of reference trajectory. This can be written as

$$\begin{bmatrix} \dot{x_r} \\ \dot{y_r} \\ \dot{\psi_r} \end{bmatrix} = \begin{bmatrix} \cos\psi_r & 0 \\ \sin\psi_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \tag{3.53}$$

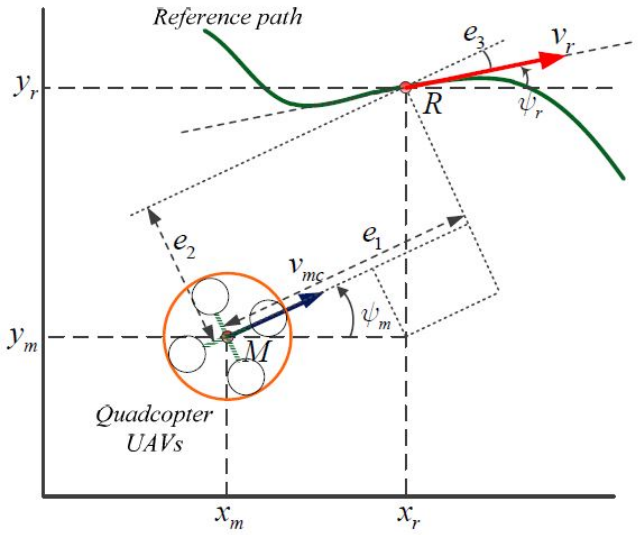

Fig. 19: Geometry of quadcopter's kinematics and reference trajectory [41]

Where (x_r, y_r, ψ_r) are reference positions and heading(Yaw), and (v_r, ω_r) are reference velocity and heading angular rate of the quadcopter respectively.

Now the position tracking errors (e_1, e_2, e_3) from Fig. 19 can be defined as

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\psi_m & \sin\psi_m & 0 \\ -\sin\psi_m & \cos\psi_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_m \\ y_r - y_m \\ \psi_r - \psi_m \end{bmatrix} \tag{3.54}$$

To converge the errors to zero, the author derived control law by taking derivative of errors $e_i (i = 1, 2, 3)$ as

$$\begin{aligned} \dot{e_1} =& (\dot{x_r} - \dot{x_m}) \cos\psi_m - \dot{\psi_m} \sin\psi_m (x_r - x_m) \\ &+ (\dot{y_r} - \dot{y_m}) \sin\psi_m + \dot{\psi_m} \cos\psi_m (y_r - y_m) \end{aligned} \tag{3.55}$$

Substituting Equations 3.52 and 3.53 into equation 3.55

$$\begin{aligned} \dot{e_1} =& v_r(\cos\psi_r \cos\psi_m + \sin\psi_r \sin\psi_m) - v_m(\cos\psi_m \cos\psi_m + \sin\psi_m \sin\psi_m) \\ &+ \dot{\psi_m} \left[-(x_r - x_m) \sin\psi_m + (y_r - y_m) \cos\psi_m\right] \end{aligned} \tag{3.56}$$

Equation 3.56 can be further simplified using trigonometric identities:

$$\dot{e_1} = v_r \cos(\psi_r - \psi_m) - v_m + \dot{\psi_m} e_2 \tag{3.57}$$

$$\dot{e_1} = v_r \cos e_3 - v_m + \omega_m e_2 \tag{3.58}$$

Now taking derivative of e_2

$$\begin{aligned}\dot{e_2} =&(\dot{x_r} - \dot{x_m})\cos\psi_m - \dot{\psi_m}\sin\psi_m(x_r - x_m) \\ &+ (\dot{y_r} - \dot{y_m})\sin\psi_m + \dot{\psi_m}\cos\psi_m(y_r - y_m)\end{aligned} \tag{3.59}$$

Substituting Equations 3.52 and 3.53 into equation 3.59

$$\begin{aligned}\dot{e_2} =&v_r(\sin\psi_r\cos\psi_m - \cos\psi_r\sin\psi_m) + v_m(\cos\psi_m\sin\psi_m - \sin\psi_m\cos\psi_m) \\ &- \dot{\psi_m}\left[(x_r - x_m)\cos\psi_m + (y_r - y_m)\sin\psi_m\right]\end{aligned} \tag{3.60}$$

Equation 3.60 can be further simplified using trigonometric identities

$$\dot{e_2} = v_r\sin(\psi_r - \psi_m) - v_m - \dot{\psi_m}e_1 \tag{3.61}$$

$$\dot{e_2} = v_r\sin e_3 - \omega_m e_1 \tag{3.62}$$

Similarly taking derivative of e_3

$$\dot{e_3} = \dot{\psi_r} - \dot{\psi_m} \tag{3.63}$$

$$\dot{e_3} = \omega_r - \omega_m \tag{3.64}$$

Now consider a Lyapunov function as:

$$V_0 = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + k_2(-\cos e_3) \tag{3.65}$$

$$\dot{V_0} = e_1\dot{e_1} + e_2\dot{e_2} + k_2 e_3\sin e_3 \tag{3.66}$$

Substituting $\dot{e_1}, \dot{e_2}, \dot{e_3}$ in equation 3.66

$$\begin{aligned}\dot{V_0} =&e_1(v_r\cos e_3 - v_m + \omega_m e_2) + e_2(v_r\sin e_3 - \omega_m e_1) \\ &+ k_2(\omega_r - \omega_m)\sin e_3\end{aligned} \tag{3.67}$$

$$\dot{V_0} = e_1(v_r\cos e_3 - v_m + \omega_m e_2) + k_2\sin e_3(\omega_r - \omega_m + \frac{v_r e_2}{k_2}) \tag{3.68}$$

To make sure that the value ofV_0 is negative, v_m, ω_m can be defined as:

$$v_m = v_r\cos e_3 + k_1 e_1 \tag{3.69}$$

$$\omega_m = \omega_{tracking} = \omega_r + \frac{v_r}{k_2}e_2 + \frac{k_3}{k_2}\sin e_3 \tag{3.70}$$

Where $k_1, k_2, k_3 > 0$.

3.4.4 Collision Avoidance Control

In any autonomous vehicle safe and stable operations is focus while executing a task. The stability of vehicle is addressed in the previous sections. This section will discuss safety of the vehicle that is to avoid

collision with any obstacle in the designated path. This will be achieved by a collision avoidance controller, main tasks of this controller are to detect the presence of an obstacle, activating collision avoidance, and maneuvering the vehicle such that collision can be avoided. The collision must be avoided not only with obstacles but also with the body of vessel. In this section control law for collision avoidance will be derived. Before starting some assumptions will be made for collision avoidance.

- The obstacle is assumed to be stationary.
- Dimensions of obstacle are already known.

Second assumption is based on the fact that obstacle detection sensor will only detect the obstacle and it will not be able to measure its dimensions. Some necessary variables will be defined based on the geometric relation from Fig. 20. These variables are defined as follows [41]

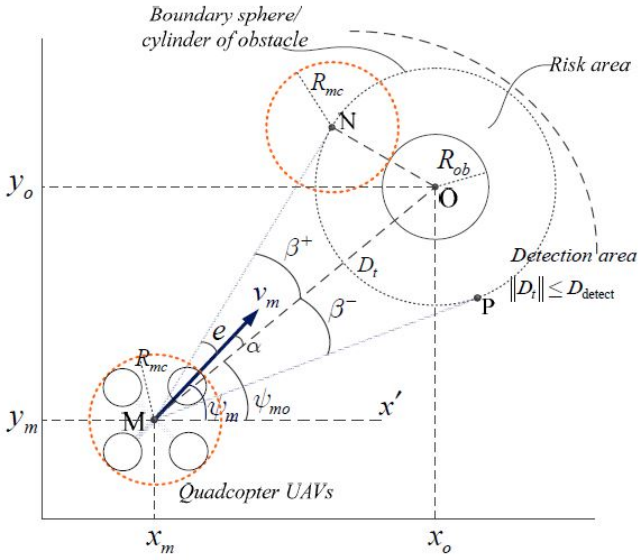

Fig. 20: Geometric relation between quadcopter and obstacle [41]

- $\boldsymbol{D_t}$: Relative distance between quadcopter and obstacle.

$$||D_t|| = \sqrt{(x_o - y_o)^2 + (x_m - y_m)^2} \tag{3.71}$$

  Where x_o, y_o are the position of obstacle.

- $\boldsymbol{R}$ is denoted as risk area, that is boundary around obstacle.

$$R = R_{safe} + R_{ob} \tag{3.72}$$

$$R_{safe} > R_{mc} \tag{3.73}$$

Where R_{safe} is safety radius, R_{mc} is radius of quadcopter including propellers, and R_{ob} is known dimension of obstacle.

- $\boldsymbol{D_{active}}$ is relative distance, at which the collision avoidance algorithm is activated, meaning it is condition to activate the collision avoidance control. Mathematically this condition can be expressed as

$$||D_t|| \leq D_{active} \tag{3.74}$$

- **Collision:** A collision occurs if relative distance between quadcopter and obstacle satisfies the following expression

$$||D_t|| < R \tag{3.75}$$

- **Collision detection:**From Fig. 21 angle between quadcopter and edges of the risk area is denoted by $\beta^{\pm}$, this angle can be expressed as

$$\beta^{\pm} = \pm \sin^{-1}\left(\frac{R}{||D_t||}\right) \tag{3.76}$$

Collision will occur if the orientation of vector v_m in inside the region made by angle β^+ and β^- that is called collision cone and shown as NMP.

- **Avoidance direction:** Avoidance direction is heading that quadcopter will make to avoid collision. This heading is the angle denoted by α between v_m and D_t, and it is computed as

$$\alpha = \psi_m - \psi_{mo} \tag{3.77}$$

$$\psi_{mo} = \tan^{-1}\left(\frac{y_o - y_m}{x_o - x_m}\right) \tag{3.78}$$

Avoidance direction is decided based on angle α. If $(0 \leq \alpha < \beta^+)$, then orientation of v_m will be inside region MNO in Fig. 20. The suitable avoidance direction is left side of obstacle. Meaning the quadcopter will turn left and α tracks β^+. Similarly if $(\beta^- < \alpha < 0)$, then orientation of v_m will be inside region PMO in Fig. 20. The suitable avoidance direction is right side of obstacle. Meaning the quadcopter will turn right and α tracks β^-.

- **Completion of collision avoidance:** The collision avoidance is completed when vector v_m is perpendicular to vector D_t, and this will happen when the following expression is valid

$$\cos\alpha \leq 0 \tag{3.79}$$

Based on the definitions and conditions described earlier it is obvious that objective of collision avoidance controller is to make α track $\beta^{\pm}$, in other words converging the error $(e = \beta^{\pm} - \alpha)$ to zero. In our case one side of the trajectory is faced by body of vessel means quadcopter will never turn to the right side to avoid obstacle or mathematically the trajectory tracking controller will never make the angle α to track β^{-}, angle α to track β^{+}. To converge this error to zero author [41] derived control law that satisfies following expressions.

$$\begin{aligned} v_m &= v_r \cos e_3 + k_1 e_1 \\ \omega_m &= \omega_{avoid} = \frac{v_m \sin e + ke}{||D_t|| \cos \beta^{\pm}} \end{aligned} \tag{3.80}$$

During simulations I found problems in stability and performance of the control law expressed in equation 3.80. There could be perfect gain tuning problem or one reason that I observed is that when collision avoidance is activated the author used same trajectory controller for v_m that is designed to force the quadcopter towards the desired path but heading rate control law try to change the heading of quadcopter by changing yaw, hence both control laws struggle but causing instability and very slow performance. This causes the quadcopter to end up into the risk area causing collisions. Another reason can be our trajectory. After analyzing results closely and doing some research about this paper [41] I found another paper [43] of this author about collision avoidance but without any trajectory controller, where the control law for both v_m and ω_m is derived considering obstacle avoidance circumstances. As discussed previously the basis purpose of heading rate controller is converge angular tracking error e to 0. Derivative of e is written as

$$\dot{e} = \dot{\beta}^{\pm} - \dot{\alpha} \tag{3.81}$$

Where $\dot{\beta}^{\pm}$ and $\dot{\alpha}$ computed in [43] are written as

$$\dot{\beta}^{\pm} = \frac{\pm R v_m}{||D_t|| \sqrt{||D_t||^2 - R^2}} \cos \alpha \tag{3.82}$$

$$\dot{\alpha} = \omega_m + \frac{1}{||D_t||} v_m \sin \alpha \tag{3.83}$$

Using equations 3.82 and 3.83 $\dot{e}$ can be obtained as

$$\dot{e} = \begin{bmatrix} \frac{\sin e}{||D_t|| \cos \beta^{\pm}} & -1 \end{bmatrix} \begin{bmatrix} v_m \\ \omega_m \end{bmatrix} \tag{3.84}$$

Now considering Lyapunov function as

$$V_o = \frac{1}{2} e^2 \tag{3.85}$$

derivative of this Lyapunov function will be

$$\dot{V}_o = \dot{e}e = e\left(\frac{\sin e}{||D_t||\cos\beta^{\pm}}v_m - \omega_m\right) \tag{3.86}$$

Negative value of $\dot{V}_o$ is achieved by setting values of (v_m, ω_m) as per expression 3.87

$$\begin{aligned} v_m &= k_1||D_t||\cos\beta^{\pm} \\ \omega_m &= \omega_{avoid} = k_1\sin e + ke \end{aligned} \tag{3.87}$$

Where $k_1 > 0$ is the first gain for normal trajectory controller and $k > 0$ is gain for avoidance controller.

To confirm that $\dot{V}_o$ will have negative value, equation 3.86 is solved by using values of (v_m, ω_m) as per expression 3.87 as

$$\dot{V}_o = e\left(\frac{\sin e}{||D_t||\cos\beta^{\pm}}(k_1||D_t||\cos\beta^{\pm}) - (k_1\sin e + ke)\right) \tag{3.88}$$

$$\dot{V}_o = e\left(k_1\sin e - k_1\sin e - ke\right) \tag{3.89}$$

$$\dot{V}_o = e\left(-ke\right) \tag{3.90}$$

$$\dot{V}_o = -ke^2 \tag{3.91}$$

From equation 3.91 it can be seen that for all the values of e the value of $\dot{V}_o$ will be negative if k is chosen such that $k > 0$.

Now when there is no collision risk with obstacle normal velocity and heading rate controller as per equations 3.69 and 3.70 will work, and when collision avoidance is activated then velocity and heading rate will be computed as per equation 3.87.

3.5 Reference path

All reference parameter x, y position, height and heading are included in path planning. This section will cover mathematical details about reference path. Reference path is generated by using cubic splines. Mathematically, splines are defined as piece-wise polynomial functions. The piece-wise functions interpolate the given data point in such a way that the curve generated passes through all these points and best fit these points. And this is referred as spline interpolation. In case of trajectory generation, a sequence of points is given to these functions, and it generates a curve with a shape that closely related to given data points. These data points are called waypoints.

Spline interpolation is different from simple polynomial interpolation in a way that polynomial interpolation generate curve through all

waypoint at once. In contrast spline interpolations generate approximated curve between approximate pairs of waypoint and then sum all the approximations and generated a final curve, that is why spline are known as piece-wise polynomial functions. Some know types of spline interpolation are linear, quadratic, and cubic splines. Type of spline is defined based on the degree of polynomial, first degree polynomials make linear splines, second degree will make quadratic splines and third degree will make cubic splines interpolation with different type of splines is shown in Fig. 21.

