McGill University

Master of Engineering

Mechanical Engineering

Quadcopter Collision Recovery Under Loss of One Propeller

Author:

Sam El Toufaili

Supervisor:

Prof. Inna Sharf

April 2019

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Engineering

Montreal, Quebec

Abstract

Quadcopters are prone to accidents and collisions with objects in urban environment such as building walls, power lines, poles and fences. During a wall collision, there is a high probability that one of the propellers on the vehicle will be damaged, as a result breaking the nominal control loop and causing the quadcopter to crash. This thesis deals with flight control recovery for when a quadcopter collides with a solid wall and experiences a loss of one propeller. Such a scenario is directly relevant to quadcopters flying in urban environments in close proximity to buildings. Following the literature review on fault-tolerant controllers for loss of one actuator, the dynamics model for a vehicle undergoing a collision with a wall is presented. A recovery flight control law using "relaxed hover" approach was selected, where the requirement for having zero angular velocity is relaxed and the vehicle hovers while rotating around a fixed axis. The recovery control strategy is made up of two cascaded controllers and consists of three phases. In the first, the impact event is identified, and the normal direction of the contact surface is estimated from accelerometer measurements. The second phase uses the surface normal direction to calculate a desired position at a safe distance away from the wall, which is used in the outer position controller to define the desired acceleration. For the case where position estimates of the vehicle are not available, an alternative velocity controller is proposed to calculate a desired acceleration and control the velocity of the vehicle during the recovery. In the last phase, the inner controller regulates the attitude of the vehicle so that the desired acceleration is achieved. This control strategy is simulated in MATLAB simulator of the system for different incoming impact velocities and attitudes, in order to demonstrate the performance of the controller and to define a bound for successful recoveries.

Résumé

Les drones sont sujets à des accidents et des collisions avec des objets dans un environnement urbain tels que les immeubles, les lignes électriques, des pôles et des clôtures. Durant une collision, il y a de fortes chances qu'un des propulseurs du véhicule soit endommagé, résultant dans une défectuosité du cycle de contrôle principal et causant la destruction du drone. La thèse présentée gère la récupération du contrôle de vol lorsque le drone entre en collision avec des obstacles solide et perd un des propulseurs. Dans un tel scénario, il est directement en lien avec des drones volant dans un environnement urbain à proximité des immeubles. En suivant la littérature sur les contrôleurs tolérants la perte d'un actionneur, le modèle dynamique d'un véhicule entrant en collision avec un mur est présenté. Un récupérateur de contrôle de vol suivant la méthode « planage relaxé » a été sélectionné. Ici, les requis accordant une vitesse angulaire nul ont été relaxées ce qui permet au véhicule d'avoir une vitesse angulaire lors du planage. La stratégie de récupération de contrôle est faite de deux contrôleurs en cascade et est constituée de trois phases. Dans la première phase, l'impact est identifié et la surface de contacte est considérée normal à la direction estimée par les mesures de l'accéléromètre. La seconde phase utilise la direction normale de la surface pour calculer une position désirée à une distance sécuritaire du mur qui est utilisée dans le contrôleur de position externe pour déterminer l'accélération nécessaire. Dans le cas où la position estimée du véhicule n'est pas disponible, un contrôleur de vitesse alternatif est proposé pour calculer une accélération optimale et contrôler la vitesse du véhicule durant le rétablissement. Dans la dernière phase, le contrôleur interne régularise le comportement du véhicule afin d'atteindre l'accélération calculée. Cette stratégie de contrôle est simulée dans MATLAB pour des impacts de différentes vitesses et

comportements dans le but de démontrer la performance du contrôleur ainsi que pour déterminer la limite des rétablissement réussis.

Acknowledgements

I would like to acknowledge my sincere gratitude to my supervisor Professor Inna Sharf for her patient guidance, constant support and most importantly her constant motivation throughout my master's degree. Her ideas, encouragement, attention to details and critiques have immensely improved and changed the way I approach problems.

A big thanks to my friends and colleagues at the Aerospace Mechatronics Lab, current and previous, who were always helpful in answering my pestering questions and sometimes laughing at my jokes. Whether it was discussions, experiments, or work, there was never a dull moment. Thank you Gareth, Bassam, Adrian, Romain, Juan, Fares, Walter, Sean, and Corey. A special thanks to Eitan for his immense help in PX4 and overall his insightful suggestions.

To my family, thank you for encouraging me in all of my pursuits and always pushing me to reach for higher limits. I am especially grateful to my parents, who supported me emotionally and financially. I always knew that you believed in me and wanted the best for me despite my occasional attempts to run off and do something impulsive like open a restaurant, start a block-chain company, or even a trust-fund at one point.

To my friends, thank you for listening, helping me balance out my life, and supporting me through this entire process. Special thanks to Tala, Maged, Luca, and Junia. The gaming nights, dinners, and memories, as well as road trips, were all much appreciated.

Finally, special thanks to the whole staff at the Department of Mechanical Engineering that made the entire process smooth especially the submission procedure.

To my parents, for their support, encouragement and sacrifices.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Quadcopters [1] and multicopters, in general, have evolved into popular and versatile flight platforms over the years. Their capability to offer great agility along with their mechanical simplicity, due to having only a few moving parts, have increased their demand. The outward positioning of the motors results in a larger torque which leads to high angular accelerations. This along with a large thrust-to-weight ratio, gives quadcopters their excellent agility. The mechanical simplicity which arises from the quadcopter having only four moving parts motors/propellers makes the repair and mechanical maintenance much easier than helicopters with their complex swash-plate mechanism. In addition, the use of relatively low cost, off-the-shelf components, makes quadcopters more economically attractive. These advantages have led to applications of quadcopters into many different fields notably as platforms to deliver packages [1] [2], humanitarian aid [3] [4], and uses in real estate [5], film [6], and agriculture [7], and many others.

A typical strategy to increase the safety of multicopters is redundancy, where multiple independent systems exist which can perform the same task. An example

[1] In this thesis, the term quadcopter is used which has the same meaning as quadrotor.

of redundancy in a multicopter is the use of a backup safety pilot in autonomous missions, where the pilot can remotely take over in case of an anomaly in the autonomous control. Another example would be the use of a hexacopter (6 propellers) or even octocopters (8 propellers) that can fly with even a lost propeller due to their redundant architecture [8], [9]. Such redundancies come at a significant cost. Safety pilots must be highly skilled, and according to current FAA and Transport Canada airspace regulations only one drone can be monitored and controlled by each pilot, making their use more expensive the longer the experiment is and the more vehicles are involved. The use of mechanical redundancies in the case of hexacopters and octocopters comes at a cost also: adding redundant propellers, which might not be needed, increasing the weight (and thus power requirements at the cost of flight time) and complexity of the vehicle.

Quadcopters are prone to crashes, especially when in autonomous flight. Wind gusts, pilot losing line of sight, pilot errors, loss of connection, and colliding with objects are all risks which can cause any UAV to lose stability and eventually crash which is hazardous to its surroundings [10]. The collision of quadcopters with obstacles is a major safety issue since many quadcopter uses require close proximity to buildings. Due to the fragility of the propellers, new quadcopter designs offer protection of their propellers with bumpers to prevent propeller damage and loss of flight in case of an accident. There is a lot of research covering obstacle avoidance but only a few researchers considered recovery from an impact with a solid wall [11]. In their paper, Dicker et al. [11] demonstrated that it is possible for a quadcopter with bumpers to recover safely from a nondestructive impact with a wall. When a quadcopter propeller impacts an object, there is a high chance it gets damaged and can not be operated anymore. The flight controller will fail since it

requires all four propellers to be operational, unless the flight controller has capabilities to recover stability and to continue the mission with the remaining three operational rotors. For instance, Freddi et al. [12] were able to re-stabilize a quadcopter following a complete motor failure where the strategy was to give up yaw control, so that the vehicle rotates freely around an axis which can be tilted for translational control. However, if a quadcopter impacts an object, the collision can cause moments and forces that will cause the quadcopter to become vertically flush with the wall and the lost propeller will add a layer of complexity to recover from the impact. To date, this type of failure has not been examined in the literature and a recovery strategy for such impacts is still to be developed.

1.2 Literature Review

This thesis touches upon various topics in quadcopter control-related research areas. The order of the literature review will be as follows: quadcopter dynamics and control; collision detection; collision recovery; fault-tolerant control strategies; and finally, the work where the recovery controller and the relaxed hover theory are presented which form the foundation of the research in this thesis.

1.2.1 Quadcopter Control

A standard quadcopter configuration has four uni-directional motors with propellers mounted on them which are placed in symmetrical formation at a distance away from the center of the vehicle. In a nominal hover attitude, these propellers generate thrust pointing upwards to counter gravity. Each pair of opposing propellers rotate in the same direction, opposite to the other pair [13]. There are two types of control formulation for quadcopters, which are commonly referred to as

the plus (+) and cross (X) configuration [14]. Different control theories and methods have been developed over the years, most notably: PID controllers [15] [16], LQR controllers [17] [18], back-stepping controllers [19] [20]. These controllers require accurate information about the position and attitude of the vehicle which is provided by measuring devices such as the GPS, gyroscope, accelerometer and other sensors [21].

1.2.2 Collision Modeling and Detection

A lot of existing research addresses the problem of obstacle avoidance to avoid collisions while very few papers cover collision detection and recovery. Tomic et al. [23] developed a model-based method for estimating an external wrench in flying robots due to a sustained low velocity impact. Gentili et al. [24] modeled the interaction of a UAV with a fixed vertical surface at a specified point using the linear Kevin-Voigt model. Galea et al. [25] designed a robust control algorithm to command a tethered quadrotor to fly to different stipple positions to make contact with a canvas using an ink soaked sponge. Vempati et al. [26] developed a control law to fly an autonomous UAV for spray painting on three-dimensional surfaces. In [25] and [26], the robot trajectory involves regular controlled collisions with a wall or flying in close proximity to a wall which present a high chance of mission failure due losing a propeller. Recent works at McGill's Aerospace Mechatronics Lab (AML) have addressed modelling collisions between a quadcopter and a vertical wall [27]. In their work, they developed a full contact model between a quadcopter platform with protective bumpers around its propellers and a wall. The model developed is the most general compared to other models covering quadcopters interaction with a stationary object. The contact model included contact

forces and moments that resulted from the impact force applied at the point of collision, that were not defined beforehand.

1.2.3 Attitude and Collision Recovery

Faessler et al. [28] developed a three-stage controller to reinitialize stable flight after an aggressive maneuver using only onboard sensors. In the first stage, the roll and pitch angles are stabilized at zero (hover orientation). Then the vertical velocity is stabilized using measurements from the barometer and the accelerometer sensors. Finally the horizontal velocity is brought to zero using a downward facing camera. The controller by Faessler et al. [28] formed the basis for the recovery control law developed by Dicker et al. [11] in McGill's Aerospace Mechatronics Lab (AML). The control law by Dicker et al. [11] along with the contact model of Chui et al. [27], which includes detection and characterization of a collision with a wall, formed the full recovery control strategy. It is important to note that the recovery control strategy developed by Dicker et al. applies to nondestructive collisions where the four propellers remain fully functional throughout the duration of the recovery period.

1.2.4 Actuator Failure Control Strategies

Freddi et al. [12] proposed the first fault-tolerant control for a quadcopter in case of an actuator failure. The goal of the fault tolerant control is to stabilize the attitude of the vehicle and make it reach the desired position in space. The fault tolerant control developed has a double loop structure, where the inner and faster loop controls the attitude and altitude of the vehicle, while the outer and slower loop controls the positions of the vehicle by modifying the desired values of the attitude angles. The

inner controller is based on the conservation of angular momentum around a vertical axis that the vehicle rotates around. When one of the actuators fails, the speed of the opposite actuator to the damaged one is adjusted such that the attitude angle controlled by this pair of rotors is zero. The quadcopter body will be parallel to the ground, spinning around the imaginary axis with a steady state rotational velocity that depends on the rotational drag. By changing the rotational speed of the healthy pair of motors, it is possible to achieve any position in the cartesian space. Therefore, once an actuator fails, the yaw control is sacrificed, and the quadcopter starts rotating around an axis. In this configuration the vehicle will be rotating around a vertical imaginary axis fixed to the body frame; to maintain the collective thrust in a desired direction, it is required to vary the thrust on each operational motor with a proper frequency proportional to the angular velocity of the vehicle. Simulation results they presented show the applicability of this fault tolerant strategy.

Lanzon et al. [29] expanded the work in [12], with a control law for the rotor failure scenario based on a model-based double control loop scheme. This model uses feedback which linearizes the system around a point where the roll and pitch angles are zero but with a constant yaw speed around the vehicle body frame z-axis. The closed-loop system relies on an H_∞ loop shaping technique to control the roll and pitch. The outer loop controls the translational displacement of the quadcopter with small angle approximations for the roll and pitch angles.

The controllers proposed by Freddi et al. [12] and Lanzon et al. [29] are challenging for the quadcopter motors and the flight computer. The two opposite healthy motors must be able to provide high thrust to accommodate the weight of the vehicle and must have high saturation levels. In addition, relying heavily on air drag to stabilize the yaw rate may not work for all vehicles, especially those with a low air drag profile.

Lippiello et al. developed a method to control a quadcopter after it loses one propeller [30]. The control law is based on the back-stepping control and requires turning off the motor opposite to the disabled one. This results in a bi-rotor that can reach any point in space but lacks the yaw control, as the one proposed earlier in [29], [12]. The back-stepping approach is used for translational movements along with a PID controller for angular control. Simulation results show the feasibility of this method. In [31], a quadcopter suffering from a lost propeller was transformed into a different system able to be controlled using only three propellers. The control law uses the generic PID controller used by the quadcopter, but redistributing the control effort to the three working propellers. However, to be able to control the tricopter a weight redistribution maneuver must be performed to shift the center of gravity of the vehicle towards the propeller opposing the faulty propeller. This is not necessarily feasible since performing a weight redistribution maneuver can increase the complexity of the vehicle, especially if the maneuver is to allow for redistributing the weight to any one of the four propellers.

Periodic solutions are used in [32] together with an LQR controller to control the quadrotor in case of single, two opposing, or three propellers failure. This will be discussed further in the next section.

1.2.5 Hover Solutions and Recovery Controller

Mueller et al. in [33] presented the novel relaxed hover condition. Here the requirement for having zero angular velocity is relaxed and instead the hover solutions are obtained such that the vehicle will be at an approximately constant position. These solutions have certain constant parameters over the hover period which means the vehicle may rotate at a constant velocity in hover. Given that the solutions are constant in the body frame the powerful techniques of time-invariant

system theory can be used for control design. In the paper, Mueller et al. discussed how to formulate and determine such relaxed hover solutions. This approach is used in [32] to compute hover solutions for a quadcopter experiencing up to three rotor failures [32]. In addition, this concept can be used to control a multicopter where the center of gravity of the vehicle is located far from the propellers' geometric center. Finally, a new type of vehicle, the spinner, was presented; here the propellers are placed in a rotationally symmetrical configuration and all rotate in the same direction. At hover these vehicles rotate at high angular yaw rates opposite to the direction of rotation of propellers.

In [32], Mueller et al. used the novel approach developed in [33] to present a control strategy for a quadcopter after losing up to three propellers, which forms the basis for the control law presented in this thesis. The strategy consists of the quadcopter spinning around a primary axis fixed with respect to the vehicle and by tilting this axis and varying the amount of produced thrust the vehicle's position can be controlled. The control strategy is made up of two cascaded controllers, a slow outer position controller that defines the desired acceleration and an inner faster loop that controls the attitude of the vehicle so that the desired acceleration is achieved.

1.3 Objectives

This thesis investigates a quadcopter recovery strategy following a rotor failure, which allows a quadcopter to recover from an impact with a wall despite the complete loss of one of its propellers. After such a failure occurs, the quadcopter can no longer maintain hover in the conventional sense, but needs to enter a new hover condition, where the vehicle's angular velocity is non-zero. This means the vehicle

will rapidly rotate about an axis, but it is still fully possible to control its position. The research assumes the wall with which the vehicle collides is flat, vertical and stationary and that coulomb friction exists between the quadcopter and the wall during impact. In addition, it is assumed that the impact with the wall occurs at one bumper only and that the corresponding actuator fails upon impact. It is assumed that all three remaining propellers are fully functional during and after the impact. This thesis will address these three main objectives:

1. The cascaded control strategy presented in [32] is applied on a custom vehicle equipped with protective bumpers and tested in simulation. The vehicle starts at a certain position, lose a propeller and be able to maintain its position and altitude and move to the desired position in space. A batch simulation will show a wide range of initial conditions from which the vehicle will be able to lose a propeller and recover into the hover solution.

