LANDING OF A QUADROTOR UAV SWARM IN A NONSTATIONARY CONFINED COMPARTMENT

by

RICK SCHIENI

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Shahab Shojaei-Zadeh

And approved by

_____

_____

_____

New Brunswick, New Jersey

OCTOBER, 2017

ABSTRACT OF THE THESIS

Landing of a Quadrotor UAV Swarm in a Nonstationary Confined Compartment

By RICK SCHIENI

Thesis Director:

Dr. Shahab Shojaei-Zadeh

Unmanned Aerial Vehicles (UAVs) have emerged as an extraordinarily useful technology in both civilian and military applications. Recent efforts have focused on expanding the capabilities of individual UAVs to the application of multiple vehicle swarms to efficiently accomplish otherwise laborious and dangerous tasks. In this work, a trajectory generation method is presented to safely land the individuals of a UAV team on a moving vessel following the performance of a team mission. It is assumed that the landing vessel has a compartment dedicated to vehicle storage which restricts the final landing maneuver to a confined space. Trajectories are generated by solving a constrained optimization problem in a computationally efficient manner by exploiting the properties of Pythagorean Hodograph Bèzier curves. A case study is presented to demonstrate the efficacy of the proposed trajectory generator. The study examines the effectiveness of the method to create successful landing trajectories for the individuals of the UAV swarm. It is shown that the method creates collision-free trajectories for multiple vehicles as they attempt to land in a confined compartment on a moving target.

Acknowledgements

I would first like to thank Dr. Shahab Shojaei-Zadeh, my advisor, for guiding me through this incredible journey of discovery and scholarship. Dr. Shojaei-Zadeh's excellent mentorship catalyzed more learning than I ever thought possible. I would also like to thank my thesis committee members, Dr. Drazer and Dr. Diez for taking the time to review this thesis.

I thank my entire research group, especially Dr. German Drazer for his wisdom and challenging inquiries posed during our group meetings. I am in a great debt to my lab-mates and fellow group members for their support: Omer Alnuaemie, Robert Weir, Siqi Du, Tianya Yin, Minglu Li, Yu Han, David Cunningham, and Isabel Liberis. Many thanks are also paid to my good friend and classmate Boris Plakalovic.

My graduate studies would not have been possible without Dr. Jerry Shan who shepherded my transition to graduate school as a young naïve undergraduate student.

Finally, I would like to thank my mother for her encouragement and undying belief that I could accomplish anything that I wanted. My graduate career has undeniably been the most interesting and transformative time of my life; without her unprecedented amount of love and care this work would never have been possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1. Introduction

1.1. Background

1.1.1. History and Application

Greek mythology tells of the aerial escape of Icarus and his father Daedalus from their entrapment in a high tower through the use of wings fashioned from feathers and wax [1]. The utility of conquering the skies and the power of flight would continue to be recognized throughout the ages. Later attempts to fly include the efforts of the 9th century poet Abbas Ibn Firnas and 11th century monk Eilmer of Malmesbury [2]. Leonardo DiVinci attempted to tackle the problem by modelling his flying technology after observations of flight in nature [3], a theme that would continue to be replicated throughout time. Successful attempts to fly aircraft would come in two forms: lighter-than-air technologies and heavier-than-air technologies. Lighter-than-air technologies are composed of "balloons" which utilize a gas less dense than air to achieve flight. Heavier-than-air technologies depend on mechanics rather than such a gas. Sustained flight using such aircraft was famously achieved through the work of the Wright brothers [4–6], whose invention would eventually evolve into the airplane of today. Other heavier-than-air vehicles such as rotor driven aircraft had been originally conceived in China around 400 BC and resulted in the first successful helicopter design in 1939 by Igor Sikorsky [7].

Since the inception of reliable fixed-wing and rotor-driven heavier-than-air aircraft, innumerable tasks have been achieved through the employment of such vehicles. For

certain tasks, both functionality and necessity fueled the desire to remove humans from being on-board the vehicles. Such motivation continues today as the field of unmanned aerial vehicles (UAVs) has been evolving rapidly. Perhaps one of the most potent motivations to develop UAVs is the removal of human beings from the battlefield. This notion was first expressed through the use of unmanned balloons by the Austrians to float over enemy territory and drop bombs in 1849 [8]. Today's military applications include Intelligence, Surveillance, and Reconnaissance (IRS) [9] as well as target identification, tracking, and prosecution [10]. UAVs have also proven to be useful in many civilian industries such as agriculture [11], disaster management [12], and natural resource management [13]. High maneuverability and mobility along with compact size allow the vehicles and their operators to explore areas that would otherwise be considered inaccessible. Installation of sensors onboard the aircraft, video cameras being one of the most popular, allow observations and measurements to be made in such hard to reach areas. Figure 1.1 presents examples of both fixed-wing and rotor driven UAV designs.

Figure 1.1: Examples of Fixed-Wing and Quadrotor UAVs (a) The Northrop Grumman RQ-4 Global Hawk Fixed-Wing UAV [14], (b) The Storm RC Hobby 550 mm RTF Quadcopter UAV [15].

With individual UAVs proving to be useful in such a wide array of applications, much effort is currently being devoted to the study of what can be achieved with large

numbers of coordinated UAVs working together to achieve a common goal. Large numbers of coordinated vehicles may be able to reflect the notion of "strength in numbers," and achieve tasks that would otherwise be impossible.

Observations made in nature demonstrate the advantages of coordinating large numbers of individuals. Schooling of fish is believed to be a defense against predation [16]. Large numbers of individual fish aids defense efforts by increasing vigilance and detection of predators [17], assessment of predators [18], predator information transfer [19–21], predator inhibition [22], and promoting evasive maneuvering [16]. Such defense mechanisms are observed in other animals such as birds and insects as well [23]. Grouping has not only served as a tool for prey, but has been utilized by predators as well. Cooperative hunting in which multiple predators will work together in order to capture prey is exhibited in mammals [24–27], marine vertebrate [28,29], and birds of prey [30]. The underlying motivation which produces such cooperative hunting in groups is the ability to capture prey that could not be captured by an individual predator. In these cases, the prey is overwhelmed by the sheer number of predators rather than the physical dominance of a single hunter.

Although an important attribute, the ability of groups to overwhelm a competitor is not the only useful function of grouping. More individuals corresponds to more tools and flexibility at the disposal of the group and thus, tasks of a more functional nature can be completed as a result of grouping behavior. Members of the insect world exhibit large amounts of self-assembly in order to construct structures in order to achieve certain tasks [31–33]. Ants in particular demonstrate such behavior well as they have been observed to use their bodies to build bridges [34] and form buoyant waterproof rafts in order to survive

floods [35]. Returning to predation, the mechanics of having a large number of cooperative predators allows such creatures to scan more area and detect prey quickly and efficiently [36]. Other mechanical advantages take the form of energy conservation such as when geese fly in formations which reduce the aerodynamic drag experienced by other individuals in the group [37].

Demonstrated to be beneficial in a variety of diverse ways in nature, such concepts and tactics have been adapted to perform tasks using groups of vehicles. Tasks which require coverage of large areas such as agriculture may benefit greatly from the utilization of UAV swarms [38]. After natural disasters, it is often necessary to perform search and rescue (SAR) or infrastructure inspection. Such tasks usually must be performed in dangerous environments and require large area coverage from the inspector which has motivated the use of UAV swarms in such situations [39–42]. The utility of unmanned vehicles as well as their rapidly decreasing cost of production has made UAVs prime candidates to be implemented in such systems.

A decrease in cost has made such vehicles disposable which is an attractive property for military applications. Swarms of vehicles may lose individual members yet still have the ability to achieve the overall goal. In addition to disposability, throughout much of history combat tactics have exploited the aforementioned idea of overwhelming prey, or in the combat sense, adversaries with large numbers of cooperating individuals [43–45]. Natural evolution and escalation implies that if one side intends on using groups of UAVs offensively then their enemy may employ such weaponry as well. Hence, groups of UAVs will be implemented not only to attack targets but to defend friendly assets as well. Military

applications are not limited to offense/ defense but can also include uses such as surveillance, target tracking, and delivery of materials among many others [44,46–50].

With the current field applications of UAVs requiring at least one off-board human operator per vehicle, the employment of large numbers of UAVs would require large numbers of human controllers. The U.S. Navy's LOCUST UAV swarming system has demonstrated the ability of groups of fixed wing UAVs to fly autonomously but are launched like a projectile from a tube requiring repackaging and reloading for subsequent missions [50]. This has motivated the desire to create swarms of UAVs which can fly autonomously while being able to land and redeploy with no human interference. Physical constraints on the UAV systems such as generally poor battery life has reinforced the desire to stop, recharge, and go as necessary both mid and post-mission.

Many military applications may benefit from autonomous landing and redeployment of UAV swarms from other vehicles such as ships and other seafaring vessels. Such a vessel would have a certain area designated to the housing and recharging of the individual UAVs. In this thesis, the landing of quadrotor UAVs into a moving landing compartment is considered. Such a compartment is modelled as a storage volume whose entrance is found on the top deck of the ship and extends downward into the ship's hull.

To ensure collisions do not occur as multiple vehicles attempt to enter the confined space dedicated to vehicle housing, observations from nature are made. In [51], a flock of approximately 1800 chimney swifts (Chaetura pelagica) are observed entering the $1.2 \times 1.0m$ rectangular entrance of a chimney that the birds had selected as their overnight roost. A circular flock shape is adopted by the birds as it solidifies a common approach

trajectory and allows individuals that fail to land to circle back and attempt to land again. The work suggests that entry into the roosting location was dictated by local interaction rules among the birds. A common flocking behavior referred to as alignment in which individuals match their velocity to that of their neighbors was employed by the birds at distances further from the entrance, but upon their approach to the landing site, more competitive behavior was observed. Becoming less cooperative and competing for space to enter the chimney, the final entrance behavior involves split-second decision making between entering the chimney and avoiding collisions. In this work, a safer and more conservative approach is adopted to the landing compartment entrance problem for application to quadrotor UAVs.

1.1.2. The Quadrotor UAV

As the potential applications of aircraft have grown, so have the desired features and requirements of the vehicles. For example, urban military applications require vehicles which are fast yet maneuverable enough to navigate confined spaces. With fixed-wing aircraft suffering from minimum turning radii and gradual changes in elevation, rotor-driven aircraft have become the popular choice for such applications. The basic flight mechanics of a quadrotor UAV are presented in Figure 1.3 to illustrate why such vehicles have been employed in tasks requiring high maneuverability.

This thesis will only consider fixed-pitch multirotor driven aircraft where each rotor is fixed with respect to the rigid body of the vehicle. The rotors are assumed to be orientated so that the thrusts created by each of the four rotors act in the same direction. Figure 1.2 illustrates a simplified version of such an aircraft.

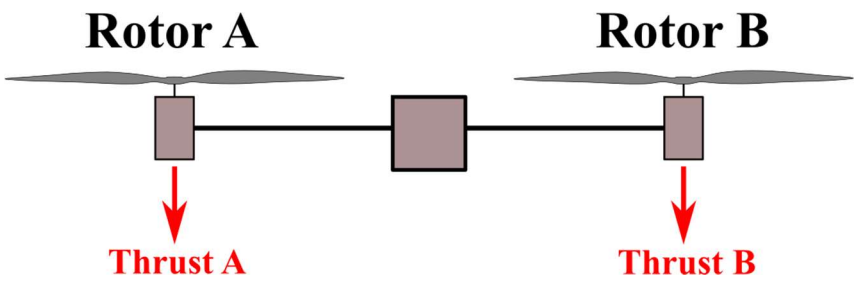


Figure 1.2: Side View of Quadcopter Creating Thrust.

As follows from its name, a quadrotor UAV, sometimes called a quadrotor helicopter or quadcopter, is propelled by four independent rotors which produce a thrust force along and a moment about the axis of rotation. Propelled by four rotors yet having three translational degrees of freedom (DOF) and three rotational DOF, the quadcopter is an underactuated system. General quadcopter motion is illustrated in Figure 1.3.

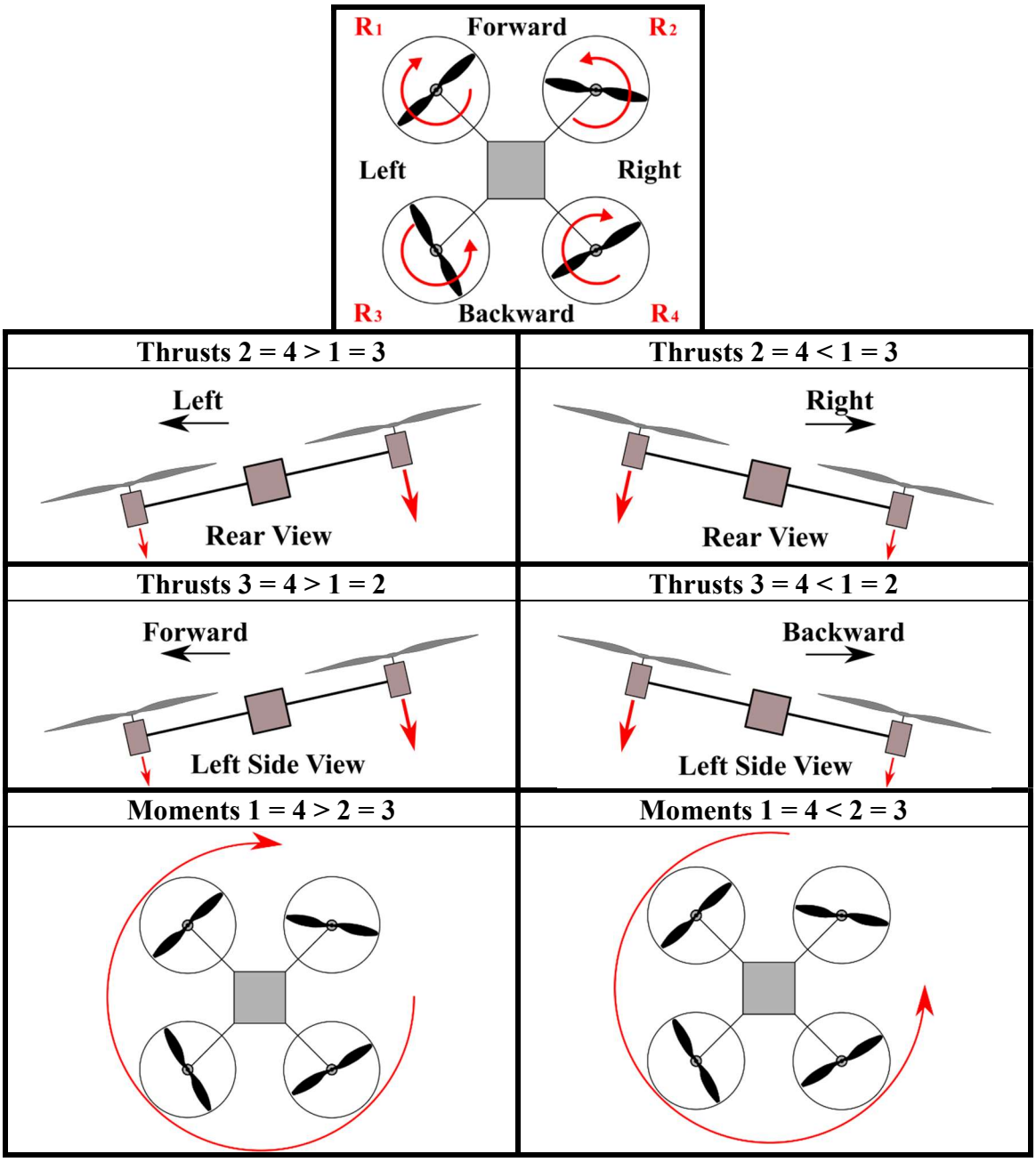


Figure 1.3: Quadcopter Flight Mechanics.

Of course, the quadcopter is not constrained to the illustrated imbalances in rotor thrusts. The thrust of each rotor can be controlled independent of the other rotors and thus, any combination of the above motions can be achieved at any time.

1.2. Literature Review

Upon completion of their mission abroad, the UAVs of the swarm will need to return to an appropriate area to prepare for the next mission. Returning to their maintenance docks entails the safe landing and recovery of the vehicles from abroad. Vehicle motion is the result of interactions between environmental forces acting on the vehicle and those forces produced by the vehicle's own actuators. These actuators are controlled by the vehicle control system which attempts to have the vehicle achieve some desired state. Often, the desired state is entered into the system by a human operator but the substantial number of vehicles and the desire for autonomy motivates the need to have a UAV determine its own desired state [52]. It is for this reason that the manner in which a UAV may best dictate its own desired state which achieves the overall goal of a safe and efficient landing of the entire swarm is investigated.

Until 1987, the aggregate motion of natural systems such as schools of fish or flocks of birds was difficult to replicate in computer animation. Instead of programming each agent's path individually, Craig Reynolds developed a distributed behavioral model which achieved aggregate flocking behavior through simple individual behaviors [53]. Looking again to nature for inspiration we see that groups of animals are essentially systems of individual autonomous agents whose simple behaviors gives rise to complexity within systems such as flocks, schools or herds [54–56]. In his work, Reynolds introduces the following three simple behaviors called flocking rules that each individual, or "boid" as he referred to them, abides by: collision avoidance, velocity matching and flock centering. The rules simply state that an agent will evaluate its neighbors in its flock and attempt to match their velocity and remain close to its neighbors but not so close that a collision may

occur. Later work went on to show that the emergent behavior of the group may be determined by who an agent considers to be its neighbor [57] and how much relative effect each force has on the individuals motion [58–62]. Such differences in the emergent behavior include the flocks forming into a “march” or becoming stuck in a state of perpetual cyclical motion [63].

Upon the inception of the idea of flocking/ swarming UAVs, emergent flocking behaviors were naturally proposed to provide the necessary intelligence to achieve flocking behavior. Building upon the numerous existing Reynolds inspired flocking algorithms [64], simulations of flocking rule driven UAV swarms which include dynamic and aerodynamic effects have been developed [65–71]. Simulations have shown that UAVs can successfully be guided by the flocking rules in in certain situations. Clark et al. extended such work and performed flight tests of aircraft guided by the flocking rules, more formally referred to as boid guidance algorithms [72]. The flight tests included only two fixed-wing unmanned aircrafts but illustrated the capability to control more than one UAV by employing boid guidance algorithms.

The lack of verification associated with the emergent flocking behavior makes it difficult and unlikely to be implemented in tasks requiring high precision such as navigating buildings and landing of the vehicles [73]. Several multi-vehicle UAV testbeds and projects such as the UltraSwarm [74], SwarMAV [75], the GRASP testbed [76], MIT’s Indoor Multi-Vehicle Flight Testbed [77,78], SUUAVE [79], Airshield [80], and the BEAR project [81] have demonstrated the ability of UAV swarms to navigate difficult terrains and obstacles without collision. These works employ complex control systems which inhibit their ability to be incorporated in real-world systems. For example, in [82]

the authors demonstrate their system's ability to navigate formations of UAVs through and around obstacles but relies upon a complex motion capture system to provide the feedback necessary to control the group. However, from the impressive guidance systems which allow these systems to move with such precision we derive information on how to plan the vehicle's path of travel precisely enough to be applied in a landing situation.

The aforementioned control-driven multiple vehicle systems often employ a path planner and/or a trajectory planner to provide the vehicle with a reference plan of travel to be followed. A differentiation exists between path planning and trajectory planning. UAV path planning addresses the problem of finding an optimal collision-free path from one point to another in three-dimensional space while trajectory planning does the same while considering the time in which the path is traversed [83,84].

Yang et al. have recently provided a survey of 3D robot path planning algorithms [85] as an expansion to their earlier paper which reviewed the literature of 3D UAV path planning [86]. Although the authors present algorithms for the 3D path planning of a single UAV, the multiple vehicle case is most often treated as an expansion of the single vehicle case and thus the review work presented is worth exploring. A taxonomy of the path planning algorithms was presented and broke the works into five main categories: sampling based algorithms, node based optimal algorithms, bio-inspired algorithms, and multi-fusion based algorithms. Sampling based algorithms generally either sample and map the system environment or just randomly search for an optimal path. Examples of sampling based algorithms include 3D Voronoi graphs [87], Rapidly-exploring Random Graph [88], Rapidly-exploring Random Tree (RRT) [89], artificial potential fields [90] and more. Node based optimal algorithms generate paths by navigating a set of nodes and include Dijkstra's

algorithm [91], A-Star (A*) [92], and all of their variations. Mathematical based algorithms model the environment as well as the vehicle and minimize a cost function that has bounded by the appropriate equalities and inequalities. Linear programming [93–96] and optimal control [97] fall under this category of algorithms. Bio-inspired algorithms mimic naturally occurring phenomenon and includes evolutionary algorithms [98–100] and colony optimization [101,102]. The final class, multi-fusion based algorithms, are those algorithms which involve combining members of any of the aforementioned categories to solve a path planning problem.

In many cases trajectory planning takes the path of travel generated by the path planner and tries to determine how a robot can feasibly traverse the path. This is accomplished by connecting a series of waypoints determined from the path planning stage and connecting them by a time-parameterized polynomial [85]. Polynomials are a natural choice to generate smooth trajectories which can be followed by a quadrotor UAV. In [103], Mellinger et al. demonstrated how minimizing the fourth derivative, or snap, of polynomial trajectory creates smooth, flyable trajectories. This idea has been expanded upon in many works, notably [104] where minimum snap trajectories were generated along with the time needed to traverse them being optimized. Minimum snap polynomial trajectories were extended to a special class of curves, called Bèzier curves, in [105]. Bèzier curves are particularly useful for trajectory generation as they exhibit properties that allow system dynamics to be easily incorporated in the trajectory generation procedure.

Bèzier curves have been employed by researchers to generate trajectories for multiple UAV systems which makes them attractive for the applications of this thesis. In [106,107], Bèzier curves were used to generate trajectories while avoiding collisions for

teams of cooperating vehicles. [108] specifically examines collision prediction and avoidance, making the necessary changes in the flight plan to avoid the collision. The algorithm proposed in [108] is extended in [109] to be able to re-plan piecewise trajectories resulting in a more dynamic approach. Dynamic re-planning of trajectories is also employed in [110] where Bèzier curves modelled the trajectory of one robot attempting to intercept another. Such interception is a task reminiscent of the landing on a moving platform problem.

1.3. Scenario Description

As discussed earlier, swarms of quadrotor UAVs have the potential to be employed in a variety of different applications. One special class of applications that this work will focus on are military applications in which the quadcopters are launched from and return to a large seafaring military vehicle. In this study, it is assumed the swarm has been deployed and has executed its intended mission abroad. The vehicles must return from the mission site and be safely recovered. Recovery entails the UAVs autonomously landing in a specified landing area on the vehicle from which they were launched. Within the landing area are docks which are equipped with the proper tooling to recharge and relaunch the vehicles with no human assistance.