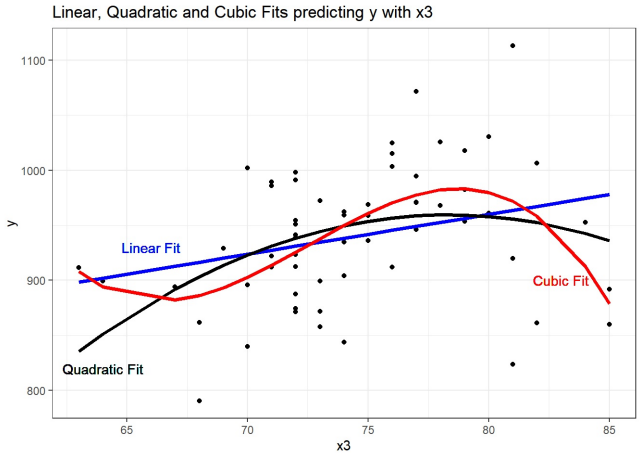

Fig. 21: Different types of interpolation on data points [44]

In this study cubic splines are considered because of the property that both first order and second order derivative of the polynomial at any waypoint are continuous hence making a very smooth trajectory [45]. General idea to generate a trajectory using cubic spline presented in [46]is to have a piece-wise function through all waypoint of the form

$$S(x) = \begin{cases} s_1(x) & if \quad x_1 \leq x < x_2 \\ s_2(x) & if \quad x_2 \leq x < x_3 \\ & \cdot \\ & \cdot \\ & \cdot \\ s_{n-1}(x) & if \quad x_{n-1} \leq x < x_n \end{cases} \tag{3.92}$$

Where $s_i(x)$ is cubic polynomial for $i = 1, 2, ..., n-1$ that is written as

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{3.93}$$

First and second order derivatives of $s_i(x)$ are written as

$$s_i^{'}(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i)^3 + c_i \tag{3.94}$$

$$s_i^{''}(x) = 6a_i(x - x_i) + 2b_i \tag{3.95}$$

As $S(x)$ will fit all the way points, so the corresponding value at given way-point can be written as

$$S(x_i) = y_i \tag{3.96}$$

$$y_i = s_i(x_i) \tag{3.97}$$

$$y_i = a_i(x_i - x_i)^3 + b_i(x_i - x_i)^2 + c_i(x_i - x_i) + d_i \tag{3.98}$$

$$y_i = d_i \tag{3.99}$$

As curve is continuous on allover the interval, it can be assumed that piece-wise function will join together at respective data points, as

$$s_i(x_i) = s_{i-1}(x_i) \tag{3.100}$$

and

$$s_i(x_i) = y_i = d_i \tag{3.101}$$

$$s_{i-1}(x_i) = a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} \tag{3.102}$$

$$d_i = a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} \tag{3.103}$$

Let $h = x_i - x_{i-1}$, then equation 3.103 will be

$$d_i = a_{i-1}h^3 + b_{i-1}h^2 + c_{i-1}h + d_{i-1} \tag{3.104}$$

Similarly due to smoothness the derivative also be equal at these points

$$s_i^{'}(x_i) = s_{i-1}^{'}(x_i) \tag{3.105}$$

using equation 3.94

$$s_i^{'}(x_i) = c_i \tag{3.106}$$

$$s_{i-1}^{'}(x_i) = 3a_{i-1}(x - x_{i-1})^2 + 2b_{i-1}(x - x_{i-1})^3 + c_{i-1} \tag{3.107}$$

$$c_i = 3a_{i-1}(x - x_{i-1})^2 + 2b_{i-1}(x - x_{i-1})^3 + c_{i-1} \tag{3.108}$$

Again substitute $h = x_i - x_{i-1}$, then equation 3.108 will be written as

$$c_i = 3a_{i-1}h^2 + 2b_{i-1}h^3 + c_{i-1} \tag{3.109}$$

From equation 3.95

$$s_i^{''}(x) = 6a_i(x - x_i) + 2b_i \tag{3.110}$$
$$s_i^{''}(x_i) = 6a_i(x_i - x_i) + 2b_i$$
$$s_i^{''}(x_i) = 2b_i \tag{3.111}$$

Assuming continuous second derivative means $s_i^{''}(x_i) = s_{i+1}^{''}(x_i)$. Now equation 3.110 can be written as

$$s_i^{''}(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \tag{3.112}$$
$$s_{i+1}^{''}(x_{i+1}) = 6a_i(x_{i+1} - x_i) + 2b_i \tag{3.113}$$
$$h = x_{i+1} - x_i$$
$$s_{i+1}^{''}(x_{i+1}) = 6a_ih + 2b_i \tag{3.114}$$
$$2b_{i+1} = 6a_ih + 2b_i \tag{3.115}$$

Coefficients of polynomial 3.93 can be determined easily by substituting $M_i = s_i^{''}(x_i)$

$$s_i^{''}(x_i) = 2b_i \tag{3.116}$$
$$M_i = 2b_i \tag{3.117}$$
$$b_i = \frac{M_i}{2} \tag{3.118}$$

From equation 3.111 and substituting value of b_i from equation 3.118 a_i can be written as

$$a_i = \frac{M_{i+1} - M_i}{6h} \tag{3.119}$$

and a_i can also be rewritten as

$$c_i = \frac{y_{i+1} - y_i}{h} - \frac{M_{i+1} + 2M_i}{6}h \tag{3.120}$$

Now all the coefficients $n - 1$ polynomial 3.93 can be determined.

In our case as trajectory will be around the vessel, so the way points are selected by approximating shape of vessel as shown in Fig. 22.

For splines the waypoints have to be in increasing order which means there should be no repeating coordinate in a waypoint, but in Fig. 22 it can be seen that vertical line is made by these way-point causing the value of x coordinates same across this vertical line. This problem can be resolved by introducing an independent monotonically increasing variable such as time that will be equal to the size of waypoints. Means the waypoints will be separated into x and y points in always

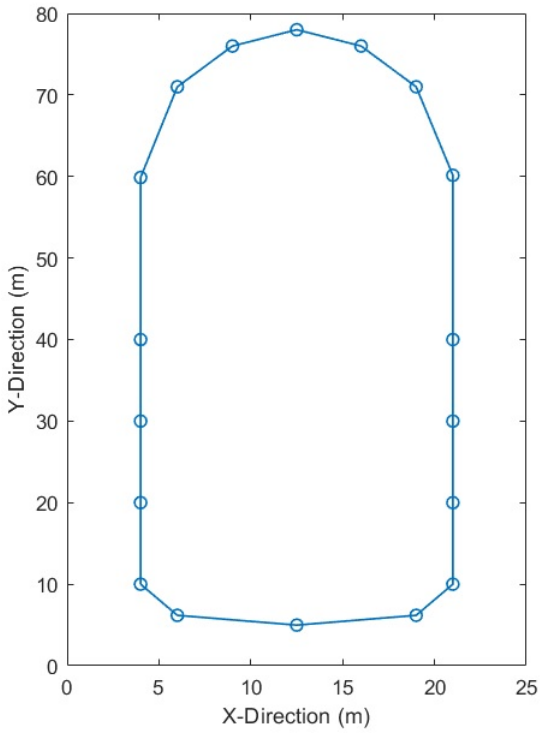

Fig. 22: Way-points for trajectory

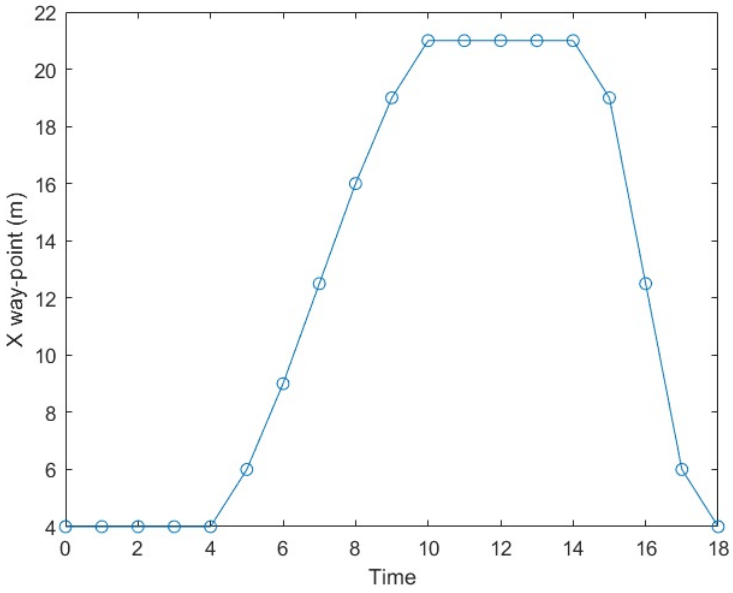

Fig. 23: X way-points w.r.t time

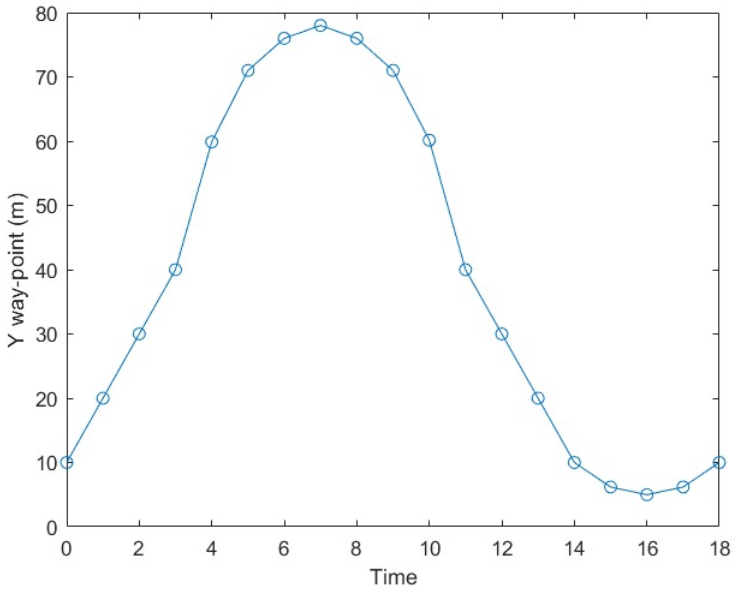

Fig. 24: Y way-points w.r.t time

increasing order with respect to a time variable. Fig. 23 shows the separated x-way points w.r.t time, and Fig. 24 shows the separated y-way points w.r.t time.

Now two separate cubic splines are generated for all waypoints, one by considering only x waypoints and second by considering only y waypoints. These spline are shown in Fig. 25 and Fig. 26.

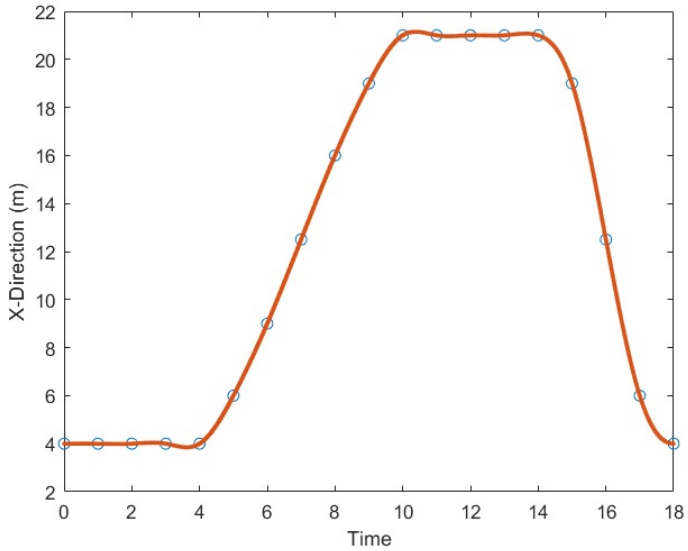

Fig. 25: Spline for x way-points

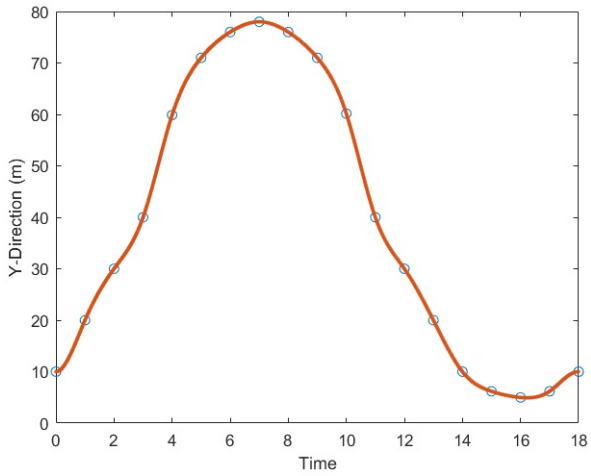

Fig. 26: Spline for y way-points

Plots shown in Fig. 25 and Fig. 26 are actually x and y coordinates of generated splines. If these coordinates are plotted together they will show the cubic spline fitting that fit and passes through all the way-points, and it is shown in Fig. 27.

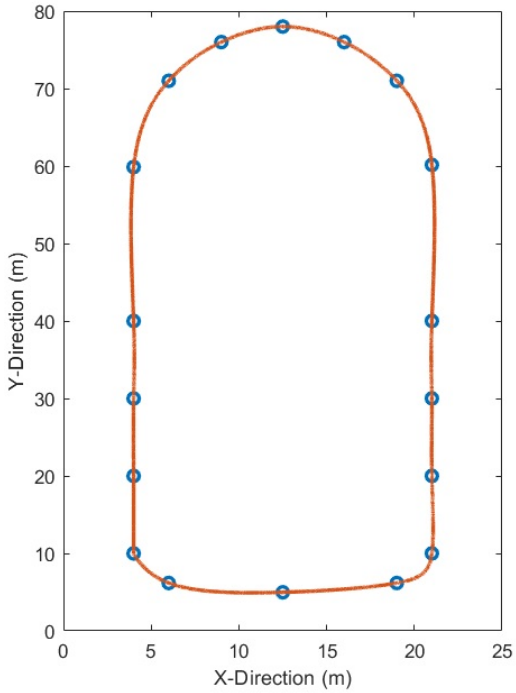

Fig. 27: Cubic spline passing through all way-points

4 Design specifications

In this section we will discuss the design requirements of the proposed multi-rotor. This will include the requirement for the development of multi-rotor as well other products required for the safe and successful operation.

4.1 Gimbal

Gimbal is a tool that allows rotation of object (that is fixed to a moving base) at least on a single axis. And these axes are roll, pitch and yaw. Gimbal act as a stabilizer for cameras as it can stabilize any motion of the camera. Gimbals have various applications ranging from videography to space navigation. But we will only focus on the video stabilizing.

A gimbal is made up of moving pivots along with a mount for the object (such as Camera). It cancels out any unwanted movement and make the object very stable. There are both mechanical (Counterweight) and electronic gimbals, but we will only consider electronics gimbals as they are light weight and can be controlled electronically. An electronic gimbal has brush-less motors along rotation axis of pivots and these motors cancel out any movement or disturbance by making slight adjustments to stabilize the mounted object. In electronic gimbals there two most known types of gimbals that are single axis, 2-axis and 3-axis gimbals. Difference is in terms of axis along which it can cancel the movements. A 2-axis gimbal can only stabilize in two axes roll and pitch axis whereas a 3-axis gimbal can stabilize in all three axes roll, pitch and yaw axis.

Selection of the type of gimbal depends on the overall design of multi-rotor and the need of stabilization. In the design of any multi-rotor weight and power are the most important considerations and the weight, power consumption and area reduce as the numbers of axis decrease in a gimbal. As the task is to capture the draft readings on the vessel by making a round trip around the vessel, and the most important area to capture the stable and clear shot is the draft marking. During the survey the camera will always need to be facing the body of the ship, and slight disturbances in the yaw angle will not make the footage blur or unreadable.

A good option for a 1-axis gimbal is quark stabilizer [47] and shown in Fig. 28. It is a 140g in weight and compatible for all type of cameras.

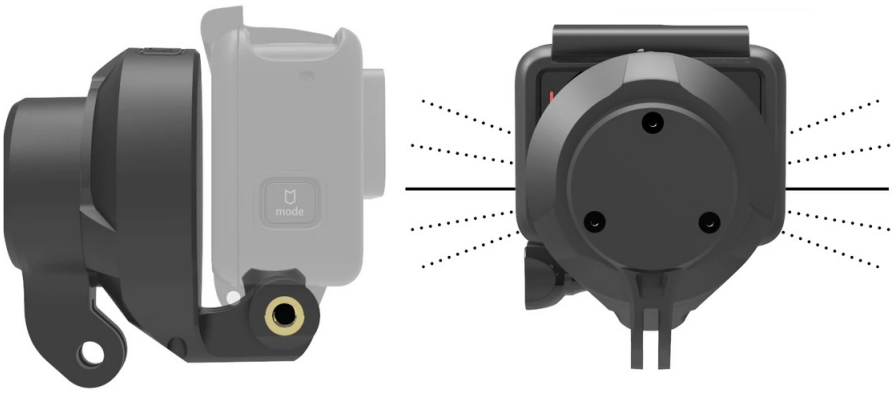

Fig. 28: 1-axis Gimbal [47]

4.2 Lights

As discussed in section 1.0.4 that reading draft in nighttime, shadows or low light will be challenging. To overcome this issue light will be added in the design specification of the multi-rotor. Selection of a specific light will be made with considering less weight of light and lumens enough to get a clear and readable view of draft markings. A lot of drone lighting options are available that include single search light, flashlight, array search light and other long range search light. For a night picture or video 200-1000 lumens of light is recommenced [48]. And for our scenario the distance between the vessel draft markings and multi-rotor will be between 0.5m to 1m.