2. Next, a recovery control strategy using the controller presented earlier is developed in simulation on the same custom quadcopter used in the first objective. The simulation uses the contact and dynamics model presented in [27] which includes detection of the collision with a wall and the contact forces due to the collision. A batch simulation is presented which shows a range of initial conditions for which the recovery controller succeeds in recovering from the impact with the wall using three propellers. These results are compared with the ones presented in [11] which is a recovery controller for a non-destructive collision.

3. Finally, a less robust velocity controller is developed that does not rely on position estimate (either from GPS or motion capture system like VICON) to

the problem of recovering from an impact with a wall after losing a propeller. The results are compared to the ones from the second objective.

1.4 Thesis Outline

The organization of the thesis is as follows. Chapter 2 covers the quadcopter dynamics and contact modeling used in the simulation. Chapter 3 presents the control law which will be referred to throughout this thesis and covers objective one. Chapter 4 presents and analyzes the simulation results of the controller presented in Chapter 3. Chapter 5 introduces the two developed recovery control strategies for a quadcopter recovering with only three propellers; as well as the results from a batch simulation of collisions and discusses in which range of initial conditions the quadcopter can and can not recover. Finally, Chapter 6 summarizes this thesis with several conclusions and potential improvements to the recovery controller.

Chapter 2

Quadcopter Dynamics and Contact Modelling

A quadcopter is made up of four motors on which propellers are fixed. These motors are mounted on an X-shaped frame, where usually the arms make a 90-degree angle with one another. The speed of rotation of the motors, which produces the lift force, can be separately controlled, therefore changing the attitude of the vehicle, which allows the quadcopter to move in 3D space.

The dynamics equations for the quadcopter presented here assume certain physical properties that are an approximation, which greatly simplifies the model. These particular assumptions are:

1. The quadcopter is a single rigid body that cannot be deformed. In case of a collision, the impact forces and moments get transmitted to the single rigid body, even though in reality the bumpers experience deformations.

2. The actuators and the motors are identical in geometry, mass and lift power.

3. Propellers are rigid.

2.1 Attitude Representation

A quadcopter's attitude can be described in many ways. The most popular and common is by using Euler angles because they separate the attitude into three separate rotations. The three rotations in order are yaw rotation, pitch rotation and roll rotation. This modeling works when the four propellers are pointing upwards which is the normal operating condition for the vehicle (i.e., regular hover condition). However, if 90-degree rotations occur (pitching or rolling up to 90 degrees), Euler angles fail which is known as gimbal lock [34]. This occurs when any two of the three Euler angle rotations describe a rotation about the same axis. For this reason, a quaternion system, with four elements, must be used to describe the quadcopter attitude in case the vehicle has a possibility of pitching or rolling to extreme angles.

2.2 Quadcopter Dynamics Nomenclature

The quadcopter vehicle is modeled using the convention shown in Figure 2.2, where two opposite motors spin in the same direction opposite to those adjacent to them.

Two frames are used to study the system motion. A frame integral with the earth which is the inertial frame, labeled $\mathcal{F}_I = \{\mathbf{e}_X\ \mathbf{e}_Y\ \mathbf{e}_Z\}$ is centered at an arbitrary point $\mathcal{O}_I$ and follows the east-north-up convection (ENU); a body-fixed frame $\mathcal{F}_Q = \{\mathbf{e}_x\ \mathbf{e}_y\ \mathbf{e}_z\}$ where $\mathcal{O}_Q$ is fixed to the center of mass of the quadcopter and follows the north-east-down convection (NED).

The body-fixed frame in this thesis is defined by the X-configuration which allows for more stable flight control [35]. As is shown in Figure 2.1, the difference

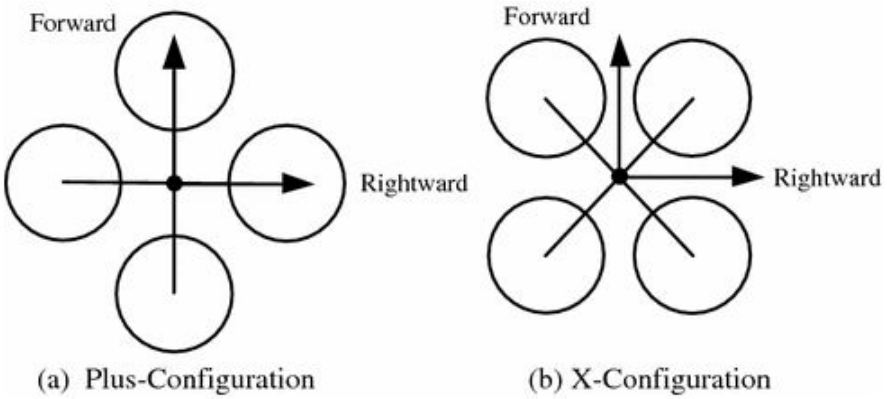


Figure 2.1: Plus and X-Configurations for a Quadcopter

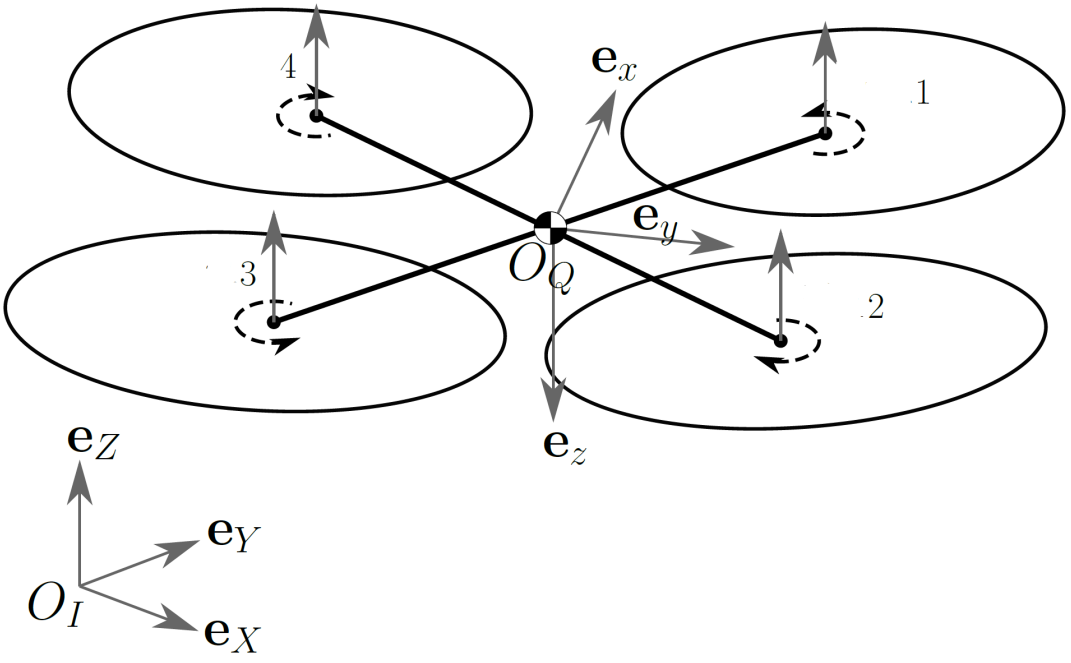


Figure 2.2: Inertial and body-fixed frames

between the two configuration is in which direction the forward flight is. In the X-configuration, the pitch and roll axes can be controlled by all the propellers; while, the plus configuration only has a pair of propellers to control each axis. Hence, the X configuration is more stable in movement and easier to fly intuitively.

Therefore, $\mathbf{e}_x$ is defined as pointing outwards from the front of the quadcopter bisecting the two front propeller arms, and $\mathbf{e}_y$ is set according to the right-hand rule, taking into account that $\mathbf{e}_z$ points downwards (see Figure 2.2).

The attitude of the quadcopter relative to $\mathcal{F}_I$ in quaternion form is given as $\boldsymbol{q} = [q_w \; q_x \; q_y \; q_z]^T$ where the subscripts represent the elements of the quaternion. Although Euler angles, which are denoted by Greek letters $[\phi \; \theta \; \psi]$ for roll, pitch, and yaw respectively, are not directly used because of the gimbal lock issue, they are used in the analysis section to intuitively understand the vehicle's performance. They can be computed from the quaternion form of the attitude using:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \tan^{-1}(2(q_w q_x + q_y q_z)/(1 - 2({q_x}^2 + {q_y}^2))) \\ \sin^{-1}(2(q_w q_y - q_z q_x) \\ \tan^{-1}(2(q_w q_z + q_x q_y)/(1 - 2({q_y}^2 + {q_z}^2))) \end{bmatrix} \tag{2.1}$$

To compute the Euler angle rates from the angular velocity of the body, defined in the body frame $\mathcal{F}_Q$ as $\boldsymbol{\omega} = [p \; q \; r]^T$, a transformation as shown below can be used.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\cos\phi\sin\theta \\ 0 & 1 & \sin\theta \\ \sin\theta & 0 & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \tag{2.2}$$

The position of the quadcopter in $\mathcal{F}_I$ is defined as $\boldsymbol{p} = [X \; Y \; Z]^T$, its velocity in the body frame $\mathcal{F}_Q$ as $\boldsymbol{v} = [u \; v \; w]^T$, its velocity in the inertial frame $\mathcal{F}_I$ as $\dot{\boldsymbol{p}} = [\dot{X} \; \dot{Y} \; \dot{Z}]^T$, its acceleration in the inertial frame as $\ddot{\boldsymbol{p}} = [\ddot{X} \; \ddot{Y} \; \ddot{Z}]^T$.

2.3 Equations of Motion

The quadcopter rotational and translational dynamics are modeled using the Newton-Euler formulation for a single rigid body expressed in the body frame $\mathcal{F}_Q$. The equations presented here are used in the simulations presented in Chapter 4 and 5 to validate the control law developed in Chapter 3.

$$m\dot{\boldsymbol{v}} + m\boldsymbol{\omega} \times \boldsymbol{v} = \boldsymbol{F_G} + \boldsymbol{F_T} + \boldsymbol{F_C} \tag{2.3}$$

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + (\sum_{j=1}^{4} \boldsymbol{r_{Tj}} \times \boldsymbol{F_{Tj}}) + \boldsymbol{M_T} + \boldsymbol{M_\Omega} + \boldsymbol{M_C} + \boldsymbol{M_A} \tag{2.4}$$

The parameters m, $\boldsymbol{I}$, and $\boldsymbol{r}_{T_j}$ represent, respectively, the mass, moment of inertia matrix the center of mass, and the position of the motor j with respect to the center of mass. The cross-product operator is represented by $\times$.

The pose kinematics and quadcopter velocities vary as follow:

$$\dot{\boldsymbol{p}} = \boldsymbol{q} \odot \boldsymbol{v} \tag{2.5}$$

$$\dot{\boldsymbol{q}} = -\frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \otimes \boldsymbol{q} \tag{2.6}$$

Vectors can be rotated between the two frames $\mathcal{F}_Q$ and $\mathcal{F}_I$ by the quaternion rotation operator $\odot$. The reverse rotation from $\mathcal{F}_I$ to $\mathcal{F}_Q$ is represented by an inverse quaternion rotation (i.e. $\boldsymbol{v} = \boldsymbol{q}^{-1} \odot \dot{\boldsymbol{p}}$). The $\otimes$ operator in Equation (2.6) refers to the quaternion multiplication opertor.

The forces $\boldsymbol{F}$ and moments $\boldsymbol{M}$ in (2.3) and (2.4) are referred to by the subscripts G, T, Ω, A and C respectively to represent the gravitational, thrust, gyroscopic, aerodynamic and contact forces and moments. In the following two sections, we will define each one of these forces and moments.

2.3.1 Standard Forces on a Quadcopter

The gravitational force in $\mathcal{F}_Q$ is:

$$\boldsymbol{F_G} = \boldsymbol{q^{-1}} \odot m\boldsymbol{g} \tag{2.7}$$

where $\boldsymbol{g} = [0 \quad 0 - g]$ is the gravity vector in $\mathcal{F}_I$ and g is the acceleration due to gravity.

The thrust force is expressed as:

$$\boldsymbol{F_T} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{4}} \boldsymbol{F_{Tj}} = \sum_{j=1}^{4} \begin{bmatrix} 0 \\ 0 \\ -k_t \Omega_j^2 \end{bmatrix} \tag{2.8}$$

Here k_t represents the propeller thrust coefficient which relates the propeller thrust $\boldsymbol{F_j}$ to the angular velocity Ω_j squared. The thrust moment is expressed as:

$$\boldsymbol{M_T} = \begin{bmatrix} 0 \\ 0 \\ k_{torq} \sum_{j=1}^{4} (-1)^j \Omega_j^2 - J_r \sum_{j=1}^{4} (-1)^j \dot{\Omega}_j^2 \end{bmatrix} \tag{2.9}$$

where k_{torq} represents the drag torque coefficient relating the drag torque generated by the rotation of propeller j to Ω_j. The second term in (2.9) represents the moment due to the change of propeller angular momentum, which uses the propeller moment of inertia around its rotational axis J_r.

The gyroscopic moment is expressed as:

$$\boldsymbol{M}_{\boldsymbol{\Omega}} = \begin{bmatrix} -qJ_r \sum_{j=1}^{4} (-1)^j \Omega_j \\ pJ_r \sum_{j=1}^{4} (-1)^j \Omega_j \\ 0 \end{bmatrix} \tag{2.10}$$

The $(-1)^j$ term in the moment equations (2.9) and (2.10) is there to account for the rotation direction of each pair of propellers: the propeller pair $(1,3)$ rotates in one direction opposite to that of the propeller pair $(2,4)$ (see Figure 2.2). Note that all propellers generate thrust upwards when in regular hover condition but rotate in opposite directions to maintain moment equilibrium about the yaw axis..

In the present model, the aerodynamic drag of the whole vehicle is assumed to act only opposite to the yaw rate r with a drag coefficient k_d. The aerodynamic drag is expressed as:

$$\boldsymbol{M}_{\boldsymbol{A}} = \begin{bmatrix} 0 \\ 0 \\ -k_d r \end{bmatrix} \tag{2.11}$$

The drag moment is assumed to be linear in the yaw rate axis. This is justified in [21] by the asymmetric relative air velocity over the advancing and retreating propeller blades. Other aerodynamic forces were omitted from the model since they are insignificant in an indoor environment.

2.3.2 Contact Forces

The contact forces used in this dynamics model were developed previously in [27]. The effect of impact on a single bumper results in a normal and a friction force at the contact point. The vertical wall is defined by $\mathbf{e}_Z, \mathbf{e}_T$ and $\mathbf{e}_N$ as shown in Figure 2.3. The wall normal vector $\mathbf{e}_N$ is the unit vector normal to the wall, pointing

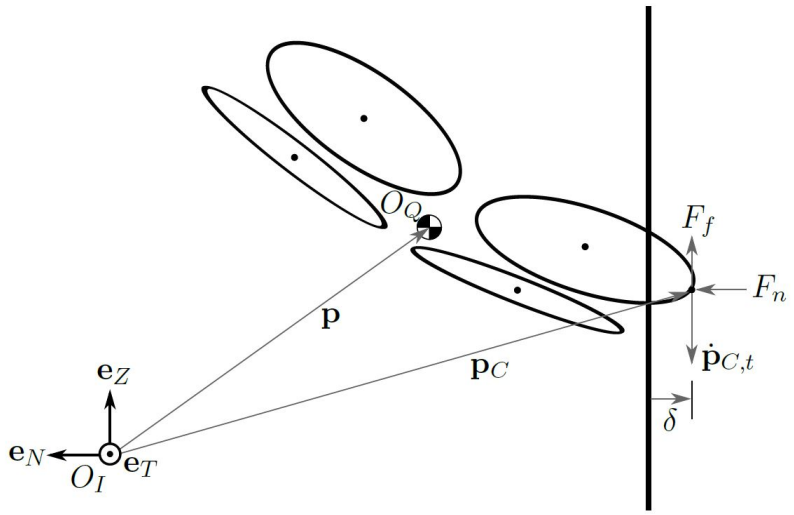


Figure 2.3: Contact force model at one bumper impact as derived in [27]

outwards from the wall to the open space. A contact occurs between a quadcopter and a vertical wall at a point $\mathbf{p}_C$. The non-linear compliant model first introduced by Hunt and Crossley [22] is used to explicitly model the normal force F_n which is directed in the direction of $\boldsymbol{e_N}$ applied at the contact point $\mathbf{p}_C$ and is expressed as:

$$F_n = \lambda\delta\dot{\delta} + k\delta \tag{2.12}$$

Here, δ stands for deformation (a.k.a., penetration) of the bumper at $\mathbf{p}_C$, k the coefficient of stiffness, and λ the damping coefficient.