1.4. Contributions of the Thesis

As the missions which can be accomplished by teams of multiple quadrotor UAVs continue to become more diverse, the methods by which the groups of UAVs are deployed and are recovered must become more diverse and robust as well. Motivated by the need to deploy and recover UAV swarms from seafaring vehicles which are limited in the amount of room available to house the UAVs, this thesis formulates a trajectory generation method

specific to the landing of multiple quadcopters in a moving confined compartment. The method is capable of guiding the UAVs to their landing position on a moving landing vessel without colliding with the walls of the compartment in which they are to be stored. Inter-vehicle collisions are guaranteed to be avoided as well. Generated trajectories are ensured to be flyable by taking the dynamic limitations of the vehicle into account during the trajectory generation stage.

1.5. Organization of the Thesis

This chapter has provided some historical perspective and general motivation for the study of landing quadrotor UAV swarms on moving watercraft. To present the steps taken to address this problem, the thesis is organized as follows. Chapter 2 introduces a trajectory generation method for quadrotor UAVS. Chapter 3 explores how the trajectory generator may be specifically applied to the landing of a quadcopter swarm in a nonstationary confined compartment. Next, Chapter 4 demonstrates the efficacy of the trajectory generation method by examining a UAV swarm landing case study. Finally, Chapter 5 presents the conclusions made from the investigation and recommends further steps to be taken to better understand the nature of landing a swarm of quadcopters on a seafaring vessel.

Chapter 2

MODEL FORMULATION

2. Model Formulation

2.1. Quadrotor UAV Model

2.1.1. Reference Frames

A single quadrotor UAV can be described by first defining two frames of reference: the inertial global frame, *G*, and a body frame, *B*. Three orthonormal axes $\overrightarrow{x_G}$, $\overrightarrow{y_G}$, and $\overrightarrow{z_G}$ constitute the global frame where $\overrightarrow{z_G}$ is defined to point upwards, opposite the direction of gravitational acceleration, as can be seen in Figure 2.1. The origin of the body frame is fixed to the center of mass of the quadcopter with $\overrightarrow{z_B}$ defined to point in the direction of the thrust created by the four constant-pitch rotors. Axes $\overrightarrow{x_B}$ and $\overrightarrow{y_B}$ are then assigned arbitrarily along two perpendicular arms of the quadcopter. As derived and illustrated in-depth in Appendix A, Z – X – Y Euler angles are employed to map the rotational relationship between the two frames. To translate from *G* to *B*, a rotation is performed about $\overrightarrow{z_G}$ by the yaw angle, φ, then around the intermediary x-axis by the roll angle, ϕ, and finally about the second intermediary y-axis by the pitch angle, θ. This relationship results in the following rotation matrix which transforms vectors expressed in the body frame to their equivalent expression in the global frame,

$$ {}^{G}_{B}\boldsymbol{R} = \begin{bmatrix} c\varphi c\theta - s\varphi s\theta s\phi & -c\phi s\varphi & s\theta c\varphi - c\theta s\varphi s\phi \\ c\theta s\varphi + s\theta s\phi c\varphi & c\phi c\varphi & s\varphi s\phi - c\varphi c\theta s\phi \\ -c\phi s\theta & s\phi & c\theta c\phi \end{bmatrix}. \tag{2.1} $$

In Equation (2.1), cφ and sφ are abbreviations for the cosine and sine of the yaw angle, φ, and similarly for the roll and pitch angles. Including the translational displacement of the body frame, the complete transformation matrix [111] can be expressed as:

$$ {}^{G}_{B}\boldsymbol{T} = \begin{bmatrix} c\varphi c\theta - s\varphi s\theta s\phi & -c\phi s\varphi & s\theta c\varphi - c\theta s\varphi s\phi & {}^{G}x_{BO} \\ c\theta s\varphi + s\theta s\phi c\varphi & c\phi c\varphi & s\varphi s\phi - c\varphi c\theta s\phi & {}^{G}y_{BO} \\ -c\phi s\theta & s\phi & c\theta c\phi & {}^{G}z_{BO} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.2} $$

Where ${}^{G}x_{BO}$, ${}^{G}y_{BO}$, and ${}^{G}z_{BO}$ are the x, y, and z coordinates of the UAV's center of mass in terms of the global frame *G* respectively.

The angular velocity of the vehicle with respect to the global frame is synonymous with the angular velocity of frame *B* in frame *G*. Denoting the components of the angular velocity in the body frame as p, q, and r, the angular velocity as observed in the global frame is expressed as:

$$ {}^{G}_{B}\vec{\omega} = p\,{}^{G}\overrightarrow{x_B} + q\,{}^{G}\overrightarrow{y_B} + r\,{}^{G}\overrightarrow{z_B} \tag{2.3} $$

where ${}^{G}\overrightarrow{x_B}$, ${}^{G}\overrightarrow{y_B}$, and ${}^{G}\overrightarrow{z_B}$ are the vectors describing the body frame axes in the global frame. According to the rotational relationships described previously, the angular velocity can be expressed in terms of the time derivatives of the roll, pitch, and yaw angles as

$$ \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix}. \tag{2.4} $$

2.1.2. Quadcopter Dynamics

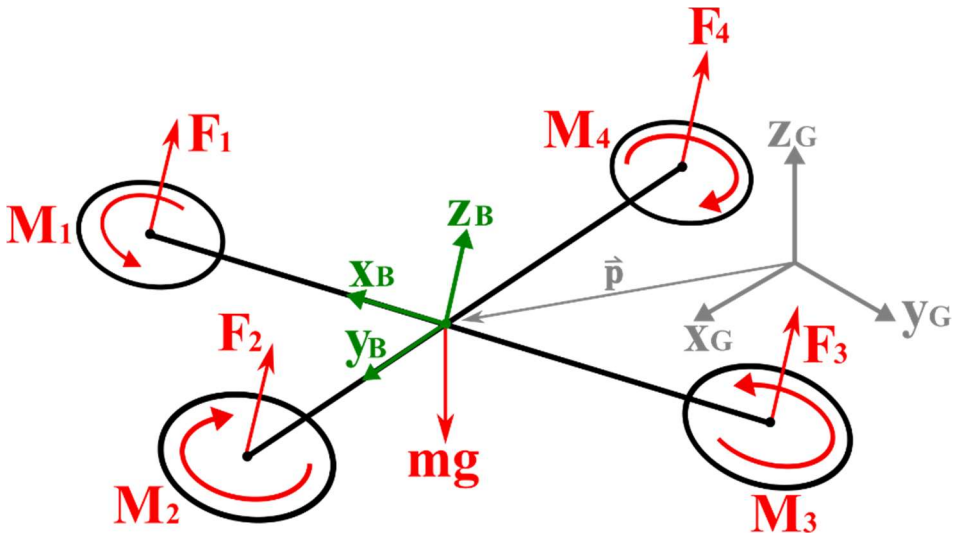


Figure 2.1: Quadcopter Forces and the Reference Frames.

In this model, the external forces acting on the vehicle are gravity in the negative $\overrightarrow{z_G}$ direction and the thrust from each of the four rotors in the positive $\overrightarrow{z_B}$ direction. The acceleration of the vehicle's center of mass can thus be described by Newton's equations in frame *G* as:

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = {}^{G}_{B}\boldsymbol{R}\begin{bmatrix}0\\0\\T\end{bmatrix} + \begin{bmatrix}0\\0\\-mg\end{bmatrix} \tag{2.5}$$

where m is the mass of the UAV, g the acceleration of gravity, and T the aggregate thrust created by the rotors.

External moments are created about $\overrightarrow{x_B}$ and $\overrightarrow{y_B}$ due to the displacement of the rotors from the vehicle's center of mass while moments about $\overrightarrow{z_B}$ stem from the rotors' ability to create a tendency to rotate about their own axis of rotation. Rotor 1 is defined to be the counterclockwise spinning actuator on the arm that runs along $\overrightarrow{x_B}$, Rotor 2 to be the clockwise spinning actuator along $\overrightarrow{y_B}$, Rotor 3 to spin counterclockwise on the $-\overrightarrow{x_B}$ axis, and Rotor 4 to be spinning clockwise on the $-\overrightarrow{y_B}$ arm. Denoting the equal lengths of the

quadcopter arms as L, and the quadcopter's inertial tensor in B as **I**, application of Euler's equations to the system in the body frame results in the following rotational relationships:

$$\mathbf{I}\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix} L(F_2 - F_4)\\ L(F_3 - F_1)\\ M_1 - M_2 + M_3 - M_4\end{bmatrix} - \begin{bmatrix}p\\ q\\ r\end{bmatrix} \times \boldsymbol{I}\begin{bmatrix}p\\ q\\ r\end{bmatrix}. \tag{2.6}$$

Throughout this thesis, the vehicle is assumed to be axisymmetric thus, the only nonzero elements of **I** are the diagonal entries I_{xx}, I_{yy}, and I_{zz}.

The state of the quadrotor system is completely defined by twelve state variables consisting of both the linear and rotational displacements and their respective velocities,

$$\vec{X} = [x \quad y \quad z \quad \phi \quad \theta \quad \varphi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\varphi}]^{\mathrm{T}}.$$

2.1.3. System Inputs

Capable of six degree of freedom (DOF) travel yet propelled by only four actuators, the quadrotor UAV is an underactuated system. Although its underactuated nature leads to complicated dynamics, the quadcopter is fortunate in the fact that a simple mapping exists between the rotor speeds and the forces/ moments created. For each rotor with an angular velocity denoted ω_i, a force in the $\overrightarrow{z_B}$ direction is created according to

$$F_i = k_F \omega_i^2. \tag{2.7}$$

In this equation, k_F is an empirically determined constant. Another such constant, k_M, exists to model the moment produced by each rotor with angular velocity ω_i such that

$$M_i = k_M \omega_i^2. \tag{2.8}$$

Although not considered in this dissertation, a vector of control inputs, $\vec{u}$, is defined for illustrative purposes. Defining the first control input as the total thrust generated and the following inputs to be the body moments about $\overrightarrow{x_B}$, $\overrightarrow{y_B}$, and $\overrightarrow{z_B}$ respectively, the control inputs are expressed as

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}. \tag{2.9}$$

2.1.4. Differential Flatness

A system's dynamics is said to be differentially flat if there exists a set of chosen parameters, called flat outputs, such that the state, $\vec{X}$, and the control inputs, $\vec{\mathrm{u}}$, can be expressed as algebraic functions of the flat outputs and their derivatives. By definition, a dynamic system $\dot{\vec{X}} = f(\vec{X}, \vec{\mathrm{u}})$, where f is a smooth vector field, is differentially flat if the vector of flat outputs, $\vec{\rho}$, can be expressed as

$$\vec{\rho} = h(\vec{X}, \vec{\mathrm{u}}, \dot{\vec{\mathrm{u}}}, \ddot{\vec{\mathrm{u}}}, \cdots, \vec{\mathrm{u}}^{(r)}) \tag{2.10 a}$$

such that

$$\vec{X} = \alpha(\vec{\rho}, \dot{\vec{\rho}}, \ddot{\vec{\rho}}, \cdots, \vec{\rho}^{(q)}) \tag{2.10 b}$$

$$\vec{\mathrm{u}} = \delta(\vec{\rho}, \dot{\vec{\rho}}, \ddot{\vec{\rho}}, \cdots, \vec{\rho}^{(q)}) \tag{2.10 c}$$

where h, α, and δ are smooth functions. For the case of a quadrotor UAV, the flat outputs, ρ_i, are defined to be the components of the vehicle's position in G and the yaw angle, φ, resulting in the vector

$$\vec{\rho} = [x \quad y \quad z \quad \varphi]^{\mathrm{T}}.$$

A detailed account of the relationships between the proposed flat outputs and the vehicle's state and inputs are appended in Appendix B. In Chapter 3, the differentially flat nature will allow the dynamic constraints of a quadcopter to be expressed in terms of the vehicle's trajectory and its derivatives.

2.2. The Landing Region

Motivated by the real-world scenario of landing a swarm of quadcopter UAVs on a seafaring military vehicle such as an aircraft carrier, a mathematical representation of a moving "ship" with a region dedicated to the landing and storage of the UAVs is introduced. In this thesis, scenarios of the vehicles landing in cylindrical compartments in the ship's top deck will be addressed.

2.2.1. The Ship Frame of Reference

A new reference frame, *S*, is added to the system to relate the ship's motion to the global frame. The frame consists of the three orthonormal axes $\overrightarrow{x_S}$, $\overrightarrow{y_S}$, and $\overrightarrow{z_S}$ and is restricted in its movements with respect to the global frame. For simplicity, the origin is defined to lay at the imaginary center of gravity of the ship which will be assumed to lay at the center of the ship's top deck. The height of the top deck above sea level is denoted H_S such that in the global frame, the origin of the ship frame exists at a constant altitude of ${}^{G}z_{SO} = H_S$. Thus, the constant altitude implies that the origin of frame *S* is restricted to translational motion in the $\overrightarrow{x_G}$-$\overrightarrow{y_G}$ plane. Figure 2.2 illustrates the relationship between the Global and Ship frames.

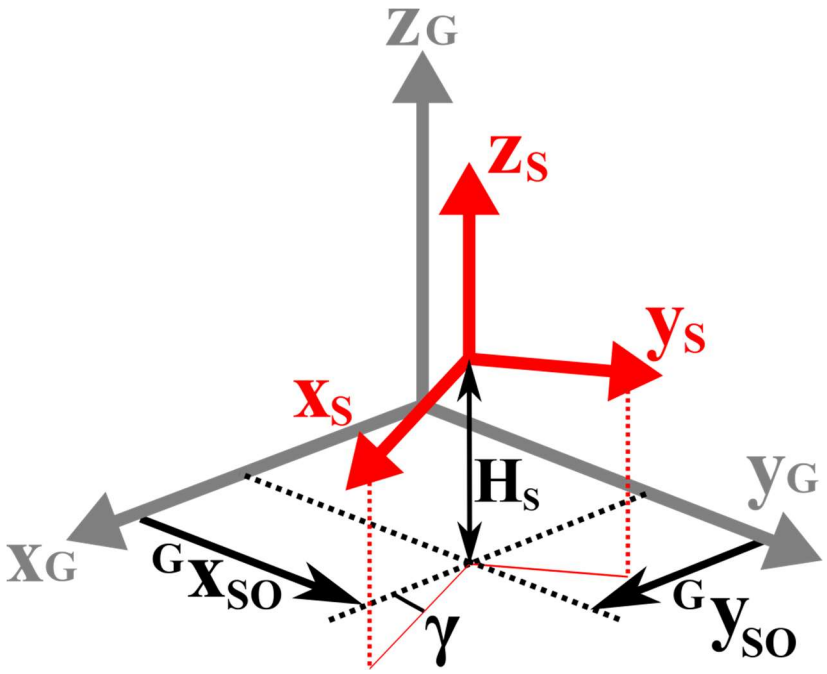


Figure 2.2: The Ship Frame of Reference.

Axis $\overrightarrow{x_S}$ of *S* extends in the direction of vessel motion and axis $\overrightarrow{y_S}$ extends through the port side of the ship. Axis $\overrightarrow{z_S}$ is defined upwards, opposite gravity, and remains in such an orientation since the ship frame is unable to rotate about the global $\overrightarrow{x_S}$ or $\overrightarrow{y_S}$ axes. The orientation of the ship in the $\overrightarrow{x_G}$-$\overrightarrow{y_G}$ plane is represented by the angle γ, measured between $\overrightarrow{x_G}$ and $\overrightarrow{x_S}$. Such relationships are enforced mathematically according to the transformation matrix:

$$ {}^{G}_{S}\boldsymbol{T} = \begin{bmatrix} c\gamma & s\gamma & 0 & {}^{G}x_{SO} \\ -s\gamma & c\gamma & 0 & {}^{G}y_{SO} \\ 0 & 0 & 1 & H_S \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.11} $$

2.2.2. Landing Docks

It is assumed that in a real-world scenario, a ship would dedicate a certain area of its deck, and the volume below such areas, to the landing and storage of multiple UAVs. Capable of caring for N_V quadcopters, the landing region contains N_D landing docks described in the ship frame of reference, where $N_D = N_V$ such that each quadcopter may have a place to land. In this thesis, the landing region will be modelled as a compartment within the ship's hull which contains landing docks whose locations are fixed in the ship frame of reference.

2.3. Trajectory Generation

Concerned with the matter of landing multiple quadcopter UAVs on a moving platform, there is an inherent desire to understand the vehicle's motion along its final approach and the motion of the landing maneuver itself. Thus, for N_V vehicles the goal is to generate N_V trajectories that satisfy the dynamic constraints of the vehicle as well as the constraints imposed by the nature of the mission. For each vehicle "i" the goal is to generate

$$\vec{p}_i: [0, t_i^F] \rightarrow \mathbb{R}^3 \qquad i = 1,2,\cdots, N_V \tag{2.12}$$

wwww`.

Following from the work performed in [112,113], trajectories are generated by decomposing the problem into finding a purely geometric spatial component and a separate temporal component. This method has been shown to increase flexibility in the trajectory generation method when multiple vehicles must be separated by a distance at any point in time along with satisfying boundary conditions, dynamic constraints, and mission-specific constraints [107]. To develop the spatial path, the dimensionless parameter $\zeta_i \epsilon [0,1]$ is

introduced to replace time as the parameter upon which the vehicle location will depend. It then follows that the spatial path is defined by the mapping

$$\vec{p}_i(\zeta_i): \rightarrow \mathbb{R}^3 \qquad \zeta_i \epsilon [0,1] \qquad i = 1,2,\cdots,N_V. \tag{2.13}$$

Of course, the nature of the real-world scenario requires that the vehicle position be described in terms of world clock time, *t*, thus a relationship between ζ_i and *t* is a necessity. To accomplish this, a timing law, denoted $\theta(\cdot)$, is introduced to dictate how ζ_i evolves with time. For this purpose, the timing law for each vehicle is defined as

$$\theta_i(\cdot) = \frac{d\zeta_i}{dt} \tag{2.14}$$

where $\theta(\cdot)$ is a smooth, non-negative function of its arguments.

2.3.1. Bèzier Curves

Following from the desire to generate trajectories in a computationally efficient manner, the family of parametric curves known as Bèzier curves is chosen to represent the relationships used in this model. A Bèzier curve of degree *n* is defined by a set of *n*+1 control points and is expressed represented by

$$C(\zeta) = \sum_{k=0}^{n} \beta_k b_k^n(\zeta) \qquad 0 \le \zeta \le 1, \tag{2.15}$$

where the coefficients, β_i, are the control points and b_k^n are the basis Bernstein polynomials. The Bernstein polynomials are expressed as

$$b_k^n(\zeta) = \binom{n}{k}(1-\zeta)^{n-k}\zeta^k \, . \tag{2.16 a}$$

In this expression, $\binom{n}{k}$ is the binomial coefficient and its value is given by

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}. \tag{2.16 b}$$

2.3.1.1. de Casteljau Algorithm

Bèzier curves are derived from a computational process known as the de Casteljau algorithm, a procedure that can be thought of as a generalization of line interpolation.

Defining n+1 control points for a Bèzier curve of degree n, consecutive control points are connected by lines to form what is called the control polygon. An example of a third degree Bèzier curve and its control polygon is illustrated in Figure 2.3.

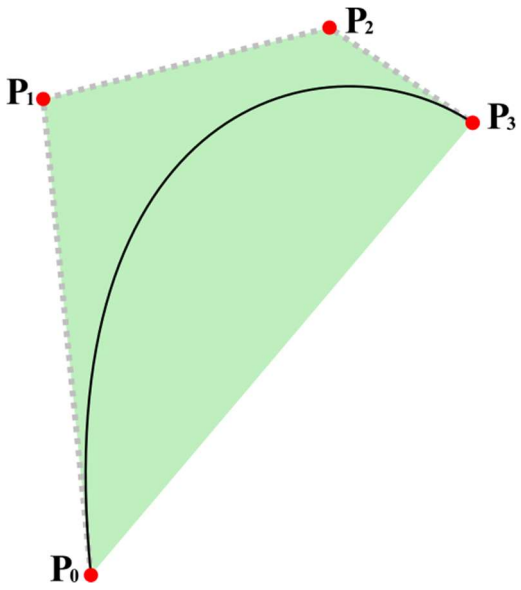


Figure 2.3: Third Degree Bèzier and Control Polygon.

To form the Bèzier curve from its control polygon, the parametric equation of a line is used to calculate points along the sides of the control polygon. Denoting the original control points as $P_{i,0}$, then the points along the side of the control polygon can be described by

$$P_{i,1} = P_{i,0}(1-t) + P_{i+1,0}t \tag{2.17}$$

where t is the parameterizing variable such that $t \in [0,1]$. These new points on the sides of the control polygon are connected consecutively with similar lines parameterized by t and this process is repeated with each new line being described by

$$P_{i,j} = P_{i,j-1}(1-t) + P_{i+1,j-1}t \tag{2.18}$$

$$t \in [0,1],\ j \in \{1,\cdots,n\},\ i \in \{1,\cdots,n-j\}$$

where j indicates the "level" of the parametric line and n is the degree of the Bèzier curve. This procedure begins with n lines forming the control polygon and continues until only one parametric line remains. The evolution of a point along this final line lies on the Bèzier curve. Figure 2.4 updates Figure 2.3 to include the new "levels" notation as well as the intermediate lines. In Figure 2.4, the Bèzier curve is described as $C(t)$.

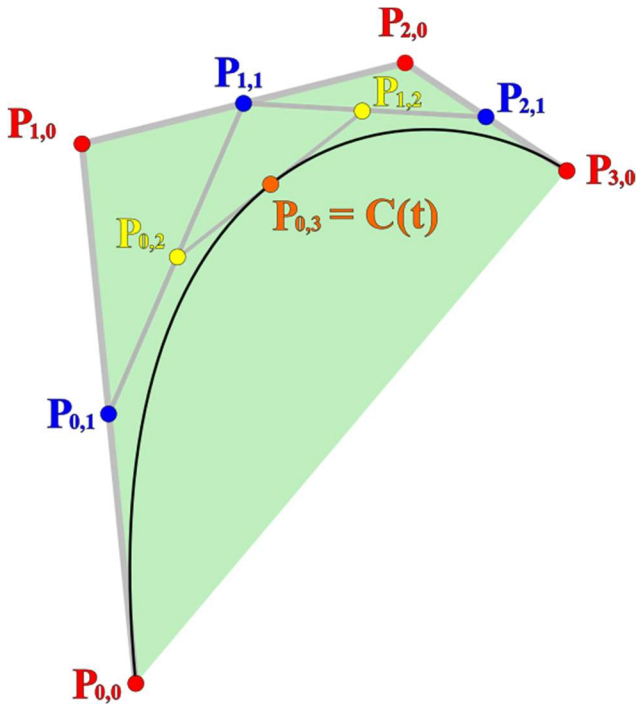


Figure 2.4: de Casteljau Algorithm.