Based on the above mentioned limitations and requirements the considered light is (ULANZI L1 PRO LED CUBE) shown in Fig. 29. This light is very small and compact in size, light weight, and provide enough illumination for night photography. Specifications of light are shown in Table 1.

This light will be mounted at the bottom side of the drone as the draft reading need to be taken close to the surface of water.

Table 1: Specifications of Light [48]

Beam Angle	60 degree
Voltage, Current	DC5V, $\geq 0.5A$
Illumination	up to 900 Lm at 0.5m, 300 Lm at 1.0m
Item Size	$3.8 \times 3.8 \times 3.6$cm
Item Weight	60g
Battery Time	80 minutes (maximum brightness)
Battery Type	3.7V, 1000mAh lithium-polymer

Fig. 29: Light for drone [48]

4.3 Collision avoidance sensors

Collision avoidance sensors are used to detect any obstacle in the path of flying or moving vehicle that can cause crash or damage upon collision with the obstacle. There are several sensors available such as ultrasonic ranging sensors, millimeter wavelength radars, laser ranging sensors or infrared ranging sensors. These sensors provide the range between the moving vehicle and the obstacle in the path to the collision avoidance system and this system will then stop or change the path of the moving vehicle. These sensors have their own advantages and drawbacks. Like ultrasonic sensors need enough amount of perpendicular surface area for reflection of the ultrasonic wave and they have low sampling rate as compared to other sensors. On the other hand, infrared and laser sensor might be unable to detect a bad reflecting color like black, but they have fast sampling rate as compared to ultrasonic, but IR sensors are vulnerable to sunlight. Similarly, millimeter wavelength radars are very sensitive and sometimes give false detection as well as they are susceptible to electrical interference.

For our application sensor should be mounted on the sides or on the top of multi-rotor to cover the surrounding area. To achieve complete safety 360-degree coverage is required. Based on the above-mentioned reasons the options found is TeraRanger Evo 3m Time of flight LIDAR by TERABEE [49]. This sensor is compatible with controllers like Arduino, Raspberry Pi, Pixhawk and compatible with ROS. Terabee provide several options such as to use single sensors separately or to create an array of 4 or 8 sensors using a TeraRanger Hub Evo (with UART board) [50] that provides USB/UART communication. Both sensor and

hub are shown in Fig. 30. Technical specifications of both sensor and hub are listed in Table 2 and Table 3 respectively.

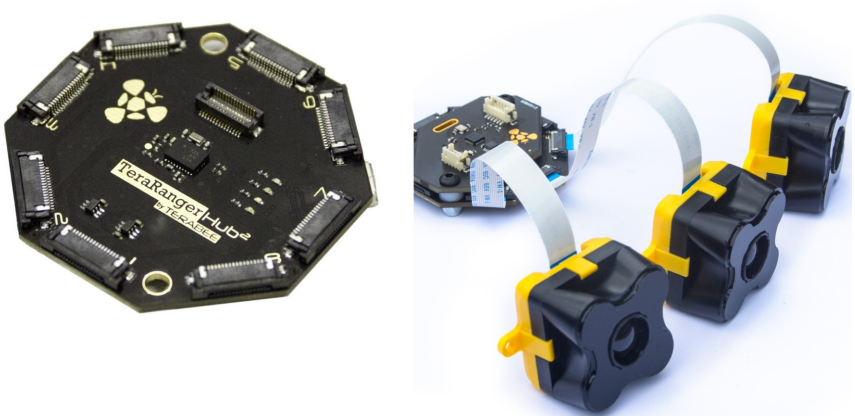

Fig. 30: TeraRanger Hub Evo and array of TeraRanger Evo 3m sensors [49]

Table 2: Specifications of TeraRanger Evo 3m [49]

Range	0.1m up to 3m
Accuracy	±2cm
Supply voltage	5V DC
Supply current	70mA - 250mA
Field of View	Approx. 2°
Weight	12 g

Table 3: Specifications of TeraRanger Hub EVO with UART board [50]

Supply voltage	12V DC
Supply current	25mA
Interfaces	USB 2.0, UART
Supply current	70mA - 250mA
Connectors	Micro USB, 4pin/6pin Hirose DF13
Weight	20 g

4.4 Height sensor

We need one sensor to maintain certain height while flying, that should be mounted at the bottom of multi-rotor. For this purpose, ultrasonic sensor is a good option because the operation of multi-rotor is mostly

over the sea water and IR and laser light can get absorbed in the sea water resulting bad reflection. Although a lot of ultrasonic range sensors are available but for our application we have selected EZ4 sensor by Maxbotix [51] as shown in Fig. 31, that is small and have very narrow beam width. Specifications of this sensor are mentioned in Table 4.

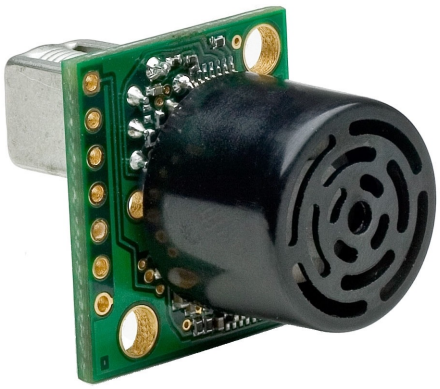

Fig. 31: Ultrasonic Rangefinder[51]

Table 4: Specifications of MB1040 LV-MaxSonar-EZ4 [51]

Resolution	1 inch or 0.0254m
Voltage, Current	DC 2.5-5.5V, 2.0mA
Maximum Range	6.45m
Sampling rate	20Hz
Outputs	Analog Voltage, RS232 Serial, Pulse Width
Weight	4.3g

4.5 Camera

Purpose of camera is to record a clear and readable video of complete survey so that the draft marking is than clearly be noted from the recorded video. Although a verity of high-quality camera are available in the market with advanced functionality like slow motion recording, high resolution, high frame rate, video stabilization, motion detection and color enhancing. For our application the main constraint is weight and size of the camera, but at the same time a good resolution and frame rate is considered to capture a good quality video as our goal is to read

the draft markings, so, considering a very high-quality overpriced camera would be waste of money. Based on these points we have opted for light weight camera by Hawkeye that have several camera products specially for FPV drones. From list of their products our requirements can be full-filled by **Hawkeye Firefly Micro 2 camera** [52] as shown in Fig. 32. It comes with a pre installed battery in it and this will remove the need of separate power for camera, and a memory card slot that will save the video. Specifications of this camera are listed in Table 5.

Fig. 32: Hawkeye Firefly Micro 2 camera[52]

Table 5: Specifications of Hawkeye Firefly Micro 2 camera [52]

Resolutions	2.5K 30FPS/1080P 60/30FPS/720P 60FPS
Voltage, Current	DC5V, 400mA
SD card supported	8-64Gb
Battery	600mAH
Weight	31 gram
Working time	About 60 minutes
Angle	Diagonal 160 degrees

4.6 Multi-rotor frame

Apart from the specifications mentioned above a multi-rotor itself is composed of different fundamental components that are required in manufacturing of any multi-rotor such as propellers, motors, frame, flight controller, power managements boards, communications, and batteries. In this section we will discuss all these components. With the advancements in drone technology and their use in either performing dedicated tasks for any organization or for personal use, many vendors now a days are manufacturing ready to fly kits that come with

all the necessary components and can be assembled easily. The size of the frame and number of rotors depend on the weightlifting requirement. A quadcopter consist of four rotor is enough in our case as we have limited the weight of the components that will be attached to the frame, by selecting light weigh components. Although a hexacopter can be selected that will have more lifting capacity but at the same time the battery requirement to keep it flying and the size of heaxacopter will put challenges like operating for a long period of time or flying in an area with compact space.

Holybro is one of these vendors that are manufacturing these drone kits. There new release **X500 V2 Kit** [53] that comes with some pre-installed components. All components of this kits are explained as follows:

4.6.1 Frame

The X500 V2 Frame is light weight as it is made with carbon fiber making it robust. The carbon fiber arms have fiber reinforced nylon connectors that make it very easy to connect motors. It has two platform board to attach flight controller and other peripherals and the spacing between these two platform boards is enough to fit devices. Also, the frame comes with 2 battery mounting straps. This frame has some pre-installed components that makes it very convenient in assembling and a ready to fly drone. Along with these pre-installed components this frame weighs 610 grams. The frame is shown in Fig. 33. All pre-installed components are explained below:

Fig. 33: Holybro X500 V2 Frame [53]

4.6.1.1 Motors

There are 4 **Holybro 2216 KV920** Motors included in this kit and this motor is shown in Fig. 34. It comes with pre-soldered ready to plug connectors. This is motor has high thrust to weight ratio. Specification of this motor for propellers T1045 at 30%, 50% and 100% throttle are listed in Table 6.

Fig. 34: Holybro 2216 KV920 motor [53]

Table 6: Specification of Holybro 2216 KV920 motor [54]

Throttle *%*	**Voltage** *V*	**Current** *A*	**Thrust** *g*	**Torque** *N.m*	**Motor speed** *RPM*	**Power** *W*	**Motor efficiency** *%*
30	16.1	1.46	214.74	0.03	4269	23.47	0.63
50	16.08	3.82	470.2	0.07	6252	61.51	0.71
100	15.92	17.42	1367.04	0.19	10703	277.32	0.75

4.6.1.2 ESC-Electronic speed controller

This kit comes with four BLHeli S ESC 20A [55] ESC's that are installed in the carbon fiber arms of the drone frame. ESC is shown in Fig. 35. It has long wires to connect to motor and control boards, and pre-soldered connectors to make a connection with the power from battery. Specifications of this ESC are mention in Table 7.

4.6.1.3 Propellers

Six pieces of **1045 Propellers** comes with this kit. They have wide and thin blades that makes them flexible, and they will not break easily in crashes. These have greater aerodynamic efficiency and good lifting

Table 7: Specifications of BLHeli S ESC 20A [55]

Constant current	20A
Burst Current	30A for 10 Sec
Battery	2-4S Lipo
Weight	6 gram
BEC(Batterly eliminator circuit)	No

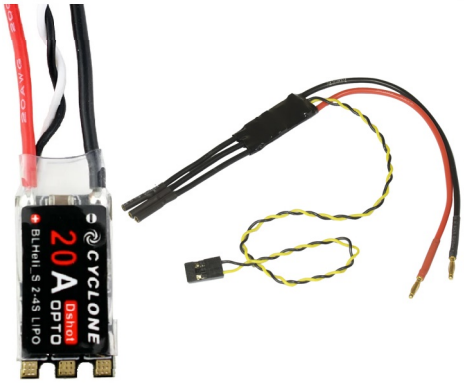

Fig. 35: BLHeli S ESC 20A [55]

capacity. These propellers are shown in Fig. 36 and their specification are mentioned in Table 8.

Fig. 36: 1045 Propellers[56]

4.6.1.4 Power distribution board

Power Distribution Board make it easy and convenient to supply power to all devices such as motors by just plugging in the connector. Drone kit comes with Power Distribution Board PDB01 with pre soldered connectors. This will avoid making mess in the wiring as all connections will be made with this board that can easily be removed anytime, and it helps in tracing the fault in case of failure. This board is shown in Fig.

Table 8: Specifications of 1045 Propellers [57]

Diameter	10 inches (25.4 cm)
Slope	4.5 inches (11.43 cm)
Shaft Diameter	6 mm
Weight	14 gram

37.

Fig. 37: Holybro power distribution board PDB01 [58]

4.7 Flight controller

A flight controller is combination of an electronic circuit board and a software, and it act as CPU (Central procession unit) for drone or any autonomous vehicle. Flight controller's tasks include communication, monitoring and controlling all parameters of a drone that include position, orientation, stability, safety, and power management. It performs these tasks using the data from different sensors such as height, speed, orientation and position. Most common sensors are height sensor, Inertial Measurement Unit (IMU), Global Positioning System (GPS) Sensors and distance sensor. All the data generated by these sensors is processed (noise filtration, fusion and estimation) by flight controller and then it is controlled by a control algorithm to perform a designated task.

There are many different options available for flight controllers. They range from very basic and cheap to very advanced and expensive. In this study Pixhawk 5X [59] flight controller is considered as it easily compatible with X500 V2 kit and it is an open source autopilot flight control software that can easily be calibrated and programmed with different hardware configurations. Pixhawk 5X is the latest flight controller in Pixhawk family. It comes with new updates that include redundancy,

isolated sensor, smart battery monitoring ports, independent power control, vibration isolation of IMUs, better stabilization, high speed performance and reliability. This flight controller is shown in Fig. 38. General specifications of Pixhawk 5x are mentioned in Table 9.

Fig. 38: Pixhawk 5X flight controller board [59]

4.8 Global positioning system (GPS) sensor

GPS sensor is a receiver with antenna that determine its location using navigation system that is based on satellites. In this study Holybro M8N GPS [60] sensor is considered. This sensor is coming along with flight controller Pixhawk 5x kit and it has ports for connection with the control board and 26cm cable and it weighs 32 grams. It is shown in Fig. 39.

4.9 Battery

Battery is essential component of a quadcopter as it is responsible for providing continuous power throughout the flight operation. The manufacturer of Holybro X500 V2 Kit has recommended to use a 4S 2000-5000mAh Lipo battery [53]. In this study 4s 5000mAh - 30C - Spectrum LiPo [61] battery is considered. It is shown in Fig. 40 and specifications are mentioned in Table 10.

4.10 Power Module

To assure safe and continuous operation of all the components in the drone a constant and stable power is necessary. This is achieved by

Table 9: Specifications of Pixhawk 5X flight controller [59]

FMU Processor	STM32F765 32 Bit Arm® Cortex®-M7 2MB Flash memory, 512KB RAM
IO Processor	32 Bit Arm® Cortex®-M3 24MHz, 8KB SRAM
Voltage ratings	Max input voltage: 6V USB Power Input: 4.75 5.25V Servo Rail Input: 0 36V
Sensors	3 Accel/Gyro 1 Magnetometer 2 Barometer
Weight	Flight Controller Module: 23g Standard Baseboard: 51g
Ports	4 general purpose serial ports 2 GPS ports,2 Power input ports 1 I2C port, 1 Ethernet port 1 SPI bus, 2 CAN Buses

Fig. 39: Holybro M8N GPS sensor [59]

Fig. 40: 4s 5000mAh - 30C - Spectrum LiPo battery [61]

Table 10: Specifications of LiPo battery [61]

Capacity	5000mAh
Number of cells	4
Voltage	14.8V
C-Rating	30C
Weight	540g

using a power module that act a voltage regulator as well as this allows continuous battery voltage level measurement for flight controller. PM02D power module [62] is selected option as it is compatible with Pixhawk 5x and voltage and current specifications meet the needed requirements. It also can perform without calibration with the flight controller by using default setting. Power module is shown in Fig. 41 and specifications are listed in Table 11.

Fig. 41: PM02D Power Module [62]

Table 11: Specifications of PM02D Power Module [62]

Max input voltage	36V
Rated current	60A
Max current	120A
Supported battery	Up to 6S
Weight	20g

4.11 RC radio transmitter and receiver

Remote controlled radio transmitter along with a receiver that is mounted on the quadcopter and connected to the flight controller will allow manual control. This allows the deployment of drone at any location manually also in case of any failure of any specific requirement an operator can manually control the drone. Different range of varieties are available for both transmitters and receivers, the most convenient and af-

fordable radio transmitter having 2km of range and a display option for settings is TX12 Radio Controller [63] and receiver from the same vendor with same range and compatibility with both the transmitter and Pixhawk 5x is R81 Receiver [64], both are shown in Fig. 42 and Fig. 43.

Fig. 42: TX12 Radio transmitter/controller [63]

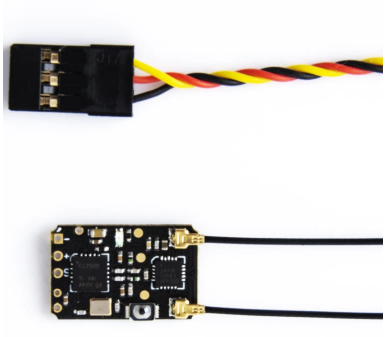

Fig. 43: R81 radio receiver [64]

4.12 Floating for quadcopter

Any aerial vehicle flying over water is always subjected to risk of sinking into the water. Specially in case of an autonomous drone the risk is more high. As in case of any system failure or power outage the drone will fell and end up sinking into the water. This will result in huge loss even if the drone is somehow recovered from the water because all the electrical and additional components attached to the drone will be damaged. Now a days different vendors are equipping their drones with a

floating device [65] that can detect a failure and float immediately to provide bouncy, and some vendors are attaching pool noodle or styrofoam balls to the leg of drones to make it float on water [66]. These two concepts are shown in Fig. 44 and Fig. 45.