The friction force F_f is applied at the point of contact $\mathbf{p}_C$ in the direction opposite to $\dot{\mathbf{p}}_{C,t}$, which is component of the velocity of $\mathbf{p}_C$ projected on the wall. This projection is computed as:

$$\dot{\mathbf{p}}_{C,t} = (\dot{\mathbf{p}}_C^{\mathbf{T}}\mathbf{e}_T)\mathbf{e}_T + (\dot{\mathbf{p}}_C^{\mathbf{T}}\mathbf{e}_Z)\mathbf{e}_Z \tag{2.13}$$

The friction force F_f is modelled with the modified Coulomb friction model is presented below:

$$F_f = \mu F_n \tag{2.14}$$

In the above, μ stands for the coefficient of friction defined as:

$$\mu = \begin{cases} \frac{\mu_c}{v_{th}} \parallel \dot{\mathbf{p}}_{C,t} \parallel & \text{if } \parallel \dot{\mathbf{p}}_{C,t} \parallel \leq v_{th} \\ \mu_c & otherwise \end{cases} \tag{2.15}$$

where v_{th} is the threshold velocity of $\dot{\mathbf{p}}_{C,t}$ after which μ remains constant, and μ_c is a constant Coulomb friction coefficient for sliding.

With the normal and friction forces presented in Equation (2.14) and Equation (2.12), a full contact model can be presented for a bumper impact with a wall:

$$\boldsymbol{F_C} = \boldsymbol{q^{-1}} \odot (F_n \mathbf{e}_N - F_f \frac{\dot{\mathbf{p}}_{C,t}}{\|\dot{\mathbf{p}}_{C,t}\|}) \tag{2.16}$$

$$\boldsymbol{M_C} = \boldsymbol{r_C} \times \boldsymbol{F_C} \tag{2.17}$$

where $\boldsymbol{r_C}$ represents the position of contact point relative to $\mathcal{O}_Q$.

2.4 Reduced Attitude Kinematics

A quadcopter has a total of six degrees of freedom between the Euclidean linear space and rotation space. On the other hand, a quadcopter has only four control inputs which correspond to the four motor commands. This makes the quadcopter an under-actuated system where it is impossible to control all six degrees of freedom simultaneously. Therefore, only four degrees of freedom can be controlled by the four control inputs and usually, these are the quadcopter Euler angels $[\phi\ \theta\ \psi]$

and the altitude z. Therefore, a typical quadcopter controller allows the control of the quadcopter full attitude to some desired attitude.

In the event of a propeller failure, which is the focus of the present thesis, the quadcopter loses one control input and hence can only control three degrees of freedom. Furthermore, the propeller failure implies the loss of controllability of one degree of freedom among roll, pitch, yaw and altitude. The most important variables to control are the roll, pitch, and altitude. Regulation of the roll and pitch angles is necessary in order for the vehicle to maintain its stability, in addition these angles affect the longitudinal and lateral displacements in the inertial Earth frame. On the other hand, altitude is a critical variable to control since it must always be kept a minimum positive value to avoid a collision with the ground. The loss of yaw control however, implies not being able to affect the heading of the quadcopter which is not of vital importance for quadcopter recovering from a propeller loss.

The strategy adopted in this thesis is to give up control of the full attitude of the vehicle, once a propeller has failed, and instead only control the pitch and roll of the quadcopter's attitude. This is referred to as reduced attitude [36]. In Chapter 3, the controller for a propeller loss case will be presented.

Chapter 3

Quadcopter Control with One Actuator Failure

This chapter focuses on presenting the control law that can recover and fly a quadcopter with only three propellers without a collision. First, the relaxed hover concept is presented, followed by solving for the equilibrium periodic solution for our custom vehicle NAVI. A control law is presented based on the control law in [32], that attempts to fly and control a quadcopter after losing a propeller. This control law is a cascaded system of a slower outer position controller that tracks a certain position in space and a faster inner reduced attitude controller that tracks a reference attitude.

3.1 Hover Condition

3.1.1 Conventional Hover

A typical quadcopter has four propellers that produce thrust in the same direction. The propellers are made to rotate in opposite directions and arranged such that the torques they produce can be made to sum to zero, while the propeller thrust supports the weight of the vehicle. Therefore, in a conventional hover condition, a

vehicle is at the same position in space with zero acceleration, zero angular velocity, and the total thrust vector pointing opposite to gravity. For a symmetrical quadcopter in conventional hover, this means each propeller produces a quarter of the total thrust required to balance gravity. The propellers' angular momenta also sum to zero which leads to the vehicle's total angular momentum to be zero in hover.

3.1.2 Relaxed Hover

Relaxed hover refers to a flight condition where we relax the constraint on the conventional hover conditions. Here, the vehicle remains *near* the same point in space, but the vehicle has non-zero angular velocity. The translational acceleration is non-zero as well; but, it must average out to zero along the periodic solution, in order for the vehicle to remain substantially at the same location. The solutions that will be considered are constant when described in the body-fixed frame; in addition, the angular velocity vector of the vehicle must be constant.

When a propeller is lost, the total forces and torques acting on the quadcopter will not be zero, and therefore static moment equilibrium will not be achieved around the center of mass. This prevents the vehicle from hovering in the conventional way, but it can settle in a relaxed hover condition. In this relaxed hover, the reduced attitude of the vehicle will be constant. In the next section, the periodic hover solution for our vehicle will be derived.

3.2 Equilibrium Solution

This section presents the equilibrium periodic solution that will be used for the position and attitude controller for a vehicle with a disabled propeller. An overbar refers to values that are constant along the periodic hover solution.

The goal is to find a periodic solution where a primary axis $\overline{\boldsymbol{n}}$ is fixed with respect to the body fixed frame. The evolution of this primary axis $\overline{\boldsymbol{n}}$ in the inertial frame is governed by the following differential equation

$$\dot{\boldsymbol{n}} = -\boldsymbol{\omega} \times \boldsymbol{n} \tag{3.1}$$

In the relaxed hover condition, the vehicle will be in a periodic state where it is rotating around this imaginary axis $\overline{\boldsymbol{n}}$ with a constant angular velocity $\overline{\boldsymbol{\omega}}$. The primary axis is fixed with respect to the body frame and is defined as $\overline{\boldsymbol{n}} = [\overline{n}_x, \overline{n}_y, \overline{n}_z]^T$; since the axis is constant along the periodic solution, we have that

$$\dot{\boldsymbol{n}} = 0 \tag{3.2}$$

and from the properties of the cross product, it follows from equation (3.1) that $\overline{\boldsymbol{n}}$ has to be parallel to the angular velocity of the body $\overline{\boldsymbol{\omega}}$ such that

$$\overline{\boldsymbol{n}} = \epsilon \overline{\boldsymbol{\omega}} \tag{3.3}$$

Since $\overline{\boldsymbol{n}}$ is a unit vector, the norm constraint is expressed as

$$\| \overline{\boldsymbol{n}} \| = \| \epsilon \overline{\boldsymbol{\omega}} \| = 1 \tag{3.4}$$

The primary axis $\overline{\boldsymbol{n}}$ is chosen to point opposite to gravity along the periodic solution. The period of the periodic hover solution can be obtained from

$$T_{hover} = \frac{2\pi}{\| \overline{\boldsymbol{\omega}} \|} \tag{3.5}$$

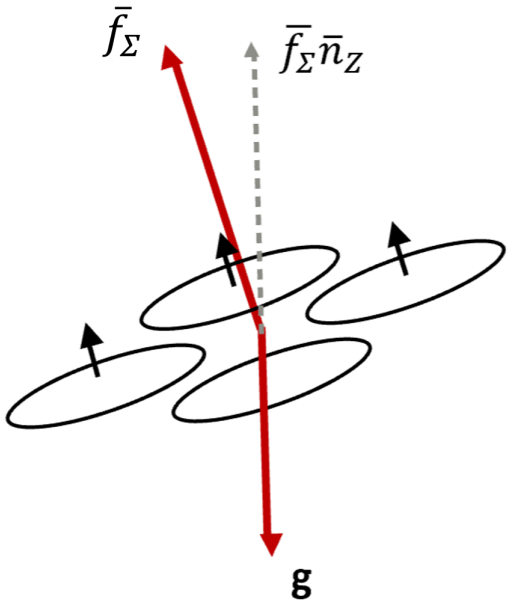


Figure 3.1: A quadcopter with a lost propeller where the total thrust vector is not pointing opposite to gravity, but has a component in the horizontal direction

Since the vehicle must not accelerate in the gravity direction, the fraction of the total thrust force $\overline{f}_\Sigma = \sum_{j=1}^{4} \overline{f}_j$ pointing opposite to gravity is $\overline{f}_\Sigma \overline{n}_z$, such that

$$\overline{f}_\Sigma = \frac{m \parallel \boldsymbol{g} \parallel}{\overline{n}_z} \tag{3.6}$$

where $\overline{n}_z$ is the z-component of the primary axis.

An interpretation of equation (3.6) is illustrated in figure (3.1) which shows that the fraction of the total thrust required by the vehicle during the hover condition that is pointing opposite to gravity must equal $\overline{n}_z$.

3.3 Vehicle Trajectory

If $\overline{n}_z < 1$, then from equation (3.6) the total thrust force is not pointing in the gravity direction but has a component perpendicular to it. This portion of the thrust

is directed in the horizontal direction which will result in an acceleration in that direction. The horizontal component $\overline{a}_{horiz}$ of this acceleration can be calculated from:

$$(m\overline{a}_{horiz})^2 + (m \parallel \boldsymbol{g} \parallel)^2 = (\overline{f}_{\sum})^2 \tag{3.7}$$

This horizontal component of the force will make the vehicle move in a circular horizontal trajectory of radius R_{circle}. To solve for this radius R_{circle} the centripetal acceleration equation $\parallel \boldsymbol{a}_{cent} \parallel = \parallel \boldsymbol{\omega} \parallel^2 R_{circle}$ will be utilized and by replacing $\overline{f}_{\sum}$ in equation (3.7) by the expression found in equation (3.6) and solving for R_{circle}, we obtain:

$$R_{circle} = \frac{\parallel \boldsymbol{g} \parallel}{\parallel \boldsymbol{\omega} \parallel^2} \frac{\sqrt{1 - \overline{n}_z{}^2}}{\overline{n}_z} \tag{3.8}$$

Therefore the vehicle will move along a circular horizontal trajectory of radius R_{circle} given by equation (3.8) in the hover solution.

3.4 Solution for Hover Equilibrium

Without loss of generality, we assume the fourth propeller (4 in Figure 2.2) has failed which means

$$\overline{f}_4 = 0 \tag{3.9}$$

Therefore the periodic hover solution is defined by the following 10 unknowns

$$\overline{n}_x, \overline{n}_y, \overline{n}_z, \overline{p}, \overline{q}, \overline{r}, \epsilon, \overline{f}_1, \overline{f}_2, \overline{f}_3$$

where $\overline{p}, \overline{q}, \overline{r}$ refer to the constant periodic components of the angular velocity of the body $\overline{\omega}$.

3.4.1 Simplifying Assumptions

Since the vehicle is one propeller deficient, the three remaining propellers must generate the thrust required without causing the vehicle to flip. One solution is to have the two opposing working propellers generate equal thrust to prevent the vehicle from having instabilities or a flipping moment along that direction as well. Therefore, a new constraint is introduced such that

$$\overline{f}_1 = \overline{f}_3 \tag{3.10}$$

The final constraint is on the force ratio between $\overline{f}_2$ and $\overline{f}_1$, where a tuning factor ρ is introduced such that

$$\rho = \overline{f}_2 / \overline{f}_1 \tag{3.11}$$

where ρ can be used to generate a large range of possible values for the equilibrium solution.

An optimal ρ value can be chosen based on the application. For example, an optimal hover solution can be found either to minimize the power required by the vehicle, minimize the rotation of the quadcopter in hover, or maximize the difference between the propeller thrust force generated in hover and the limits of the propellers.

In the case of a quadcopter recovering after impacting a wall and losing a propeller, the top priority is to minimize the rotation of the quadcopter and be at a safe threshold from the propeller maximum limits to allow for good maneuverability. This is important because after the quadcopter impacts a wall, it has to stabilize to the hover solution as fast as possible and be able to move away from the wall. In addition, having the equilibrium thrusts of the propellers at a safe margin from the maximum limits helps in recovering and moving the vehicle away from wall, rather

than saturating the propellers which may lead to the loss of lift and crash towards the wall.

Therefore the 10 unknowns presented earlier in this section that define the equilibrium solution can be computed by solving the 9 equations (3.3), (3.4), (3.10), (3.11) as well as equation (2.4) with the angular acceleration $\dot{\boldsymbol{\omega}}$ set to zero. These equations are presented below:

$$\overline{\boldsymbol{n}} = \epsilon\overline{\boldsymbol{\omega}} \tag{3.12}$$

$$\| \overline{\boldsymbol{n}} \| = \| \epsilon\overline{\boldsymbol{\omega}} \| = 1 \tag{3.13}$$

$$\overline{f}_1 = \overline{f}_3 \tag{3.14}$$

$$\rho = \overline{f}_2 / \overline{f}_1 \tag{3.15}$$

$$-\overline{\boldsymbol{\omega}} \times \boldsymbol{I}\overline{\boldsymbol{\omega}} + (\sum_{\boldsymbol{j=1}}^{\boldsymbol{3}} \boldsymbol{r_{Tj}} \times \overline{\boldsymbol{f_{T_j}}}) + \boldsymbol{M_T} + \boldsymbol{M_\Omega} + \boldsymbol{M_A} = \boldsymbol{0} \tag{3.16}$$

Note that equations (3.12), and (3.16) each represent 3 scalar equations which makes for a total of 9 equations. Here ρ will be an additional variable that will be studied further in the next section.

3.4.2 Force Ratio Analysis

The equilibrium solutions of our custom quadcopter NAVI for different values of ρ, which represents the force ratio of the two adjacent healthy propellers, are shown in Figure 3.2.