The Bèzier curve begins at the first control point of the control polygon and terminates at the last. At both the start and end points, the Bèzier is tangent to the control polygon. Never encountering the intermediate control points, the Bèzier curve always lies within the convex hull of its control polygon. Along with always laying inside their convex hulls, Bèzier curves have the additional favorable characteristics of being closed under addition, subtraction, multiplication, differentiation, integration, and composition. With Bèzier curves being completely defined by their control points, many computational

procedures benefit from the ability to be computed strictly in terms of the constant control points. Such operations include the arithmetic operations and as will be seen in the following section, differentiation and integration.

de Casteljau algorithm also can subdivide an n^{th} degree Bèzier curve into two separate segments both of which are themselves n^{th} degree Bèzier curves defined on $[0,1]$. The union of these two segments results in the original curve. This procedure can be performed at any point, τ, on the Bèzier curve $C(t)$ where $\tau \in (0,1)$ and the new control points are found via Equation (2.18). As an example, if Figure 2.4 is envisioned to be illustrated at $\tau = t$, the original curve would be subdivided at the point $\mathbf{P}_{0,3}$. The first sub-segment would originate at $\mathbf{P}_{0,0}$, terminate at $\mathbf{P}_{0,3}$, and contain the intermediate points $\mathbf{P}_{0,1}$ and $\mathbf{P}_{0,2}$ according to Equation (2.18). Similarly, the second sub-segment would begin at $\mathbf{P}_{0,3}$, terminate at $\mathbf{P}_{3,0}$, and include $\mathbf{P}_{1,2}$ and $\mathbf{P}_{2,1}$.

2.3.1.2. Derivatives of Bèzier Curves

With the spatial path, timing law, constraints, and cost function that will be used to generate trajectories all to be defined as Bèzier curves in future sections, the goal of this section is to provide insight into the computational efficiency of calculating the derivative of a Bèzier curve. Recalling that an n^{th} degree Bèzier curve is defined by n+1 control points, the derivative of a Bèzier curve as expressed in Equation 2.15 is

$$C'(\zeta) = \sum_{i=0}^{n-1}[n(\beta_{i+1} - \beta_i)]b_i^{n-1}(\zeta), \qquad 0 \le \zeta \le 1. \qquad (2.19\text{ a})$$

To help illustrate the derivation of higher order derivatives, finite difference is employed and the control points are defined to be the 0^{th} level differences denoted D_i^0 and such that $D_i^0 = \beta_i$. Equation (2.19 a) can then be written as

$$C'(\zeta) = \sum_{i=0}^{n-1}[n(D_{i+1}^0 - D_i^0)]b_i^{n-1}(\zeta), \qquad 0 \le \zeta \le 1. \qquad (2.19\text{ b})$$

The difference between two consecutive 0th level differences is then defined as a 1st level difference denoted D_i^1. This trend continues as necessary with each higher level being composed of the differences of the elements of the level below it. Such a trend can be illustrated using matrices as can be seen in the following.

$$\begin{bmatrix} D_1^1 \\ D_2^1 \\ D_3^1 \\ D_4^1 \\ D_5^1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} D_1^0 \\ D_2^0 \\ D_3^0 \\ D_4^0 \\ D_5^0 \\ D_6^0 \end{bmatrix}$$

Higher levels of finite differences can be found by multiplying the product by a finite difference matrix similar to the one illustrated, whose size has been adjusted for the one less element found in each higher level.

With the levels of finite difference defined and represented by D_i^k, where k represents the difference level, a generalized expression for the higher order derivatives of Bèzier curves is provided in Equation (2.20).

$$C^{(r)}(\zeta) = n(n-1)(n-2)\cdots(n-r+1)\sum_{i=0}^{n-r} \mathrm{D}_i^r b_i^{n-r}(\zeta) \qquad (2.20)$$

It follows that the derivative of a Bèzier curve is itself a Bèzier curve and when regarded as its own curve, the derivative is referred to as the hodograph of the original curve.

2.3.1.3. Pythagorean Hodograph (PH) Bèzier Curves

Often when evaluating a vehicle's path of travel, it is useful to calculate the arc length of the path to evaluate the distance travelled by the vehicle. For a general path denoted $\vec{r}(\zeta)$, the arc length, $s(\zeta)$, is expressed as

$$s(\zeta) = \int_0^{\zeta} \|\vec{r}'(\tau)\| d\tau = \int_0^{\zeta} \sqrt{\vec{x}'(\tau)^2 + \vec{y}'(\tau)^2 + \vec{z}'(\tau)^2}\, d\tau. \tag{2.21}$$

Due to the square root in Equation (2.21), it is often that $s(\zeta)$ cannot be found in closed-form. To help solve this issue, the spatial paths will be described as a special type of Bèzier curve known as Pythagorean Hodograph (PH) Bèzier curves.

A curve is a Pythagorean Hodograph curve should its hodograph, $\vec{r}'(\zeta)$, satisfy the following Pythagorean condition

$$x'^2(\zeta) + y'^2(\zeta) + z'^2(\zeta) = \sigma^2(\zeta) \tag{2.22}$$

for some polynomial $\sigma(\zeta)$. For a PH curve, the arc length equation expressed in (2.21) becomes

$$s(\zeta) = \int_0^{\zeta} \sigma(\tau) d\tau. \tag{2.23}$$

For an in-depth conversation on PH curves, the reader is referred to [114]. Theorem 21.1 in [114] states that spatial Pythagorean hodograph curve must be expressible in terms of other real polynomials such that

$$x'(\zeta) = u^2(\zeta) + v^2(\zeta) - p^2(\zeta) - q^2(\zeta) \tag{2.24 a}$$

$$y'(\zeta) = 2[u(\zeta)q(\zeta) + v(\zeta)p(\zeta)] \tag{2.24 b}$$

$$z'(\zeta) = 2[v(\zeta)q(\zeta) - u(\zeta)p(\zeta)] \tag{2.24 c}$$

with parametric speed

$$\sigma(\zeta) = u^2(\zeta) + v^2(\zeta) + p^2(\zeta) + q^2(\zeta). \tag{2.25}$$

Appendix C demonstrates that the expressions of Equations (2.24) and (2.25) allow spatial PH Bèzier curves to be expressed compactly using a quaternion representation.

2.3.2. Trajectory Generation Through Cost Minimization

With the basic mathematical entities employed in this thesis summarized, attention is now turned to the problem of finding trajectories for the UAVs to follow. In the simplest terms, the trajectory generation problem requires the connection of two points in space by a smooth curve that can be followed by a quadcopter with an adequate control system. These smooth curves will be designed with the goal of minimizing an appropriate such global cost function for the system denoted $J(\cdot)$.

Though the boundary conditions will take on a non-constant nature later in this thesis, for this section it will be assumed that the boundary conditions between which a smooth curve will be designed are known. As with many trajectory generation problems, we assume that the initial and final vectors for the vehicle's position and velocity are known. Noting that the trajectories will be described by Bèzier curves which only exist on the interval $[0,1]$, the following boundary conditions are imposed on the system

$$\vec{p}_i(\zeta_i = 0) = \vec{p}_i^I \qquad \vec{p}_i(\zeta_i = 1) = \vec{p}_i^F \qquad \text{(2.26 a,b)}$$

$$\vec{v}_i(t = 0) = \vec{v}_i^I \qquad \vec{v}_i(t = t_i^F) = \vec{v}_i^F. \qquad \text{(2.26 c,d)}$$

As will be seen in the development of the timing law, the beginning of the trajectory in terms of the spatial parameter, ζ_i, will correspond with $t = 0$ and the end of the trajectory at $\zeta_i = 1$ will correspond to the end of the trajectory traversal time and thus correlate to $t = t_i^F$.

As described earlier, that trajectory will come in the form of a spatial path and a separate timing law. The following method generates the spatial path as a quintic PH Bèzier curve and thus, has the form

$$\vec{p}_i(\zeta_i) = \sum_{k=0}^{5} \overrightarrow{\bar{p}_{\iota,k}}\, b_k^5(\zeta_i) \tag{2.27}$$

for each vehicle, as $i = 1,2,\cdots,N_V$. The vectors $\overrightarrow{\bar{p}_{\iota,k}}$ represent the control points which completely define the spatial curve. Quintic PH Bèzier curves have been studied extensively in [114–116]; the fifth order PH Bèzier is the most feasible selection as its higher-order counterparts can become unpredictable and more computationally intensive [112].

To begin to find the control points of the spatial path, $\vec{p}_i(\zeta_i)$, the implications of the boundary conditions of Equation (2.26) are first examined. The initial and final positions, evaluated at $\zeta_i = 0$ and $\zeta_i = 1$ imply that $\overrightarrow{\bar{p}_{\iota,0}} = \vec{p}_i^I$ and $\overrightarrow{\bar{p}_{\iota,5}} = \vec{p}_i^F$. While prescribed velocities may initially appear to be a temporal constraint, it is observed that the initial and final first parametric derivatives of $\vec{p}_i(\zeta_i)$ must be vectors with the same directions as $\vec{v}_i^I$ and $\vec{v}_i^F$ respectively. Thus, with $\vec{v}_i^I$ and $\vec{v}_i^F$ known, their directional cosines are calculated and expressed as α_x, α_y, and α_z. It follows that the first parametric derivatives of the spatial paths can be expressed as

$$\frac{d\vec{p}_i(0)}{d\zeta_i} = \left\|\vec{p}_i{}'(0)\right\| \cdot \begin{bmatrix} \alpha_x(0) \\ \alpha_y(0) \\ \alpha_z(0) \end{bmatrix}, \qquad \frac{d\vec{p}_i(1)}{d\zeta_i} = \left\|\vec{p}_i{}'(1)\right\| \cdot \begin{bmatrix} \alpha_x(1) \\ \alpha_y(1) \\ \alpha_z(1) \end{bmatrix}. \tag{2.28 a,b}$$

It is important to note that $\left\|\vec{p}_i{}'(0)\right\|$ and $\left\|\vec{p}_i{}'(1)\right\|$ are properties of the spatial curve and are not the magnitude of velocity. Should these values be known, a smooth curve satisfying these endpoints and derivatives could be found as a first-order Hermite interpolation problem.

As alluded to earlier, a benefit of spatial quintic PH Bèzier curves is their ability to be written compactly using quadratic quaternion Bèzier curves, $\mathring{A}_i(\zeta_i)$. These curves are equivalent to the Bèzier curves of Equation (2.15) except that the control points are quaternions $\overline{\mathring{A}}_{i,k} \in \mathbb{H}$ for $k = 0,1,2$. A quaternion can be expressed as a scalar value and a vector such that a general quaternion, Å, may be represented as $\mathring{A} = a + b\hat{\imath} + c\hat{\jmath} + d\hat{k}$ and that quaternion's conjugate, denoted $\mathring{A}^*$, is defined such that $\mathring{A}^* = a - b\hat{\imath} - c\hat{\jmath} - d\hat{k}$. As derived in [114] and explained in more detail in Appendix C, the following relationship exists between the control points of the spatial quintic PH Bèzier curve and the control points of its quaternion counterpart

$$\vec{\bar{p}}_{i,1} = \vec{\bar{p}}_{i,0} + \frac{1}{5}\overline{\mathring{A}_{i,0}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,0}} \tag{2.29 a}$$

$$\vec{\bar{p}}_{i,2} = \vec{\bar{p}}_{i,1} + \frac{1}{10}\left(\overline{\mathring{A}_{i,0}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,1}} + \overline{\mathring{A}_{i,1}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,0}}\right) \tag{2.29 b}$$

$$\vec{\bar{p}}_{i,3} = \vec{\bar{p}}_{i,2} + \frac{1}{30}\left(\overline{\mathring{A}_{i,0}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,2}} + 4\overline{\mathring{A}_{i,1}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,1}} + \overline{\mathring{A}_{i,2}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,0}}\right) \tag{2.29 c}$$

$$\vec{\bar{p}}_{i,4} = \vec{\bar{p}}_{i,3} + \frac{1}{10}\left(\overline{\mathring{A}_{i,1}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,2}} + \overline{\mathring{A}_{i,2}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,1}}\right) \tag{2.29 d}$$

$$\vec{\bar{p}}_{i,5} = \vec{\bar{p}}_{i,4} + \frac{1}{5}\overline{\mathring{A}_{i,2}}\,\hat{\imath}\,\overline{\mathring{A}^*_{i,2}} \tag{2.29 e}$$

where $\hat{\imath}$ is the unit quaternion $[0 \quad 1 \quad 0 \quad 0]$, and the control points $\overrightarrow{\bar{p}_{i,0}}$ and $\overrightarrow{\bar{p}_{i,5}}$ are defined by the boundary conditions. It follows that the four remaining control points can be written in terms of the three quaternions, $\overline{\mathring{A}_{i,0}}$, $\overline{\mathring{A}_{i,1}}$, and $\overline{\mathring{A}_{i,2}}$. Referring again to Appendix C for further detail, the three quaternions can be found in terms of $\|\vec{p}_i{}'(0)\|$, $\|\vec{p}_i{}'(1)\|$, and two angular parameters β_i^0 and β_i^2.These four parameters are the first four decision variables that will be solved for in the cost minimization process. Once they have been found according to the cost minimization, the spatial path is completely defined.

In addition to the spatial path which is determined by the four aforementioned decision variables, a timing law was introduced in Equation (2.14) to account for the temporal specifications of the trajectory. Continuing to benefit from the properties of Bèzier curves, the timing law is defined as a quadratic Bèzier curve of the form

$$\theta_i(\hat{t}_i) = \sum_{k=0}^{2} \bar{\theta}_{i,k} b_k^2(\hat{t}_i) \qquad i = 1,2,\cdots,N_V \tag{2.30}$$

where $\hat{t}_i$ represents the dimensionless time variable necessary to satisfy the Bèzier curve's nature of being defined on the interval $[0,1]$. This dimensionless time variable is defined as

$$\hat{t}_i = \frac{t}{t_i^F} \tag{2.31}$$

and it follows that

$$d\hat{t}_i = \frac{1}{t_i^F} dt. \tag{2.32}$$

The introduction of this dimensionless time variable necessitates an update in the definition of the timing law such that

$$\theta_i(\hat{t}_i) = \frac{d\zeta_i}{d\hat{t}_i}. \qquad \hat{t}_i \in [0,1] \tag{2.33}$$

Should the occasion arise where the change in the spatial parameter ζ_i with respect to clock time, *t*, be found, substitution of Equation (2.32) into (2.33) reveals the following relationship

$$\frac{d\zeta_i}{dt} = \frac{1}{t_i^F}\frac{d\zeta_i}{d\hat{t}_i} = \frac{1}{t_i^F}\theta_i(\hat{t}_i). \tag{2.34}$$

From the definition of the timing law as expressed in Equation (2.33) an expression relating the spatial parameter ζ_i to the dimensionless time, $\hat{t}_i$, can be found through integration as

$$\zeta_i(\hat{t}_i) = \int_0^{\hat{t}_i} \theta_i(\tau)\, d\tau. \tag{2.35}$$

Earlier it was mentioned that the beginning of the spatial path at $\zeta_i = 0$ would correspond to $t = 0$ and the end of the trajectory at $\zeta_i = 1$ would correspond to $t = t_i^F$. Since $\theta_i(\hat{t}_i)$ is a second degree Bèzier curve, it follows from the integration of Equation (2.34) that $\zeta_i(\hat{t}_i)$ is a third degree Bèzier curve and is thus defined by four control points as

$$\zeta_i(\hat{t}_i) = \sum_{k=0}^{3} \bar{\zeta}_{i,k} b_k^3(\hat{t}_i). \tag{2.36}$$

Enforcing that $\zeta_i = 0$ when $t = 0$ implies that the $\bar{\zeta}_{i,0} = 0$ and constraining ζ_i to have a value of one at $t = t_i^F$ implies $\bar{\zeta}_{i,3} = 1$. The two remaining control points can be found in terms of the timing law control points from the definition of the derivative of a Bèzier curve as expressed in Equation (2.19 a) to be

$$\bar{\zeta}_{i,1} = \frac{1}{3}\bar{\theta}_{i,0} + \bar{\zeta}_{i,0} = \frac{1}{3}\bar{\theta}_{i,0} \tag{2.37 a}$$

$$\bar{\zeta}_{i,2} = \frac{1}{3}(\bar{\theta}_{i,0} + \bar{\theta}_{i,1}). \tag{2.37 b}$$

It then follows that it is necessary to determine the control points of $\theta_i(\hat{t}_i)$ to define the timing law. Continuing to exploit the integrative relationship used to derive the expressions of Equation (2.37), it can be found that the control point $\bar{\theta}_{i,1}$ can be written in terms of $\bar{\theta}_{i,0}$ and $\bar{\theta}_{i,2}$. Employing Equation (2.19 a) to find an expression for $\bar{\theta}_{i,2}$, the following relationship is presented

$$\bar{\theta}_{i,2} = 3\left(\bar{\zeta}_{i,3} - \bar{\zeta}_{i,2}\right) = 3\left[1 - \left(\frac{1}{3}(\bar{\theta}_{i,0} + \bar{\theta}_{i,1})\right)\right], \tag{2.38 a}$$

rearranging reveals

$$\bar{\theta}_{i,1} = 3 - \bar{\theta}_{i,2} - \bar{\theta}_{i,0}. \tag{2.38 b}$$

To find the remaining control points, the implications of the boundary conditions of Equation (2.26) are employed. The directions of the initial and final velocity vectors were used to solve for the spatial path but the magnitude of velocity has yet to be enforced upon

the system. To prescribe these conditions, the following expression for the speed profile is derived

$$\|\vec{v}(\hat{t}_i)\| = \left\|\frac{d\vec{p}_i(\zeta_i)}{dt}\right\| = \left\|\frac{\partial\vec{p}_i(\zeta_i)}{\partial\zeta_i}\frac{d\zeta_i}{dt}\right\| = \|\vec{p}_i{}'(\zeta_i(\hat{t}_i))\|\frac{1}{t_i^F}\theta_i(\hat{t}_i), \qquad (2.39)$$

where $\zeta_i(\hat{t}_i)$ can be brought outside of the norm due to its positive nature. Prescribing the initial and final values of Equation (2.39) as implied by the boundary velocity vectors results in the following relationships

$$\|\vec{p}_i{}'(0)\|\frac{1}{t_i^F}\bar{\theta}_{i,0} = \|\vec{v}_i^I\| \qquad (2.40\text{ a})$$

$$\|\vec{p}_i{}'(1)\|\frac{1}{t_i^F}\bar{\theta}_{i,2} = \|\vec{v}_i^F\|. \qquad (2.40\text{ b})$$

Noting that $\|\vec{p}_i{}'(0)\|$ and $\|\vec{p}_i{}'(1)\|$ are the same decision variables needed to determine the spatial path, t_i^F is introduced as the fifth and final decision variable such that the timing law control points can be found in terms of boundary conditions and decision variables according to

$$\bar{\theta}_{i,0} = \frac{t_i^F\|\vec{v}_i^I\|}{\|\vec{p}_i{}'(0)\|} \qquad (2.41\text{ a})$$

$$\bar{\theta}_{i,2} = \frac{t_i^F\|\vec{v}_i^F\|}{\|\vec{p}_i{}'(1)\|}. \qquad (2.41\text{ b})$$

With the inclusion of t_i^F into the family of decision variables, a complete decision variable vector for each vehicle can be expressed as

$$\boldsymbol{\Xi}_i = \begin{bmatrix} \|\vec{p}_i{}'(0)\| \\ \|\vec{p}_i{}'(1)\| \\ \beta_i^0 \\ \beta_i^2 \\ t_i^F \end{bmatrix}. \qquad (2.42)$$

Combining the set of decision variables as expressed in Equation (2.42) with the boundary conditions of Equation (2.26) a smooth curve can be found as a first-order Hermite interpolation problem.

A set of decision variables are found for each vehicle through the minimization of a global cost minimization process. Since the boundary conditions are inherently satisfied from the way the problem has been setup, the goal becomes to generate trajectories by solving the following constrained optimization problem

$$\min_{\Xi_i \times \cdots \times \Xi_{N_V}} J(\cdot) \tag{2.43}$$

s.t. Dynamic Constraints

Deconfliction Constraints

Mission Specific Constraints.

The cost function and constraints of Equation (2.43) are explored in detail as part of Chapter 3.

Chapter 2 has detailed the trajectory generation method to be employed throughout the rest of this thesis. Given the boundary conditions of Equation (2.26) and solving for the decision variables of Equation (2.42) to minimize some desired cost function $J(\cdot)$, a smooth curve is found according to the first-order Hermite interpolation problem. In the next chapter, the constraints of Equation (2.43) are detailed to investigate the landing of quadcopter UAV swarms on a moving vessel.

Chapter 3

LANDING SCENARIO CONSTRAINTS

3. Landing Scenario Constraints

This chapter aims to summarize the methods employed to explore the UAV swarm landing problem using the model proposed in Chapter 2. The trajectory generation method proposed in Chapter 2 aims to find trajectories that minimize a global cost function by solving a constrained optimization problem. In this chapter, the constraints dictated by the nature of the landing problem are defined so that the trajectory generation method may find safe trajectories for multiple UAVs to land in the vehicle storage compartment of a moving seafaring vessel.

In the interest of computational efficiency, the constraints of the following section are encouraged to be in Bèzier form wherever possible so that the appropriate global maxima and minima of constrained parameters may be determined quickly and efficiently. This is achieved through a modification of the work presented in [117]. In this work, Chang et al. present a culling-based algorithm for computing the minimum distance between two Bèzier curves (or surfaces). Taking advantage of the famous Gilbert-Johnson-Keerthi (GJK) algorithm [118,119] to efficiently compute the distance between the convex hulls of two Bèzier curves, a MATLAB implementation of Chang et al.'s method was created for this thesis which can quickly ensure that constraints in the Bèzier form have not been violated.

This thesis employs the MATLAB R2017a mathematics software as developed by MathWorks, Inc. to model the trajectory generation framework. To solve the constrained

optimization problem, the MPBNGC 2.0 proximal bundle method optimization Fortran subroutine [120] was integrated into the MATLAB scripts. A bundle method solver is employed to address the nonsmooth nature of the proposed constrained optimization problem that will be revealed within this chapter. The MPBNGC 2.0 Fortran code was kindly by provided by Prof. Marko M. Mäkelä of the University of Turku, Finland. All necessary tools and software are installed on an Acer Aspire R5 laptop computer operating with a 64-bit Windows 10 operating system. The unit has an Intel Core i7 (7^{th} generation) CPU rated to perform at 2.70 GHz. 12 GB of RAM memory are installed on the machine and a user-defined portion of the 1 TB HDD can be reassigned to function as RAM memory through a virtual memory option. An NVIDIA GeForce 940 MX GPU processes all visual simulations and graphics.