Fig. 44: Emergency floating device [65]

Fig. 45: Floating landing gear [66]

These available floating devices are for smaller drones, but our drone is bigger than that. As the solution is simple to attach a lightweight material that will provide buoyancy to the drone, so the idea is to attach two long rectangular styrofoam on both landing gears of the air frame of quadcopter. Stryfoam is very lightweight and also it does not require any kind of electronics to detect a failure but it will work all the time. Dimensions of these rectangular styrofoams will depend on the stability of quadcopter on water surface, and it can be determined easily by testing.

5 Main result

In this section the model of proposed algorithms for control, trajectory tracking, and collision avoidance are presented. All the models are implemented on MATLAB/Simulink using equations of the proposed algorithms. General block diagrams of collision avoidance algorithm and control structure are presented in Fig. 46 and Fig. 47 respectively. In Fig. 46 parallelogram block is indicating data inputs and diamond blocks are conditions to select in between normal or collision avoidance controller, rectangular blocks are control laws for both controllers that will be selected based on conditions. If distance between quadcopter and obstacle is less than or equal to activation distance then avoidance controller will be activated and it will keep check on the condition of completion of avoidance. When the condition is satisfied that is $(\cos\alpha \leq 0)$ the avoidance controller will stop and normal trajectory tracking controller will start again.

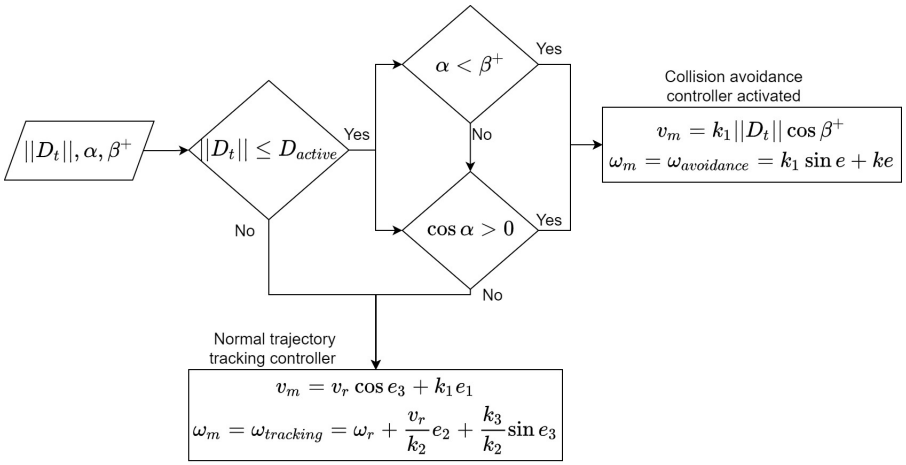

Fig. 46: General block diagram of collision avoidance algorithm

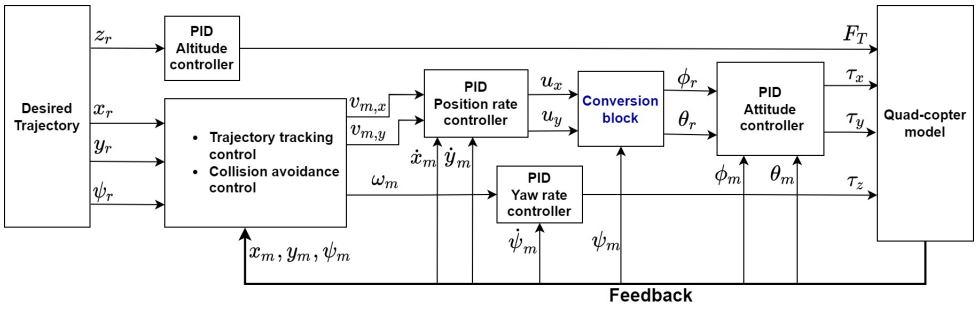

Fig. 47: General block diagram of control structure

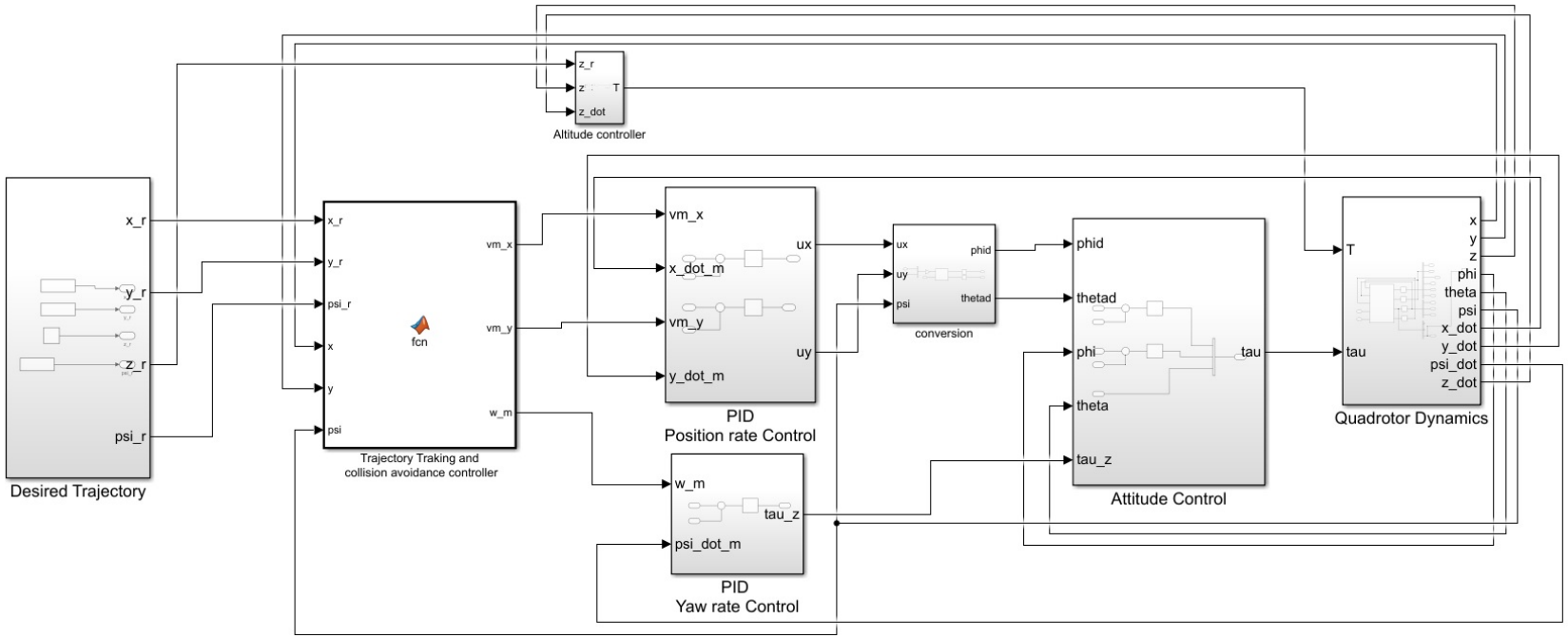

Fig. 48: Complete simulink model

5.1 Simulink Model

General block diagram represented in previous sections is implemented on Simulink as shown in Fig. 48 using equations described in section 4. All these equations are implemented using MATLAB functions, subsystems, and available blocks in the library. Overall model is divided into different sub blocks based on hierarchy, data flow and dependency of different variables. All blocks of this models are separately explained below.

5.2 Desired trajectory block

Reference position $[x_r, y_r, z_r]$ and yaw angle $\psi_r$that quadcopter needs to follow are provided by this section. Block is shown in Fig. 49. An already computed trajectory by using blocks from library called (from workspace) is loaded in it, both X and Y coordinates of computed flight path are imported in the simulation and passed as output. Height and yaw are set as a constant for complete flight path using a constant block from library. Sampling time of all sub blocks in it is set to -1 (inherited sample time), that sample time is automatically generated by the simulator by checking other blocks in the model. Reason to use this time is to make sure that next reference point of the trajectory is loaded in the simulation exactly when it is required.

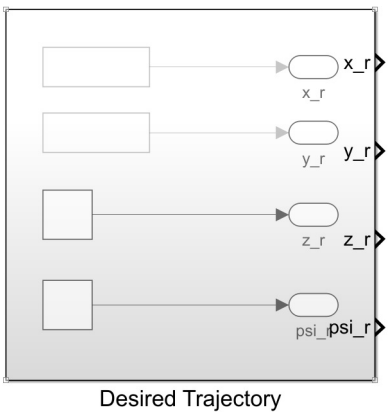

Fig. 49: Desired trajectory simulink block

5.3 Trajectory tracking and collision avoidance controller block

Reference position and yaw along with current position and yaw are input for this block. It will work in between to states, trajectory tracking

and collision avoidance. State is dependent on condition of activating the avoidance algorithm. When there is no risk of collision normal trajectory tracking controller will compute heading angular rate ω_m and velocity v_m but when obstacle is detected and there is a risk of collision then collision avoidance controller will be activated and now it computes both v_m and ω_m to maneuver quadcopter away from obstacle. Once obstacle avoidance is achieved the state of system will be change back to normal trajectory tracking controller. These switching between states is implemented inside the Matlab function using if and else conditions. Equations of trajectory tracking algorithm from section 4.4.3 and equations for collision avoidance control algorithm from section 4.4.4 are implemented in it. Using these equation it will generate velocity vector v_m that has both X and Y components $v_{mx} = v_m \cos\psi_m$ and $v_{my} = v_m \sin\psi_m)$ and heading angular rate ω_m depending on its working state. Actual position $[x, y, z]$ and yaw angle ψ and reference $[x_r, y_r, z_r]$ and yaw angle ψ_r generated by previous block are fed as input in it. This block is shown in Fig. 50.

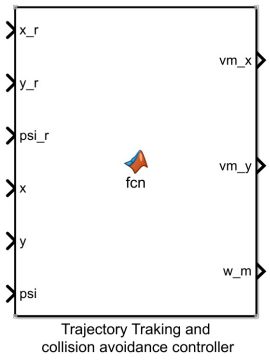

Fig. 50: Trajectory tracking and collision avoidance controller simulink block

5.4 Position rate control block

Velocity vector's x and y components and actual x and y position derivatives (velocities) of quadcopter are fed as input to this block and a Simulink built-in PID block is used to generate a controlled outputs. Position rate control block is shown in Fig. 51 and inside of this block is shown in Fig. 52.

Two separate PID blocks are used for x and y position, gains for both PID controllers are set to same values. Error difference between reference and actual velocities is calculated by summation block and then fed into the PID controller blocks. Controller will uses the proportional, integral and derivative gains to generate stable and controller output

signals (u_x, u_y). PID blocks are easy to use and just need to set P, I and D gains. PID block implements a PID controller in it and generate a gained sum of input signal, integral of the input and derivative of the input signal. Input is gained by the set proportional, integral, and derivative gains in the properties of PID block.

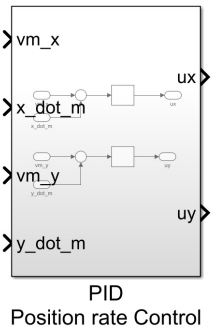

Fig. 51: Position rate controller Simulink block

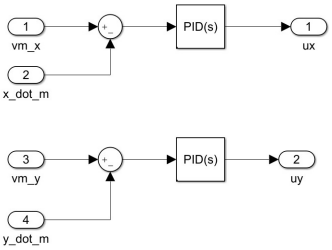

Fig. 52: Inside of position rate controller block

5.5 Conversion block

As previously described in section 4.4.1 that a desired position is indirectly controlled by providing a reference roll and pitch angles based on the output from position control block. Quadcopter will make changes in roll and pitch angles according to this conversion from position control signal (u_x, u_y) to reference angles ϕ_r, θ_r and will move towards the goal position. This conversion from PID position rate control output to angles is done in this bock that is shown in Fig. 53.

5.6 Yaw rate control block

Heading angular rate ω_m generated from trajectory tracking and collision avoidance controller block, and actual yaw angular rate $(\dot{\psi}_m)$via feedback from quadcopter model are input for this block. Error between

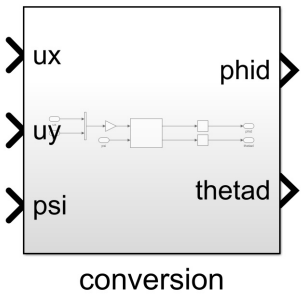

Fig. 53: Position control to desired angels converter Simulink block

these two angular rates is computed using summation block which then fed into the PID block. Using P, I and D gains it will generate a controlled and stable torque along z-axis (τ_z, τ_ψ). As all torques are responsible for controlling the attitude of quadcopters mean technically this controller can also be placed inside the attitude controller block. But here it is used separately as it generates the controlled torque along z-axis and it has dependency on trajectory tacking and avoidance control so the gains for PID inside of it are different, also it works in the same loop in which position rate control block works. Desired heading angular rate and actual heading angular rate is fed into this block that is shown in Fig. 54 and inside this block error difference between these inputs is computed and fed into the PID block that generates the controlled torque as shown in Fig. 55.

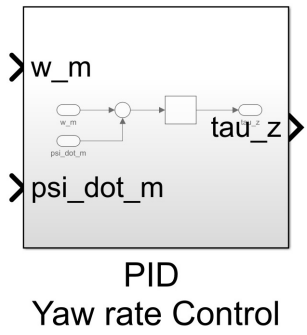

Fig. 54: Yaw rate controller Simulink block

5.7 Attitude control block

Quadcopter must follow the reference angles generated from the conversion block, and a controller is needed to converge the error between desired and actual roll and pitch angles. These Reference roll angles from the conversion block are fed as input to the attitude control block

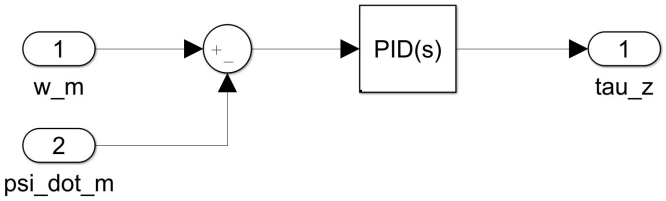

Fig. 55: Inside of yaw rate controller block

and this block generates controlled torques along x and y axis as shown in Fig. 56. It computes error between reference and actual inputs and then fed these errors to PID blocks that generate the output. It is the inner most control loop and time constants are set by keeping in mind that it should be at least five time faster than the position control blocks. As seen in Fig. 57 the torque for yaw generated from yaw rate control block is mixed with the outputs of attitude control block using a multiplexer and vector of torques named $\tau = [\tau_x, \tau_y, \tau_z]$ is sent to output port.

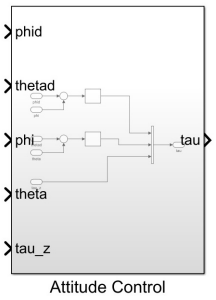

Fig. 56: Attitude controller Simulink block

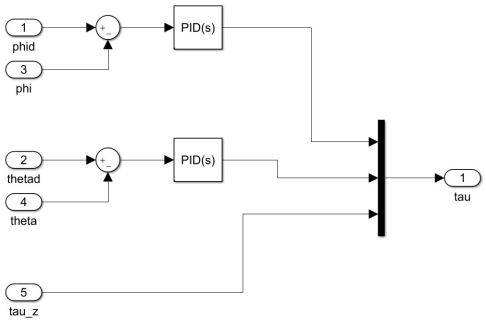

Fig. 57: Inside of attitude controller block

5.8 Altitude control block

Although it has PID controller in its other block, but it is implemented differently by cascading two PID blocks. Reason for doing this is that altitude controller is a first order system and adding differential gain for damping purpose will not have enough effect on the output, instead using two cascaded PID a stable and more robust output can be achieved. Altitude control block and the inside implementation of two cascaded PID blocks are shown in Fig. 58 and Fig. 59 respectively. Error between required z_r and actual height z is fed into first PID which will generate controlled acceleration in height $\dot{z}$. Generated acceleration is reference as input to the second PID, error between reference and actual acceleration is input and it will generate a stable and smooth output. Later this signal is used to calculate thrust by simple mathematical operation such as multiplication with mass of quadcopter and compensating the gravitational force by adding value of gravity $g = 9.81m/s^2$ to the output from second PID as per equation 3.28. Only proportional gain is used and gain of secondary PID is set five times as of primary controller. This will remove overshoot in the thrust when quadcopter will tilt to move in x or y direction.