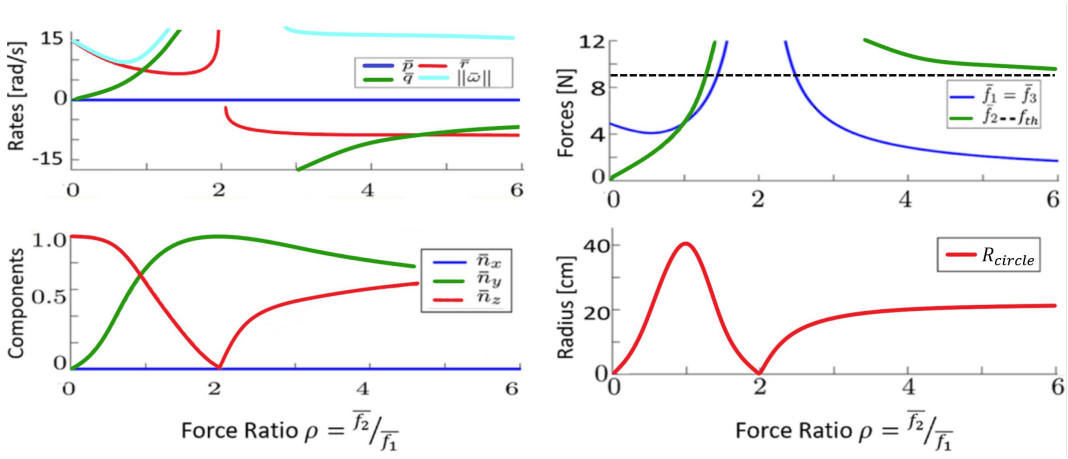


Figure 3.2: The hover solution variation as a function of the force ratio $\rho = \bar{f}_2/\bar{f}_1$. The plots show the angular velocity, primary axis components, propeller forces with the motor thresholds, and the radius of the circular trajectory as a function of the force ratio

At a force ratio of zero, f_2, the propeller facing the one that was damaged is shut off. Therefore only two of the vehicle's propellers are producing thrust. For this solution, the angular velocity of the vehicle is maximum at a value of 15 rad/s and the vehicle's primary axis has only a component in the z-direction (i.e., n_z). From equation (3.6), for $n_z = 1$ all the thrust of the vehicle is pointing opposite to gravity and the vehicle will have no force in the horizontal direction, hence the radius of the circular trajectory is zero as shown in the fourth plot in figure (3.2). As the force ratio increases from zero, which means the second propeller is producing a thrust, the vehicle primary axis moves away from the body z-axis and n_z decreases while the n_y component increases. The pitch rate starts increasing as the thrust from the second propeller increases. A minimum for the magnitude of the angular velocity of the vehicle occurs at around 9.5 rad/s at a force ratio of 0.55 as well as a minimum of the propeller thrusts f_1 and f_3 occurs at a force ratio of 0.53. A discontinuity occurs at a force ratio of 2, where the vehicle is standing upright in a vertical position (i.e. pitched 90°) which can be seen in the components of primary axis, that is $n_y = 1$ while $n_z = 0$. As ρ increases away from the discontinuity, the yaw

rotation of the vehicle switches direction as can be confirmed by the negative yaw rate r in the first plot. An important consideration to keep in mind is the maximum thrust of the propellers. The required thrust for some of the equilibrium solutions is not achievable, which is shown by the dashed line in the third plot. All the equilibrium solutions beyond the value of approximately $\rho = 1.2$ are not possible due to the limitation on the thrusts produced. Therefore we are confined to solutions corresponding to $\rho = [0 - 1.2]$ interval.

Since the minima of the angular velocity and of the force f_1 occur in close proximity, a value of force ratio ρ = 0.51 which lies in between the two minima was chosen to be the equilibrium solution. This gives enough difference between the propeller thrusts in hover and the maximum thrust limits for the vehicle to be able to maneuver and converge to the solution. In addition, it is preferable to choose a low angular velocity for the vehicle in the hover solution, which helps the vehicle to converge faster to the hover solution, recover faster after losing the propeller, and be more stable during the periodic solution. It is interesting to incorporate several equilibrium solutions in the recovery, for instance during landing it would be optimal to transition to a solution with a zero radius of circular trajectory (which is the solution at $\rho = 0$). For simplicity only one solution will be used in the recovery.

Therefore, the equilibrium solution that will be used in this thesis is defined by

$$\overline{\boldsymbol{n}} = (0, 0.14, 0.99)^T$$

$$\overline{\boldsymbol{f}} = (4.2, 2.1, 4.2, 0)^T$$

$$\overline{\boldsymbol{\omega}} = (0, 1.4, 10)^T$$

3.5 Control Law

The control law presented in this thesis from [32] incorporates a cascaded control strategy with a faster outer position module and a slower inner module that regulates the reduced attitude of the vehicle. Figure (3.3) illustrates the block diagram of this controller. The goal is to converge to the periodic hover solution, to be able to maintain the relaxed hover condition, and to control the vehicle's position and reduced attitude. The following section will go over each block of the cascaded controller, starting with the position controller and followed by the reduced attitude controller.

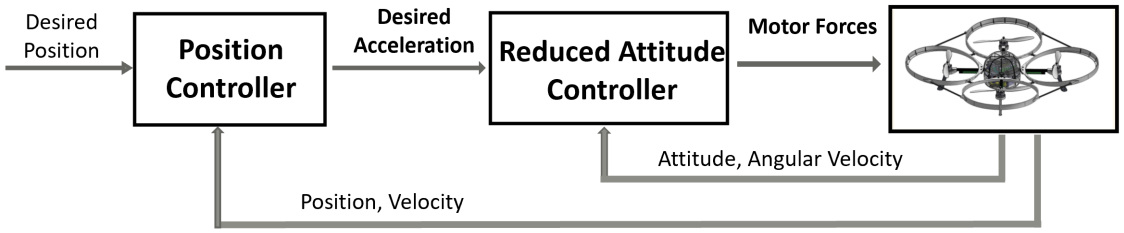


Figure 3.3: Block diagram showing the cascaded control design used in this thesis. An outer controller regulates the positions and a faster inner controller adjusts the reduced attitude. The inner controller computes the required thrust by each propeller.

3.5.1 Position Controller

The deviation of the center of mass of the quadcopter from a desired point is space in the inertial frame will be referred to as $\boldsymbol{d}$. This deviation $\boldsymbol{d}$ can be considered as the error in the position of the vehicle, i.e., the difference between the current position and the desired position. Similarly, the deviation in the velocity of the vehicle can be expressed as $\dot{\boldsymbol{d}}$. The goal of the position controller is to make this deviation $\boldsymbol{d}$ behave as a 2^{nd} order system as defined by the following dynamics:

$$\ddot{\boldsymbol{d}}_{des} + 2\xi\omega_n\dot{\boldsymbol{d}} + {\omega_n}^2\boldsymbol{d} = 0 \tag{3.17}$$

where the damping ratio and the natural frequency are chosen such that the position controller is slower than the inner control loop.

As described earlier, the body fixed vector $\boldsymbol{n}$ is parallel to the angular velocity of the body and is directed to point upwards; $\boldsymbol{n}$ can be thought of as the thrust direction of the vehicle averaged over one rotation of the vehicle. Therefore, by rotating and controlling $\boldsymbol{n}$, it is possible to control the acceleration and the position of the vehicle.

A new vector $\boldsymbol{n_{des}}(t)$ is introduced which defines the desired acceleration direction. The desired acceleration denoted as $\ddot{\boldsymbol{d}}_{\boldsymbol{des}}$ from equation (3.17) can be used to solve for the total thrust force $f_{\sum}(t)$ and the desired primary axis $\boldsymbol{n_{des}}(t)$ using the following equations:

$$f_{\sum}(t) = \frac{m}{\overline{n}_z} \parallel (\ddot{\boldsymbol{d}}_{\boldsymbol{des}}(\boldsymbol{t}) - \boldsymbol{g}) \parallel \tag{3.18}$$

$$\boldsymbol{n_{des}}(t)\overline{n}_z f_{\sum}(t) = m\boldsymbol{q}^{-\mathbf{1}} \odot (\ddot{\boldsymbol{d}}_{\boldsymbol{des}}(\boldsymbol{t}) - \boldsymbol{g}) \tag{3.19}$$

In equation (3.18), after calculating $\ddot{\boldsymbol{d}}_{\boldsymbol{des}}$ and compensating for gravity, the desired total thrust force $f_{\sum}$ can be calculated. Similarly in equation (3.19), the desired total acceleration is rotated to the body frame using the inverse quaternion rotation, from which the desired primary axis $\boldsymbol{n_{des}}$ can be computed after substituting $f_{\sum}$ that was just computed. Note that $\overline{n}_z$ is a constant computed as part of the equilibrium. The relationship between relevant accelerations and direction of $\boldsymbol{n}_{des}$ is illustrated in Figure 3.4.

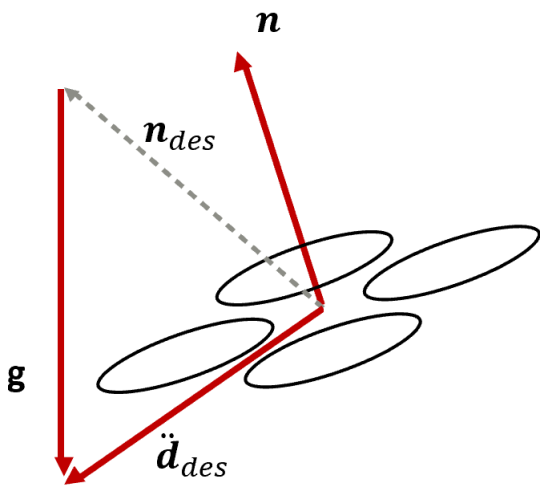


Figure 3.4: A diagram showing the relationship between the vectors used in the position control. The desired acceleration computed by the position controller from equation (3.17) followed by computing the desired primary axis from equation (3.19).

The goal of the reduced attitude controller is to make the body primary axis $\boldsymbol{n}$ align with the desired primary axis $\boldsymbol{n_{des}}$ while producing a total thrust $f_{\sum}$. This will be expanded on further in the next section.

3.5.2 Reduced Attitude Controller

The quadcopter reduced attitude has to be controllable near the equilibrium hover solution. This is achieved by linearizing the system near the equilibrium solution by using the time invariant nature of the hover solution. Given that the total force $f_{\sum}$ can be chosen, and the direction of the primary axis $\boldsymbol{n}$ can be specified, the vehicles acceleration can be controlled and hence also its position.

The vehicle's reduced attitude can be described by the state vector $\boldsymbol{x} = (p, q, n_x, n_y)$. Since the outer controller computes the desired acceleration based on the desired position, the reduced attitude controller will exploit the attitude deviation from the

equilibrium hover point. In other words the goal of the attitude controller is to drive the primary axis of the vehicle $\boldsymbol{n}$ to $\overline{\boldsymbol{n}}$ and $\dot{\boldsymbol{n}}$ to zero. This is expressed by the state deviation

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} - \overline{\boldsymbol{x}} \tag{3.20}$$

where $\boldsymbol{x}$ is the current state and $\overline{\boldsymbol{x}}$ represents the periodic hover solution derived in Section (3.4.2).

Linearising about the equilibrium solution yields the following system in state-space form:

$$\dot{\widetilde{\boldsymbol{x}}} = \boldsymbol{A}\widetilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} \tag{3.21}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & \overline{K} & 0 & 0 \\ -\overline{K} & 0 & 0 & 0 \\ 0 & -\overline{n}_z & 0 & \overline{r} \\ \overline{n}_z & 0 & -\overline{r} & 0 \end{bmatrix}_{x=\overline{x}}, \boldsymbol{B} = \frac{l}{I_{xx}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{x=\overline{x}} \tag{3.22}$$

In the above, l is the distance from the center of mass of the vehicle to each propeller arm, I_{xx} the first component of the inertia matrix of the quadcopter about its center of mass, $\overline{r}$ is the yaw rate equilibrium solution solved for in Section (3.4.2), and $\overline{K}$ is a constant that emerges from the linearisation of the system defined as follows:

$$\overline{K} = \frac{I_{xx} - J_r}{I_{xx}}\overline{r} - \frac{J_r}{I_{xx}}(\overline{\omega}_1 + \overline{\omega}_2 + \overline{\omega}_3) \tag{3.23}$$

where J_r is the moment of inertia of the propeller around its rotational axis.

Since the total thrust is computed at each simulation time step, one of the three input variables is used to control the altitude; the 3 functional propellers have only 2 free input variables which can be used to control the attitude. These form the input vector $\boldsymbol{u}$, where $\boldsymbol{u} = (u_1, u_2)$ which represents the deviations of the propeller

forces from the equilibrium hover solution propeller forces and is expressed with the following equation:

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} (f_3 - \overline{f}_3) - (f_1 - \overline{f}_1) \\ (f_2 - \overline{f}_2) \end{bmatrix} \tag{3.24}$$

Given the linear system of equation (3.21), a linear-quadratic regulator (LQR) controller is used to solve for the control input. In the present controller design, the LQR cost matrix $\boldsymbol{Q}$ and input matrix $\boldsymbol{R}$ are defined as follows:

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix} \tag{3.25}$$

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.26}$$

where the cost value on the deviation from the primary axis was set to 20 and the cost value on the angular rates set to 1 $s^2 rad^{-2}$. These values were optimized with the goal to minimize the vehicle's drift from the primary axis overall.

The feedback control law that minimizes the value of the cost can be defined as follows:

$$\boldsymbol{u} = -\boldsymbol{K}_{LQR}\boldsymbol{x} \tag{3.27}$$

where K_{LQR} is computed by the 'LQR' MATLAB function using the linearized system defined in Equation 3.22, the cost and the input matrices. Thus, the three propeller thursts f_1, f_2, f_3 are computed using equation (3.24) combined with the total

thrust constraint equation:

$$f_{\sum} = f_1 + f_2 + f_3 \tag{3.28}$$

Using the inner and outer controller, the vehicle can be controlled to converge to the hover solution. In the following chapter, simulation results are presented to test performance and the robustness of the controller presented here to converge to the hover solution from different initial conditions.

Chapter 4

Quadcopter Simulation for One Propeller Failure

Before applying the control theory presented in Chapter 3 to collision recovery of a vehicle with a failed propeller, it is first tested in a simulation environment for stabilizing the vehicle from a wide range of initial conditions. The motivation for testing the reduced attitude controller alone for basic stabilization, without the additional complexity of collision with an obstacle, is to validate and thoroughly test the control law and understand its limitations and strengths.

The reduced attitude control law was coded into the quadcopter dynamics simulator described in Chapter 2. Batch simulations were performed to ensure that the control law is capable of recovering the vehicle to a reduced hover state from a wide range of initial orientations. The model employed in the dynamics simulator represents the custom propeller protected quadcopter NAVI. The following sections will describe the simulation environment and the various analyses conducted to test the controller.

4.1 Quadcopter Dynamics Simulator

Simulations were performed in MATLAB using the standard MATLAB libraries along with MATLAB Aerospace Toolbox. The simulation environment was programmed using the MATLAB structure data type to log variables, to manage, and to access them efficiently.

The dynamics model used was the one presented in Chapter 2, where the most essential dynamics effects were modeled while omitting others to simplify the model. Most notably, the terms due to yaw drag, propeller acceleration and propeller saturation were modeled. Also, the thrust and drag generated by the propellers were calculated as a quadratic function of the motor speeds. All other aerodynamic effects caused by the propeller interactions were neglected.

The simulator used ODE45 to propagate the state with a time step of 200 Hz to match that of NAVI's flight controller. At each time step, the complete state of the quadcopter and the controller output were updated and stored in different data structures. Four Matlab 'struct' data types were used to store these values: Pose, Twist, Propstate, and Control. Pose and Twist were updated directly from the state of the quadcopter which is defined as

$$\boldsymbol{x} = [u\ v\ w\ p\ q\ r\ \dot{X}\ \dot{Y}\ \dot{Z}\ q_w\ q_x\ q_y\ q_z]^T \tag{4.1}$$

The vehicle parameters used in the simulation are presented in Table 4.1 below. The moments of the inertia, as well as the geometric parameters, were obtained from a detailed computer-aided design model of Navi. The DJI propeller datasheet was used to obtain the propeller moment of inertia about its rotation axis for the 8-inch propellers defined as J_r. The propeller thrust coefficient k_t, which relates the

square of the rotational speed of the propellers to the thrust generated, was experimentally obtained by using a force torque sensor to measure the thrust force generated at different propeller speeds. Similarly, the propeller drag moment coefficient k_d, was obtained via the force-torque sensor by measuring the moment about the axis of rotation at different propeller rotational speeds. The quadcopter yaw drag coefficient k_r was approximated by considering values for similar vehicles from the literature [32] [33].