3.1. Vehicle Constraints

Motivated to create the most realistic model possible, an existing quadrotor UAV is chosen to provide the parameters necessary to accurately reflect the dynamical limitations of the individual vehicles. Physical properties are adopted from existing data and it is assumed that the vehicle has an adequate control system that is capable of following any desired, dynamically feasible, trajectory.

3.1.1. The Ascending Technologies Hummingbird Quadcopter

Existing literature demonstrates an extensive use of the Ascending Technologies (AscTec) Hummingbird [121] research quadrotor UAV, pictured in Figure 3.1. In this work, the AscTec Hummingbird is adopted as the model upon which the simulated UAVs will be based. This choice benefits from the availability of the inertial and motor properties of the system being well documented in the literature as well as being an extremely flexible

and popular quadrotor for research efforts should this work be extended to real-world experimentation.

Figure 3.1: The AscTec Hummingbird Quadcopter [121].

The inertial and dimensional properties of the AscTec Hummingbird quadcopter are accepted as published in [122]. Values for the force and moment coefficients, k_F and k_M, featured in Equations (2.7) and (2.8), have been determined empirically in [76]. This referenced work also presents experimentally determined bounds on rotor speeds. All necessary vehicle properties have been tabulated in Table 3.1.

Vehicle Property	Value
Mass (m)	0.5 kg
Arm Length (L)	0.17 m
I_{xx}	0.00365 kg·m^2
I_{yy}	0.00368 kg·m^2
I_{zz}	0.00703 kg·m^2
Force Coefficient (k_F)	$6.11 \cdot 10^{-8}$ $\frac{\text{N}}{\text{rpm}^2}$
Moment Coefficient (k_M)	$1.50 \cdot 10^{-9}$ $\frac{\text{N} \cdot \text{m}}{\text{rpm}^2}$
Maximum Rotor Speed (ω_{max})	7800 rpm
Minimum Rotor Speed (ω_{min})	1200 rpm

Table 3.1: Properties of the AscTec Hummingbird.

3.1.2. Physical Limitations

To ensure that the designed trajectories can be flown by real-world quadcopters, the physical limitations of the vehicles are included in the trajectory generation process. The first and perhaps most obvious constraint is the maximum velocity at which the vehicle can move. Ascending Technologies lists the maximum velocity of the Hummingbird to be 15 $\frac{m}{s}$ [121]. However, the vehicles in this thesis shall be limited to 8 $\frac{m}{s}$ as Huang et al. [123] show that at speeds greater than this, aerodynamic effects contribute largely to the system dynamics and the dynamics begin to resemble that of a fixed-wing aircraft. Thus, the limit of 8 $\frac{m}{s}$ is enforced to ensure the accuracy at which the trajectory can be followed as well as the integrity of any dynamics-based metrics.

In Chapter 2, an expression for the speed profile of the trajectory was stated as a Bèzier curve in Equation (2.39) to impose the boundary conditions on the system. That expression is employed once more to enforce the previously described velocity constraint as

$$0 \leq \left\|\vec{p}_i{}'\left(\mu_i(\hat{t}_i)\right)\right\| \frac{1}{t_i^F} \theta_i(\hat{t}_i) \leq 8\frac{m}{s} \qquad \forall \hat{t}_i \in [0,1]. \tag{3.1}$$

The upper bound of Equation (3.1) follows from the discussion above but it may seem redundant to constrain a velocity vector's magnitude to be greater than zero. First, it should be stated that a quadcopter can achieve speeds of $0\,\frac{m}{s}$ as they do not suffer from the stalling phenomenon common in fixed-wing aircraft. Second, requiring the vehicle's speed to be positive removes the need for an additional constraint forcing the timing law to be a non-negative function. Thus, enforcing the speed profile to be positive eliminates the need to find another expression to constrain $\theta_i(\hat{t}_i)$.

To enforce constraints such as the one stated in Equation (3.1) for all $\hat{t}_i \in [0,1]$ would require an infinite number of constraints corresponding to the infinite number of values on $[0,1]$ for which the constraints must hold. This implies that the constrained optimization problem belongs to a class of problems known as Semi-Infinite Programming (SIP) problems. However, the method described earlier to calculate the global maxima and minima of Bèzier polynomials allows the constraint of Equation (3.1), and all future constraints, to be expressed as

$$0 \leq \min\left\|\vec{p}_i{}'\left(\mu_i(\hat{t}_i)\right)\right\| \frac{1}{t_i^F} \theta_i(\hat{t}_i) \tag{3.2 a}$$

$$\max\left\|\vec{p}_i{}'\left(\mu_i(\hat{t}_i)\right)\right\| \frac{1}{t_i^F} \theta_i(\hat{t}_i) \leq 8\frac{m}{s}. \tag{3.2 b}$$

This allows the SIP problem to be expressed as a *finite-dimensional* nonsmooth optimization problem. The nonsmoothness is a result of the minimum and maximum functions being nondifferentiable and motivates the choice of a bundle method solver.

Section 2.1.2 introduced the dynamic model of the quadcopter which provides the equations of motion which govern the movement of the individual vehicles. Recalling the linear dynamics of Equation (2.5),

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = {}^{G}_{B}\boldsymbol{R}\begin{bmatrix}0\\0\\T\end{bmatrix} + \begin{bmatrix}0\\0\\-mg\end{bmatrix}$$

it can be seen that with aerodynamic effects neglected, an individual vehicle's linear accelerations are dictated by gravitational forces and the net thrust created by all four rotors. Equation (2.7) illustrates that the thrust created by each rotor is proportional to the square of the rotor's speed. Physical limitations on the rotor motors implies a limit on rotor speed and thus, a limitation on thrust. Assuming the quadcopter is driven by identical rotors and motors, the maximum thrust that can be produced is

$$T_{max} = 4\, k_F\, \omega_{max}^2. \tag{3.3}$$

Employing the parameters of the Hummingbird quadcopter, the vehicles employed in this work are capable of creating a maximum thrust of about 14 N. Dividing the linear dynamic equations of (2.5) by the Hummingbird's mass and substituting the maximum value for thrust in, the maximum linear accelerations of the vehicle are given by

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = {}^{G}_{B}\boldsymbol{R}\begin{bmatrix}0\\0\\28\frac{m}{s^2}\end{bmatrix} - \begin{bmatrix}0\\0\\9.8\frac{m}{s^2}\end{bmatrix}. \tag{3.4}$$

As the rotation matrix ${}^{G}_{B}\boldsymbol{R}$ distributes the acceleration due to thrust among the three Euclidean axes depending on the vehicle's attitude, a conservative approach is taken in this work to ensure feasibility and the magnitude of the vehicle's acceleration is prescribed an upper bound of $20\,\frac{m}{s^2}$. To impose this bound, the second time derivative of $\vec{p}_{d,i}(\mu_i)$ is taken

in a manner similar to that which revealed the expression for the speed profile earlier. The acceleration profile is then stated as

$$\left\|\frac{d^2\vec{p}_i(\mu_i)}{dt^2}\right\| = \frac{1}{(t^F)^2}\left\|\vec{p}_i{}''\left(\mu_i(\hat{t}_i)\right)\theta^2(\hat{t}_i) + \vec{p}_i{}'\left(\mu_i(\hat{t}_i)\right)\dot{\theta}(\hat{t}_i)\right\| = \|\vec{a}(\hat{t}_i)\|. \tag{3.5}$$

However, due to the square root implied with the norm function, Equation (3.5) is not a Bèzier polynomial. Desiring to impose the constraints in Bèzier form for computational efficiency, the acceleration constraint is imposed on the square of the acceleration profile and thus, has a maximum value of $20^2\,\frac{m^2}{s^4}$ and the constraint becomes

$$\|\vec{a}(\hat{t}_i)\|^2 = \frac{1}{(t^F)^4}\left\|\vec{p}_{d,i}{}''\left(\mu_i(\hat{t}_i)\right)\theta^2(\hat{t}_i) + \vec{p}_{d,i}{}'\left(\mu_i(\hat{t}_i)\right)\dot{\theta}(\hat{t}_i)\right\|^2 \leq 400\,\frac{m^2}{s^4} \tag{3.6}$$

3.2. Mobile Landing Vessel

The introduction of a mobile vessel on which the individual UAVs will land interferes with the earlier description of the trajectory generator. Chapter 2 formulated trajectories based on constant boundary conditions but landing on a moving vessel implies a dependence of the final boundary conditions on time and the motion of the vessel. For the work presented in this thesis, it is assumed that the trajectory of the landing vessel is known. Actual tracking of a moving vehicle by the swarm involves locating both the UAVs and the vessel in open water, sampling times to update the positions of the vehicles, and an advanced control system to steer the UAVs to the vessel; all of which are difficult research problems in themselves and are outside the scope of this thesis. A ship such as an aircraft carrier that does not make any erratic, unpredictable movements, is thus considered in this work. It is assumed that the seafaring vessel is relatively slow-moving with respect to the speed capabilities of the UAV to ensure that the UAV has the means to catch the vessel.

With the assumption that the simple motion of the ship is described by a trajectory known to the individual UAVs, the description of the ships motion begins with an initial position and a constant velocity such that

$$\vec{S}(t) = \vec{S}_0 + \vec{V}_S * t \tag{3.7}$$

where $\vec{S}(t)$ is the ship's position as a function of clock time, $\vec{S}_0$ is the ship's original position, and $\vec{V}_S$ is its constant velocity. Equation (3.7) is simple in nature but lacks the Bèzier form that is desirable to keep the computations efficient. For a UAV to successfully land on the vessel, it must have no relative motion with respect to the ship at the end of its trajectory. Thus, for each vehicle the final boundary conditions are updated such that

$$\vec{p}_i(\mu_i = 1) = \vec{S}(t_i^F) \qquad\qquad \vec{v}_i(t = t_i^F) = \vec{V}_S. \tag{3.8 a, b}$$

Though unnecessary to impose the final boundary conditions on the vehicles, the dependence of future constraints on the ship's trajectory motivates the need to express the ship's motion in Bèzier form. As will be seen in future sections, constraints will require a comparison between an individual UAV's position and the position of the ship at some given clock time, t. To compare the values of the UAV's position with the ship's position, the equations describing the two positions must evolve with clock time identically. With the UAV's position already expressed as a Bèzier curve, the comparison between the two positions is made by describing the ship's position in the following form for that particular constraint:

$$\vec{S}(\hat{t}_i) = \vec{S}_0 + \left(\vec{V}_S \cdot t_i^F\right) \cdot \left(\frac{t}{t_i^F}\right) = \vec{S}_0 + \left(\vec{V}_S \cdot t_i^F\right) \cdot \hat{t}_i. \tag{3.9}$$

To be expressed as a Bèzier curve, Equation (3.9) is rewritten as

$$\vec{S}(\hat{t}_i) = \sum_{k=0}^{1} \vec{\tilde{S}}_{i,k}\, b_k^1(\hat{t}_i) \tag{3.10}$$

where

$$\vec{\bar{S}}_{i,0} = \vec{S}_0 \tag{3.11 a}$$

$$\vec{\bar{S}}_{i,1} = \vec{S}_0 + \vec{V}_S \cdot t_i^F . \tag{3.11 b}$$

Care should be taken when implementing Equation (3.10) as it is a UAV-specific expression for the ship's trajectory and thus, can only be employed to constrain that specific UAV's trajectory.

3.3. Compartment Constraints

It has been stated that the ship on which the team of UAVs will land features a compartment dedicated to housing all the individual vehicles. Within this compartment it is assumed there exist fixed docks where the vehicles must land to recharge and prepare for the next deployment. Though now the landing vessel is imaginary, the compartments are modeled as circular boreholes in the center of the ship's top deck which extend some prescribed depth into the hull. It is then necessary to constrain the trajectory generator such that the vehicle's do not collide with the walls of the compartment. In this section, the constraints necessary to avoid such collisions are first illustrated for the case of a compartment that has no motion then extended to the moving compartment case.

3.3.1. Fixed Compartment

To illustrate the mechanics of the novel compartment constraints that prevent collisions between the vehicles and the walls of the confined UAV storage compartment, a static compartment is first considered. For simplicity, it is assumed that the compartment is of cylindrical shape with a radius of x_R and the axis of the cylindrical compartment being coincident with the global Z-axis. The compartment opens at an altitude of ${}^G z_{SO} = H_S$, reflecting the notion of the ship's top deck being above sea level and the vehicle will land

in the center of the circular tube. An illustration of a such a compartment is provided in Figure 3.2.

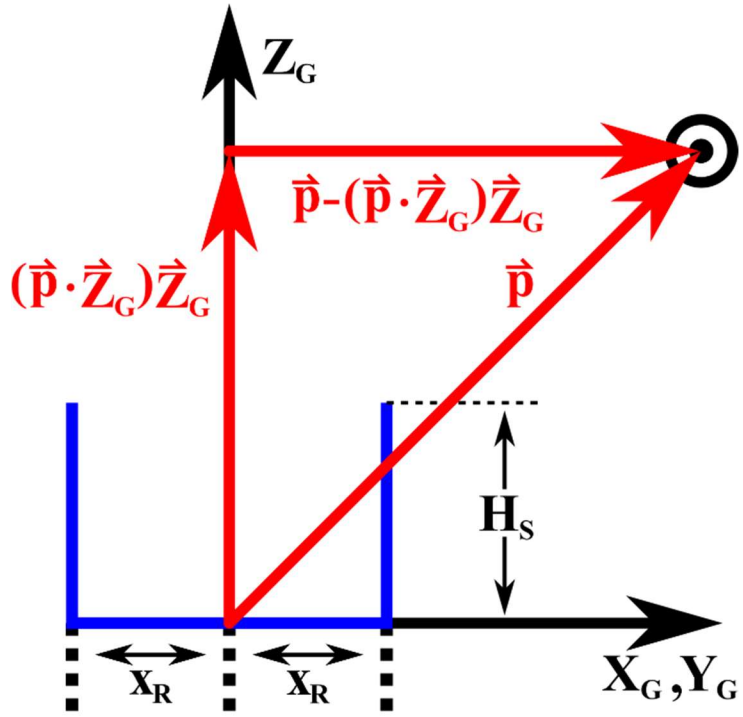


Figure 3.2: A Simplified Stationary UAV Storage Compartment.

Figure 3.2 identifies the location of the UAV by the vector $\vec{p}$ and illustrates the compartment as a blue corridor. Simple vector mathematics reveals that the vehicle's lateral distance from the cylindrical compartment's axis can be expressed as

$$\|\vec{p} - (\vec{p} \cdot \overrightarrow{z_G})\overrightarrow{z_G}\|. \tag{3.12}$$

It becomes necessary to ensure that the lateral distance described by Equation (3.12) is less than x_R when the vehicle has a global altitude less than H_S.

While there may be many ways to ensure that $\|\vec{p} - (\vec{p} \cdot \overrightarrow{z_G})\overrightarrow{z_G}\| < x_R$ for the appropriate vehicle attitudes, stress is placed on finding a constraint in Bèzier form to exploit the efficient methods available to enforce such constraints. To develop such a constraint, it is noted that the vector $\vec{p} - (\vec{p} \cdot \overrightarrow{z_G})\overrightarrow{z_G}$ can be expressed in Bèzier form as

Bèzier curves are closed under multiplication but the implied square root in the norm function of Equation (3.12) threatens the retention of the Bèzier form. Similar to the procedure performed on the acceleration constraint, both sides of the inequality are squared resulting in the following desired relationship for select altitudes:

$$\|\vec{p} - (\vec{p} \cdot \overrightarrow{z_G})\overrightarrow{z_G}\|^2 < \mathrm{x}_\mathrm{R}^2. \tag{3.13}$$

To enforce the altitude condition on when Equation (3.13) should be imposed on the system, the left side of Equation (3.13) is multiplied by an altitude-indicating coefficient of Bèzier form which also serves to make the constraint more conservative and thus, more likely to avoid collision. Introducing the aforementioned altitude-indicating coefficient, the final constraint imposed on the system is expressed as

$$\left(1 + \frac{H_S - (\vec{p} \cdot \overrightarrow{z_G})}{H_S}\right) \|\vec{p} - (\vec{p} \cdot \overrightarrow{z_G})\overrightarrow{z_G}\|^2 < \mathrm{x}_\mathrm{R}^2. \tag{3.14}$$

Realizing the quantity $(\vec{p} \cdot \overrightarrow{z_G})$ is the altitude of the vehicle, it shall be referred to as p_z here forth. Figure 3.3 improves upon the illustrations of Figure 3.2 to demonstrate the mechanics of the constraint expressed in Equation (3.14).

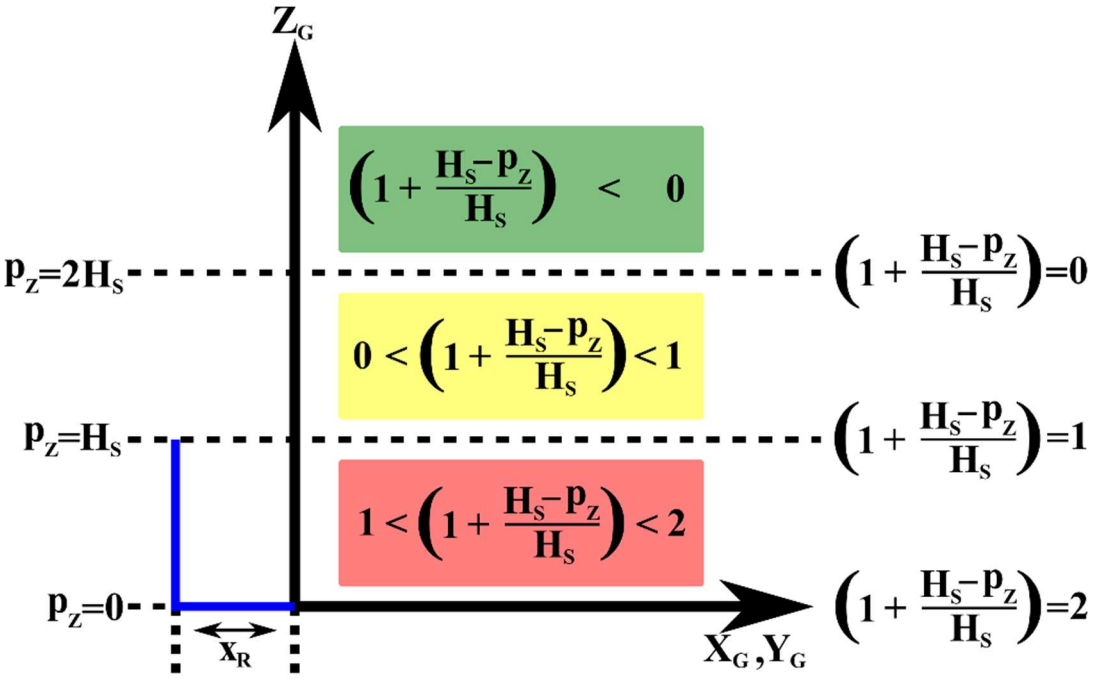


Figure 3.3: Altitude-Indicating Compartment Constraint Coefficient.

Examination of Figure 3.3 and the behavior of the altitude-indicating coefficient, reveals that for altitudes greater than twice the compartment depth the constraint of Equation (3.14) has no effect as a negative coefficient multiplied by a positive quantity will always be negative and therefore, always less than the compartment radius squared. Once the vehicle has an altitude less than twice the compartment depth, the constraint begins to influence the lateral distance of the vehicle from the cylinder axis. The constraint is considered to be conservative in the fact that it begins to pull the UAV towards the compartment's axis before the vehicle has entered and also requires the lateral distance to be much less than the cylinder radius as the vehicle proceeds further into the compartment.

3.3.2. Moving Compartment

Section 3.3.1 presented a simplified case of a stationary compartment to exhibit the mathematical mechanics behind the constraints. This simplified model is now expanded upon to ensure that collisions do not occur as the global positions of both the vehicle and

the compartment walls change with time. The differentiating factor between the stationary and moving case then comes from the motion of the center of the compartment entrance which had been fixed previously. It follows from the nature of the ship that the location of the entrance can be described by the ship location, Equation (3.10). Figure 3.4 illustrates the new vector relations where the compartment entrance is located by the ship's position, denoted $\vec{S}$.

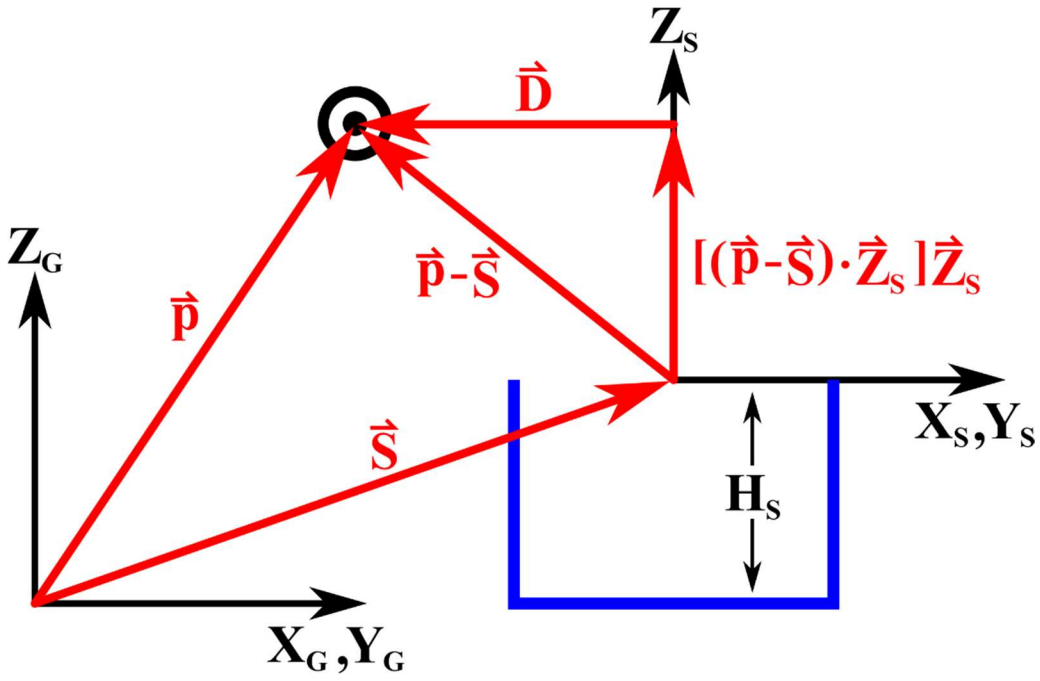


Figure 3.4: Moving Landing Compartment.