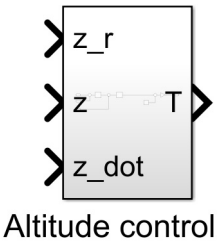

Fig. 58: Altitude controller Simulink block

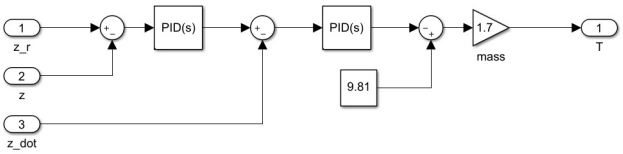

Fig. 59: Inside of altitude controller block

5.9 Quadcopter plant model block

Equations derived in section 4.1 and 4.2 are implemented in this block using MATLAB function block. Inputs are thrust force and torques generated from controllers and it outputs the positions, velocities, Euler

angles and angular rates that are used as feedback in rest of the simulation model. All system model parameters such as mass, inertia matrix, rotation matrix is also defined in this function block. Block is shown in Fig. 60. Inside this block integrator blocks are used that converts the derivative variable such as $\dot{x}$ to x. It is done according to the need of any state in the simulation.

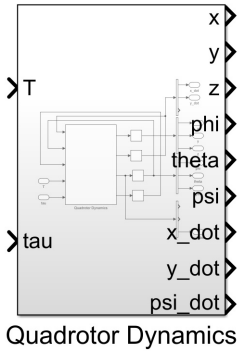

Fig. 60: Quadcopter dynamics Simulink block

It is required to set initial state of integrator block as a starting point for numerical solver otherwise it will give error in the start of simulation. Inside of this block is shown in Fig. 61

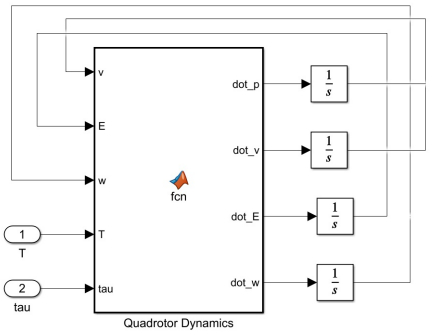

Fig. 61: Inside of Quadcopter dynamics block

6 Simulation/Experimental results

In this section the results of simulation done by using the Simulink model explained in previous section are presented. Results are categorized into different subsections based on presence of obstacle in the path, location of obstacle and single or multiple obstacles. Before presenting the results simulation parameters used by simulator needed to be discussed. Simulink solver is set to fixed-point solver because it provides a constant sample time that will fetch coordinates of desired trajectory consistently on a constant interval time steps without skipping any of the data. Shape and size of the ship are not exactly as per dimensions of a large vessel for bulk cargo instead they are scaled down to make simulation fast and to analyze graphical results of the system. General quadcopter system parameter for simulation is listed in Table 12.

Table 12: Quadcopter general system parameter for all simulations

Symbol	**Description**	**Value and Unit**
m	Total mass of quadcopter	$1.7kg$
g	Gravitational acceleration	$9.81m/s^2$
I_{xx}	Moment of inertia along x-axis	$0.1kg.m^2$
I_{yy}	Moment of inertia along y-axis	$0.1kg.m^2$
I_{zz}	Moment of inertia along z-axis	$0.2kg.m^2$
T	Simulation time	$200s$
T_{step}	Step size	$0.01s$
S	Start position	$(4, 0, 0)$

Simulation 1: Trajectory tracking only

This simulation is performed without considering any obstacle in the flight path, to check the working of trajectory tracking controller alone. Parameters for this simulation are listed in Table 13. Performance and results are presented below.

Overall trajectory tracking performance is illustrated in Fig. 62. It can be seen that quadcopter is able to follow the desired trajectory that is indicated by green line and the actual path adapted by quadcopter is indicated by doted blue line. There are slight deviation from desired trajectory at the curved paths like in Fig. 62 at the top of vessel there is some deviation and also some deviation at the curvatures in the sides of bottom of vessels.

Fig. 63 presents position errors in all x, y and z directions. It can be clearly seen that in the start when quadcopter moved from its stating

Table 13: Parameters for simulation 1

Symbol	Description	Value and Unit
v_r	Reference velocity	$1m/s$
ω_r	Reference angular rate	$0rad/s$
k_1, k_2, k_3	Trajectory tracking controller gains	$(5, 1.8, 2)$
$(P, I, D)_{Position}$	Position rate controller gains	$(2, 0, 0.8)$
$(P, I, D)_{Yaw}$	Yaw rate controller gains	$(20, 0, 0)$
$(P, I, D)_{Attitude}$	Attitude controller gains	$(10, 0, 5)$
$(P, I, D)_{Attitude}$	Altitude cascaded controller gains	$(1, 0, 0), (5, 0, 0)$

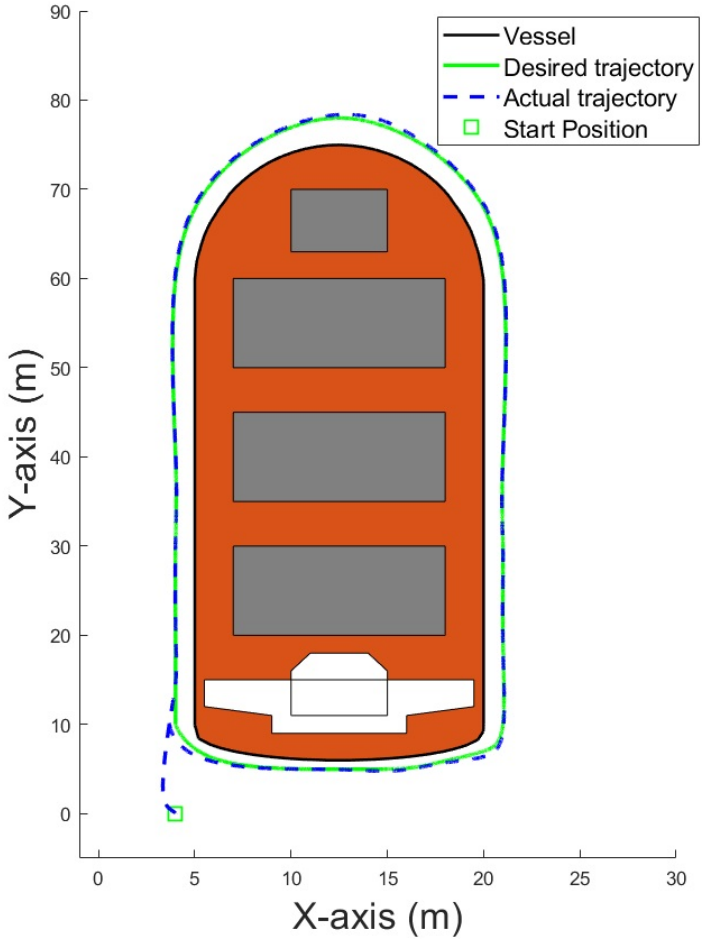

Fig. 62: Performance of working path across vessel

point the error is maximum in x, y and z axis, but it approached to 0 in a very short time and later on some fluctuations in the error that are

approximately 0.5m. These variations can be removed by fine tuning and increasing gains, but it induces some oscillations at curvatures or starting point and while testing this simulation it has been observed that high gain oscillation was causing problems for collision avoidance, that is why these gains were set.

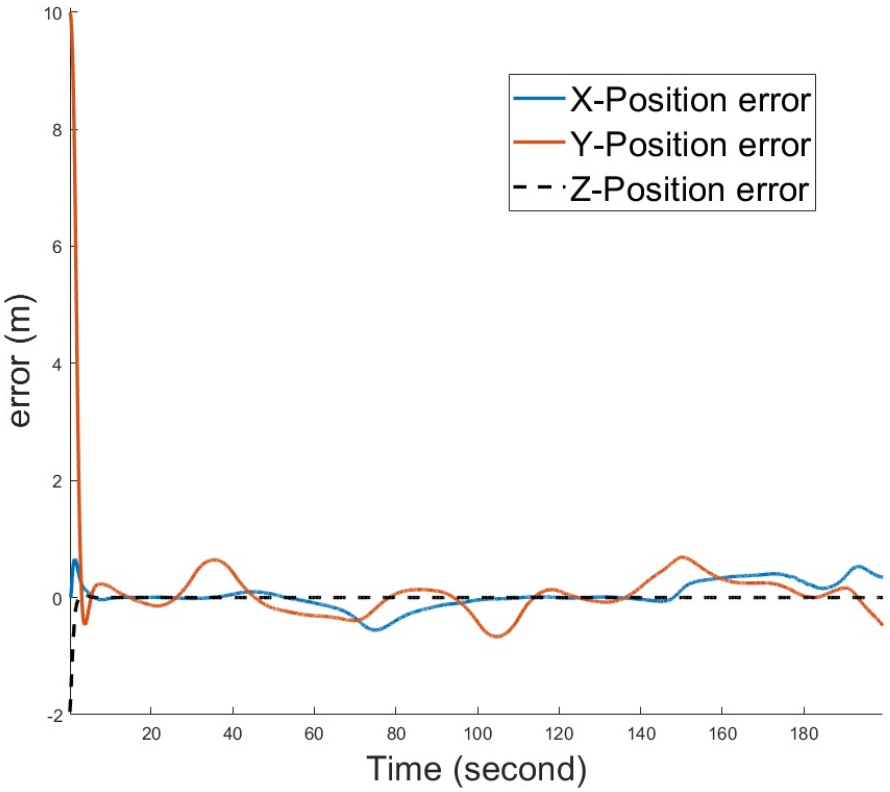

Fig. 63: Position errors

Heading angular rates from both the trajectory controller and quadcopter model are shown in Fig. 64, and it is visible that desired angular rate is achieved that was set to 0. Zoomed view of heading angular rate in Fig. 65 that shows convergence of from a large value to desired value in a very short amount of time at the beginning of simulation.

Simulation 2: Collision avoidance with 1 obstacle

This simulation is performed with considering one obstacle in the flight path, to check the working of both trajectory tracking controller and collision avoidance controller at the same time. Parameters for this simulation are listed in Table 14. Performance and results are presented below.

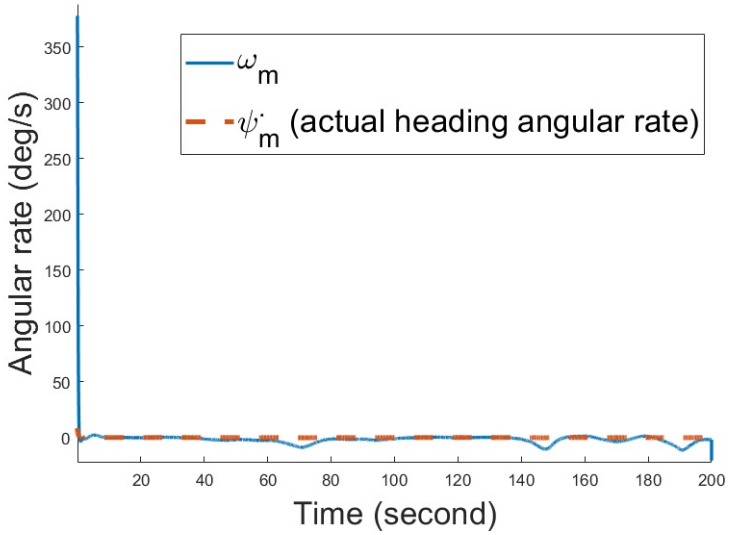

Fig. 64: Heading angular rate

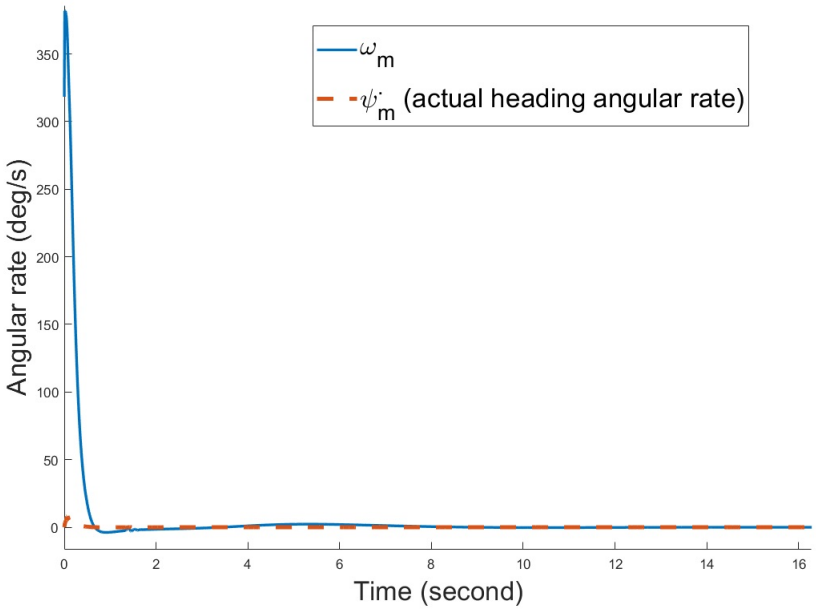

Fig. 65: Zoomed view heading angular rate

Performance of working path in the presence of obstacle is shown in Fig. 66. It is visible in this result that quadcopter safely and successfully avoided the obstacle.

Position errors for this simulation are shown in Fig. 67. Maximum error occurs when the collision avoidance is activated as it can be seen in Fig. 67 that two vertical line shows the time span in which collision

Table 14: Parameters for simulation 2

Symbol	**Description**	**Value and Unit**
v_r	Reference velocity	$1m/s$
ω_r	Reference angular rate	$0rad/s$
k_1, k_2, k_3	Trajectory tracking controller gains	$(1.8, 1.8, 2)$
k	Collision avoidance controller gain	10
$(P, I, D)_{Position}$	Position rate controller gains	$(2, 0, 0.8)$
$(P, I, D)_{Yaw}$	Yaw rate controller gains	$(20, 0, 0)$
$(P, I, D)_{Attitude}$	Attitude controller gains	$(10, 0, 5)$
$(P, I, D)_{Attitude}$	Altitude cascaded controller gains	$(1, 0, 0), (5, 0, 0)$
O	Obstacle positions	$(3.98, 59.48)$
D_{active}	Distance of activating the collision avoidance algorithm	$4.5m$
R	Safety radius	$1.43m$

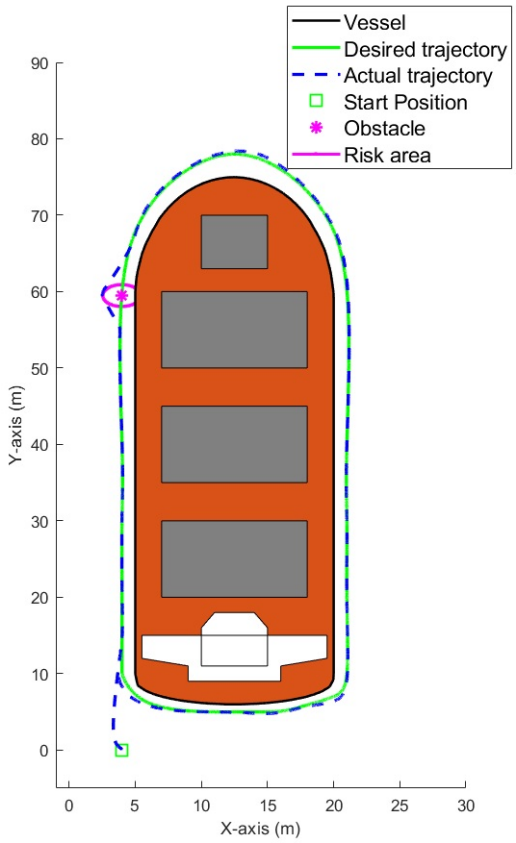

Fig. 66: Performance of working path across vessel

avoidance controller worked. Error is increased due to that reason that quadcopter is maneuvered away from obstacle, hence causing deviation from desired trajectory. Zoomed and detailed view of position error during the working of avoidance controller is shown in Fig. 68.