Table 4.1: Navi quadcopter parameters used in simulation

Parameter	Value	Unit	Description
m	1.05	kg	Mass of Navi
g	9.81	m/s^2	Gravitational acceleration
l	0.22	m	Arm length
k_t	8.7e-8	N/RPM^2	Propeller thrust coefficient
k_d	8.7e-9	Nm/RPM^2	Propeller drag coefficient
J_r	2.208e-5	$Kg{\cdot}m^2$	Propeller moment of inertia
k_r	0.08	Nms/rad	Vehicle drag coefficient
$\boldsymbol{I}$	$\begin{bmatrix} 1.122\times10^{-2} & -5.62\times10^{-5} & -1.42\times10^{-8} \\ -5.62\times10^{-5} & 1.123\times10^{-2} & -4.5\times10^{-6} \\ -1.42\times10^{-8} & -4.5\times10^{-6} & 2.108\times10^{-2} \end{bmatrix}$	$Kg\cdot m^2$	Navi moment of inertia

4.2 Simulation Results

In this section, the results of the controller presented in Chapter 3 will be tested under different initial conditions to test its robustness. Two sets of simulation results will be presented. In the first set, the goal is to make the vehicle stabilize to the reduced hover condition around the point where the vehicle lost its propeller. In the second set of simulations, the quadcopter will be commanded to reposition to a specific location away from where the propeller was lost. In both simulations,

propeller 4 (see Figure 2.1) fails at $t = 0.1$ and the recovery controller is activated immediately.

4.2.1 Case 1: Maintaining Hover

The first set of simulations tested the controller's capabilities of maintaining position control as well as to reach hover. The priority as always is for the inner controller to reach the equilibrium hover condition while the outer position controller slowly drives the vehicle to the desired location which is the location of the vehicle when the propeller failed. The initial and desired states for the first set of simulations are summarized in Table 4.2 and Table 4.3. Euler angles will be used to define the initial attitude condition because of their simpler and intuitive representation. The built-in MATLAB function 'angle2quat' offered in the Aerospace Toolbox is used to convert the initial Euler angles to quaternions to propagate the dynamics of the system.

Table 4.2: Initial conditions for the controller simulations

Description	Value	Unit
x	2	m
y	2	m
z	5	m
$\dot{X},\dot{Y},\dot{Z}$	Variable	m/s
p,q,r	0	rad/s
ϕ,θ,ψ	Variable	deg

Simulations are executed for 8 seconds which is considered sufficient time for the vehicle to converge to the equilibrium solution as well as to reach the desired position; otherwise, the vehicle will be considered to have drifted far away or crashed. Therefore, the recovery task is considered a failure if the vehicle loses more than

Table 4.3: Desired parameters for the outer position controller

Description	Value	Unit
x_{des}	2	m
y_{des}	2	m
z_{des}	5	m
$\dot{X}_{des}, \dot{Y}_{des}, \dot{Z}_{des}$	0	m/s

2.5 m in altitude, drifts horizontally by more than 4 m, or does not reach the desired position in 8 seconds.

In the first simulation, the quadcopter is initialized with zero Euler angles and no initial horizontal velocity which represents a normal hover condition.

As can be seen in the first plot of Figure 4.1, the x, y and z variables converge to the desired values with a slight error in the altitude after approximately 4 seconds. This steady state error in the altitude is due to the fact that there is no integral term in our feedback controller. The vehicle deviates by almost 0.7 m in the x-direction before converging back to the initial position. The oscillations, particularly prominent in the velocity responses (second plot in Figure 4.1) after the vehicle reaches the desired position($t > 4$ s) represent the periodic equilibrium solution; at equilibrium the vehicle wobbles around the desired position with a non-zero velocity but the average velocity over the period of oscillation is zero. In the third subplot, the vehicle's angular velocity converges in approximately one second; however, the vehicle's pitch and roll rates exhibit small offset from the equilibrium solution computed in Chapter 3. This can be explained by the simplifying assumptions and approximations used to derive the equilibrium solution. On the other hand, the angular velocity in the yaw direction r converges precisely to the equilibrium value. The propeller thrust plot shows the two opposite working propellers (f_1 and f_3) experience an increase in thrust coincident with the drop in thrust of the 2^{nd} propeller

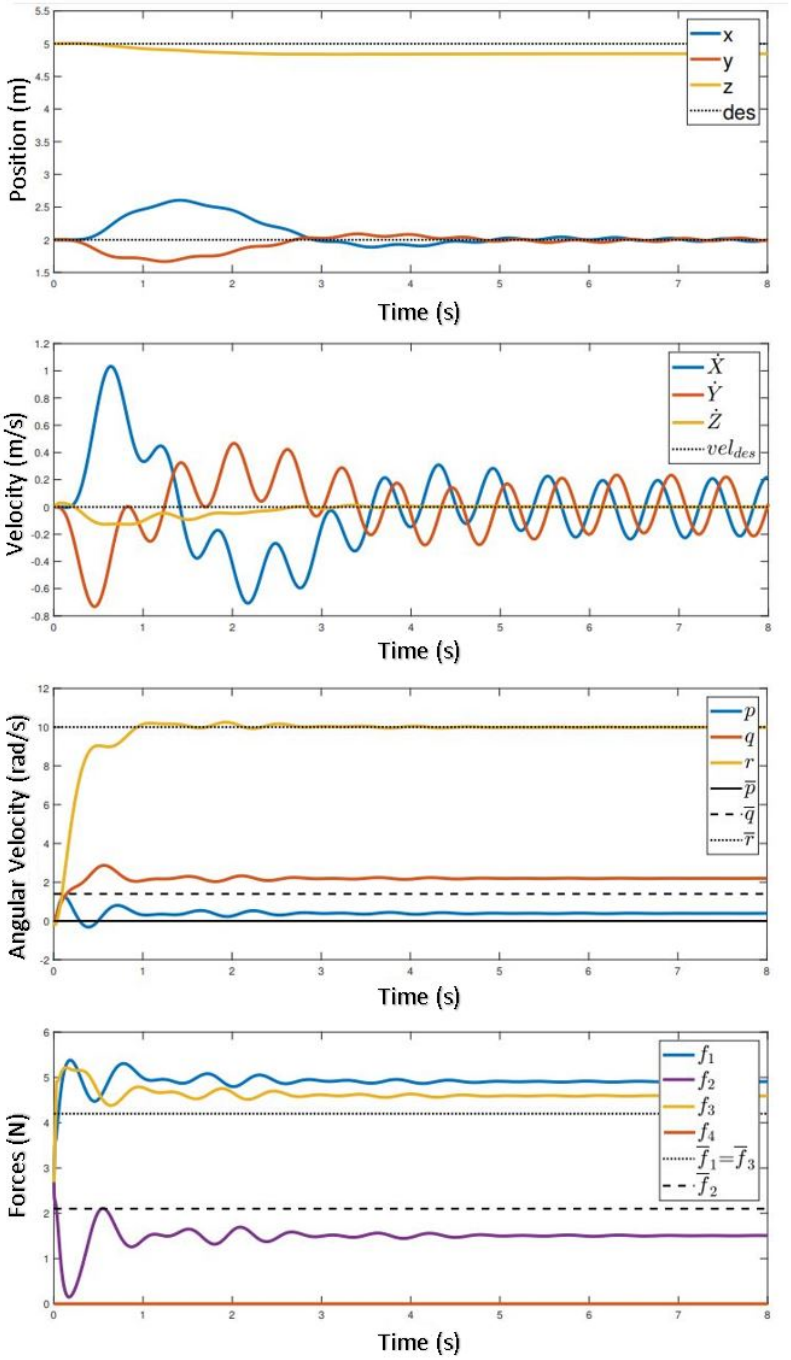


Figure 4.1: Simulation results after losing the 4th propeller where the position and desired position are shown in the first plot, velocity component variation in the second, angular velocity and the equilibrium vales for each component in the third. Propeller forces and there equilibrium values for the three propellers.

in response to loss of thrust in propeller 4. The drop in the 2^{nd} propeller thrust below the equilibrium value $\overline{f}_2$ results as the controller tries to increase the angular velocity of the vehicle about the body z-axis before activating the 2^{nd} propeller. As shown in Figure 4.2, in the absence of thrust on the 4^{th} propeller, f_2 thrust will cause the vehicle to flip.

Meanwhile, while f_2 is at a minimum in the short time immediately after failure of f_4, the thrusts of f_1 and f_3 reach a value of $\frac{mg}{2}$, thus generating the required thrust to provide the lift to cancel out the gravitational force. When the vehicle has gained sufficient angular momentum to handle a moment from f_2 without causing the vehicle to flip, f_2 converges to the equilibrium solution $\overline{f}_2$ with a slight offset and f_1 and f_3 converge to their equilibrium solutions with a slight offset. The fact that the vehicle is able to reasonably maintain its altitude with only a small drop 0.2m implies that the three working propellers are able to collectively generate sufficient thrust at all times after the loss of a propeller.

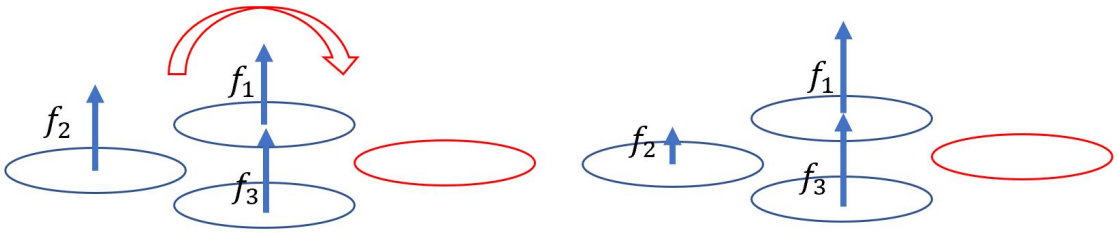


Figure 4.2: A diagram showing the reason behind the sudden drop in f_2 below the equilibrium solution value in the first second after f_4 is disabled

The next set of simulation tests is designed to demonstrate the effect of the initial conditions on the performance of the vehicle, specifically the effect of different initial Euler angles and the initial horizontal velocity.

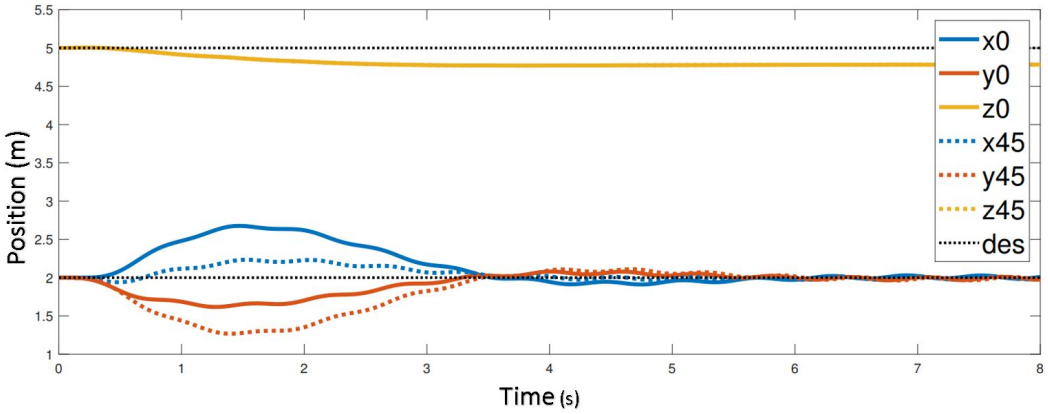
The effect of varying the yaw angle will be addressed first. To test the sensitivity of the control law to a change in the yaw angle, the remaining initial conditions are

kept the same as in the first scenario presented (zero pitch, zero roll, and zero initial velocity).

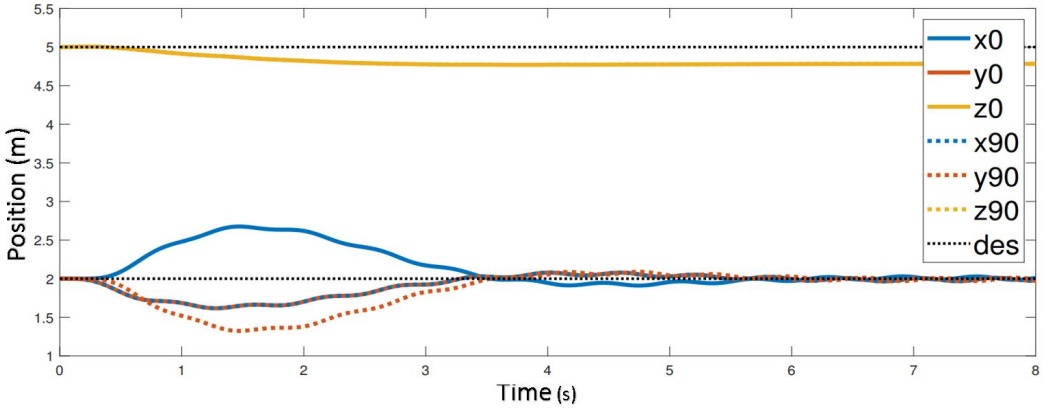
The results in Figure 4.3 compare simulated position of the vehicle for three initial yaw angles to the results presented in Figure 4.1. Since the only response that revealed interesting outcomes is the vehicle position response, the other states are not considered in this comparison. The results in Figure 4.3 show a major difference in the drift direction of the vehicle as the initial yaw angle is changed. The 3^{rd} case for the initial yaw angle of 180° shows the diametrically opposite results to the zero yaw angle case as would be expected. Therefore an interesting conclusion can be summarized from the results: as long as the other initial conditions remain the same, a change in only the yaw angle will not cause the vehicle to drift further than the absolute value of any one of the cases, but only result in a change in the direction of the drift.

Next, the impact of the initial velocity will be addressed. For simplicity, an initial velocity in the x-direction only is assumed and all the Euler angles are set to zero. The plots in Figure 4.4 show the position responses (x-axis, y-axis, z-axis components respectively). Each plot shows the responses of the vehicle for the initial velocity in the x-direction varying from 0 to 3.5 m/s. It can be seen that as the initial velocity increases, the vehicle's position drift is more substantial and increases; however, the controller is still able to stabilize and regulate the vehicle around the equilibrium solution. Although the vehicle recovered from the loss of a propeller for all velocities considered, the high drift in the horizontal direction is worrisome especially since the vehicle is uncontrollable in a sense that the pilot can not change the path of the vehicle during this time. An interesting pattern that emerged from these plots is the sequence of the drift as the velocity increases which shows that the results are not chaotic but are sensitive to many variables that can affect the

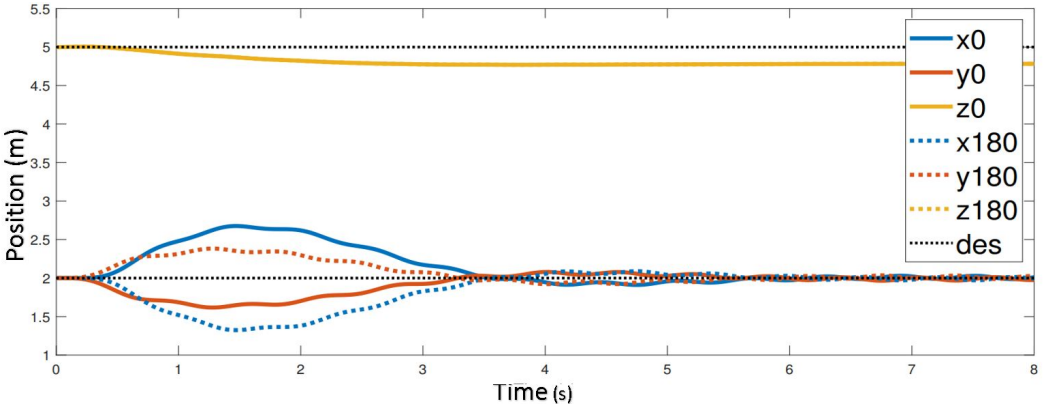
response of the vehicle.