Figure 3.4 employs the Bèzier form of the ship's trajectory as expressed in Equation (3.10). The figure also introduces a new vector, denoted $\vec{D}$, which represents the UAV's lateral distance from the axis of the cylindrical apartment. According to the figure, this vector takes on the value

$$\vec{D} = (\vec{p} - \vec{S}) - [(\vec{p} - \vec{S}) \cdot \vec{z_s}]. \tag{3.15}$$

It follows that the final compartment constraint for a moving landing volume may be expressed as

$$\left(1+\frac{H_S-(\vec{p}\cdot\overrightarrow{z_G})}{H_S}\right)\left\|\vec{D}\right\|^2 < \mathrm{x}_\mathrm{R}^2 \tag{3.16}$$

where $\vec{D}$ is defined according to Equation (3.15).

3.4. Collision Avoidance (Deconfliction)

Deconfliction between two or more vehicles can be achieved through two mechanisms: spatial deconfliction which ensures the two trajectories never come within some distance of one another and temporal deconfliction which allows the trajectories to overlap one another so long as not at the same moment in time. The tight, confined quarters of the landing compartment may require the spatial paths of the vehicle trajectories to overlap and so temporal deconfliction is employed to avoid collisions amongst the vehicles.

To ensure that collisions among vehicles are avoided, a spatial clearance, C_s, is defined as the minimum distance between the center of masses of any two UAVs at a specific moment in time. Since the constraint restricts that spatial distances between vehicles with respect to time, the spatial path $\vec{p}_i(\mu_i)$ must be re-parameterized in terms of $\hat{t}_i$. Since Bèzier curves are closed under composition, it can be shown that $\vec{p}_i(\hat{t}_i)$ can be expressed as a PH Bèzier curve of degree fifteen. Thus, $\vec{p}_i(\hat{t}_i)$ is written as

$$\vec{p}_i(\hat{t}_i) = \sum_{k=0}^{15} \vec{\tilde{p}}_{i,k} b_k^{15}(\hat{t}_i). \tag{3.17}$$

Although Equation (3.17) states that each of the trajectories can be expressed in terms of the vehicle's respective dimensionless time variable, temporal deconfliction necessitates the spatial separation of vehicles at any instant in global clock time. Thus, the issue arises that for vehicles i and j where $i, j = 1, \cdots, N_V$ and $i \neq j$, at a certain clock time t, $\hat{t}_i \neq \hat{t}_j$ since each dimensionless time is dependent on the individual vehicle's total

trajectory traversal time, t_i^F or t_j^F. For the following discussion, two trajectories, $\vec{p}_i(\hat{t}_i)$ and $\vec{p}_j(\hat{t}_j)$, are considered and it is assumed that $t_i^F > t_j^F$ without loss of generality.

When vehicle j completes its mission, the vehicle will be located in one of the landing compartment's landing docks and vehicle i will still be traversing its trajectory. Thus, for any clock time $t > t_j^F$, the trajectory $\vec{p}_i(\hat{t}_i)$ need only avoid the volume dedicated to the storage of vehicle j. To best study the landing behavior of multiple quadrotor UAVs, no constraint is imposed on the path $\vec{p}_i(\hat{t}_i)$ to avoid the location $\vec{p}_j(1)$. Instead, such considerations are taken into account when discussing observations made of swarm landing behavior in Chapter 4 as such collisions may be dependent upon the arrangement of landing docks within the compartment.

Keeping the aforementioned notion in mind, the mission of vehicle j is considered to be completed when $t > t_j^F$. The deconfliction constraint can then be stated as

$$\left\|\vec{p}_i(\hat{t}_i) - \vec{p}_j(\hat{t}_j)\right\|^2 \geq C_S^2, \qquad \forall t \in \left[0, t_j^F\right]. \tag{3.18}$$

Since, as discussed earlier, $\hat{t}_i \neq \hat{t}_j$ at a given clock time t, measures must be taken to ensure temporal deconfliction according to clock time. To achieve this, the de Casteljau algorithm described in Section 2.3.1.1 is employed to subdivide the trajectory of vehicle with the larger trajectory traversal time, vehicle i in this case. The trajectory $\vec{p}_i(\hat{t}_i)$ is subdivided at the point

$\alpha = \frac{t_j^F}{t_i^F}$. Recalling that de Casteljau algorithm separates the original curve into two separate Bèzier curves both of which are defined on $[0,1]$, the first segment corresponds to $\vec{p}_i(\hat{t}_i)$ for $t \in \left[0, t_j^F\right]$. This first new Bèzier curve is denoted $\vec{p}_i^{\alpha}(\hat{t}_i^{\alpha})$ and is defined on $\hat{t}_i^{\alpha} \in [0,1]$. If $\hat{t}_i^{\alpha}$ is allowed to evolve at the same rate as t_j^F then, as it has been shown in [112],

$$\vec{p}_i(\hat{t}_i) = \vec{p}_i^{\alpha}(\hat{t}_j), \qquad \forall t \in [0, t_j^F]. \tag{3.19}$$

The constraint proposed in Equation (3.19) can then be rewritten as

$$\left\|\vec{p}_i^{\alpha}(\hat{t}_j) - \vec{p}_j(\hat{t}_j)\right\|^2 \geq C_S^2, \qquad \forall t_j \in [0,1]. \tag{3.20}$$

Expanding this conclusion to the general case of any of the N_V vehicles, the deconfliction constraint is stated as

$$\min_{\substack{i,j=1,\cdots N_V \\ i \neq j}} \left\|\sum_{k=0}^{15} \vec{\tilde{p}}_{ij,k} b_k^{15}(\hat{t}_{ij})\right\|^2 \geq C_S^2 \tag{3.21 a}$$

where

$$\hat{t}_{ij} = \frac{t}{t_{ij}^F} \tag{3.21 b}$$

$$t_{ij}^F = \min\{t_i^F, t_j^F\} \tag{3.21 c}$$

and

$$\vec{\tilde{p}}_{ij,k} = \begin{cases} \vec{\tilde{p}}_{i,k}^{\alpha} - \vec{\tilde{p}}_{j,k}, & \text{if } t_i^F > t_j^F \\ \vec{\tilde{p}}_{i,k} - \vec{\tilde{p}}_{j,k}^{\alpha}, & \text{if } t_j^F > t_i^F \end{cases}. \tag{3.21 d, e}$$

Chapter 4

CASE STUDY AND DISCUSSION

4. Case Study and Discussion

Determined to contribute to the problem of landing multiple quadrotor UAVs in a confined compartment on a moving vehicle, the effectiveness of the proposed model of Chapters 2 and 3 is presented in this chapter. Some experimental parameters and definitions necessary to illustrate the context of the case study are first presented. The investigation then verifies the capabilities of the model for different initial UAV-landing vessel configurations, ensuring that the constraints specific to the landing problem are not violated. Observations from the case study are then analyzed to provide insight into how the trajectory generation method may best contribute to the UAV swarm landing problem.

4.1. Experimental Parameters

Before beginning to study the effectiveness of the proposed trajectory generation method, a set of experimental parameters is first defined to clarify the physical properties of the system.

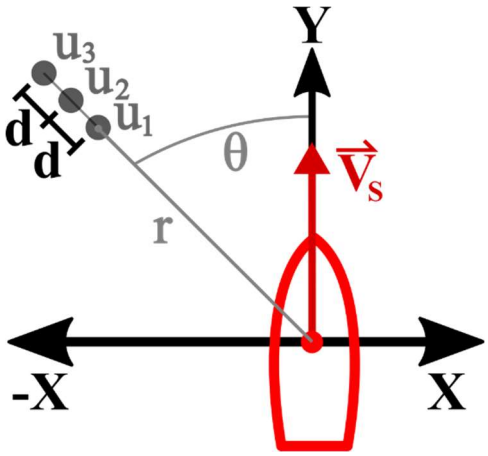
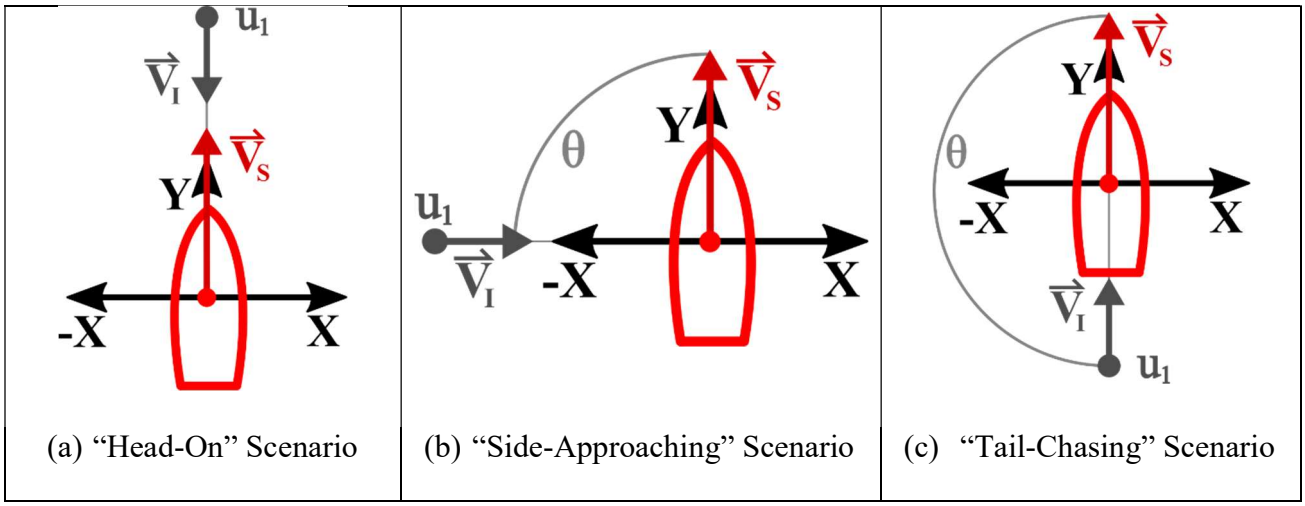


Figure 4.1: Generic Experimental Setup Schematic

Figure 4.1 presents a basic schematic of a general experimental setup. For the experiments described here within, the ship will originate at the origin of the $\overrightarrow{x_G}$-$\overrightarrow{y_G}$ plane and be designated a velocity vector, $\overrightarrow{V_S}$, consisting only of a constant, positive $\overrightarrow{y_G}$ component. It follows that the ship frame of reference, *S*, has an original position of $[0 \quad 0 \quad \mathrm{H_S}]^{\mathrm{T}}\, m$. UAVs are denoted $\mathrm{u_i}$ for $\mathrm{i} \in \{1, \cdots, N_V\}$, where $\mathrm{u_1}$ is considered the "leading UAV." The initial position of $\mathrm{u_1}$ is defined by a radius-angle pair, $[r, \theta]$ where $\theta \in [0, \pi]$ is measured counterclockwise from $\overrightarrow{y_G}$ and has some altitude, A, along $+\overrightarrow{z_G}$. Unless stated otherwise, subsequent vehicles, $\mathrm{u_i}$ for $\mathrm{i} \in \{2, \cdots, N_V\}$, are each defined by an identical displacement distance, d, from the vehicle $\mathrm{i} - 1$ in front of it along the line $[r, \theta]$. All UAVs initially share the same altitude value, A, and initial velocity vector, $\overrightarrow{V_\mathrm{I}}$.

Three main initial UAV-ship configurations will be referred to in this document. The "head-on" configuration corresponds to $\theta = 0$ and the velocities of the UAVs are in the opposite direction of that of the landing ship. A "side-approaching" scenario indicates $\theta = \frac{\pi}{2}$ and the initial UAV-ship velocities are perpendicular. Finally, when $\theta = \pi$, the configuration is referred to a "tail-chasing" approach. Figure 4.2 illustrates the basic setup of each initial configuration.



(a) "Head-On" Scenario (b) "Side-Approaching" Scenario (c) "Tail-Chasing" Scenario

Figure 4.2: Common Experiment Initial Configurations

During a real-world scenario, the landing procedure would be considered a failure should any UAVs collide with another aerial vehicle or with the walls of the compartment. Thus, the importance of vehicles not colliding with compartment walls or one another will be stressed to avoid catastrophic incidents. For this reason, it is important to describe the parameters that will be used to enforce such constraints. As stated in Table 3.1, the AscTec Hummingbird has arms that are $0.17m$ long. Considering the rotor radius of $0.10m$ in addition to the arm length, the vehicle has a total diameter of $0.54m$. It follows that to avoid intervehicle collisions, the center of masses of any two vehicles must be separated by a distance of at least $0.54m$. Inclusion of a safety factor, chosen to have a value of about 1.4, promotes the enforcement of a spatial clearance of $0.75m$ between the center of masses of any two UAVs at any point in time. This value for the required spatial clearance between vehicles shall remain constant for the remainder of the investigation.

With a total diameter of $0.54m$, the dimensions of the quadcopter also dictate the allowable dimensions of the vehicle storage volume. A storage volume shall be defined in terms of the quadcopter dimensions to reflect a sense of how confined the compartment is.

With a height of $0.0855m$, it is required that quadcopters landing at the deepest depths of the compartment land at an altitude of ${}^{G}z_{SO} = 0.04275m$. Vehicles landing nearest the top deck land at an altitude of ${}^{G}z_{SO} = (H_S - 0.04275)m$ to ensure the quadcopter is completely submerged below the ship's deck. Coupling these notions with the mandatory spatial clearance of $0.75m$, the height of a compartment designed to house N_V vehicles, is defined as

$$H_S = 0.0855m + (\mathrm{N_V} - 1) \cdot 0.75m. \tag{4.1}$$

A description of the compartment's radius, R_C, is provided by Equation (4.2) where the vehicle's diameter is scaled by a parameter k_R such that

$$R_C = k_R \cdot \frac{1}{2}(0.54m). \tag{4.2}$$

4.2. Case Study

To demonstrate the ability of the methods proposed in this work to generate trajectories which successfully land the individuals of a UAV swarm in a nonstationary confined compartment, a case study is presented. As it is unlikely that the UAV swarm will ever be approaching its destination vehicle completely "head-on" or completely from behind, a "side-approaching" experimental setup is chosen to illustrate the effectiveness of the method.

By the nature of the "side-approaching" setup, the initial angle is defined as $\theta = \frac{\pi}{2}$. The radius belonging to the radius-angle pair, $[r, \theta]$, is set at $20m$, the same value is taken for the initial altitude as well. With the system containing three UAVs total, each initially with $1m$ of separation from one another, the constrained optimization problem is formulated such that the amount of time it takes for all three UAVs to land is minimized. In accordance with Equations (4.1) and (4.2), the cylindrical compartment is defined such

that $H_S = 1.5855m$ and $R_C = 1m$. All the parameters necessary to describe the experimental setup employed in the case study have been tabulated and are presented in Table 4.1.

θ [rad]	r [m]	A [m]	Hs [m]	Rc [m]	$\overrightarrow{V_I}$ [m/s]	$\overrightarrow{V_S}$ [m/s]	UAVs [#]	d [m]
π/2	20	20	1.5855	1	[-4; 0; 0]	[0; 3; 0]	3	1

Table 4.1: Case Study Experimental Parameters.

All trajectories were generated by attempting to minimize the amount of time needed for the UAV to traverse its trajectory. The resulting trajectories generated for the experimental parameters established are plotted separately in Figure 4.3, separate plots are used only for visual clarity.

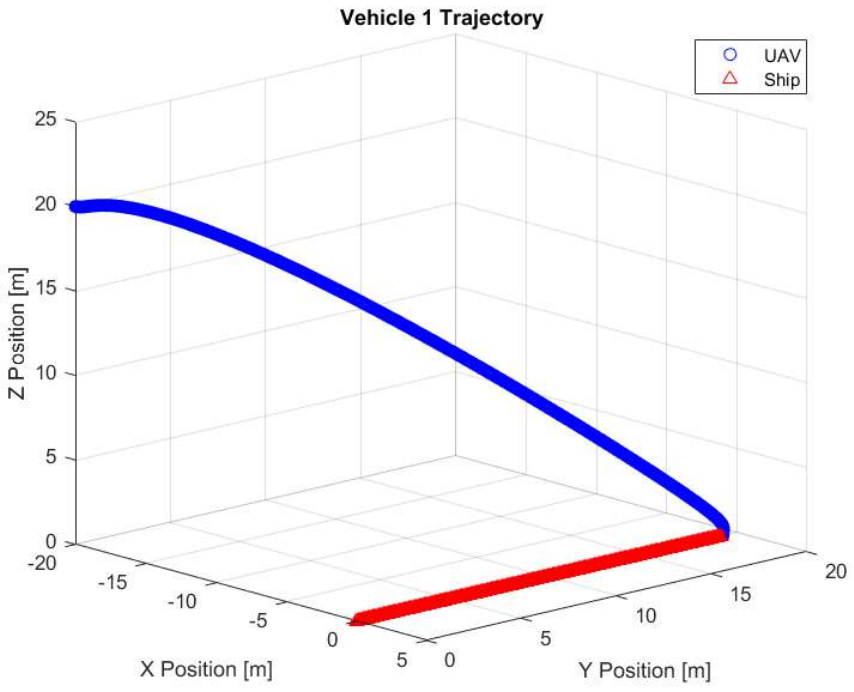


Figure 4.3 (a): Vehicle 1 Trajectory (Traversal Time of 6.39s).

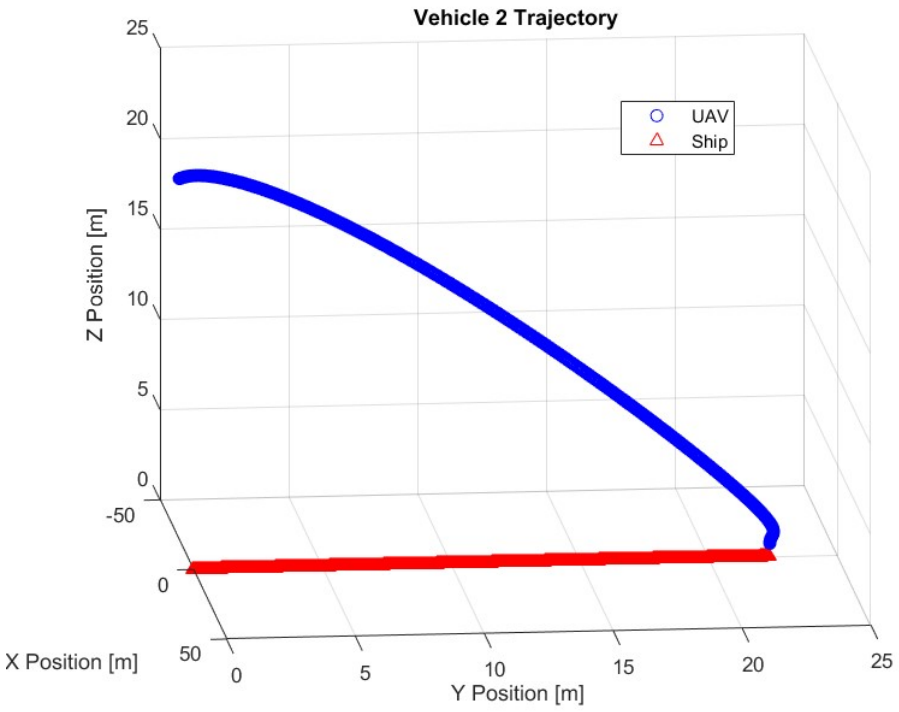


Figure 4.3 (b): Vehicle 2 Trajectory (Traversal Time of 7.19s).

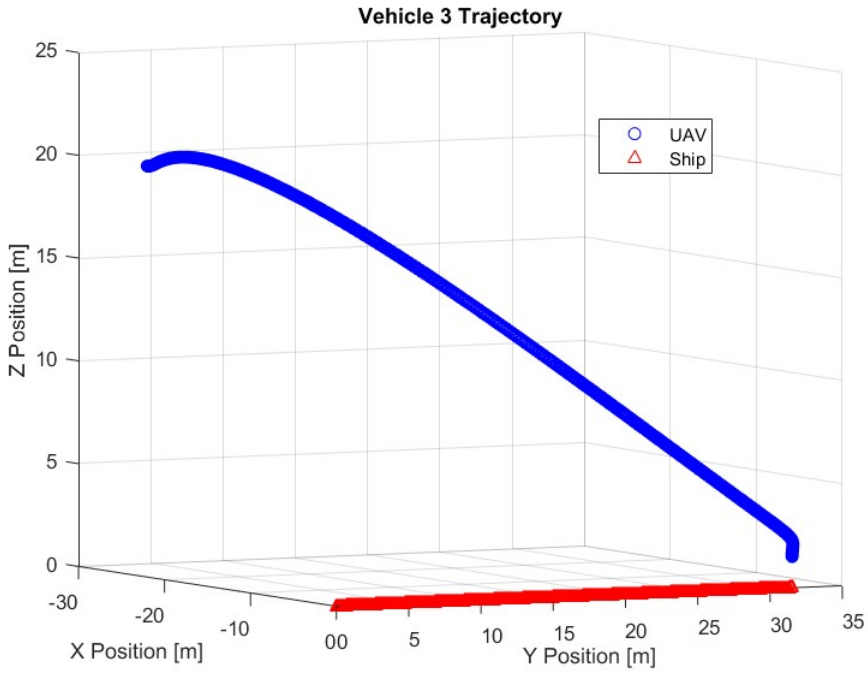


Figure 4.3 (c): Vehicle 1 Trajectory (Traversal Time of 10.51s).

All vehicles were shown to successfully intercept the moving vehicle at an altitude corresponding to a particular landing dock within the confined compartment. Overall, the trajectories took the general form of the expected shape with some minor inconsistencies, likely due to the suboptimal nature of the optimization problem solution as a time minimizing process would generate a straighter spatial path. The final landing maneuver created by the trajectory generator has the UAV following a nearly vertical path to stay within the confines of the UAV storage compartment.

Although the shape of the trajectories were not always completely smooth and the final landing maneuver is observed to be of a very rigid nature, it can be seen that the physical limitations of the vehicle are never violated. Figure 4.4 illustrates the velocities and the acceleration constraint values of the vehicles throughout their flights.