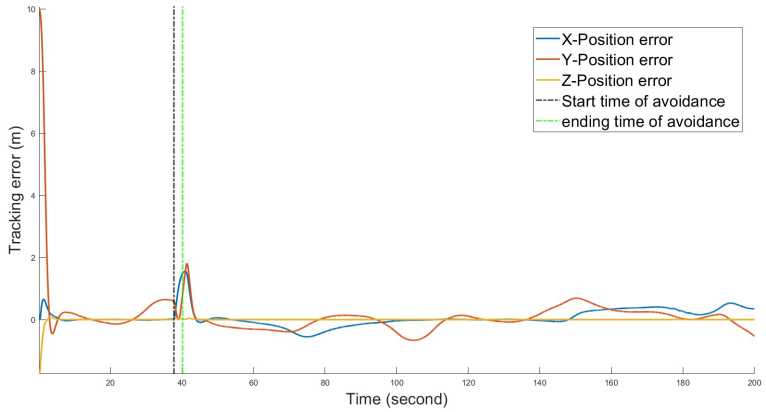

Fig. 67: Position errors

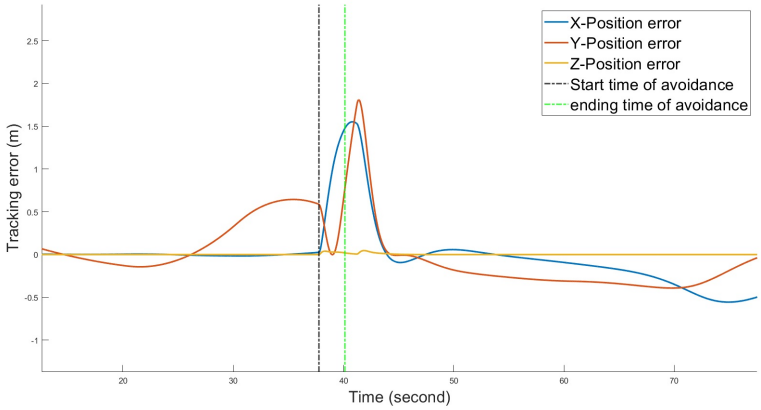

Fig. 68: Position errors zoomed view

Fig. 69 represents the two angles α and β. These angles are computed by collision avoidance controller, and they have contribution in the equations of this algorithm. In Fig. 70 that is close and zoomed view of this tracking, it can be clearly seen that when collision avoidance (this point is indicated as start time of avoidance in Fig. 70) is activated, the angle α start to track angle β using the control law. At the point when collision avoidance is achieved successfully and distance between obstacle and quadcopter is more than the limit and there is no risk of collision anymore (this point is indicated as ending time of avoidance in Fig. 70) the collision avoidance controller deactivated and

normal controller will start. From that point on-wards angle α stopped tracking angle β. The sudden shift of angle α from positive to negative value visible in Fig. 69 is because on the left side of vessel quadcopter is flying along increasing y direction and when quadcopter turned right at the top of top front of vessel it starts to move in decreasing y direction causing the heading angle in downward direction that result in negative α.

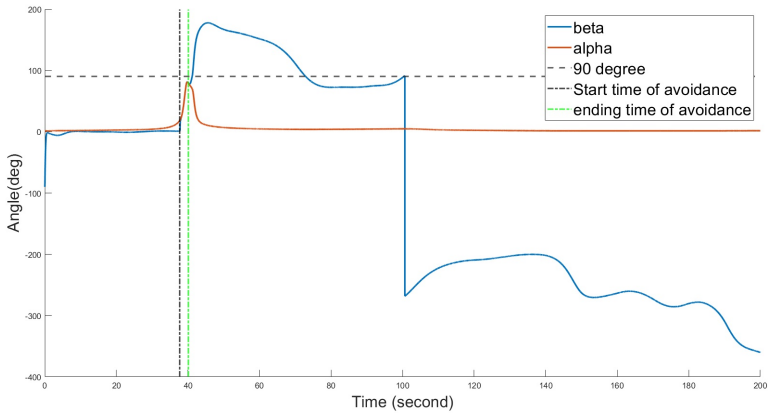

Fig. 69: Angle α tracking angle β

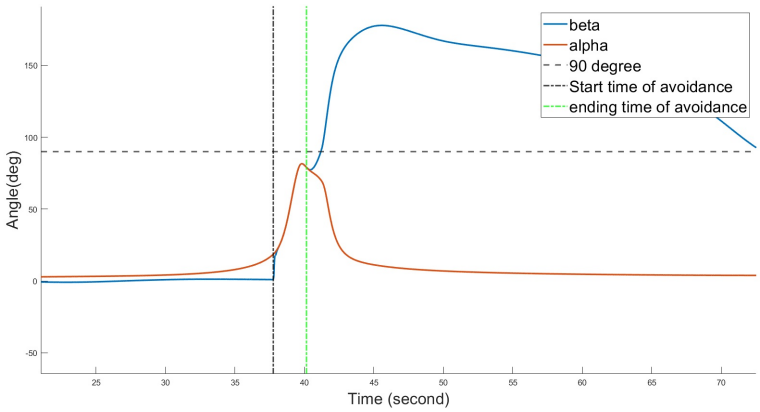

Fig. 70: Zoomed view of angle α tracking angle β

Angular tracking error for collision avoidance control that is $(e = \alpha - \beta)$ is shown in Fig. 71. It is visible in this figure that error was greater than zero prior to the start of avoidance, but when avoidance controller activated it converged that error to zero until the avoidance is completed and ended, indicating that angle α successfully tracked angle β. Convergence of error to zero is clearly shown in the zoomed view in Fig. 72. In Fig. 71 the flip in error is from negative to positive

value and reason is same as explained for angular tracking, but sign is different because the error is computed by subtracting angle α from angle β. Therefore, negative value of α subtracted from positive value of β changed the error in positive value when quadcopter moved from left to right side of vessel.

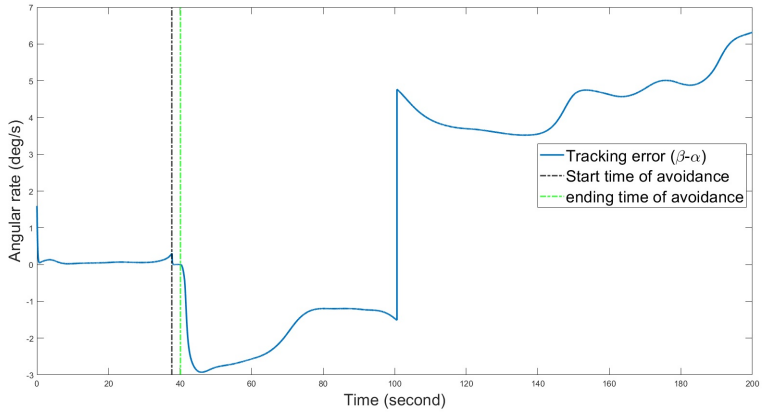

Fig. 71: Angular tracking error

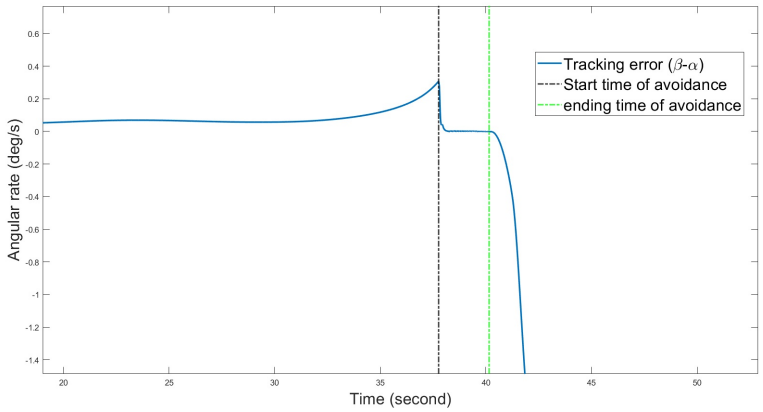

Fig. 72: Zoomed view of Angular tracking error

Relative distance between quadcopter and obstacle in comparison to safety radius and activation distance for collision avoidance for complete simulation is shown in Fig. 73, and detailed view of distance during the working of avoidance controller is shown in Fig. 74. It can be seen in Fig. 74 that in between the time span of starting and ending of avoidance the distance D_t approached close to safety radius R, but it never became less than R, indicating that controller successfully avoided obstacle.

Controller generated heading rate ω_m and actual heading rate $\dot{\psi}_m$

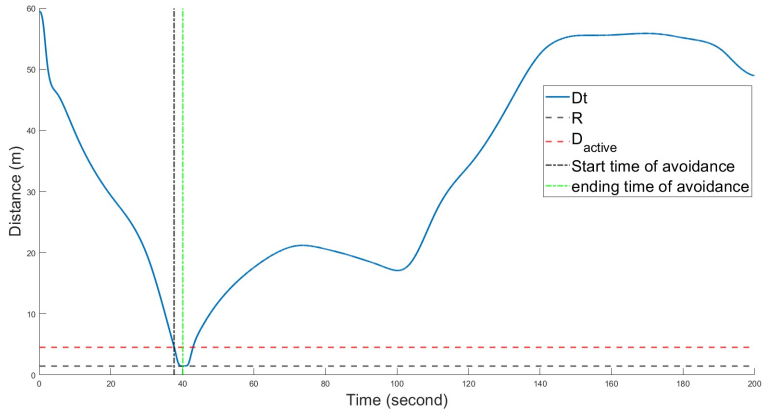

Fig. 73: Distance D_t for complete simulation

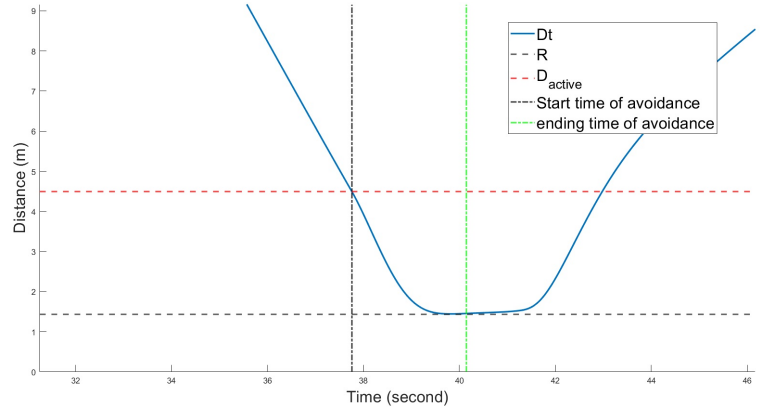

Fig. 74: Zoomed view of distance D_t

from Fig. 75 are almost equal to each other before avoidance is activated. Once avoidance is started ω_m showed a lot of variation as shown in zoomed view in Fig. 76 indicating that the avoidance controller is changing the heading of quadcopter to avoid obstacle. Even with large deviations in ω_m the actual heading angular rate of quadcopter $\dot{\psi}_m$ didn't showed much deviations from the desired value 0, because of the separate heading rate PID controller, only a small amount of change at the start of avoidance control activation indicating that quadcopter changed the heading to avoid obstacle.

Simulation 3: Collision avoidance with multiple obstacles on both sides of vessel

This simulation is performed with considering three obstacles in the flight path, one on the right side of vessel and two on the left side of ves-

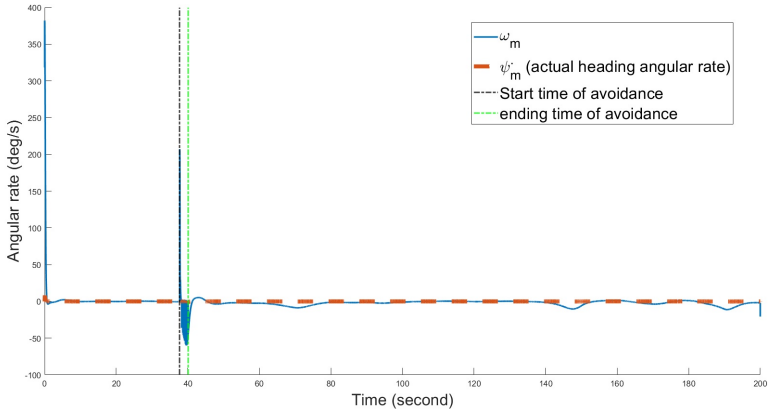

Fig. 75: Heading angular rate for complete simulation

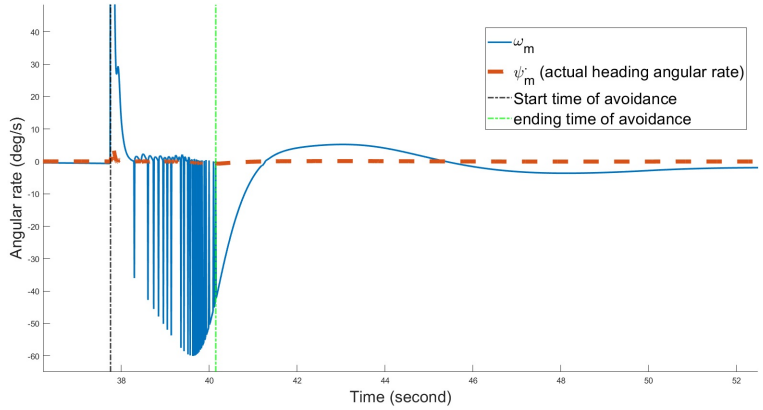

Fig. 76: Zoomed view of Heading angular rate

sel at different locations to check the working of both trajectory tracking controller and collision avoidance controller in presence of multiple obstacles. Parameters for this simulation are listed in Table 15. Performance and results are presented below.

Complete working path of quadcopter in presence of three obstacles is shown in Fig. 77. It can be seen that quadcopter has successfully achieved obstacle avoidance for all three obstacles.

In Fig. 78 position error in z, y and z directions for complete simulation are shown, it can be seen in this figure that position error in y direction was 10m at the beginning and it dropped close to 0 immediately at the start of simulation. A zoomed and detailed representation of position errors is shown in Fig. 79 that is zoomed in to skip starting spike. It can be seen in Fig. 79 that after the maximum error in position occurred during the time spans in which collision avoidance is activated, these time spans are indicated by vertical dotted line in Fig.

Table 15: Parameters for simulation 3

Symbol	Description	Value and Unit
v_r	Reference velocity	$1m/s$
ω_r	Reference angular rate	$0rad/s$
k_1, k_2, k_3	Trajectory tracking controller gains	$(1.8, 1.8, 2)$
k	Collision avoidance controller gain	10
$(P, I, D)_{Position}$	Position rate controller gains	$(2, 0, 0.8)$
$(P, I, D)_{Yaw}$	Yaw rate controller gains	$(20, 0, 0)$
$(P, I, D)_{Attitude}$	Attitude controller gains	$(10, 0, 5)$
$(P, I, D)_{Attitude}$	Altitude cascaded controller gains	$(1, 0, 0), (5, 0, 0)$
$O1$	Obstacle 1 positions	$(4, 20)$
$O2$	Obstacle 2 positions	$(3.98, 59.48)$
$O3$	Obstacle 3 positions	$(21.14, 51.01)$
D_{active}	Distance of activating the collision avoidance algorithm	$4.5m$
R	Safety radius	$1.43m$

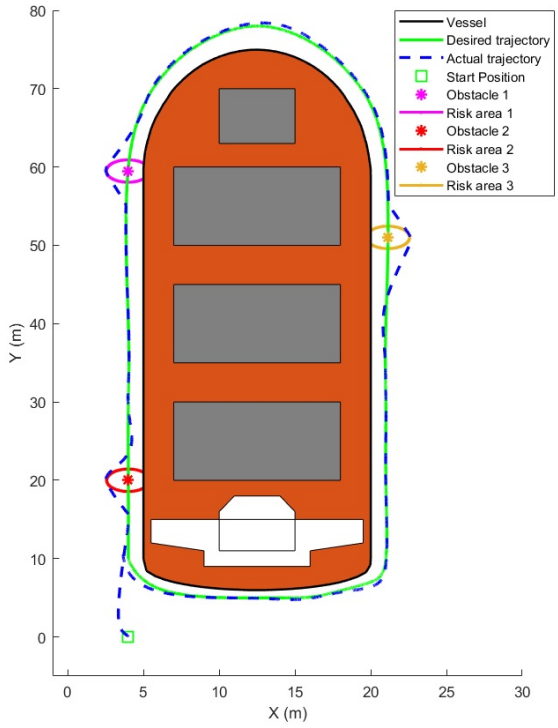

Fig. 77: Performance of working path across vessel

78, indicating that position of quadcopter is diverted away from desired trajectory and this caused error in positions. Position errors begin to rise when avoidance starts and start to approach zero when avoidance controller stops.