(a) $\psi = 45°$ vs $\psi = 0°$



(b) $\psi = 90°$ vs $\psi = 0°$



(c) $\psi = 180°$ vs $\psi = 0°$

Figure 4.3: Simulation results showing the propagation of the vehicle's position at different initial yaw angles ($45°, 90°, 180°$) compared to the first case with zero yaw deflection

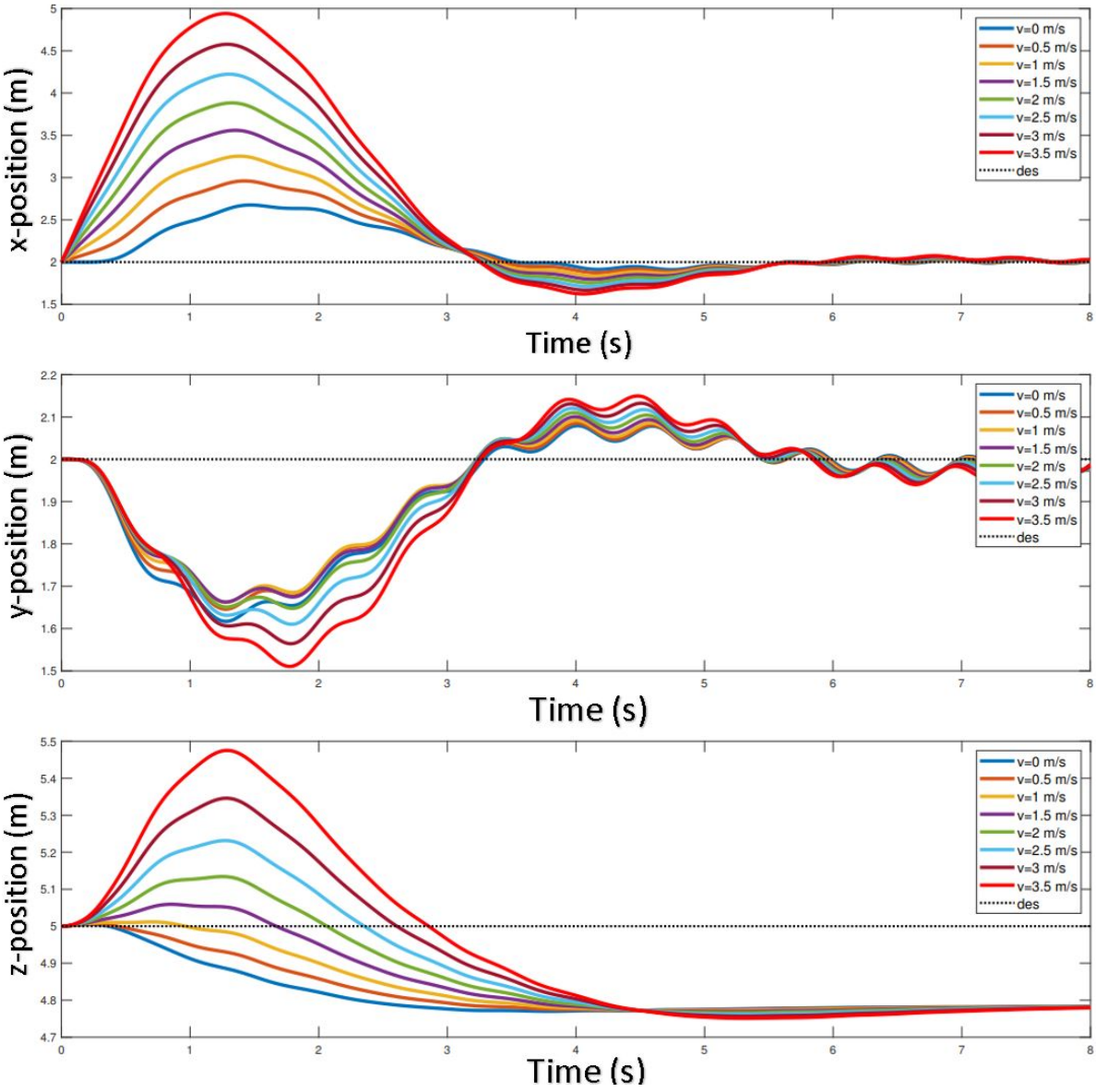
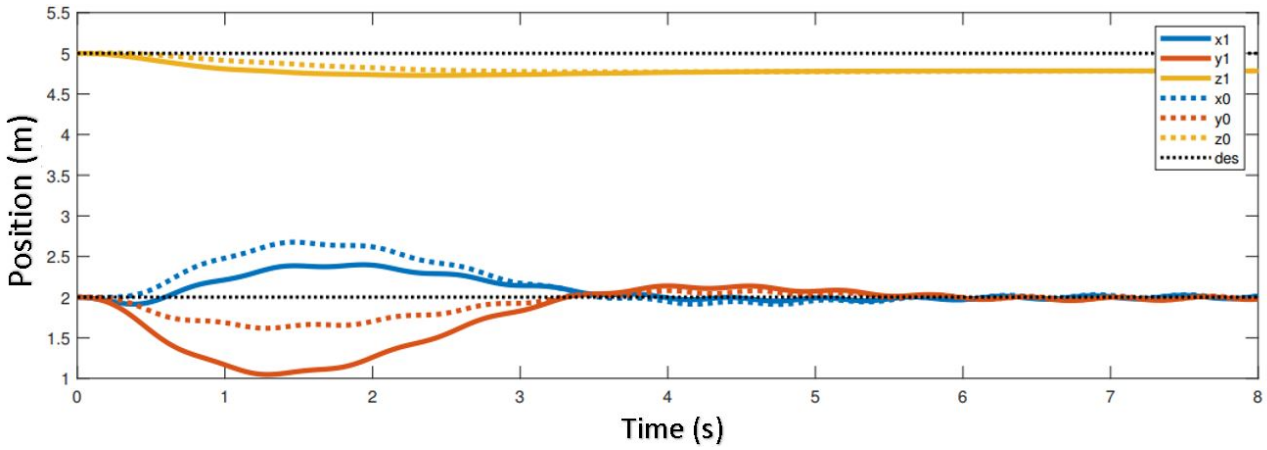


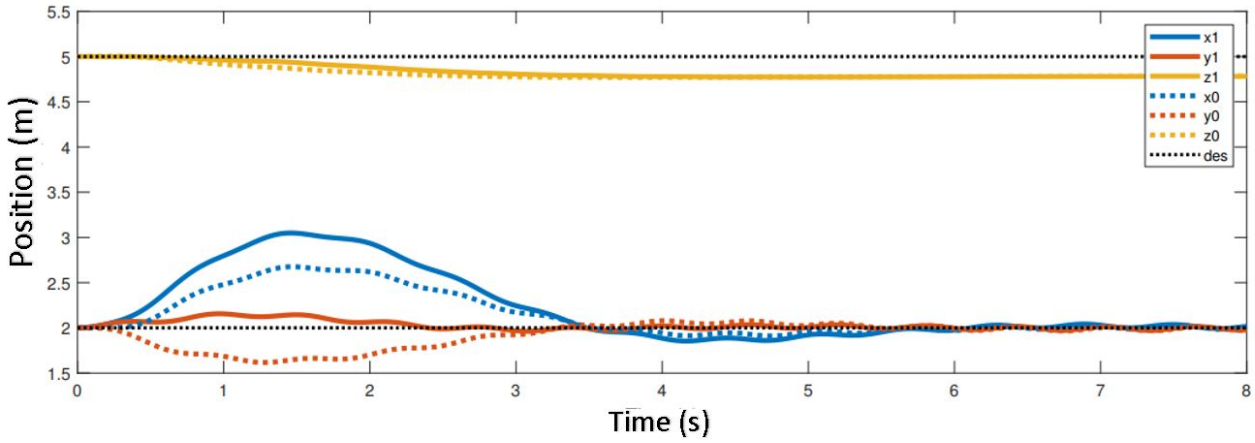
Figure 4.4: Simulation results showing the propagation of the vehicle's 3D position at different initial velocities superimposed in one plot for each axis

Finally, only two results will be presented for the change of the initial pitch and roll angle due to the complex, chaotic behaviour and the dependency of the response on the other variables (initial yaw and initial velocity). The first scenario will assume the initial roll and pitch of the vehicle are 15° each, with no yaw angle offset or a horizontal velocity. The second scenario assumes the initial roll and angle

are $-15°$. The results will be compared to the results presented in Figure 4.1 which account for the standard hover case (zero pitch, roll, and yaw).



(a) $15°$ roll and pitch



(b) $-15°$ roll and pitch

Figure 4.5: Simulation results showing the propagation of the vehicle's position at two different initial roll and pitch angles compared to the first case at normal hover

Figure 4.5a reveals that as the pitch and roll angles increase, the vehicle overall drifts further to recover. There was not much of a change in the altitude of the vehicle from the normal recovery, and it stabilizes to the same offset error. Overall the vehicle recovers at the same time (3.5 s) in both cases. For the sake of maintaining consistent results, we will always assume the 4^{th} propeller has stopped working. Figure 4.5b results differ slightly from the results presented in Figure 4.5a. The max

drift (1 m) is still the same as that in Figure 4.5a but now in the x-direction rather than the y-direction. This is logical since the location of the 4^{th} propeller relative to the orientation of the vehicle plays a big part in the path the vehicle will take to reach the recovery stage.

4.2.2 Case 2: Position Regulation

In the previous section, the response of a quadcopter losing a propeller and its ability to maintain its position were scrutinized. The dependency of the response on initial conditions was investigated. In this section, we will test the capability of the control law presented in Chapter 3 to reach any desired point in space. This is important since upon impacting a wall and losing a propeller, a reliable controller incorporated in a control recovery pipeline should be able to reach the desired point away from the wall, which will be addressed in the next chapter.

The scenario presented will be initialized at normal hover condition (zero Euler angles and no velocity) at $(x, y, z) = (2, 2, 5)$ and have a desired set point for the outer position controller at the coordinates $(x_{des}, y_{des}, z_{des}) = (4, 4, 7)$.

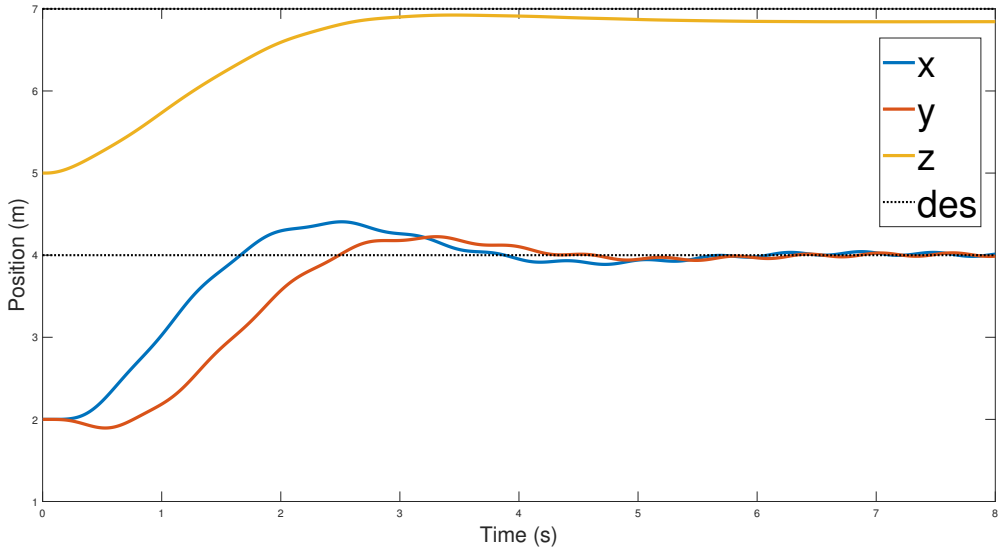


Figure 4.6: Simulation results showing the propagation of the vehicle's position starting from normal hover to a desired point of $(x_{des}, y_{des}, z_{des}) = (4, 4, 7)$

Figure 4.6 shows that the position controller is capable of reaching the desired point in space and not just maintain its position. The vehicle gained altitude to reach the desired height with a small offset without overshooting but still overshoots around 0.4 m in the horizontal direction compared to 0.7 m in Figure 4.1. The change in the desired position did not cause the controller to take more time to recover, as the recovery still took approximately 3.5 s to reach and stabilize around the desired position.

4.2.3 Batch Simulation Validation

To evaluate the recovery controller, a batch simulation was performed by varying the initial conditions of the simulation six thousand times. The batch method provides essential information about the average deviations for variables such as

drift in the horizontal direction off the path to the desired point, height loss, as well as the recovery time.

The range of initial conditions was chosen to cover a large number of possible scenarios that could happen in a normal flight. The initial linear velocity of the quadcopter in the X direction was varied from 0 m/s to 2 m/s. The angular body rates and the initial yaw of the vehicle were set to zero. The initial roll and pitch were varied from $-\pi/6$ to $\pi/6$. The initial conditions used in the batch simulation are summarized in Table 4.4.

Table 4.4: Initial conditions for the batch simulation

Range	Units	Description
$-\pi/6 < \phi < \pi/6$	rad	Initial roll
$-\pi/6 < \theta < \pi/6$	rad	Initial pitch
$\psi = 0$	rad	Initial yaw
$0 < \dot{X} < 2$	m/s	Initial linear velocities
$\boldsymbol{\omega} = \mathbf{0}$	rad/s	Initial angular velocities

The histograms of the simulation results for height loss, horizontal drift, and time to recover are presented in Figure 4.7. Simulations were run for six simulated seconds each. The recovery success rate of over 98.7% was obtained where the vehicle recovered within a 5-second time window. This demonstrates that close to the full range of possible initial conditions can be handled by the controller. The average recovery time, which is the time for the vehicle to reach the desired position hovering around the equilibrium solution, needed to reach the desired position was 4.1 seconds with a standard deviation of 0.4 seconds. The average height loss over the simulations was 0.25 m with a standard deviation of 0.02 m. This shows that the

vertical drop is not a significant issue for the range of cases considered. Finally, the average horizontal drift is 0.7 m with only around 13% of the simulated scenarios with a drift greater than 1 m. In the latter cases, the initial velocity is not zero and the attitude deviations are large.

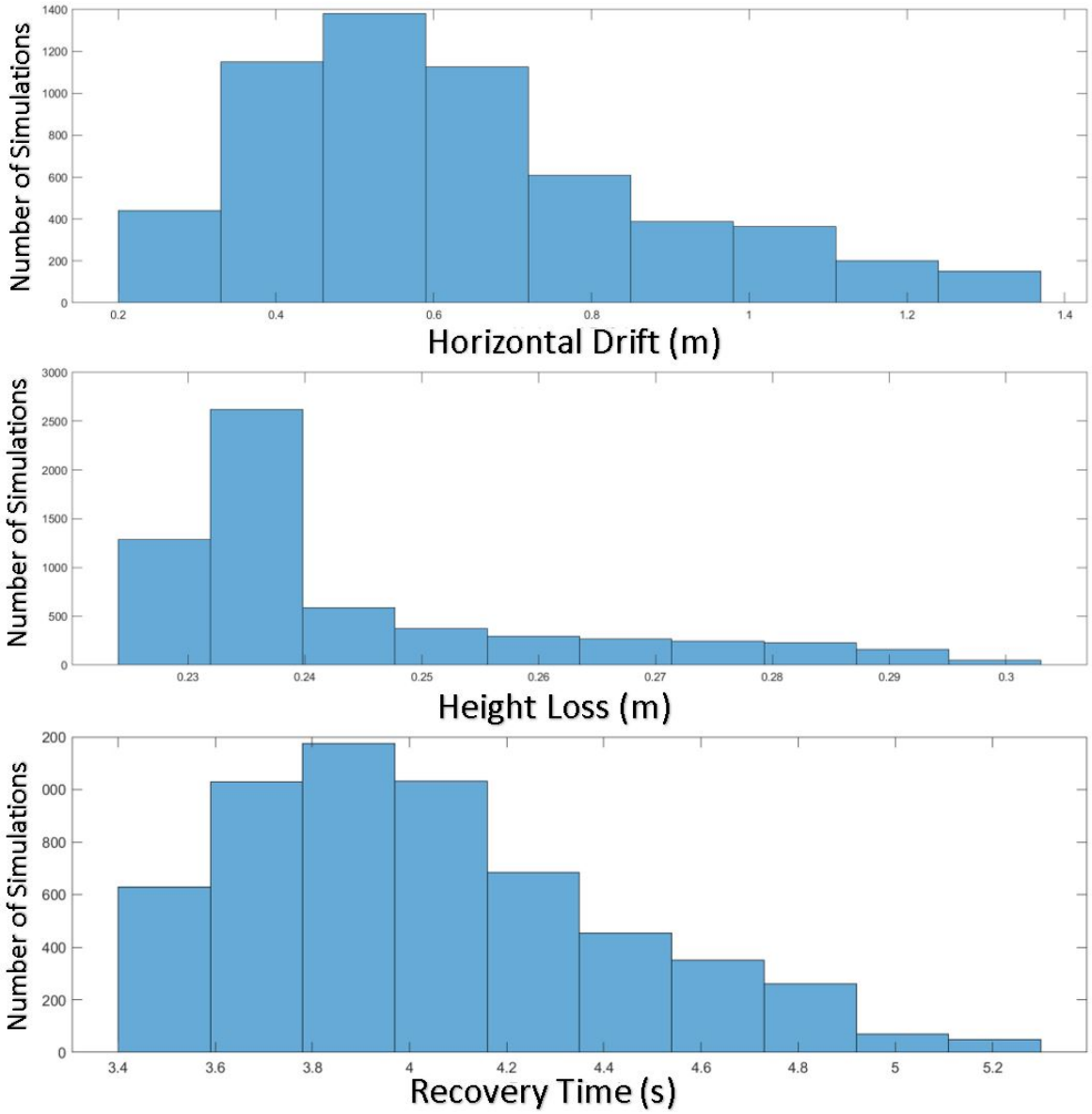


Figure 4.7: Histograms of the batch simulation results showing the number of simulations with respect to respectively the horizontal drift, height loss and recovery time.