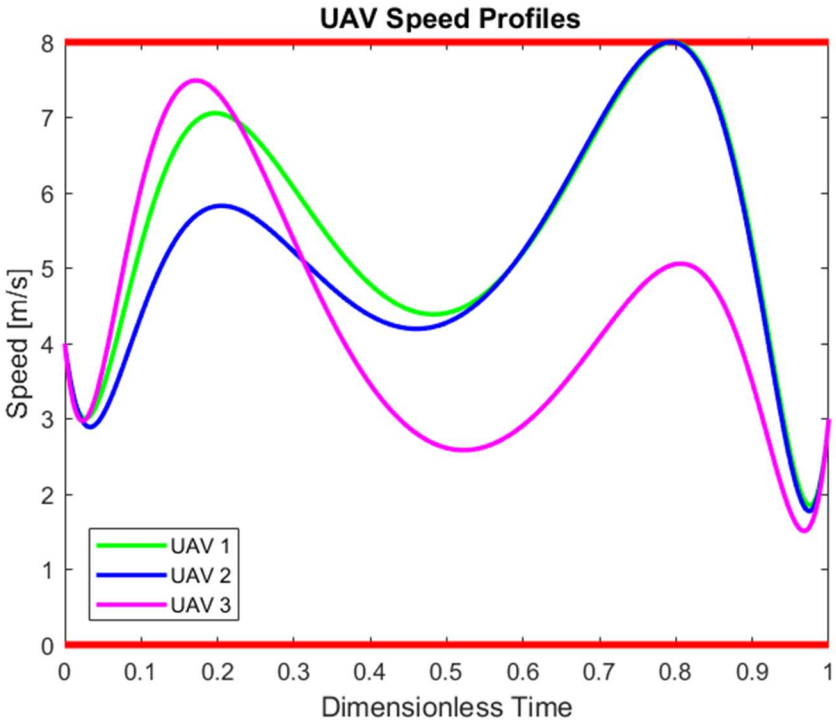


Figure 4.4 (a): UAV Speed Profiles

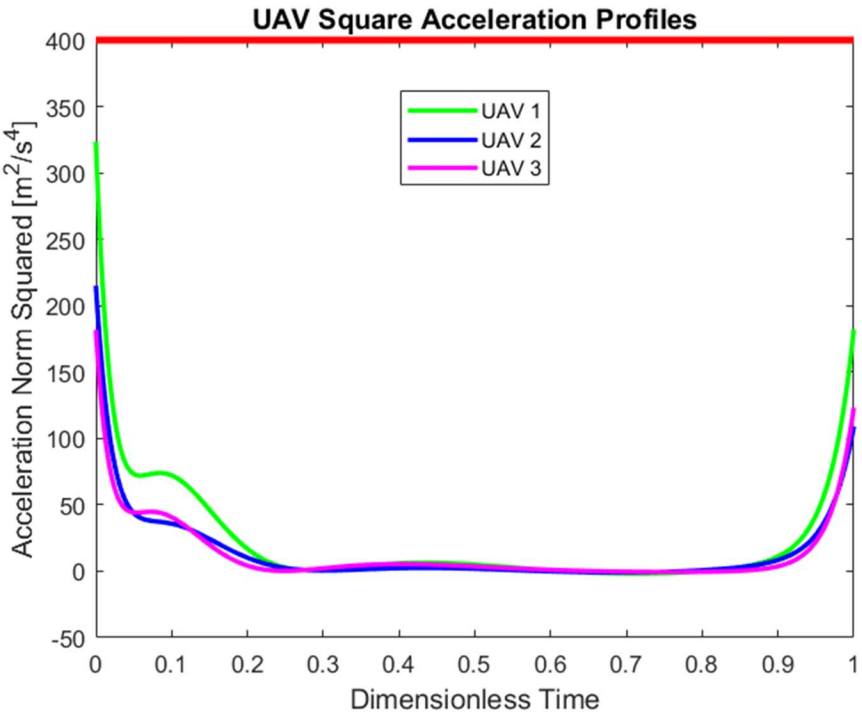


Figure 4.4 (b): UAV Acceleration Squared Profiles.

From Figure 4.4 it can be seen that the trajectories generated for the UAVs to follow do not require the vehicles to perform any maneuvers outside of the vehicle's physical limitations. Observations regarding the unpredictable trajectory shape leak into the observed dynamics as the vehicles are observed to speed up and slow down to accommodate these maneuvers. Such behavior results in the peaks and valleys in the speed profiles and corresponding acceleration values of zero.

The two main sets of constraints that were applied to the system were dynamic constraints and landing scenario specific constraints which avoided collisions amongst the vehicles and between the vehicles and the compartment walls. Continuing on the success of the dynamic constraints, no collisions were observed amongst the vehicles or between the vehicles and the compartment walls.

4.3. Additional Observations

The main difficulty in the trajectory generation process was the determination of an initial guess to supply the bundle method optimization solver. Trajectories were observed to be extremely volatile and reactive to small changes in the supplied initial starting point. The nature of the decision variables set forth by the problem make these good initial guesses difficult to determine. A better methodology for formulating the initial guess that is entered into the optimization algorithm could not only address the volatility of the trajectories generated but the sub-optimality which leads to control issues over the trajectories formed.

Chapter 5

CONCLUSION

5. Conclusion

5.1. Concluding Remarks

Motivated by the applicability of cooperative unmanned aerial vehicle teams to an innumerable amount of tasks and services in both the civilian and military domains, this thesis has proposed a method to safely recover such a team following its respective mission. Particularly focused on military applications, this thesis devoted its attention to the landing of a UAV swarm on a moving, seafaring vessel with a confined compartment dedicated to the housing of the individual team members.

A computationally efficient trajectory generation method was first proposed which planned vehicle motion by considering the vehicle's spatial path and the temporal considerations of the flight separately. Strictly geometric spatial paths were modelled using Pythagorean Hodograph Bèzier curves. The timing law which took the temporal specifications of the mission into account was modelled using a quadratic Bèzier polynomial. Trajectories were generated by solving a constrained optimization problem where the constraints ensured that the vehicle dynamics were not violated and that collisions did not occur between vehicles or between vehicles and the landing compartment. By formulating the constraints in Bèzier form, the global extrema of the constraint functions could be efficiently calculated and thus, the constraints could be imposed on the system in a computationally friendly manner.

To illustrate the efficacy of the proposed trajectory generation method, Chapter 4 detailed a case study in which the new method was employed to plan the landing trajectories of multiple UAVs attempting to land in a moving compartment. For the landing scenario detailed, the trajectory generator was shown to be successful in guiding the individual UAVs to their landing location in a moving confined compartment. The method successfully generated plans of motion which respected the vehicle's physical limitations and ensured that no collisions occurred.

Due to the nonsmooth and nonconvex nature of the optimization problem that had to be solved to generate the trajectories, the trajectories that were generated were suboptimal. This led to a lack of control over the paths generated even though the constraints were guaranteed to be satisfied. As with most nonlinear optimization programs, the bundle method employed in this work required a user-defined initial guess of the solution. Due to the sensitivity of the problem to the initial guess, resulting trajectories were volatile with respect to this input making the trajectory generation process difficult for a user, let alone an automated process in the future.

5.2. Future Work

Chapter 3 detailed the use of a simple motion model for the landing ship and cited several additional areas of current research efforts whose developments were deemed necessary to form a viable solution to the real-world UAV swarm landing problem. Moving target interception methods have been researched extensively but the concepts must be integrated with UAV control systems and finally with the proposed trajectory generation method in order to be effective.

With the trajectories generated by the proposed method being found as a solution to a constrained optimization problem, an optimization scheme custom-tailored to this specific problem would improve the results. The optimization problem proposed is nonsmooth and nonconvex which makes the problem difficult to solve. It would be ideal for such an optimization method to be decentralized and distributed over the individuals of the swarm. Decentralized approaches provide robustness to the system should an individual UAV, or any other centralized computing unit, be compromised or destroyed. A distributed method would then have to grapple with the fact that the lightweight quadrotor UAVs have generally weak computing power. One additional concern with regards to the optimization problem is the necessity of an initial guess in nonlinear solvers.

Should a suitable optimization method be available, the method proposed in this thesis has the potential to provide great insight into the swarm landing problem. The ability to generate trajectories which minimize an objective function opens the door to understanding the behaviors of landing swarms as they attempt to minimize different criteria. A comparison of different minimization criteria in various landing scenarios could provide insight into the preferred landing schemes and approaches for UAV swarms landing in a nonstationary confined compartment.

Appendix A

EULER ROTATIONS

A. Euler Rotations

The model employed to describe the physical systems to be studied in this thesis consists of multiple refence frames which are related to one another through rotational and translational matrices. In this section, we illustrate the Z-X-Y Euler angles used to map the vehicle body frame, *B*, to the global frame, *G*.

We begin by defining an intermediate frame, I^1, which lays directly on top of the global frame, *G*, sharing an origin and each of the axes laying in the same direction as its respective counterpoint on the other frame as illustrated in Figure A.1 (a).

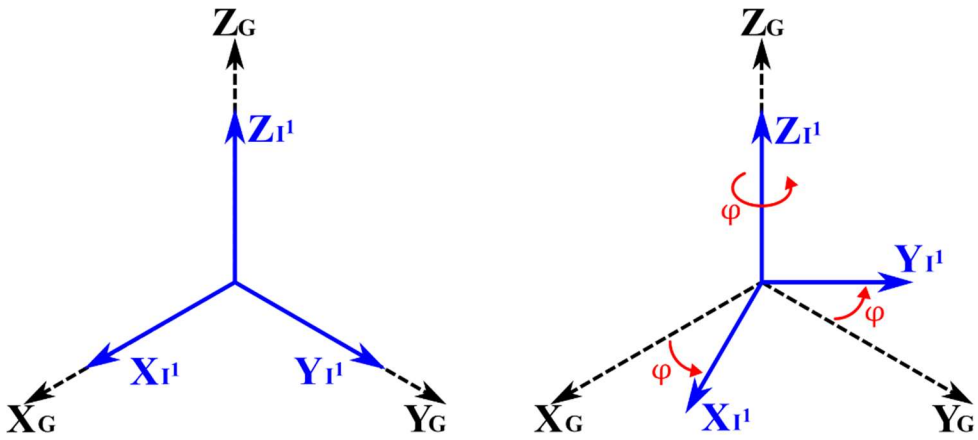


Figure A.1: (a) Initial Orientation of Frames G and I^1. (b) Yaw Rotation about $\overrightarrow{z_G}$.

Rotating the frame I^1 about the Z-axis of frame *G* by the angle φ, any vector expressed in frame I^1 can be translated into the global frame, *G*, by the rotational matrix

$$ {}^{G}_{I^1}\mathbf{R} = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.1} $$

Introducing a second intermediate frame, I^2, a similar procedure is performed only this time frame I^2 is rotated about the X-axis of frame I^1 by the roll angle ϕ.

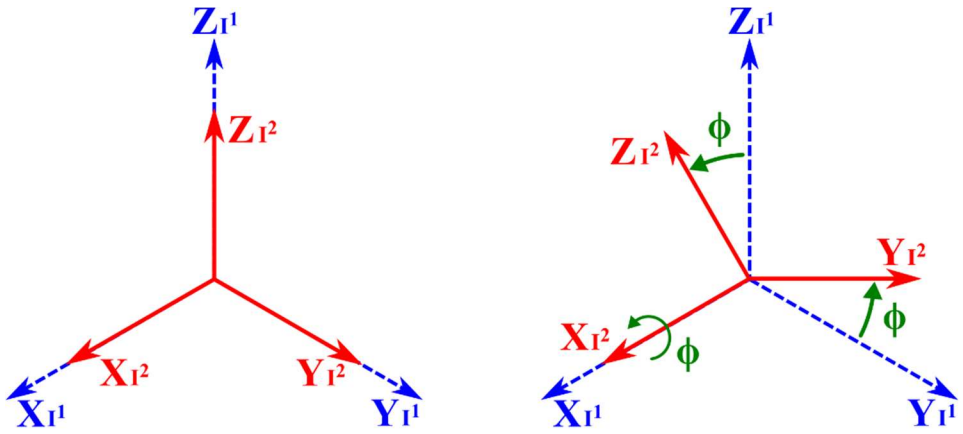


Figure A.2: (a) Initial Orientation of Frames I^1 and I^2. (b) Roll Rotation about $\overrightarrow{z_{I^1}}$.

Similar to the previously detailed rotation, the relationship between I^1 and I^2 can be expressed in the form of a rotation matrix:

$$ {}^{I^1}_{I^2}\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}. \tag{A.2} $$

Finally, the final body frame is defined and rotated about the Y-axis of I^2 by the pitch angle, θ, as illustrated in Figure A.3. The Corresponding relationships are expressed in Equation (A.3).

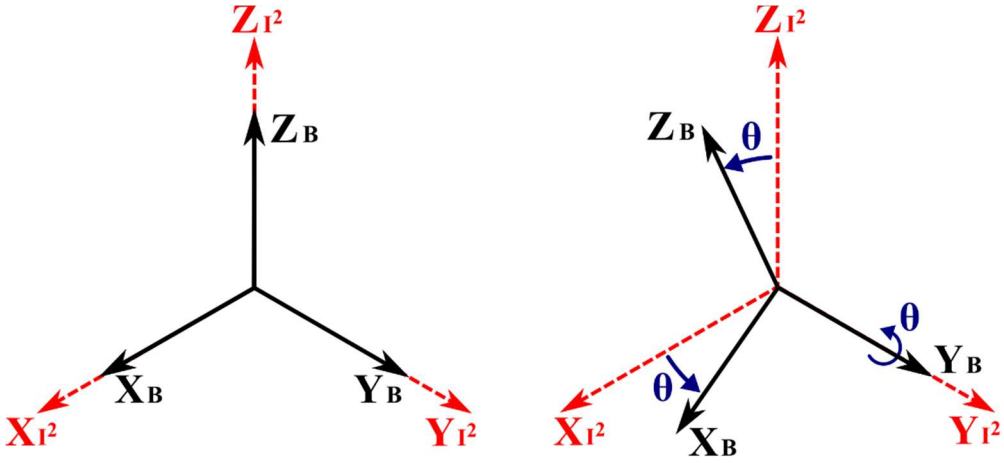


Figure A.3: (a) Initial Orientation of Frames I^2 and B. (b) Pitch Rotation about $\overrightarrow{y_{I^2}}$.

$$ {}^{I^2}_{B}\mathbf{R} = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}. \tag{A.3} $$

Each of the previously detailed rotational relationships can be used to find the final relationship between the global frame, *G*, and the body frame, *B*. Translating the expression of the body frame, *B*, in frame I^2 through the first intermediate frame and then on to the global frame, we obtain the reference frame relationship as expressed in Equation (2.1) and repeated below

$$ {}^{G}_{B}\boldsymbol{R} = {}^{G}_{I^1}\mathbf{R}\,{}^{I^1}_{I^2}\mathbf{R}\,{}^{I^2}_{B}\mathbf{R} \tag{A.4 a} $$

$$ {}^{G}_{B}\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}\begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{A.4 b} $$

$$ {}^{G}_{B}\boldsymbol{R} = \begin{bmatrix} c\varphi c\theta - s\varphi s\theta s\emptyset & -c\emptyset s\varphi & s\theta c\varphi - c\theta s\varphi s\emptyset \\ c\theta s\varphi + s\theta s\emptyset c\varphi & c\emptyset c\varphi & s\varphi s\emptyset - c\varphi c\theta s\emptyset \\ -c\emptyset s\theta & s\emptyset & c\theta c\emptyset \end{bmatrix}. \tag{A.4 c} $$

As mentioned in Chapter 2, cφ and sφ are abbreviations for the cosine and sine of the yaw angle, φ, and the same is true for the roll and pitch angles. It is worth noting that the rotation matrix from frame *B* to frame *G* can be written as a matrix of the body frame axes [124] in the global frame such that

$$ {}^{G}_{B}\boldsymbol{R} = [\overrightarrow{x_B} \quad \overrightarrow{y_B} \quad \overrightarrow{z_B}]. \tag{A.5} $$

Appendix B

DIFFERENTIAL FLATNESS

B. Differential Flatness

Chapter 2 makes the claim that the quadcopter's dynamics can be expressed as a differentially flat system. A set of flat outputs, $\vec{\rho} = [x \quad y \quad z \quad \varphi]^{\mathrm{T}}$, was defined such that the state of the quadrotor UAV and its control inputs, $\vec{u}$, could be expressed as algebraic functions of the flat outputs and their derivatives. The differentially flat expression of the quadcopter dynamics later went on to serve as the basis for what was defined as a dynamically feasible trajectory for the quadrotor UAV to follow. This section serves to illustrate the relationships between the defined flat outputs and the vehicle's state and control inputs.

B.1. Translation

To illustrate the relationship between the translation of the vehicle and the defined flat outputs, we begin by clearly defining the flat outputs in Equation (B.1)

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \varphi \end{bmatrix}. \tag{B.1}$$

It then follows that the velocities and accelerations of the flat outputs are the same as their state-space counterparts. The trivial relationships observed continue for higher order time derivatives.

$$\begin{bmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ \dot{\rho}_4 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi} \end{bmatrix} \tag{B.2}$$

$$\begin{bmatrix} \ddot{\rho}_1 \\ \ddot{\rho}_2 \\ \ddot{\rho}_3 \\ \ddot{\rho}_4 \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\varphi} \end{bmatrix} \tag{B.3}$$

B.2. Attitude

As described in Chapter 2 and derived in depth in section A.1, the description of the vehicle's rotational state is defined by the rotational matrix from the body frame, *B*, to the global frame, *G*, as expressed in Equation (2.1) and identically in Equation (A.4 c). To show that ${}^{G}_{B}\boldsymbol{R}$, and thus the complete rotational state, can be expressed in terms of the flat outputs, the translational acceleration is derived from the linear dynamics as

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = {}^{G}_{B}\boldsymbol{R} \begin{bmatrix} 0 \\ 0 \\ a_T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}. \tag{B.4}$$

In Equation (B.4), a_T represents the acceleration contribution from the thrust generated by the vehicle's rotors. For a given trajectory and its corresponding set of translational accelerations $(\ddot{x}, \ddot{y}, \ddot{z})$, we define the vector $\vec{f}$ describing the mass-normalized forces necessary to follow the trajectory such that

$$\vec{f} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{B.5 a}$$

It then follows from the combination of Equations (B.5 a) and (B.5 b) that

$$\vec{f} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} = {}^{G}_{B}\boldsymbol{R} \begin{bmatrix} 0 \\ 0 \\ a_T \end{bmatrix}. \tag{B.5 b}$$

Equation (B.5 b) indicates that the vector $\vec{f}$ describes the acceleration due to rotor thrust in terms of the global reference frame, *G*. Since the fixed rotors can only create thrust in the direction of $\overrightarrow{z_B}$, it can be found that

$$\overrightarrow{z_B} = \frac{\vec{f}}{\|\vec{f}\|}. \tag{B.6}$$

With $\vec{f}$ having been described in terms of the translational accelerations, their respective flat output relations described in Section B.2, and the gravitational acceleration constant, it follows that $\overrightarrow{z_B}$ can be completely described in terms of flat outputs.

To show that the other vectors that combine with $\overrightarrow{z_B}$ to form ${}^{G}_{B}\boldsymbol{R}$ as expressed in Equation (A.5), namely $\overrightarrow{x_B}$ and $\overrightarrow{y_B}$, attention is turned to the intermediate reference frame I^l detailed in section A.1. Given the yaw angle, ρ_4, that frame I^l was rotated about $\overrightarrow{z_G}$ by, we can see that the X-axis of I^l can be written in terms of the global frame as

$$\overrightarrow{x_{I^1}} = \begin{bmatrix} cos\rho_4 \\ sin\rho_4 \\ 0 \end{bmatrix}. \tag{B.7}$$

We now introduce a preceding superscript to indicate a vector being written in terms of any other reference frame other than the global frame, *G*. Thus, we denote the X-axis of the intermediate frame I^l in terms of *B* as ${}^{B}\overrightarrow{x_{I^1}}$. It is a property of rotational matrices that

$${}^{A}_{B}\boldsymbol{R} = {}^{B}_{A}\boldsymbol{R}^{-1} = {}^{B}_{A}\boldsymbol{R}^{T} \tag{B.8}$$

thus, Equations (A.2) and (A.3) are used to express ${}^{B}\overrightarrow{x_{I^1}}$ as

$${}^{B}\overrightarrow{x_{I^1}} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{B.9 a}$$

$${}^{B}\overrightarrow{x_{I^1}} = \begin{bmatrix} cos\theta \\ 0 \\ sin\theta \end{bmatrix}. \tag{B.9 b}$$

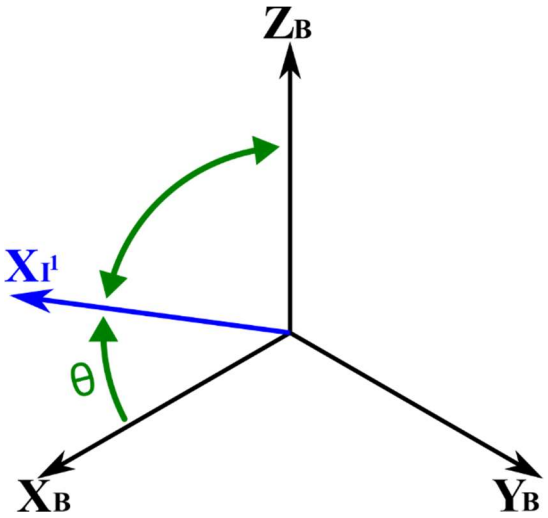


Figure B.1: $\overrightarrow{x_{I^1}}$ in the body frame *B*.

As expressed in Equation (B.9 b) and illustrated in Figure B.1, $\overrightarrow{x_{I^1}}$ is always in the $\overrightarrow{x_B}$ - $\overrightarrow{z_B}$ plane, just rotated about $\overrightarrow{y_B}$ by the pitch angle θ. It can then be reasoned that

$$\overrightarrow{y_B} = \frac{\overrightarrow{z_B} \times \overrightarrow{x_{I^1}}}{\|\overrightarrow{z_B} \times \overrightarrow{x_{I^1}}\|} \tag{B.10}$$

provided that $\overrightarrow{z_B} \times \overrightarrow{x_{I^1}} \neq 0$. Following from Equations (B.6) and (B.7), Equation (B.10) shows that $\overrightarrow{y_B}$ can be expressed in terms of flat outputs. Since the body frame consists of three orthonormal axes, $\overrightarrow{x_B}$ can also be written in terms of flat outputs according to

$$\overrightarrow{x_B} = \overrightarrow{y_B} \times \overrightarrow{z_B}. \tag{B.11}$$

With $\overrightarrow{x_B}$, $\overrightarrow{y_B}$, and $\overrightarrow{z_B}$ all written in terms of flat outputs, Equation (A.5) implies that ${}^G_B\boldsymbol{R}$, and thus the quadcopter's orientation, can be expressed completely in terms of flat outputs.