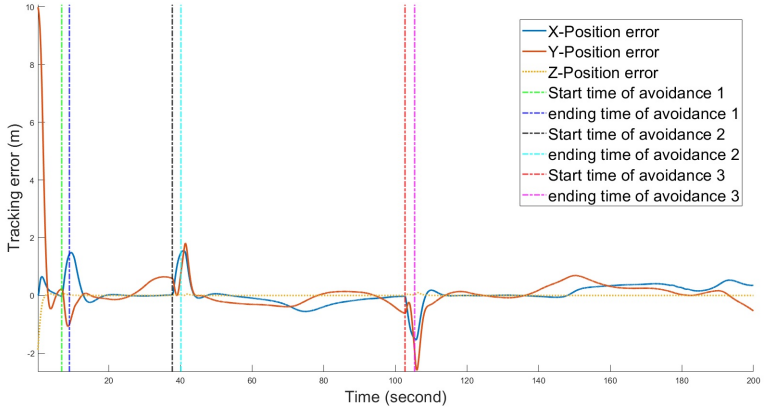

Fig. 78: Position errors

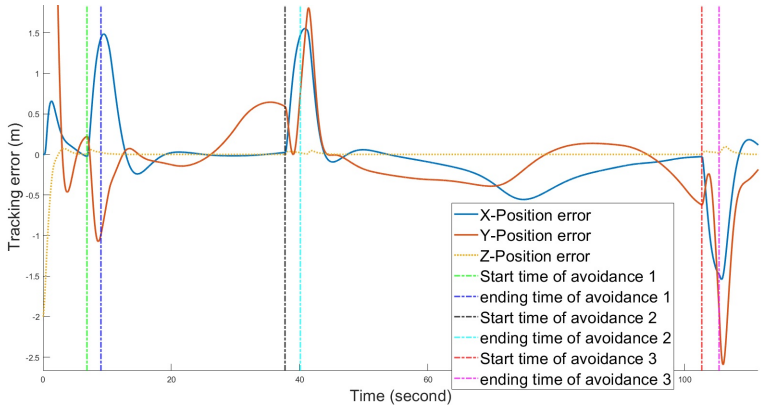

Fig. 79: Position errors zoomed view

Whenever avoidance controller is activated it makes the angle α to track angle β, this angular tracking is shown in Fig. 80. As there are three obstacles in this simulation therefore angle α tracked angle β, a zoomed view with more details is shown in Fig. 81. Large flip in the angle α from positive to negative that is visible in Fig. 80 is skipped by zooming the result as shown in Fig. 81, also large spikes in α are also cropped in Fig. 81 to have a detailed view. This flip in α is caused due to the reason that we are only tracking β^+ because of the reason that on β^- side there is body of vessel, and on the right side of vessel the quadcopter moves downward or negative y-direction resulting angle α in negative. But as said earlier that tracking will be for β^+ that is why this

sudden flip occurred at the start of avoidance for third obstacle and it again tuned to its original negative value after the ending of avoidance for third obstacle. For first two obstacle α is also tracking β.

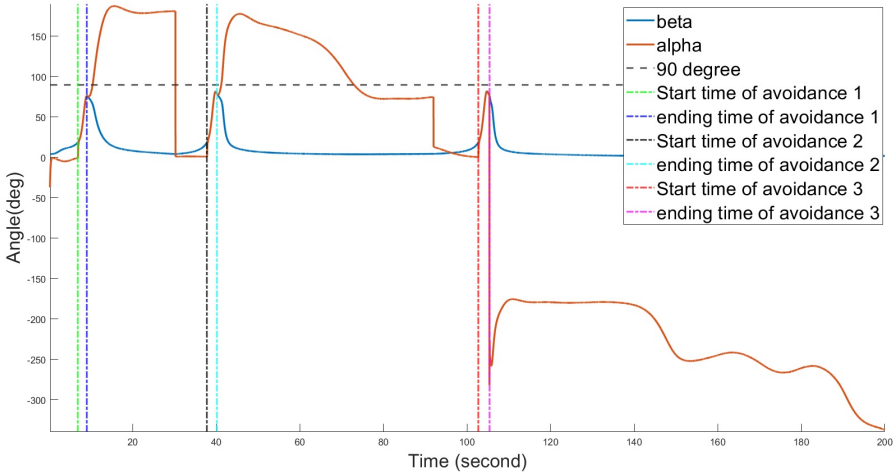

Fig. 80: Angle α tracking angle β

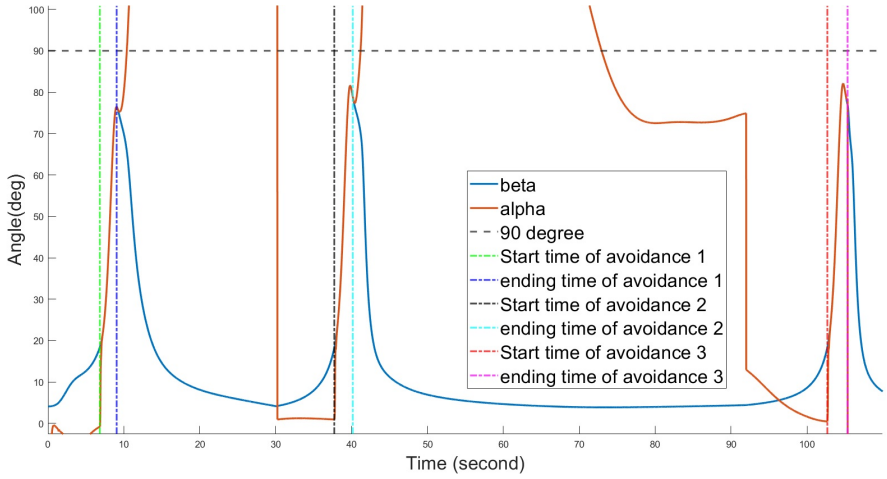

Fig. 81: Zoomed view of angle α tracking angle β

As discussed in last paragraph about angle α tracking β, this is done by avoidance controller that achieve this goal by converging the angular tracking error $(e = \beta - \alpha)$ to 0. This convergence of angular tracking error to 0 for complete simulation is shown in Fig. 82. A zoomed view ignoring large spikes due to the same reason explained in previous paragraph is shown in Fig. 82. As in Fig. 82 flip in error in from negative to positive but in angle α in Fig. 80 flip is from positive to negative, this is due to the formula for error that is $(e = \beta - \alpha)$ which make error

positive when negative α is subtracted from positive β. In Fig. 83 it can be seen that error dropped to 0 immediately when avoidance started for all three obstacles.

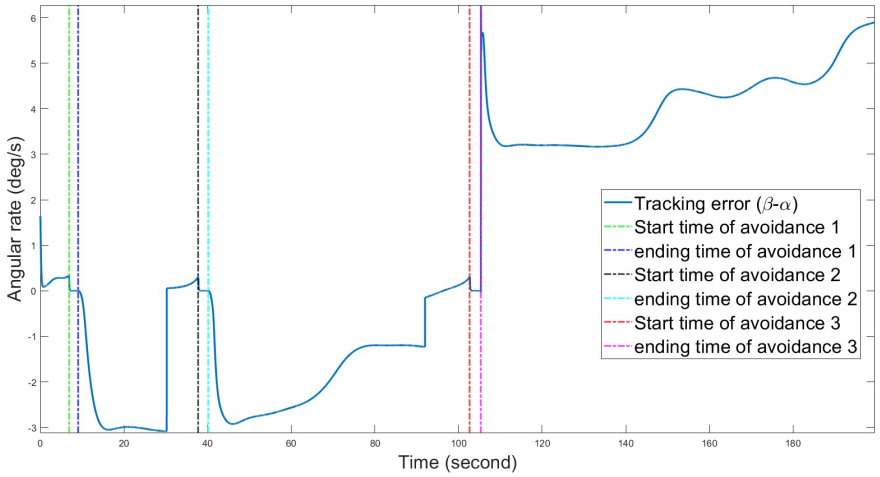

Fig. 82: Angular tracking error

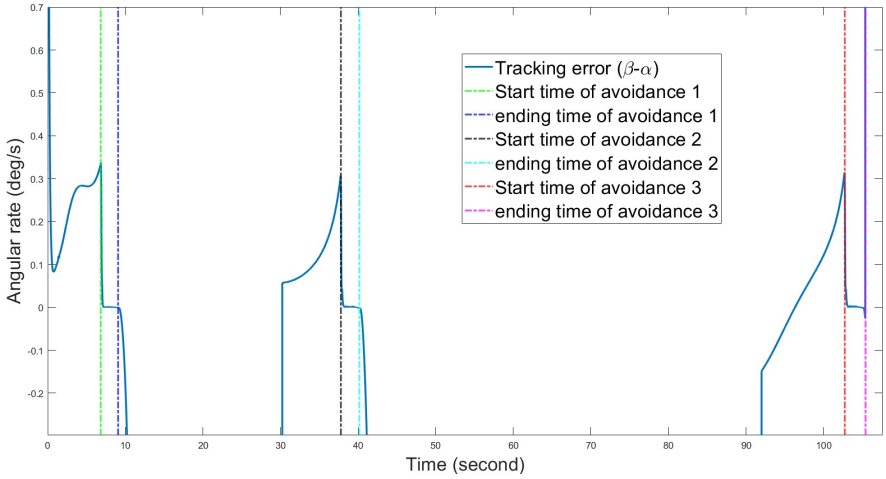

Fig. 83: Zoomed view of Angular tracking error

Main goal of avoidance controller is to make sure that relative distance D_t between quadcopter and obstacle should never became less than safety radius R, this is demonstrated in Fig. 84 for complete simulation and a zoomed view is shown in Fig. 85. It can be seen in Fig. 85 that whenever distance D_t approached activation distance D_{active} the collision avoidance started that is indicated by the vertical dotted lines as start and ending of obstacle avoidance for each obstacle. After starting of avoidance D_t approached R but never dropped below R for all

three obstacles indicating that avoidance is achieved.

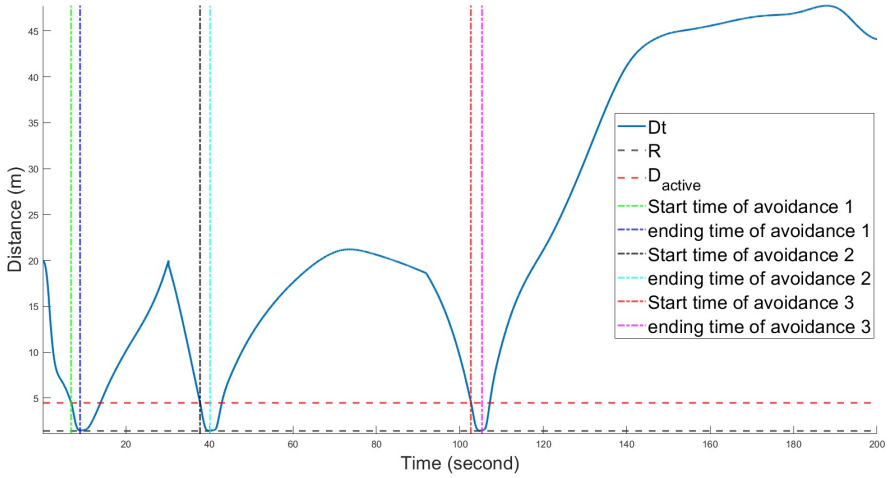

Fig. 84: Distance D_t for complete simulation

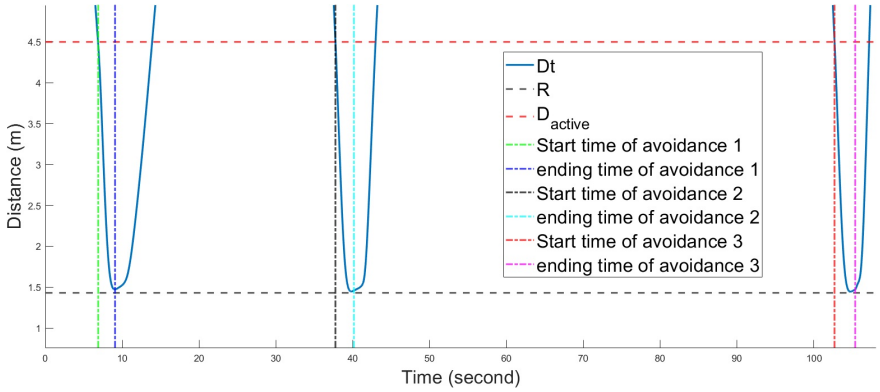

Fig. 85: Zoomed view of distance D_t

Changes in the heading angular rate that avoidance controller induced during avoidance are shown in Fig. 86. This heading angular rate is responsible for generating the torque required to change heading of quadcopter by changing its yaw angle. Zoomed view of heading angular rate is shown in Fig. 87. It shows a large spike at the start of collision avoidance for all three obstacles, this spike is because of the reason that with normal trajectory tracking controller working it is converging heading angular rate to desired value that is set to 0, but when avoidance activated the avoidance controller immediately start to change the heading rate but then PID for heading angular rate control try to settle it down to normal values. But variations in the heading angular rate can be seen in the time difference between start and end of

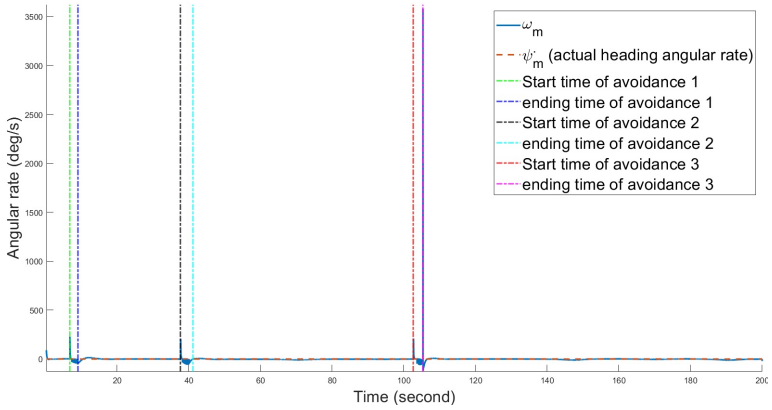

Fig. 86: Heading angular rate for complete simulation

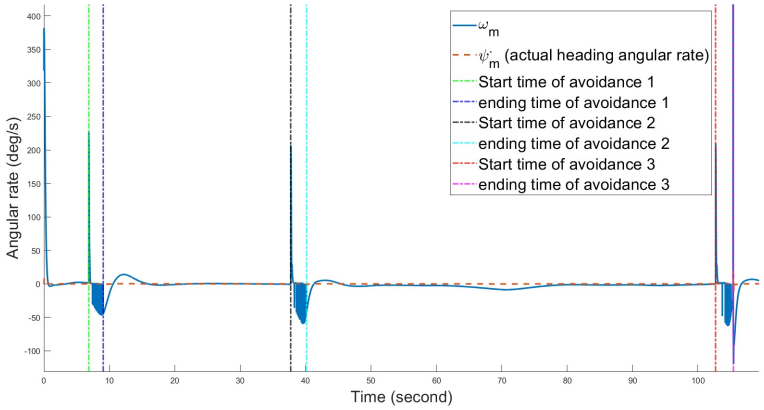

Fig. 87: Zoomed view of heading angular rate

activation, to continuously turn the quadcopter to make sure D_t never become less than safety radius R and also the quadcopter make a turn around the safety radius to approach desired trajectory when avoidance controller will stop again.

7 Discussion

Many challenges and problems were faced during the work on this thesis. At the beginning stages it was difficult to found enough literature available on the internet about draft surveys. Literature was mostly about calculation and marine industry related. Also, no previous work was found about application of drones in autonomous draft surveys. But this was covered by available literature about autonomous inspections with drones in different needs.

Moreover, full filling the design specification was also challenging because cost of the product needs to be as minimum as possible, but this should not degrade the performance and reliability to the product. Sensors for collision avoidance opted are Lidar based, limiting the collision avoidance to obstacle detection only whereas shape and size are required make collision avoidance more efficient, therefor the dimensions of obstacle are already known in this work. Dimension of vessel were not exactly same as of a true vessel but are scaled down for easiness in the implementation, hence trajectory planned is also a scaled version. One other factor was weight of the whole drone should be as low as possible so that a reasonable flight time can be achieved. And this led to looking for components such gimbal and camera with small weights. It may cause problems in the quality of video captured by camera or instability in the video as single axis light weight and low-cost gimbal is considered. Actual hardware was not tested due to short amount of time, implementing this idea on actual hardware will discover some more problems that was not covered in this study, and this can be explored by doing experimentation with hardware in a lab, as well as in actual working environment.