4.3 Hardware in the Loop Simulation

Once the recovery control law was tested in simulation, it was programmed onto the Pixracer micro-controller of the NAVI vehicle. Pixracer is a micro-controller optimized for a wide variety of small racing quadcopters and planes. The firmware flight stack called 'PX4' runs the flight control, state estimation, and all middleware functions. The PX4 flight stack is open source and the most well maintained and developed option for developers. It uses the Mavlink protocol to communicate with other devices and modules. A desktop ground station application called QGroundControl, which also uses Mavlink, is used for parameter setting, flight time debugging, and calibration.

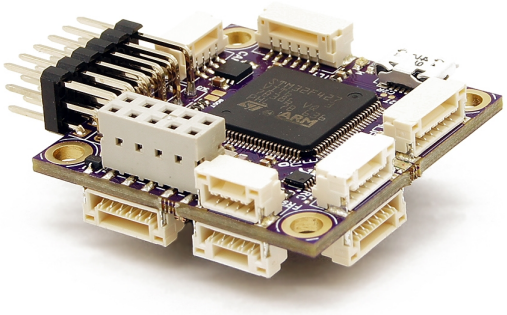


Figure 4.8: Pixracer microcontroller

Due to safety concerns of testing with a real vehicle, firmware validation and parameter tunning were carried out via hardware in the loop which is a simulation mode which runs PX4 firmware on the real flight controller Pixracer. Hardware in the loop is run in conjunction with Gazebo, a powerful 3D simulation environment

for autonomous robots that runs on the desktop ground station. A micro-USB connects the controller running Hardware in the loop to the ground station running Gazebo and QGroundControl. A model of our vehicle including its physical and motor properties was input into Gazebo. This model helps Gazebo in simulating the effect of the environment on the vehicle while in flight.

One of the PX4 flight controller modules 'mc_att_control' was modified to run the recovery controller in conjunction with the standard attitude and manual controllers. The recovery controller is triggered upon the pilot switching one of the switches on the radio control. The goal is to have the vehicle lift off to a certain altitude and hover there using the altitude control mode. The pilot then switches using the manual button on the radio controller to the recovery controller. Since the vehicle will not actually be losing a motor, one of the four propeller thrusts will be set to zero and the three remaining propeller thrusts will be supplied by the controller output. For safety reasons, PX4 sets all zero thrusts signals to the minimum arming rpm; therefore, the minimum arming rpm was set to zero as well as other fail-safes had to be removed in the 'motor_rpm' module in the PX4 firmware. This ensures that when the recovery control switch is triggered, only three propellers will produce thrust.

Activation of the recovery controller always led to crashes in the Gazebo simulation. After cross examining the crashes to the Matlab simulation it was found that the crashes were due to a missing yaw drag term in the Gazebo dynamics model. Gazebo is designed such that it is only possible to change the vehicle physical dimensions and motor properties and not the dynamics of the environment; since the yaw drag is rarely used for general aircraft simulations it was never coded into the Gazebo environment. Due to the risks of vehicle damage and difficulties with

tethering a vehicle that is yawing at high rates, we did not pursue further experiments. It was critical to verify that the yaw drag plays a big role in the recovery, especially in the first seconds of the recovery. Having sufficient drag in the yaw direction helps the vehicle to stabilize and reach the equilibrium solution. It is important to also note that the value for the yaw drag used in the Matlab simulation is an estimated value, and the equilibrium solution was derived based on that specific value.

4.4 Summary

The simulation results presented in this chapter show that the developed control law is capable of recovering the vehicle after losing a propeller from a wide range of initial conditions. Also, the vehicle was able to recover and reach a desired point in space in any direction. However, higher initial velocities and higher initial pitch or roll angles increase the horizontal drift of the vehicle. The batch simulations provided consistent results on the efficiency of the control law in recovering the vehicle in a certain period of time for a wide range of initial conditions. The time for recovery was not affected by the initial conditions. Finally, Hardware in the loop simulation failed due to a missing yaw drag term in the Gazebo environment which prevented us from conducting an experimental evaluation of the recovery controller.

Chapter 5

Collision Control with One Actuator Failure at Impact

In this chapter, the recovery controller as described and tested in the previous chapters was applied to the research objective of recovering from a collision with a vertical wall after losing one actuator. A batch simulation was performed on a wide range of initial conditions to test and validate the control strategy adopted. The results were also compared to the controller developed in [11] to recover from an impact with a wall but without losing any actuator, that is, where the vehicle uses all four propellers to recover.

Section 5.1 goes over the control pipeline, or strategy, that was developed to recover from a destructive impact with a wall that caused a propeller failure. The pipeline consists of three stages, starting with impact detection to position control and finally the reduced attitude controller. However, to allow for the case of loss of the GPS signal, which is easily compromised by tall buildings and big structures [37] [38], the position controller is replaced by a velocity controller with an alternative overall control strategy; this strategy is less effective but it is independent of the vehicle position estimate.

The contact dynamics model introduced in [27], that defined the interaction between the quadcopter's bumpers around the propellers and the wall, was used in the MATLAB simulator. This simulator was utilized to generate results with a batch simulation for a wide range of initial conditions to identify the most challenging scenarios to recover from, as well as to find the critical initial conditions and parameters where the vehicle was not able to recover with a lost propeller. The setup for these simulations is described in Section 5.2. The two control strategies suggested in this chapter were also compared to identify the strength and weaknesses of each. Finally Section 5.3 contains the analysis of simulation results which uncovers that majority of impacts were recoverable up to a certain critical angle.

5.1 Collision Recovery Strategy

The collision recovery strategy is composed of three consecutive stages as shown in Figure 5.1. The first stage detects that an impact has occurred. The second stage allows for two possible options depending on whether there exists a reliable position estimate of the vehicle. If there is a vehicle position estimate, the position controller computes the desired acceleration required to move away from the wall based on the output of the first stage; otherwise, the velocity control computes the desired acceleration based on velocity estimates. Finally, the last stage recovers the vehicle's reduced attitude while producing the desired thrust of each propeller to match the desired acceleration. The third stage differs slightly based on which of the two aforementioned options was taken in the second stage.

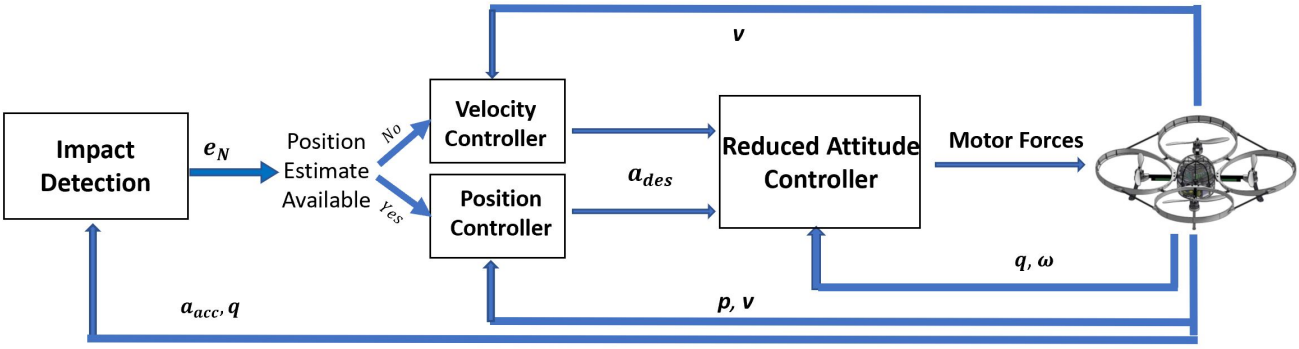


Figure 5.1: A block diagram showing the proposed three-stage recovery strategy of a vehicle after impacting a wall and losing a propeller.

5.1.1 Collision Detection

It is assumed that impacts only occur with vertical walls, which can be detected by a sudden increase in the vehicle horizontal acceleration. The absolute acceleration can be calculated by rotating the acceleration measurement $\boldsymbol{a}_{acc}$ into the inertial frame, and compensating for gravity as shown below:

$$\boldsymbol{a} = \boldsymbol{q} \odot \boldsymbol{a}_{acc} + \mathbf{g} \tag{5.1}$$

where $\boldsymbol{a} = [a_x, a_y, a_z]^T$ is the vehicle's center of mass acceleration in $\mathcal{F}_I$.

A collision is detected whenever the magnitude of the horizontal components of the acceleration exceeds a certain threshold k as shown below:

$$CollisionDetected = \begin{cases} 1 & if \parallel [a_x, a_y]^T \parallel > k \\ 0 & otherwise \end{cases} \tag{5.2}$$

At the time of collision detection, the wall normal $\boldsymbol{e}_N$, to be used in subsequent stages can be estimated as follows:

$$\boldsymbol{e}_N = \frac{[a_x, a_y]^T}{\parallel [a_x, a_y]^T \parallel} \tag{5.3}$$

5.1.2 Desired Acceleration Calculation

After a collision is detected, the second phase is triggered where the controller computes the desired acceleration to send to the inner controller to achieve. Since GPS signal is not very reliable in indoor environments or adjacent to tall buildings [38] [39], a position estimate is not always available; therefore, a velocity controller is incorporated as an alternative to the position controller.

Starting with the position controller, and using the computed wall normal $\boldsymbol{e}_N = [e_{Nx}, e_{Ny}, 0]^T$, a desired position can be found that lies along the normal direction at the same altitude as the impact, at a certain distance away from the wall. To make the recovery time faster and have enough clearance away from the wall, a clearance variable d_{clear} was introduced. Therefore the desired position is computed such that:

$$\boldsymbol{p_{des}} = d_{clear} \begin{bmatrix} e_{Nx} \\ e_{Ny} \\ 0 \end{bmatrix} + \begin{bmatrix} x_{impact} \\ y_{impact} \\ z_{impact} \end{bmatrix} \tag{5.4}$$

where z_{impact} is the altitude at the time of impact which can be easily obtained from the barometer measurement and x_{impact} and y_{impact} are obtained from the GPS signal. In the results presented here, d_{clear} was chosen to be 3 m. The desired velocity $\dot{\boldsymbol{p}}_{\boldsymbol{des}}$ is always set to zero. Afterwards, Equation (3.17) can be rearranged to calculate the desired acceleration $\boldsymbol{a}_{des}$ as shown below

$$\boldsymbol{a}_{des} = -2\xi\omega_n\dot{\boldsymbol{p}} - {\omega_n}^2(\boldsymbol{p_{des}} - \boldsymbol{p}) \tag{5.5}$$

In case of a missing position estimate, Equation (5.5) cannot be used to compute the desired acceleration, which means the vehicle cannot be tracked during the recovery; therefore, an alternative recovery strategy is needed.

Velocity estimates are obtained from various sources and do not solely rely on a GPS signal or a position estimate [40] [41]. Integrating the accelerometer measurements would result in an approximate velocity estimate which may be good enough over a short time frame immediately after impact to recover the vehicle. However, if there is no position feedback, the vehicle can drift from any external perturbations and it is impossible to command the vehicle to a certain position in space. Therefore, using the wall normal vector $\boldsymbol{e_N}$, we can compute the desired acceleration that scales over time to recover to a safe distance. Since we can not control the vehicle's position, we can command the vehicle to head in a certain direction through the desired acceleration while its velocity is controlled.

Starting with the wall normal, which is estimated from acceleration measurements only, and compensating for gravity, we define the desired acceleration vector to incorporate a velocity dependent term as follows:

$$\boldsymbol{a}_{des} = -2\xi\omega_n\dot{\boldsymbol{p}} + (\boldsymbol{e}_N + \mathbf{g}) \tag{5.6}$$

However, the second term in the above equation, $(\boldsymbol{e}_N + \mathbf{g})$, is not changing as time progresses and as the vehicle points away from the wall. Therefore, a scaling of the desired acceleration is proposed, to autonomously scale the acceleration vector as the vehicle moves away from the wall, as illustrated in Figure (5.2). A maximum scaling time T_{scale} of 2 seconds was chosen through trial and error; a higher value would result in the vehicle drifting far from the wall and a value less than 2 seconds risks the vehicle to remain too close to the wall. Since the final desired acceleration vector at the end of the T_{scale} period should be pointing opposite to gravity, that is the final acceleration vector should only have a z-component, the following time-dependant expression for a_{des} is proposed

$$\boldsymbol{a}_{des} = \begin{cases} -2\xi\omega_n\dot{\boldsymbol{p}} + K_N(\boldsymbol{e_N})(1 - t/T_{scale}) + \mathbf{g} & \text{if t} \leq T_{scale} \\ -2\xi\omega_n\dot{\boldsymbol{p}} + \mathbf{g} & otherwise \end{cases} \tag{5.7}$$

where the $\boldsymbol{e_N}$ term decreases linearly with time to zero at $t = T_{scale}$. The coefficient K_N allows to adjust the magnitude of the horizontal component of the acceleration vector to help move the vehicle away from the wall more aggressively.

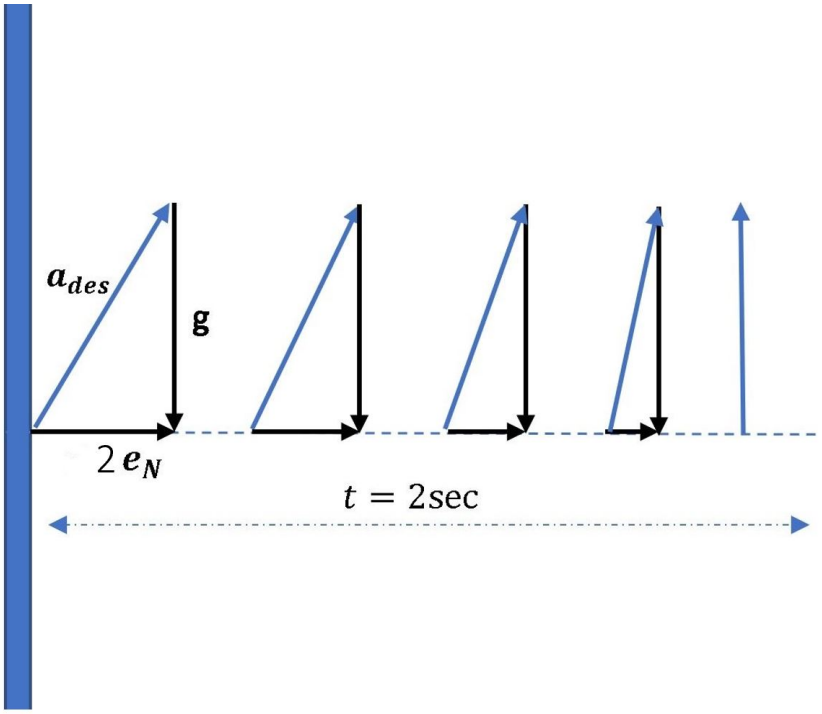


Figure 5.2: A diagram showing how the desired acceleration changes over the duration of the recovery (t=T_{scale} s)

5.1.3 Reduced Attitude Controller

The third stage is similar to the reduced attitude controller presented in Chapter 3. The desired acceleration computed in stage 2 is used to find the total thrust force f_Σ which can be calculated using Equation (3.18) and the desired primary axis $\boldsymbol{n_{des}}$ using Equation (3.19).