B.3. Rotational Velocity

Multiplying the vector $\vec{f}$ by mass to describe the forces in totality and rewriting Equation (B.5 b) the following relationship is exhibited

$$m\vec{f} = m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} = T\overrightarrow{z_B}. \tag{B.12}$$

Projecting these forces along $\overrightarrow{z_B}$ and taking the derivative with respect to time, the jerk of the vehicle's center of mass is governed by

$$m\begin{bmatrix} \dddot{x} \\ \dddot{y} \\ \dddot{z} \end{bmatrix} \cdot \overrightarrow{z_B} = \dot{T}. \tag{B.13}$$

The time derivative of the linear dynamics of Equation (2.5) is then taken to express the jerk of the trajectory as

$$m\dot{\vec{a}} = \dot{T}\overrightarrow{z_B} + {}^{G}_{B}\vec{\omega} \times T\overrightarrow{z_B}. \tag{B.14}$$

Reflecting Equation (B.14) onto the axis $\overrightarrow{z_B}$ and using the relationship expressed in Equation (B.13), it is found that

$$m\left[\dot{\vec{a}} - \left(\overrightarrow{z_B} \cdot \dot{\vec{a}}\right)\overrightarrow{z_B}\right] = T({}^{G}_{B}\vec{\omega} \times \overrightarrow{z_B}). \tag{B.15}$$

Realizing that Equation (B.15) represents the projection of $m\dot{\vec{a}}$ onto the $\overrightarrow{x_B} - \overrightarrow{y_B}$ plane, both sides are divided by the magnitude of the thrust force and the cross product of the right-hand side is evaluated in terms of the body frame, *B*

$$\frac{m}{T}\left[\dot{\vec{a}} - \left(\overrightarrow{z_B} \cdot \dot{\vec{a}}\right)\overrightarrow{z_B}\right] = {}^{B}\begin{bmatrix} q \\ -p \\ 0 \end{bmatrix}. \tag{B.16}$$

Reflecting the left-hand side of Equation (B.16) onto the $\overrightarrow{x_B}$ and $\overrightarrow{y_B}$ axes respectively, the body frame components of the rotational velocity are found according to

$$q = \frac{m}{T}\left[\dot{\vec{a}} - \left(\overrightarrow{z_B} \cdot \dot{\vec{a}}\right)\overrightarrow{z_B}\right] \cdot \overrightarrow{x_B} \tag{B.17 a}$$

$$p = -\frac{m}{T}\left[\dot{\vec{a}} - \left(\overrightarrow{z_B} \cdot \dot{\vec{a}}\right)\overrightarrow{z_B}\right] \cdot \overrightarrow{y_B}. \tag{B.17 b}$$

Each of the components on the right side of the expressions in Equation (B.17) have been shown previously to be functions of the flat outputs.

To demonstrate how the angular velocity about the axis $\overrightarrow{z_B}$ is dependent upon flat outputs, it is noted how Figure A.1 illustrates that the yaw rotation is a rotation about the stationary axis $\overrightarrow{z_G}$ of the inertial frame. The magnitude of the angular velocity of this rotation about $\overrightarrow{z_B}$ can thus be stated as a projection of the rotation in the global frame onto $\overrightarrow{z_B}$ as

$$r = \dot{\varphi}\,\overrightarrow{z_G} \cdot \overrightarrow{z_B}. \tag{B.18}$$

With the vectors in Equation (B.18) being written in the global frame, there is no need to demonstrate the dependence of $\overrightarrow{z_G}$ on flat outputs.

B.4. Rotational Acceleration

Demonstration the dependence of the angular accelerations in terms of flat outputs and their derivatives proceeds similarly to that of Section B.3 and commences with taking the second derivative of the linear dynamics, the first derivative of Equation (B.14)

$$m\ddot{\vec{a}} = \ddot{T}\overrightarrow{z_B} + 2\,{}^{G}_{B}\vec{\omega} \times \dot{T}\overrightarrow{z_B} + {}^{G}_{B}\vec{\omega} \times {}^{G}_{B}\vec{\omega} \times T\,\overrightarrow{z_B} + {}^{G}_{B}\vec{\alpha} \times T\,\overrightarrow{z_B} \tag{B.19}$$

where ${}^{G}_{B}\vec{\alpha}$ represents the angular acceleration in the body frame. Rearranging Equation (B.19) and dividing by the magnitude of thrust reveals the following relationship:

$${}^{G}_{B}\vec{\alpha} \times \overrightarrow{z_B} = \frac{m\ddot{\vec{a}}}{T} - \frac{1}{T}\ddot{T}\overrightarrow{z_B} - \frac{2}{T}\,{}^{G}_{B}\vec{\omega} \times \dot{T}\overrightarrow{z_B} - {}^{G}_{B}\vec{\omega} \times {}^{G}_{B}\vec{\omega} \times \overrightarrow{z_B} \tag{B.20}$$

Equation (B.13) provides an expression for the first derivative of thrust. To find an expression for the second derivative of thrust, Equation (B.19) is projected along $\overrightarrow{z_B}$ with the zero terms eliminated and rearranged to find that

$$\ddot{T} = \overrightarrow{z_B} \cdot m\ddot{\vec{a}} - \overrightarrow{z_B} \cdot ({}^{G}_{B}\vec{\omega} \times {}^{G}_{B}\vec{\omega} \times T\,\overrightarrow{z_B}). \tag{B.21}$$

Minding the fact that Equations (B.13) and (B.19) provide the necessary expressions for the derivatives of thrust, Equation (B.20) can be rewritten similar to (B.16) as

$$^{B}\begin{bmatrix} \dot{q} \\ -\dot{p} \\ 0 \end{bmatrix} = \frac{m\ddot{\vec{a}}}{T} - \frac{1}{T}\ddot{T}\overrightarrow{z_B} - \frac{2}{T}\,{}^{G}_{B}\vec{\omega} \times \dot{T}\overrightarrow{z_B} - {}^{G}_{B}\vec{\omega} \times {}^{G}_{B}\vec{\omega} \times \overrightarrow{z_B}\,. \quad \text{(B.22)}$$

It then follows that the scalar values of the two angular accelerations can be found according to

$$\dot{p} = -\left(\frac{1}{T}m\ddot{\vec{a}} - \frac{1}{T}\ddot{T}\overrightarrow{z_B} - \frac{2}{T}\,{}^{G}_{B}\vec{\omega} \times \dot{T}\overrightarrow{z_B} - {}^{G}_{B}\vec{\omega} \times {}^{G}_{B}\vec{\omega} \times \overrightarrow{z_B}\right) \cdot \overrightarrow{y_B} \quad \text{(B.23 a)}$$

$$\dot{q} = \left(\frac{1}{T}m\ddot{\vec{a}} - \frac{1}{T}\ddot{T}\overrightarrow{z_B} - \frac{2}{T}\,{}^{G}_{B}\vec{\omega} \times \dot{T}\overrightarrow{z_B} - {}^{G}_{B}\vec{\omega} \times {}^{G}_{B}\vec{\omega} \times \overrightarrow{z_B}\right) \cdot \overrightarrow{x_B} \quad \text{(B.23 b)}$$

The dependence of the angular acceleration about $\overrightarrow{z_B}$ follows from the same logic presented in Section B.3. Following from the same notions which allowed the formulation of Equation (B.18), it can be shown that

$$\dot{r} = \ddot{\varphi}\,\overrightarrow{z_G} \cdot \overrightarrow{z_B}. \quad \text{(B.24)}$$

B.5. Control Inputs

For completeness, Chapter 2 introduced a set of control inputs defined in Equation (2.9). The first control input, the total thrust generated by all four rotors, was shown to be completely dependent on flat outputs through the relationship express in Equation (B.12).

It then remains to show that the other three control inputs can be written in terms of flat outputs. To achieve this, it is noted that each of the remaining inputs are defined as the moments about each of the axes of the body frame, *B*. Thus, the Euler's equations of Equation (2.6) are rewritten in terms of the control inputs as

$$\mathbf{I}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \text{(B.25 a)}$$

where **I** is the inertia matrix of scalars. Having shown the differentially flat properties of the angular velocities and accelerations, it follows that the last three control inputs can be written in terms of flat outputs as

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \mathbf{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{B.25 b}$$

Appendix C

SPATIAL PH BÉZIER CURVES IN QUATERNION FORM

C. PH Bèzier Curves in Quaternion Form

Chapter 2 states that the spatial quintic Pythagorean hodograph curves benefit from the ability to be represented compactly using quadratic quaternion Bèzier curves, $\mathring{\mathrm{A}}_i(\zeta_i)$. The chapter goes on to show how the spatial path of a UAV's trajectory can be determined by complimenting the boundary conditions with decision variables $\|\vec{p}_i{}'(0)\|$, $\|\vec{p}_i{}'(1)\|$, β_i^0 and β_i^2 to solve for the quaternions $\overline{\mathring{\mathrm{A}}_{\iota,0}}$, $\overline{\mathring{\mathrm{A}}_{\iota,1}}$, and $\overline{\mathring{\mathrm{A}}_{\iota,2}}$ and thus, completely define the spatial path. This appendix serves to summarize the necessary information from [114] to supplement Chapter 2 in describing the necessary relationships needed to describe spatial PH Bèzier curves in quaternion form.

Quaternions were briefly described in Section 2.3.2 as four-dimensional numbers composed of a scalar part and a vector part such that a general quaternion, $\mathring{\mathrm{A}} \in \mathbb{H}$, may be of the form

$$\mathring{\mathrm{A}} = (a, \vec{a}) = a + b\hat{\imath} + c\hat{\jmath} + d\hat{k} \tag{C.1 a}$$

where

$$\vec{a} = b\hat{\imath} + c\hat{\jmath} + d\hat{k}. \tag{C.1 b}$$

The conjugate of a quaternion was also introduced and is defined as

$$\mathring{\mathrm{A}}^* = (a, -\vec{a}) = a - b\hat{\imath} - c\hat{\jmath} - d\hat{k}. \tag{C.2}$$

Pure vector quaternions are those whose scalar has no value and thus can be written

$$\mathrm{V} = (0, \vec{v}) = v_x\hat{\imath} + v_y\hat{\jmath} + v_z\hat{k}. \tag{C.3}$$

Much of the quaternion's renown has stemmed from its application to spatial rotations. Given some vector $\vec{n}$, a pure vector quaternion may be rotated about $\vec{n}$ by an angle θ through quaternion multiplication; the resulting pure vector quaternion, R, is expressed as

$$R = \mathcal{U}\mathcal{V}\mathcal{U}^* \tag{C.4}$$

where $\mathcal{U}$ is the unit quaternion such that $|\mathcal{U}| = 1$ and

$$\mathcal{U} = \left[\cos\tfrac{1}{2}\theta \quad \vec{n}\sin\tfrac{1}{2}\theta\right]. \tag{C.5}$$

Any quaternion can be expressed in terms of the unit vector as $\mathring{A} = |\mathring{A}|\mathcal{U}$, where

$$\left|\mathring{A}\right|^2 = a^2 + \vec{a}^2 = \mathring{A}\mathring{A}^* \tag{C.6}$$

is the square of the magnitude of the quaternion Å.

A polynomial equation with quaternion coefficients, dubbed a quaternion polynomial, may take the form

$$\mathring{A}(\zeta) = u(\zeta) + v(\zeta)\hat{\imath} + p(\zeta)\hat{\jmath} + q(\zeta)\hat{k}. \tag{C.7}$$

Performing a rotation of $\mathring{A}(\zeta)$ about the unit vector $\hat{\imath}$ results according to Equation (C.4) always results in a pure vector quaternion written as

$$\begin{aligned}\mathring{A}(\zeta)\hat{\imath}\mathring{A}^*(\zeta) = {} & [u^2(\zeta) + v^2(\zeta) - p^2(\zeta) - q^2(\zeta)]\hat{\imath} \\ & +2[u(\zeta)q(\zeta) + v(\zeta)p(\zeta)]\hat{\jmath} + 2[v(\zeta)q(\zeta) - u(\zeta)p(\zeta)]\hat{k}.\end{aligned} \tag{C.8}$$

Comparison of Equation (C.8) to the Equation (2.24) alludes to the notion that was identified by Choi et al. [125] that spatial Pythagorean hodographs could be represented as rotations of the unit vector $\hat{\imath}$ about the vector $\vec{n}(\zeta)$ by an angle $\theta(\zeta)$ and scaled by $\left|\mathring{A}(\zeta)\right|^2$. The choice to model spatial PH curves as a spatial rotation of $\hat{\imath}$ is purely conventional; any unit vector can be used in place of $\hat{\imath}$ because of the rotation invariance of the spatial PH

curve quaternion form. It then follows that the hodograph of some spatial PH curve, $\vec{r}(\zeta)$ may be stated as

$$r'(\zeta) = \text{Å}(\zeta)\hat{\imath}\text{Å}^*(\zeta) = [u^2(\zeta) + v^2(\zeta) - p^2(\zeta) - q^2(\zeta)]\hat{\imath} + 2[u(\zeta)q(\zeta) + v(\zeta)p(\zeta)]\hat{\jmath} + 2[v(\zeta)q(\zeta) - u(\zeta)p(\zeta)]\hat{k}. \quad \text{(C.9)}$$

With the desire to obtain a spatial PH curve of 5th degree, the hodograph of the spatial path must be of degree four. Following from the nature of spatial rotations expressed in Equation (C.4), $\text{Å}(\zeta)$ is then described as a quadratic quaternion polynomial. Expressing $\text{Å}(\zeta)$ in Bèzier form,

$$\text{Å}(\zeta) = \sum_{k=0}^{2} \overline{\text{Å}}_k b_k^2(\zeta), \quad \text{(C.10)}$$

where $\overline{\text{Å}}_k$ are the quaternion control points, the polynomials $u(\zeta)$, $v(\zeta)$, $p(\zeta)$, and $q(\zeta)$ are expressed in terms of Bernstein coefficients on $\zeta \in [0,1]$. The integration of Equation (C.9) then leads to the relationships between the control points of the spatial PH Bèzier curve and the quaternion polynomial coefficients as described in Equation (2.29) and repeated below

$$\vec{\bar{p}}_{i,1} = \vec{\bar{p}}_{i,0} + \frac{1}{5}\overline{\text{Å}_{\iota,0}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,0}}$$

$$\vec{\bar{p}}_{i,2} = \vec{\bar{p}}_{i,1} + \frac{1}{10}\left(\overline{\text{Å}_{\iota,0}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,1}} + \overline{\text{Å}_{\iota,1}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,0}}\right)$$

$$\vec{\bar{p}}_{i,3} = \vec{\bar{p}}_{i,2} + \frac{1}{30}\left(\overline{\text{Å}_{\iota,0}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,2}} + 4\overline{\text{Å}_{\iota,1}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,1}} + \overline{\text{Å}_{\iota,2}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,0}}\right)$$

$$\vec{\bar{p}}_{i,4} = \vec{\bar{p}}_{i,3} + \frac{1}{10}\left(\overline{\text{Å}_{\iota,1}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,2}} + \overline{\text{Å}_{\iota,2}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,1}}\right)$$

$$\vec{\bar{p}}_{i,5} = \vec{\bar{p}}_{i,4} + \frac{1}{5}\overline{\text{Å}_{\iota,2}}\,\hat{\imath}\,\overline{\text{Å}^*_{\iota,2}}.$$

From imposition of the boundary conditions described in Section 2.3.2 it is implied that $\overrightarrow{\bar{p}_{\iota,0}} = \vec{p}_i^{\,I}$ and $\overrightarrow{\bar{p}_{\iota,5}} = \vec{p}_i^{\,F}$ thus, it remains to find $\overline{\text{Å}_{\iota,0}}$, $\overline{\text{Å}_{\iota,1}}$, and $\overline{\text{Å}_{\iota,2}}$ to completely find

the spatial path. Recalling that the magnitudes of the hodograph at the initial and final points are to be determined as decision variables in the cost minimization problem, it becomes necessary to evaluate the following

$$\mathring{A}_i(0)\hat{\imath}\mathring{A}_i^*(0) = \overline{\mathring{A}_{\iota,0}}\hat{\imath}\overline{\mathring{A}_{\iota,0}^*} = \vec{p}_i{}'(0) \tag{C.11 a}$$

$$\mathring{A}_i(1)\hat{\imath}\mathring{A}_i^*(1) = \overline{\mathring{A}_{\iota,2}}\hat{\imath}\overline{\mathring{A}_{\iota,2}^*} = \vec{p}_i{}'(1). \tag{C.11 b}$$

Equation (C.11) represents a mapping of the unit basis vector $\hat{\imath}$ to the vectors $\vec{p}_i{}'(0)$ and $\vec{p}_i{}'(1)$. Posing the problem generally in terms of some vector $\vec{G}$ in $\mathbb{R}^3$ such that

$$\mathring{A}\hat{\imath}\mathring{A}^* = \vec{G}, \tag{C.12}$$

the solution can be expressed in terms of the magnitude of $\vec{G}$, its directional cosines (λ, μ, ν), and an angular parameter β as

$$\mathring{A} = \sqrt{\frac{1}{2}(1+\lambda)\|\vec{G}\|}\left(-\sin\beta + \cos\beta\,\hat{\imath} + \frac{\mu\cos\beta + \nu\sin}{1+\lambda}\hat{\jmath} + \frac{\nu\cos\beta + \mu\sin}{1+\lambda}\hat{k}\right). \tag{C.13}$$

Application of Equation (C.13) to the expressions of (C.11) allow for $\overline{\mathring{A}_{\iota,0}}$ and $\overline{\mathring{A}_{\iota,2}}$ to be solved for from the problem's boundary conditions and decision variables.

With $\overline{\mathring{A}_{\iota,0}}$ and $\overline{\mathring{A}_{\iota,2}}$ accounted for it remains to find $\overline{\mathring{A}_{\iota,1}}$ and thus, completely define the spatial path. Interpolation of the endpoint conditions gives

$$\begin{aligned}
&\int_0^1 \mathring{A}_i(\zeta)\hat{\imath}\mathring{A}_i^*(\zeta)\,d\zeta = \vec{p}_i^F - \vec{p}_i^I \\
&= \frac{1}{5}\overline{\mathring{A}_{\iota,0}}\hat{\imath}\overline{\mathring{A}_{\iota,0}^*} + \frac{1}{10}\left(\overline{\mathring{A}_{\iota,1}}\hat{\imath}\overline{\mathring{A}_{\iota,0}^*} + \overline{\mathring{A}_{\iota,0}}\hat{\imath}\overline{\mathring{A}_{\iota,1}^*}\right) + \frac{1}{30}\left(\overline{\mathring{A}_{\iota,0}}\hat{\imath}\overline{\mathring{A}_{\iota,2}^*} + 4\overline{\mathring{A}_{\iota,1}}\hat{\imath}\overline{\mathring{A}_{\iota,1}^*} + \overline{\mathring{A}_{\iota,2}}\hat{\imath}\overline{\mathring{A}_{\iota,0}^*}\right) + \\
&\qquad \frac{1}{10}\left(\overline{\mathring{A}_{\iota,1}}\hat{\imath}\overline{\mathring{A}_{\iota,2}^*} + \overline{\mathring{A}_{\iota,2}}\hat{\imath}\overline{\mathring{A}_{\iota,1}^*}\right) + \frac{1}{5}\overline{\mathring{A}_{\iota,2}}\hat{\imath}\overline{\mathring{A}_{\iota,2}^*}.
\end{aligned} \tag{C.14}$$

Substituting Equation (C.11) into (C.14) and rearranging yields the relationship

$$\left(3\overline{\mathring{\mathrm{A}}_{\iota,0}} + 4\overline{\mathring{\mathrm{A}}_{\iota,1}} + 3\overline{\mathring{\mathrm{A}}_{\iota,2}}\right)\hat{\imath}\left(3\overline{\mathring{\mathrm{A}}_{\iota,0}} + 4\overline{\mathring{\mathrm{A}}_{\iota,1}} + 3\overline{\mathring{\mathrm{A}}_{\iota,2}}\right)^{*}$$

$$= 120(\vec{p}_i^{\,F} - \vec{p}_i^{\,I}) - 15\left(\vec{p}_i{}'(0) + \vec{p}_i{}'(1)\right) + 5\left(\overline{\mathring{\mathrm{A}}_{\iota,0}}\hat{\imath}\overline{\mathring{\mathrm{A}}_{\iota,2}^{*}} + \overline{\mathring{\mathrm{A}}_{\iota,2}}\hat{\imath}\overline{\mathring{\mathrm{A}}_{\iota,0}^{*}}\right) = \vec{G}. \quad \text{(C.15)}$$

Noting that Equation (C.15) is of the same form as (C.12), the general solution to (C.15) in terms of $\vec{G}$ is stated as

$$\overline{\mathring{\mathrm{A}}_{\iota,1}}(\beta_i^1) = -\frac{3}{4}\left(\overline{\mathring{\mathrm{A}}_{\iota,0}} + \overline{\mathring{\mathrm{A}}_{\iota,2}}\right) + \frac{\sqrt{\frac{1}{2}(1+\lambda)\|\vec{G}\|}}{4}(-\sin\beta_i^1 + \cos\beta_i^1\,\hat{\imath}$$

$$+ \frac{\mu\cos\beta_i^1 + \nu\sin\beta_i^1}{1+\lambda}\hat{\jmath} + \frac{\nu\cos\beta_i^1 + \mu\sin\beta_i^1}{1+\lambda}\hat{k}). \quad \text{(C.16)}$$

In [115] it is shown that the curve shape depends only on the differences of the angles β_i^0, β_i^1, and β_i^2 and thus, Equation (C.16) can be simplified to depend only on $\overline{\mathring{\mathrm{A}}_{\iota,0}}$ and $\overline{\mathring{\mathrm{A}}_{\iota,2}}$ by choosing a value of β_i^1.

References

[1] Thompson M. *Icarus* [Internet]. Encyclopedia Mythica Online. 2017. Available from: http://www.pantheon.org/articles/i/icarus.html

[2] White L. *Eilmer of Malmesbury, an Eleventh Century Aviator: A Case Study of Technological Innovation, Its Context and Tradition*. Technol Cult [Internet]. 1961;2(2):97–111. Available from: http://www.jstor.org/stable/3101411

[3] Kemp M. *Leonardo da Vinci: experience, experiment and design*. Princeton University Press Princeton, New Jersey; 2006.

[4] *The Wright Brothers & The Invention of the Aerial Age*. Smithsonian Institute.

[5] Johnson MA. *On the Aviation Trail in the Wright Brothers' West Side Neighborhood in Dayton , Ohio*. 2001;

[6] Harbidge J. *Flying through the ages* [Internet]. BBC News Online. 1999. Available from: http://news.bbc.co.uk/2/hi/special_report/1998/11/98/great_balloon_challenge/29956.stm

[7] Giges NS. *Igor Sikorsky: Aviation Pioneer*. ASME.org. 2013.

[8] Kennett LB. *A History of Strategic Bombing* [Internet]. Scribner; 1982. Available from: https://books.google.com/books?id=kXmAQgAACAAJ

[9] Mark Pomerleau. *How technology has changed intelligence collection* [Internet]. defencesystems.com. 2015 [cited 2017 Feb 6]. Available from: https://defensesystems.com/articles/2015/04/22/technology-has-changed-intelligence-gathering.aspx

[10] Bhalla P. *Emerging Trends in Unmanned Aerial Systems*. 1991;86–94.