Modelling of this system is done by certain assumptions and ignoring details about aerodynamic effects and power modelling of the system such as battery consumption by motors and other supporting components such as gimbal, lights, and extra sensor. External environment factors such as winds, waves, and weather conditions such as rain or strong sunlight were also not considered. Approximated Weight and other dimensions of the complete drones with the help of available data on the website of drone and other components is used in simulation. Most of the literature available about collision avoidance is based on advance camera, sensors, and algorithms such as image processing, artificial intelligence, and other search algorithms.

In this simulation a simple geometric collision avoidance controller is implemented. Tuning the PID controller for drone, trajectory tracking controller and collision avoidance controller was really challenging. Specially the avoidance controller turned out to be very unstable even with negligible variation in the gains of any other controller in

the system caused performance errors. Also, a tuned controller starts to behave different in different scenarios like changing trajectory can lead to different performance. Another challenging thing was simulator (Simulink) as a lot of time after detailed inspection of faulty results the cause was found to be numerical error in the solver. Fixed-step solver was used in simulation because of a constant stepping in time to load the consistent coordinates of trajectory, and this also made simulation unstable sometimes but changing the solver to variable-step. Although it resolved stability and performance issues most of the time but with the problem of missing data points in trajectory. But with hit and trial experiments and fine tuning resulted in good performance of designed system for different scenarios.

Implementing these controllers on a flight control can be challenging but for open-source controller like Pixhawk it is comparatively easy as most of the controllers are already designed in this package. This work can be implemented on actual hardware and with few experimentations and improvements in controller tuning a good performance can be achieved.

8 Conclusion

A new concept about the use of drone in maritime industry for measuring the draft of large vessels is summarized in this thesis. This study has also covered the supportive components to make a drone operation more efficient during different lighting conditions. Details related to design specification requirements of a drone are also covered in this thesis. Some state-of-the-art techniques for controller design and use of these different controllers for autonomous drones' operations are shown in this thesis with the help of mathematical expressions and simulation result. Generation of trajectory for vessel survey and inspection is also covered in this study. Other challenges that occur for any autonomous flying vehicle are collision avoidance and stable control of the vehicle, this work demonstrated an approach to tackle this problem by covering mathematical details and doing some modifications in an existing work. All the work is implemented on MATLAB/Simulink simulation, and that can be used by anyone to study and analyze the working of such drone-based operations or to take this concept on step further by improving it or adding new concepts. Therefore, desired objectives of this study are fully tried to cover in this thesis, even though not with full perfection and precision. It will provide adequate solutions and ways forward that anyone can utilize this knowledge and data to progress into a ready to use product or a working prototype.

8.1 Future work

Advance sensors utilization for obstacle detection and recognizing the shape of and dimensions of obstacle would be very interesting as it will make it work for any kind of obstacle. And advance simulator that take into count external environmental factors, aerodynamic effects and better numerical computations is an important thing to be considered for future work. Vision-based navigation can also be implemented on this idea, and it will make it more autonomous and general for use. Another future work that can be considered is to use image procession techniques to recognize draft markings on the vessel and stopping the drone exactly at draft marking at taking high quality pictures and feeding these pictures to the operator room in real-time. Lastly a more advanced future work can be to use high performance processor to perform image processing, data processing and computations to estimate a draft reading by the drone itself.

References

[1] Frederiksen, Marianne Harbo, Knudsen, and Mette Præst. "Drones for offshore and maritime missions: Opportunities and barriers". In: *Innovation Fund Denmark* (2018).

[2] *Narvik harbour*. URL: https://www.narvikhavn.no/om-narvik-havn.aspx. (accessed: 03.25.2022).

[3] *LKAB*. URL: https://en.wikipedia.org/wiki/LKAB. (accessed: 03.25.2022).

[4] *LKAB*. URL: https://www.lkab.com/en/about-lkab/from-mine-to-port/transport/harbours/. (accessed: 03.25.2022).

[5] *LKAB Narvik harbour*. URL: https://www.skipsrevyen.no/article/oekt-malmutskiping-i-narvik/. (accessed: 03.25.2022).

[6] *Types of cargo*. URL: https://transportgeography.org/contents/chapter5/maritime-transportation/types-maritime-cargo//. (accessed: 03.25.2022).

[7] *Draft or Draught of ships*. URL: https://bit.ly/3s3IWxF. (accessed: 06.012.2021).

[8] *Draft Surveys Importance*. URL: https://bit.ly/3s4lFvJ. (accessed: 06.012.2021.

[9] *US Coast guard Maritime safety alert. accessed: 06.012.2021*. URL: https://bit.ly/3ysyiBP. accessed: 06.012.2021.

[10] *Basic ship terms, accessed: 06.012.2021*. URL: https://mullowaydive.wordpress.com/2011/04/09/77/. (accessed: 06.012.2021.

[11] *A general draft survey, accessed: 06.012.2021*. URL: https://www.gozetim.com/survey/deniz-survey/draft-survey/. (accessed: 06.012.2021.

[12] W.J. Dibble, P. Mitchell, North Of England P&I Association Staff, and North of England P & I Association. *Draught Surveys: A Guide to Good Practice*. Loss prevention guides. North of England P&I Association Limited, 2009. ISBN: 9780955825750. URL: https://books.google.no/books?id=f89CcAAACAAJ.

[13] *Narvik Day light Chart, accessed: 24.01.2022*. URL: https://www.gaisma.com/en/location/narvik.html. (accessed: 24.01.2022.

[14] Takahiro Tsujii, Hiromi Yoshida, and Youji Iiguni. "Automatic draft reading based on image processing". In: *Optical Engineering* 55 (2016).

[15] *The International Institute Of Marine Surveyors: Unit 5 Draught Surveying*. URL: https://www.iims.org.uk/product/unit-05-draught-surveying/. (accessed: 06.012.2021.

[16] D.Ray, R.Wallace, W.Eugene, and J.Michael. *Portable draft measurement device and method of use therefor*. US Patent 6347461. 2002. URL: https://patents.google.com/patent/US6347461.

[17] Huayao Zheng, Yunqian Huang, and Yinzhong Ye. "New level sensor system for ship stability analysis and monitor". In: *IEEE Transactions on Instrumentation and Measurement* 48.6 (1999), pp. 1014–1017. DOI: 10.1109/19.816106.

[18] RANXSHIC Baojia. "Ship Draft Detection Based on Machine Vision". In: ().

[19] *Autonomous drone inspections, Accessed: 06.012.2021*. URL: https://www.dnv.com/news/autonomous-drone-inspections-move-step-closer-after-successful-test--177264. (accessed: 06.012.2021.

[20] U Dahana and Raja Oloan Saut Gurning. "Maritime Aerial Surveillance: Integration Manual Identification System to Automatic Identification System". In: *IOP Conference Series: Earth and Environmental Science* 557 (Sept. 2020), p. 012014. DOI: 10.1088/1755-1315/557/1/012014.

[21] *Drone Rescue*. URL: https://www.bbc.com/news/world-australia-42731112. (accessed: 06.012.2021.

[22] *EMSA civilian maritime drone, Accessed: 06.012.2021*. URL: https://www.suasnews.com/2021/07/nordic-unmanned-and-textron-wins-eur-20-million-contract-with-emsa/. (accessed: 06.012.2021.

[23] Agata Krystosik-Gromadzińska. "The use of drones in the maritime sector–areas and benefits". In: *Zeszyty Naukowe Akademii Morskiej w Szczecinie* (2021), p. 16.

[24] *Equinor Drone Delivery, Accessed: 06.012.2021*. URL: https://www.equinor.com/en/news/20200828-drone-transport-troll.htm. (accessed: 06.012.2021.

[25] Xiangling Li, Wei Feng, Jue Wang, Yunfei Chen, Ning Ge, and Cheng-Xiang Wang. "Enabling 5G on the Ocean: A Hybrid Satellite-UAV-Terrestrial Network Solution". In: *IEEE Wireless Communications* 27.6 (2020), pp. 116–121. DOI: 10.1109/MWC.001.2000076.

[26] *Prevention of Air Pollution from Ships*. URL: https://www.equinor.com/en/news/20200828-drone-transport-troll.htm. (accessed: 06.012.2021.

[27] Ashraf Saleem, Ahmed Al Maashri, Omer Eldirdiry, Jawhar Ghommam, Hadj Bourdoucen, Amran Al-Kamzari, Ghazi Al Rawas, and Ahmed Ammari. "Detection of Oil Spill Pollution in Seawater Using Drones: Simulation amp; Lab-based Experimental Study". In: *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. 2021, pp. 1–5. DOI: 10.1109/IEMTRONICS52119.2021.9422576.

[28] *Nordic Unmanned Oilspill Monitoring*. URL: https://nordicunmanned.com/green-solutions/oilspill-monitoring/. (accessed: 06.012.2021.

[29] James Albus, Elena Messina, Robert Wade, and Woody English. "Specifying autonomy levels for unmanned systems: interim report". In: *Conference* 5422 (Sept. 2004). DOI: 10.1117/12.552074.

[30] Jie Chen, Junjie Wu, Gang Chen, Wei Dong, and Xinjun Sheng. "Design and Development of a Multi-rotor Unmanned Aerial Vehicle System for Bridge Inspection". In: vol. 9834. Aug. 2016, pp. 498–510. ISBN: 978-3-319-43505-3. DOI: 10.1007/978-3-319-43506-0_44.

[31] Hongcan Guan, Xiliang Sun, Yanjun Su, Tianyu Hu, Haitao Wang, Heping Wang, Chigang Peng, and Qinghua Guo. "UAV-lidar aids automatic intelligent powerline inspection". In: *International Journal of Electrical Power Energy Systems* 130 (2021), p. 106987. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2021.106987. URL: https://www.sciencedirect.com/science/article/pii/S0142061521002271.

[32] Tong He, Yihui Zeng, and Zhuangli Hu. "Research of Multi-Rotor UAVs Detailed Autonomous Inspection Technology of Transmission Lines Based on Route Planning". In: *IEEE Access* 7 (2019).

[33] Francisco Bonnin-Pascual, Alberto Ortiz, Emilio Garcia-Fidalgo, and Joan P Company-Corcoles. "A reconfigurable framework to turn a MAV into an effective tool for vessel inspection". In: *Robotics and Computer-Integrated Manufacturing* 56 (2019), pp. 191–211.

[34] Yuncheng Lu, Zhucun Xue, Gui-Song Xia, and Liangpei Zhang. "A survey on vision-based UAV navigation". In: *Geo-spatial information science* 21.1 (2018), pp. 21–32.

[35] Taha Elmokadem and Andrey V Savkin. "Towards Fully Autonomous UAVs: A Survey". In: *Sensors* 21.18 (2021), p. 6223.

[36] Alexander Brown, Mario Felizola, Ryan Harrington, and Daniel Mayben. "The Automatic Flying Security Drone (AFSD)". In: ().

[37] Victor Manuel Aboytes Reséndiz and Edgar Alejandro Rivas-Araiza. “System Identification of a Quad-rotor in X Configuration from Experimental Data”. In: *Res. Comput. Sci.* 118 (2016), pp. 77–86.

[38] *Quadcopter Modelling and Control With MATLAB/Simulink Implementation. Bachelor's thesis*. URL: https://www.theseus.fi/bitstream/handle/10024/333992/Usman_Muhammad.pdf?sequence=2&isAllowed=y. (accessed: 02.09.2022).

[39] Amir Hussein and Rayyan Abdallah. “Autopilot Design for a Quadcopter”. PhD thesis. Oct. 2017. DOI: 10.13140/RG.2.2.17020.80008.

[40] Chinedu Amadi and Willie Smit. “Design and Implementation of Model Predictive Control on Pixhawk Flight Controller”. PhD thesis. Dec. 2018.

[41] Ha Thanh, Bui, and Hong. “Nonlinear Control for Autonomous Trajectory Tracking while Considering Collision Avoidance of UAVs Based on Geometric Relations”. In: *Energies* 12 (Apr. 2019), p. 1551. DOI: 10.3390/en12081551.

[42] *General structure of PID controller*. URL: https://bit.ly/37JMZaP. (accessed: 04.25.2022).

[43] Ha Thanh, Nguyen Phi, and Sung Hong. “Simple nonlinear control of quadcopter for collision avoidance based on geometric approach in static environment”. In: *International Journal of Advanced Robotic Systems* 15 (Mar. 2018), p. 172988141876757. DOI: 10.1177/1729881418767575.

[44] *Different types of spline interpolation*. URL: https://thomaselove.github.io/432-notes/adding-non-linear-terms-to-a-linear-regression-model.html. (accessed: 03.25.2022).

[45] *Cubic splines*. URL: https://www.orcina.com/webhelp/OrcaFlex/Content/html/Interpolationmethods.htm. (accessed: 03.25.2022).

[46] *Cubic splines*. URL: https://www.rajgunesh.com/resources/downloads/numerical/cubicsplineinterpol.pdf. (accessed: 03.25.2022).

[47] *Single Axis Gimbal*. URL: https://noirmatter.com/pages/quark. (accessed: 01.31.2022).

[48] *Light Recommendations for night photography*. URL: https://jmlobert.blogspot.com/2016/07/flashlight-recommendations-for-night_9.html. (accessed: 01.27.2022).

[49] *TeraRanger Hub Evo for multi-sensor applications*. URL: https://www.terabee.com/shop/accessories/teraranger-hub-evo/. (accessed: 02.04.2022).

[50] *Collision avoidance sensor by terabee*. URL: https://www.terabee.com/shop/lidar-tof-range-finders/teraranger-evo-3m/. (accessed: 02.04.2022).

[51] *Ultrasonic Rangefinder*. URL: https://www.maxbotix.com/ultrasonic_sensors/mb1240.htm. (accessed: 02.06.2022).

[52] *Hawkeye camera*. URL: https://www.fireflycameras.com/products/hawkeye-firefly-micro-2-preorder?variant=14999759028290. (accessed: 02.07.2022).

[53] *Holybro X500 V2 Kit*. URL: http://www.holybro.com/product/x500-v2-kit/. (accessed: 02.09.2022).

[54] *R81 ReceiverHolybro 2216 KV920 Motor specifications*. URL: https://bit.ly/3stkJAs. (accessed: 03.25.2022).

[55] *Holybro ESC BLHeli S ESC 20A*. URL: https://www.flyingtech.co.uk/electronics/holybro-blheli-s-esc-20a-2-4s-speed-controller. (accessed: 02.10.2022).

[56] *1045 Propellers*. URL: https://www.3dxr.co.uk/multirotor-c3/multirotor-frames-c97/holybro-propellers-1045-2-pair-for-x500-frame-kit-p3725. (accessed: 02.10.2022).

[57] *1045 Propellers*. URL: https://www.jsumo.com/10-propeller-1045-1045r-pair. (accessed: 02.10.2022).

[58] *Holybro Power distribution board*. URL: https://www.3dxr.co.uk/electronics-c78/power-management-c91/power-distribution-boards-pdbs-c191/holybro-pdb-board-for-x500-v2-kit-p4983. (accessed: 02.10.2022).

[59] *Pixhawk 5x flight controller*. URL: https://shop.holybro.com/pixhawk-5x_p1279.html. (accessed: 04.25.2022).

[60] *Holybro M8N GPS sensor*. URL: https://shop.holybro.com/holybro-m8n-gps_p1094.html. (accessed: 04.25.2022).

[61] *4s 5000mAh - 30C - Spectrum LiPo*. URL: https://www.elefun.no/p/prod.aspx?v=51518/. (accessed: 02.04.2022).

[62] *PM02D Power Module*. URL: https://shop.holybro.com/pm02d-power-module_p1285.html/. (accessed: 02.04.2022).

[63] *TX12 Radio Controller*. URL: https://www.radiomasterrc.com/collections/transmitter/products/tx12-radio-controller. (accessed: 03.25.2022).

[64] *R81 Receiver*. URL: https://www.radiomasterrc.com/collections/receivers-1/products/r81-receiver. (accessed: 03.25.2022).

[65] *Quadcopter Foldable Floating System*. URL: https://www.amazon.com/Thekkiinngg-Gear-Quadcopter-System-Set-Inflatable-Compatible/dp/B0784TY3NG. (accessed: 03.25.2022).

[66] *Floating landing gear*. URL: https://www.amazon.com/Owoda-Floating-Extension-Extender-Training/dp/B08Z44YF5G?th=1. (accessed: 03.25.2022).