[11] Engel A, Teichert B. *The Photogrammetric Potential of Low-Cost UAVs In Forestry and Agriculture*. Int Arch Photogramm Remote Sens Spat Inf Sci. 2008;XXXVII(B1):1207–14.

[12] Adams SM, Friedland CJ. *A Survey of Unmanned Aerial Vehicle (UAV) Usage for Imagery Collection in Disaster Research and Management*. In: 9th International Workshop on Remote Sensing for Disaster Response. Stanford University; 2011.

[13] Horcher A, Visser RJM. *Unmanned Aerial Vehicles : Applications for Natural Resource Management and Monitoring*. 2004;

[14] *Northrop Grumman RQ-4A Global Hawk* [Internet]. UAVGlobal. Available from: http://www.uavglobal.com/rq-4a-global-hawk/

[15] *550mm RTF Quadcopter UAV* [Internet]. Robotshop.com. Available from: http://www.robotshop.com/en/550mm-rtf-quadcopter-uav.html

[16] Magurran AE. *The adaptive significance of schooling as an anti-predator defence in fish*. Ann Zool Fenn. 1990;27:51–66.

[17] Andrews S, Columbia B. *Vigilant Behaviour and Shoal Size in Minnows*. 1985;(February 2016).

[18] Magurran AE. *Predator inspection behaviour in minnow shoals: differences between populations and individuals*. Behav Ecol Sociobiol. 1986;19(4):267–73.

[19] Magurran AE, Higham A. *Information Transfer across Fish Shoals under Predator Threat*. Ethology [Internet]. 1988;78(2):153–8. Available from:

http://dx.doi.org/10.1111/j.1439-0310.1988.tb00226.x
[20] Smith RJF, Smith MJ. *Predator-recognition Behaviour in Two Species of Gobiid Fishes, Asterropteryx semipunctatus and Gnatholepis anjerensis*. Ethology [Internet]. 1989;83(1):19–30. Available from: http://dx.doi.org/10.1111/j.1439-0310.1989.tb00516.x
[21] Handegard NO, Boswell KM, Ioannou CC, Leblanc SP, Tjøstheim DB, Couzin ID. *Report The Dynamics of Coordinated Group Hunting and Collective Information Transfer among Schooling Prey*. CURBIO [Internet]. 2012;22(13):1213–7. Available from: http://dx.doi.org/10.1016/j.cub.2012.04.050
[22] George CJW. *Behavioral interaction in the pickerel (Esox niger LeSueur and Esox americanus LeSueur) and the mosquito-fish (Gambusia patruelis)*. Harvard University; 1960.
[23] Krause J, Ruxton GD. *Living in groups*. Oxford University Press; 2002.
[24] Packer C, Scheel D, Pusey AE. *Why Lions Form Groups: Food is Not Enough*. Am Nat [Internet]. 1990;136(1):1–19. Available from: http://www.jstor.org/stable/2556343
[25] Creel S. *Communal hunting and pack size in African wild dogs , Lycaon pictus*. 1995;1325–39.
[26] Macdonald DW. *The ecology of carnivore social behaviour*. Nature [Internet]. 1983 Feb 3;301(5899):379–84. Available from: http://dx.doi.org/10.1038/301379a0
[27] Boesch C. *Cooperative hunting in wild chimpanzees*. Anim Behav [Internet]. 1994 Sep;48(3):653–67. Available from: http://www.sciencedirect.com/science/article/pii/S0003347284712851
[28] Vail AL, Manica A, Bshary R. *Referential gestures in fish collaborative hunting*. Nat Commun [Internet]. 2013;4:1765–7. Available from: http://dx.doi.org/10.1038/ncomms2781
[29] Bshary R, Hohner A, Ait-el-djoudi K, Fricke H. *Interspecific Communicative and Coordinated Hunting between Groupers and Giant Moray Eels in the Red Sea*. 2006;4(12).
[30] Hector DP. *Cooperative Hunting and its Relationship to Foraging Success and Prey Size in an Avian Predator*. Ethology [Internet]. 1986;73(3):247–57. Available from: http://dx.doi.org/10.1111/j.1439-0310.1986.tb00915.x
[31] Camazine S. *Self-organization in biological systems*. Princeton University Press; 2003.
[32] Anderson C, Theraulaz G, Deneubourg J. *Self-assemblages in insect societies*. 2002;49:99–110.
[33] Gaudiano P, Shargel B, Bonabeau E, Clough B. *Control of UAV Swarms: What the Bugs Can Teach Us*. 2nd AIAA "Unmanned Unlimited" Conf Work Exhib [Internet]. 2003;1–11. Available from: http://arc.aiaa.org/doi/10.2514/6.2003-6624
[34] Reid CR, Lutz MJ, Powell S, Kao AB, Couzin ID, Garnier S. *Army ants dynamically adjust living bridges in response to a cost – benefit trade-off*. 2015;112(49):15113–8.
[35] Mlot NJ, Tovey CA, Hu DL. *Fire ants self-assemble into waterproof rafts to survive floods*. 2011;1–5.
[36] Beauchamp G. *Chapter 1 - Finding and Exploiting Food in Groups BT - Social*

Predation. In San Diego: Academic Press; 2014. p. 3–27. Available from: http://www.sciencedirect.com/science/article/pii/B9780124072282000019

[37] Weimerskirch H, Martin J, Clerquin Y, Alexandre P, Jiraskova S. *Energy saving in flight formation*. Nature [Internet]. 2001 Oct 18;413(6857):697–8. Available from: http://dx.doi.org/10.1038/35099670

[38] Braun T, Pessin G, Costa FG, Os FS, S PUSP, Bern B. *THE USE OF UNMANNED AERIAL VEHICLES AND WIRELESS SENSOR NETWORK IN AGRICULTURAL APPLICATIONS Institute of Mathematics and Computer Science (ICMC) University of S ˜ Institute of Computer Science and Applied Mathematics School of Mathematical and Computer*. 2012;5045–8.

[39] *A Survey of Unmanned Aerial Vehicle (UAV) Usage for Imagery Collection in Disaster Research and Management*. 2015;(February).

[40] Dominici D, Baiocchi V, Zavino A, Alicandro M, Elaiopoulos M. *Micro UAV for post seismic hazards surveying in old city center of L'Aquila*. FIG Work Week 2012 - Knowing to Manag Territ Prot Environ Eval Cult Herit. 2012;(May):15.

[41] Ackerman E. *Global Hawk UAV May Be Able to Peek inside Damaged Reactors.* [Internet]. IEEE Spectrum. 2011 [cited 2017 Feb 6]. Available from: http://spectrum.ieee.org/automaton/robotics/military-robots/global-hawk-uav-may-be-able-to-peek-inside-damaged-reactors

[42] Crowe S. *Can These Swarming Drones Improve Disaster Response?* [Internet]. Robotics Trends. 2015 [cited 2017 Feb 6]. Available from: http://www.roboticstrends.com/article/can_these_swarming_drones_improve_disaster_response

[43] Arquilla J, Ronfeldt D. *Swarming and the future of conflict*.

[44] Edwards SJA. *Swarming and the Future of Warfare*. Pardee RAND Graduate School; 2005.

[45] Jankovic R. *Computer Simulation of an Armoured Battalion Swarming Computer Simulation of an Armoured Battalion Swarming*. 2017;(April).

[46] Schumacher C, Chandler PR, Rasmussen SJ, Walker D. *Task allocation for wide area search munitions with variable path length*. Proc 2003 Am Control Conf 2003. 2003;4:3472–7.

[47] Schumacher C, Chandler PR, Rasmussen SR, Chandler PR, Rasmussen SR. *SEARCH MUNITIONS VIA ITERATIVE NETWORK FLOW*. 2002;

[48] Nigam N, Bieniawski S, Kroo I, Vian J. *Control of multiple UAVs for persistent surveillance: Algorithm and flight test results*. IEEE Trans Control Syst Technol. 2012;20(5):1236–51.

[49] Dono TF, Chung TH. *Optimized transit planning and landing of aerial robotic swarms*. Proc - IEEE Int Conf Robot Autom. 2013;1843–50.

[50] Smalley D. *LOCUST: Autonomous, swarming UAVs fly into the future* [Internet]. Arlington, VA; 2015. Available from: https://www.onr.navy.mil/Media-Center/Press-Releases/2015/LOCUST-low-cost-UAV-swarm-ONR.aspx

[51] Evangelista DJ, Ray DD, Raja SK, Hedrick TL. *Three-dimensional trajectories and network analyses of group behaviour within chimney swift flocks during approaches to the roost*. Proc R Soc B. 2017;

[52] Bürkle A, Segor F, Kollmann M. *Towards autonomous micro UAV swarms*. J Intell Robot Syst Theory Appl. 2011;61(1–4):339–53.

[53] Reynolds CW. *Flocks, herds and schools: A distributed behavioral model*. ACM SIGGRAPH Comput Graph [Internet]. 1987;21(4):25–34. Available from: http://portal.acm.org/citation.cfm?doid=37402.37406
[54] Waldrop MM. *Complexity: The Emerging Science at the Edge of Order and Chaos* [Internet]. Simon & Schuster; 1993. (A Touchstone Book). Available from: https://books.google.com/books?id=JTRJxYK_tZsC
[55] Bonabeau E, Dorigo M, Therauluz G. *Swarm Intelligence From Natural to Artificial Systems*. Santa Fe Institute Studies in the Sciences of Complexity. New York: Oxford University Press; 1999.
[56] Clough B. *UAV Swarming? So What Are Those Swarms, What Are The Implications, And How Do We Handle Them?* Read. 2002;
[57] Couzin ID, Krause J, James R, Ruxton GD, Franks NR. *Collective memory and spacial sorting in animal groups*. J Theor Biol. 2002;218(1):1–11.
[58] Reuter H, Breckling B. *Selforganization of fish schools: an object-oriented model*. Ecol Modell [Internet]. 1994;75:147–59. Available from: http://www.sciencedirect.com/science/article/pii/0304380094900140
[59] Mirabet V, Auger P, Lett C. *Spatial structures in simulations of animal grouping*. Ecol Modell [Internet]. 2007;201(3):468–76. Available from: http://www.sciencedirect.com/science/article/pii/S0304380006005102
[60] Hemelrijk CK, Hildenbrandt H. *Self-organized shape and frontal density of fish schools*. Ethology. 2008;114(3):245–54.
[61] Hildenbrandt H, Carere C, Hemelrijk CK. *Self-organized aerial displays of thousands of starlings: A model*. Behav Ecol. 2010;21(6):1349–59.
[62] Hemelrijk CK, Hildenbrandt H. *Schools of fish and flocks of birds: their shape and internal structure by self-organization*. Interface Focus. 2012;2(6):726–37.
[63] Kubo Y, Iwasa Y. *Phase diagram of a multiple forces model for animal group formation: marches versus circles determined by the relative strength of alignment and cohesion*. Popul Ecol. 2016;58(3):357–70.
[64] Reynolds CW. *Boids - Background and Update* [Internet]. 2001 [cited 2017 Feb 6]. Available from: http://www.red3d.com/cwr/boids/
[65] Watson NR, John NW, Crowther WJ. *Simulation of unmanned air vehicle flocking*. Proc - Theory Pract Comput Graph TPCG 2003. 2003;130–7.
[66] Crowther B, Riviere X. *Flocking of autonomous unmanned air vehicles*. 17th UAV System conference, Bristol UK. 2002.
[67] Crowther WJ. *Rule-based guidance for flight vehicle flocking*. Proc Inst Mech Eng Part G-Journal Aerosp Eng. 2004;218(September):111–24.
[68] Virágh C, Vásárhelyi G, Tarcai N, Szörényi T, Somorjai G, Nepusz T, et al. *Flocking algorithm for autonomous flying robots.* Bioinspir Biomim [Internet]. 2014;9(2):25012. Available from: http://www.ncbi.nlm.nih.gov/pubmed/24852272
[69] Symington A, De Nardi R, Julier S, Hailes S. *Simulating quadrotor UAVs in outdoor scenarios*. IEEE Int Conf Intell Robot Syst. 2014;(Iros):3382–8.
[70] Sharma R, Ghose D. *Swarm intelligence based collision avoidance between realistically modelled UAV clusters*. Proc Am Control Conf. 2007;(7):3892–7.
[71] Sharma RK, Ghose D. *Collision avoidance between UAV clusters using swarm intelligence techniques*. Int J Syst Sci [Internet]. 2009;40(5):521–38. Available from: http://www.tandfonline.com/doi/abs/10.1080/00207720902750003

[72] Clark JB, Jacques DR. *Flight test results for UAVs using boid guidance algorithms*. Procedia Comput Sci [Internet]. 2012;8:232–8. Available from: http://dx.doi.org/10.1016/j.procs.2012.01.048

[73] Bennet DJ, McInnes CR, Suzuki M, Uchiyama K. *Autonomous Three-Dimensional Formation Flight for a Swarm of Unmanned Aerial Vehicles*. J Guid Control Dyn. 2011;34}(6):1899–1908}.

[74] O.~Holland, J.~Woods, R.~De~Nardi, A.~Clark. *Beyond swarm intellegence: The UltraSwarm*. Proc 2005 IEEE Swarm Intell Symp. 2005;217–24.

[75] De Nardi R, Holland O, Woods J, Clark A. *SwarMAV: A Swarm of Miniature Aerial Vehicles*. Proc 21st Bristol Int UAV Syst Conf [Internet]. 2006;(April 2006). Available from: http://cogprints.org/5569/

[76] Michael N, Mellinger D, Lindsey Q, Kumar V. *The GRASP multiple micro-UAV testbed*. IEEE Robot Autom Mag. 2010;17(3):56–65.

[77] Valenti M, Bethke B, Dale D, Frank A, McGrew J, Ahrens S, et al. *The MIT indoor multi-vehicle flight testbed*. Proc - IEEE Int Conf Robot Autom. 2007;2758–9.

[78] Valenti M, Bethke B, Fiore G, How JP. *Indoor Multi-Vehicle Flight Testbed for Fault Detection , Isolation , and Recovery*. AIAA Guid Navig Control Conf Exhib. 2006;(8):1–18.

[79] Teacy WTL, Nie J, McClean S, Parr G, Hailes S, Julier S, et al. *Collaborative Sensing by Unmanned Aerial Vehicles*. Communication [Internet]. 2009;13–6. Available from: http://eprints.soton.ac.uk/267290/

[80] Daniel K, Dusza B, Lewandowski A, Wietfeld C. *Airshield: A system-of-systems muav remote sensing architecture for disaster response*. 2009 IEEE Int Syst Conf Proc. 2009;196–200.

[81] Shim DH, Kim HJ, Sastry S. *A flight control system for aerial robots: Algorithms and experiments*. IFAC Proc Vol. 2002;15(1):241–6.

[82] Kushleyev A, Mellinger D, Powers C, Kumar V. *Towards a swarm of agile micro quadrotors*. Auton Robots. 2013;35(4):287–300.

[83] Choset HM. *Principles of robot motion: theory, algorithms, and implementation*. MIT press; 2005.

[84] Latombe J-C. *Robot motion planning*. Vol. 124. Springer Science & Business Media; 1991.

[85] Yang L, Qi J, Song D, Xiao J, Han J, Xia Y. *Survey of Robot 3D Path Planning Algorithms*. J Control Sci Eng. 2016;2016.

[86] Yang L, Qi J, Xiao J, Yong X. *A literature review of UAV 3D path planning*. Proc World Congr Intell Control Autom. 2015;2015–March(March):2376–81.

[87] Cho Y, Kim DD, Diagram V, Kim DD. *Topology Representation for the Voronoi Diagram of 3D Spheres*. Int J CAD/CAM. 2005;5(1):59~68.

[88] Karaman S, Frazzoli E. *Incremental Sampling-based Algorithms for Optimal Motion Planning*. Robot Sci Syst [Internet]. 2010;20. Available from: http://arxiv.org/abs/1005.0416

[89] Yang K, Sukkarieh S. *3D smooth path planning for a UAV in cluttered natural environments*. 2008 IEEE/RSJ Int Conf Intell Robot Syst IROS. 2008;794–800.

[90] Hwang YK, Ahuja N. *A potential field approach to path planning*. IEEE Trans Robot Autom [Internet]. 1992;8(1):23–32. Available from: files/1274/Hwang and

Ahuja - 1992 - A potential field approach to path planning.pdf%5Cnfiles/1275/abs_all.html

[91] Dijkstra EW. *A Note on Two Probles in Connexion with Graphs*. Numer Math. 1959;1(1):269–71.

[92] Hart PE, Nilsson NJ, Raphael B. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths* [Internet]. ACM SIGART Bulletin. 1972. p. 28–9. Available from: http://portal.acm.org/citation.cfm?doid=1056777.1056779

[93] Schumacher C, Chandler P, Pachter M, Pachter L. *UAV Task Assignment With Mixed-Integer Linear Programming*. Am Control Conf. 2004;

[94] Masehian E, Habibi G. *Robot Path Planning in 3D Space Using Binary Integer Programming*. Int J Mech Syst Sci Eng. 2007;1(5):1255–60.

[95] C´abbas Chamseddine Youmin Zhang CAR, Theilliol EJD. *Flatness-Based Trajectory Planning / Replanning for a Quadrotor Unmanned Aerial Vehicle*. Ieee Trans Aerosp Electron Syst. 2012;48(4):2832–48.

[96] Biegler LT, Borrelli F, Subramanian D. *MILP and NLP Techniques for centralized trajectory planning of multiple unmanned air vehicles A comparison between MILP and NLP Techniques for Centralized Trajectory Planning of Multiple Unmanned Air Vehicles*. 2006;(July 2016).

[97] Miller B, Stepanyan K, Miller A, Andreev M. *3D path planning in a threat environment*. Proc IEEE Conf Decis Control. 2011;6864–9.

[98] Jia D, Vagners J. *Parallel evolutionary algorithms for uav path planning*. 2004;(September).

[99] Besada-Portas E, De La Torre L, De La Cruz JM, De Andrés-Toro B. *Evolutionary trajectory planner for multiple UAVs in realistic scenarios*. IEEE Trans Robot. 2010;26(4):619–34.

[100] Hasircioglu I. *3-D Path Planning for the Navigation of Unmanned Aerial Vehicles by Using Evolutionary Algorithms*. 10th Annu Conf Genet Evol Comput. 2008;1499–506.

[101] Cheng CT, Fallahi K, Leung H, Tse CK. *Cooperative path planner for UAVs using ACO algorithm with gaussian distribution functions*. Proc - IEEE Int Symp Circuits Syst. 2009;173–6.

[102] Shang K, Karungaru S, Feng Z, Ke L, Terada K. *A GA-ACO Hybrid Algorithm for the Multi-UAV Mission Planning Problem*. 2014;243–8.

[103] Mellinger D, Kumar V. *Minimum Snap Trajectory Generation and Control for Quadrotors*. 2011;2520–5.

[104] Richter C, Bry A, Roy N. *Polynomial Trajectory Planning for Quadrotor Flight*. In: Robotics Research Springer Tracts in Advanced Robotics. 2016. p. 649–66.

[105] Lakshmanan A. *Piecewise Bezier Curve Trajectory Generation and Control for Quadrotors*. University of Illinois at Urbana-Champaign; 2016.

[106] Choe R, Puig J, Cichella V, Xargay E, Hovakimyan N. *Trajectory generation using spatial Pythagorean Hodograph Bézier curves*. In: AIAA Guidance, Navigation, and Control Conference. 2015. p. 597.

[107] Choe R, Cichella V, Xargay E, Hovakimyan N, Trujillo AC, Kaminer I. *A trajectory-generation framework for time-critical cooperative missions*. In: AIAA Infotech@ Aerospace (I@ A) Conference. 2013. p. 4582.

[108] Mehdi SB, Choe R, Cichella V, Hovakimyan N. *Collision avoidance through path*

replanning using Bézier curves. In: AIAA Guidance, Navigation, and Control Conference. 2015. p. 598.

[109] Mehdi SB, Choe R, Hovakimyan N. *Avoiding multiple collisions through trajectory replanning using piecewise B??zier curves*. Proc IEEE Conf Decis Control. 2016;2016–Febru(Cdc):2755–60.

[110] Deshmukh V. *Dynamic trajectory planning for mobile robot intercepting a moving target using Bezier curve*. In: Proceedings of the 2016 IEEE Students' Technology Symposium Dynamic. 2016. p. 322–7.

[111] Craig JJ. *Introduction to Robotics: Mechanics and Control* [Internet]. Pearson/Prentice Hall; 2005. (Addison-Wesley series in electrical and computer engineering: control engineering). Available from: https://books.google.com/books?id=MqMeAQAAIAAJ

[112] Choe R, Puig-navarro J, Cichella V, Xargay E, Hovakimyan N. *Cooperative Trajectory Generation Using Pythagorean Hodograph Bézier Curves*. 2016;39(8).

[113] Cichella V, Choe R, Mehdi SB, Xargay E, Hovakimyan N, Dobrokhodov V, et al. *Safe Coordinated Maneuvering of Teams of Multirotor UAVs*. 2016;1–69.

[114] Farouki RT. *Pythagorean- Hodograph Curves*.

[115] Farouki RT. *Spatial Pythagorean hodographs , quaternions , and rotations in R 3 and R 4*.

[116] Farouki RT. *The Elastic Bending Energy of Pythagorean-hodograph Curves*. Comput Aided Geom Des [Internet]. 1996 Apr;13(3):227–41. Available from: http://dx.doi.org/10.1016/0167-8396(95)00024-0

[117] Chang J, Choi Y, Kim M, Wang W. *Computers & Graphics Computation of the minimum distance between two Be*. 2011;35:677–84.

[118] Gilbert EG, Keerthi SS. *A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space*. 1988;4(2).

[119] Lindemann P. *The Gilbert-Johnson-Keerthi Distance Algorithm*. 2009;

[120] Mäkelä MM. *Multiobjective Proximal Bundle Method for Nonconvex Nonsmooth Optimization: Fortran Subroutine MPBNGC 2.0*. Reports of the Department of Mathematical Information Technology, Series B, Scientific Computing. Jyväskylä; 2003.

[121] Ascending Technologies. *AscTec Hummingbird* [Internet]. Available from: http://www.asctec.de/en/uav-uas-drones-rpas-roav/asctec-hummingbird/

[122] Neeley W, Fierro R, Abdallah C, Neeley W. *Design and Development of a High-Performance Quadrotor Control Architecture Based on Feedback Linearization by*. 2015;175.

[123] Hoffmann GM, Waslander SL, Tomlin CJ. *Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering*.

[124] Spong MW, Hutchinson S, Vidyasagar M. *Robot modeling and control*. Vol. 3. Wiley New York; 2006.

[125] In H, Seok D, Pyo H. *Clifford algebra , spin representation , and rational parameterization of curves and surfaces* . 2002;5–48.