The overall goal of the control strategy is slightly different from the controller presented in Chapter 3, where the priority was to stabilize the attitude of the vehicle and subsequently move it to a desired position. The goal now is to move the vehicle away from the wall and to achieve convergence to the equilibrium solution at the same time. It can be said that the recovery control in this chapter is characterized by a two-point failure as compared to the controller of Chapter 3. If the vehicle is not able to reach the equilibrium solution for whatever reason, or it collides again into the wall while in recovery, then the vehicle crashes. Therefore, a slight adjustment has to be made to the parameters of the outer controller to make both objectives of the controller a priority. This is explored further in the simulations.

5.2 Simulation Parameters

The collision recovery strategy was validated by running thousands of simulations in batch through a wide range of initial collision conditions. The wall is assumed to be a vertical plane that the vehicle impacts at a 45-degree yaw angle so that only one propeller bumper comes in contact with the wall during impact. Before presenting the results, we will define the initial conditions and the parameters used in the simulations.

5.2.1 Initial Conditions

From the previous chapter, it was concluded that the initial velocity, as well as the attitude of the vehicle (Euler angles) have an effect on the response of the vehicle after losing a propeller. Similarly, the incoming velocity and the inclination of the vehicle with respect to the wall have a significant impact on the post-collision response of the vehicle. The yaw angle of the vehicle dictates whether one or two

propellers impact the wall and as already stated, the yaw angle is fixed at 45 degrees for all the simulations. Initial angular velocities were set to zero for all the simulations because a typical quadcopter flying into a wall would not generally have large body rates. The incoming velocity for the vehicle was set to be along the $\mathbf{e}_X$ axis of the inertial frame. Since the vehicle is symmetrical about the roll and pitch axes, we varied the orientation of the vehicle pre-collision such that it is pitched into or away from the wall. However, if both pitch and roll angles are nonzero, the pitch angle is not a measure of the inclination of the vehicle into or away from the wall. Therefore, a new parameter γ defined as the inclination angle is introduced, which is the angle between the projection of the body fixed $-$ z-axis onto the vertical plane normal to the wall, and the inertial Z-axis. The inclination is positive when the vehicle is pitched into the wall, and negative if directed away. A wall tangent vector $\boldsymbol{e}_T$ is defined in the inertial frame as:

$$\boldsymbol{e}_T = \boldsymbol{e}_Z \times \boldsymbol{e}_N \tag{5.8}$$

The body fixed $-$z axis can be described in the inertial frame as:

$$\boldsymbol{e}_z^I = \boldsymbol{q} \odot (-\boldsymbol{e}_z) \tag{5.9}$$

So that γ can be computed as:

$$\gamma = sign(\gamma) \cdot \cos^{-1}\left(\frac{\left(\boldsymbol{e}_z^I - \left((\boldsymbol{e}_z^I)^T \boldsymbol{e}_T\right)\boldsymbol{e}_T\right)^T \boldsymbol{e}_Z}{\parallel \boldsymbol{e}_z^I - ((\boldsymbol{e}_z^I)^T \boldsymbol{e}_T)\boldsymbol{e}_T \parallel}\right) \tag{5.10}$$

Using the inclination angle γ allows to combine the two variables ϕ and θ into one value which helps in analyzing the simulation results; besides, it provides an intuitive way to think about the orientation of the vehicle with respect to the wall.

The batch simulation initial conditions and parameters are summarized in Table 5.1.

Table 5.1: Initial conditions for the batch simulation

Range	Units	Description
$\xi = 0.7$	-	Damping of the outer controller
$\omega_n = 1.2$	-	Natural frequency of the outer controller
$K_{det} = 9.81$	m/s^2	Acceleration threshold for collision detection
$-20 < \gamma < 40$	deg	Inclination Angle
$\psi = 45$	deg	Initial yaw
$0.5 < \dot{X} < 4$	m/s	Incoming linear velocities along e_X axis
$\omega = 0$	rad/s	Initial angular velocities

5.2.2 Bumper Deflection Critical Value

The simulation starts with all four motor speeds set to their hover thrust value and the vehicle placed close to the wall starting at the initial conditions predefined. The collision occurs within milliseconds of the simulation depending on the initial attitude and incoming velocity of the scenario. Although we are only setting the initial velocity of the vehicle pre-collision and not at impact, for the analysis of the results it will be assumed that the vehicle's state at collision is approximately the same as the initial state of the vehicle pre-collision.

After the collision occurs, the propeller with the bumper impacting the wall is assumed to be damaged and is shut off in the simulation (by setting its thrust output to zero). The timing of when to shut off the propeller was considered, either when

the collision is detected or slightly later. In reality, the propeller is likely to be damaged at the instant when the bumper deflects the maximum amount and interferes with the propeller blades, causing the motor to stall or to shutdown and possibly damaging the blades. Numerous simulations of quadcopter impacts showed that positive inclination impacts caused a bumper deflection of over 0.01m; on the other hand, most negative inclination angles (where the vehicle points away from the wall) impacts caused a peak bumper deflection of less than 0.01 m. Taking this into consideration, a bumper deflection of 0.005 m was used as a threshold to turn off the propeller to ensure the propeller is shut off after the collision in all the cases considered.

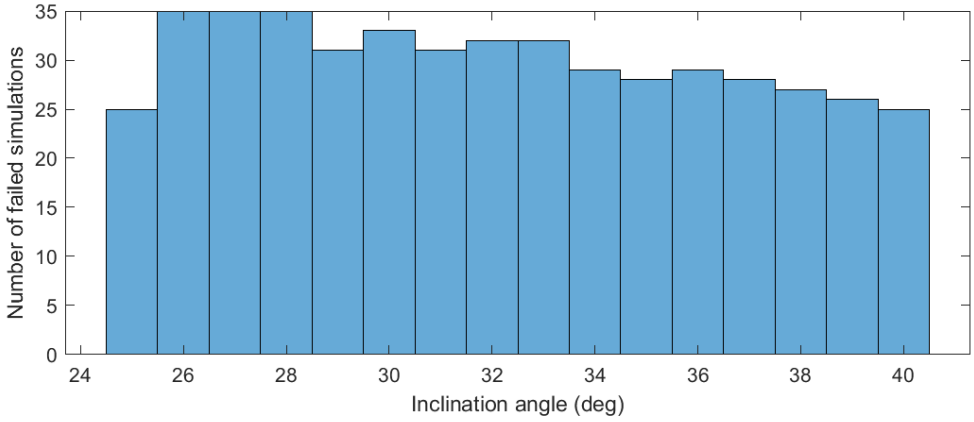
5.3 Results and Analysis

In this section, the performance of the impact recovery controllers presented earlier in this chapter is evaluated. First, the batch results for a wide range of initial conditions as summarized in Table 5.1 are presented followed by an analysis of several interesting cases and observations. Three main results will be presented: a set for the position controller, a comparison between the position controller and the velocity controller, and finally a comparison between our recovery controller and the recovery controller developed in [11] that was successful in recovering a quadcopter from an impact with a wall without losing a propeller.

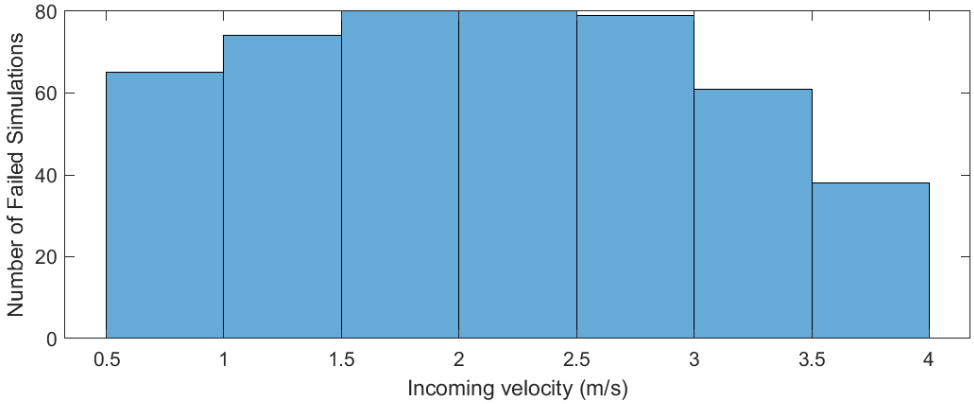
Each simulation ran for 5-8 seconds depending on which controller was used, which is considered ample time for the vehicle to recover and move away from the wall. A failure is defined to occur when the vehicle crashes (falls to ground), or drifts by more than 3 m from the final desired location at the end of the simulation time.

5.3.1 Impact Recovery with Position Control

The results of the batch simulations for the position controller are summarized in Figure 5.3. Out of 2700 simulations, 81.75 % of were successful in recovering from the impact. All of the failures were due to the vehicle crashing to the ground, and no failures occurred due to the vehicle drifting too far from the goal position.



(a) Number of failed simulations vs. inclination angle



(b) Number of failed simulations vs. Incoming velocity

Figure 5.3: Histograms of the failed trials of the batch simulation results.

All the trials were successful for impacts at negative inclination angles (when the vehicle is pitched away from the wall). Failures started to occur in the positive spectrum of the inclination angles considered. As can be seen in Figure 5.3,

the failure rate heavily depends on the inclination angle, where failures begin to occur when the inclination angle exceeds 25°. On the other hand, no strong correlation was found between incoming velocity (in the range a considered here) and failure rate, as can be seen in Figure 5.3b, where some failures were occurring at all incoming velocities.

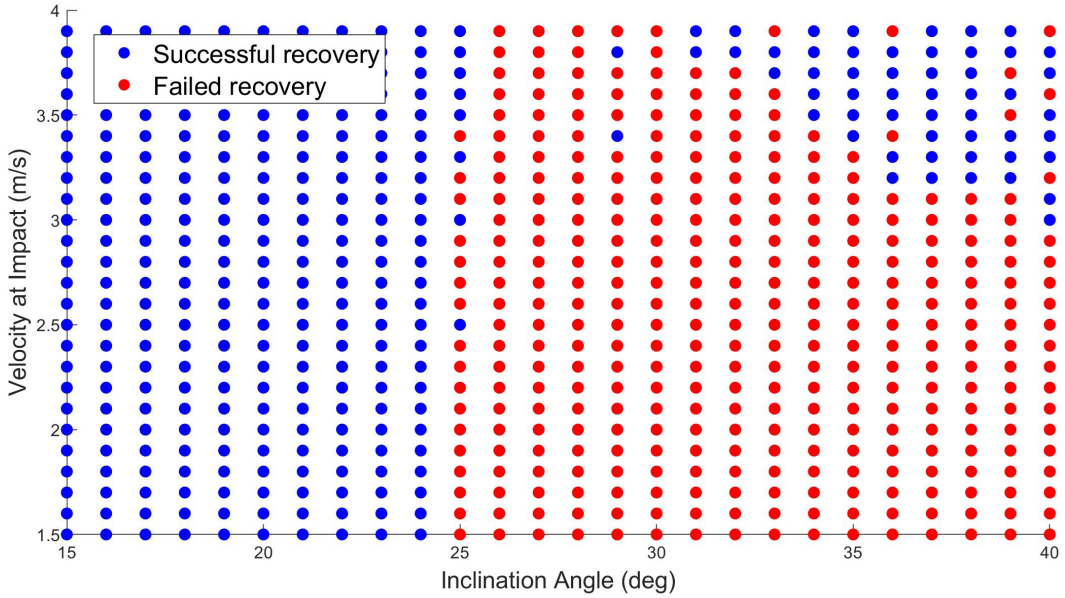


Figure 5.4: Simulated failures vs inclination and velocity

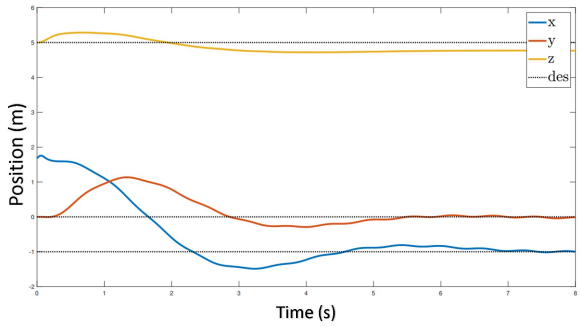
Figure 5.4 summarizes the trial outcome (success or failure) as a function of the incoming velocity and positive inclination angles. Starting at an inclination of 25°, the vehicle crashed in most of the simulations. A critical angle is reached at 26°, where all simulations failed regardless of the incoming velocity. Interestingly, after an inclination of 28° a trend can be seen where as the inclination angle increases the failed recovery rate decreases at high incoming velocities. This will be explored further in the following section.

To further understand the recovery controller performance, a successful recovery with γ=22° and incoming velocity of 2 m/s is presented and analyzed in Figure 5.5.

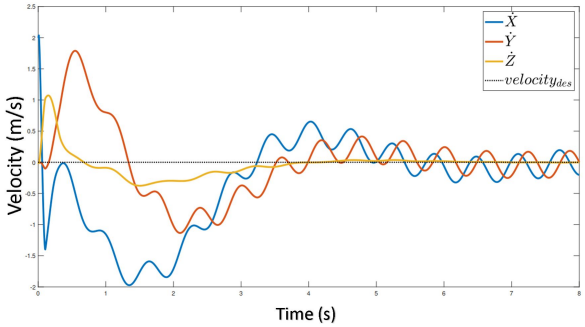
Figure 5.5a shows the position response of the vehicle and the z-component demonstrates that the vehicle was able to maintain its altitude throughout the recovery. It is interesting to see that the altitude of the vehicle increases immediately after the impact. This can be explained in Figure 5.5d where after losing the impacted propeller, the two opposing propellers f_1 and f_3 peak in thrust to spin the vehicle in the yaw direction to reach the equilibrium solution for the yaw rate which causes the vehicle to gain altitude. The vehicle was able to reach the desired position within 6.5 seconds after impact with an overshoot on the y-axis.

Figure 5.5b reveals that initially the vehicle had a positive velocity heading towards the wall; upon the impact, the vehicle bounces away from the wall, and the velocity component of the vehicle in the x-axis points away from the wall. This helped the controller to reach the equilibrium solution without drifting into the wall's plane but instead away from it towards the desired position. Therefore, having a bumper around the propeller, even though it did not help in preventing the actuator from getting damaged, did help in preventing the vehicle from collapsing towards the wall and giving the controller the required conditions to move away from the wall.

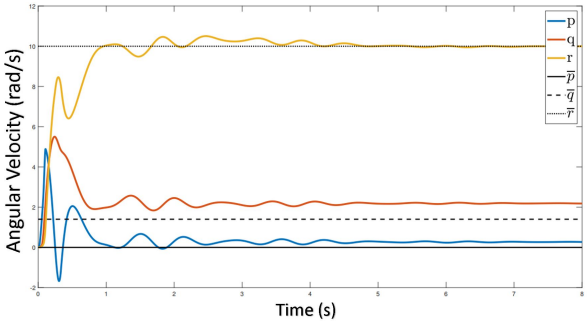
Figure 5.5c shows that the controller converged to the equilibrium angular velocity solution within the first 2 seconds; even though, angular velocity response exhibits larger fluctuations compared to that without impact, as the one shown in Figure 4.1. This is expected as the impact with a wall induces additional moments on the vehicle. To visualize the vehicle's response throughout the recovery, several snapshots of the vehicle during the first seconds are presented in Figure 5.6.



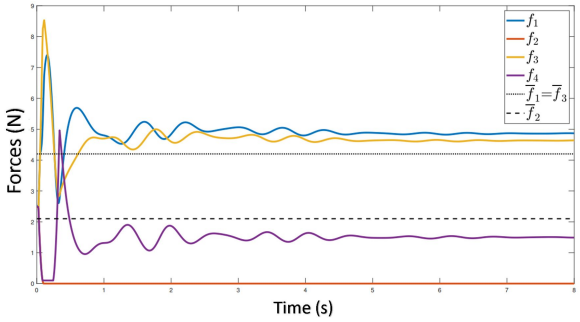
(a) Position



(b) Velocity



(c) Angular velocity



(d) Propeller thrusts

Figure 5.5: Simulation results of a successful impact recovery at γ =22° and incoming velocity of 2 m/s where the vehicle starts at 0.2 m away from the wall and loses the propeller upon impact.

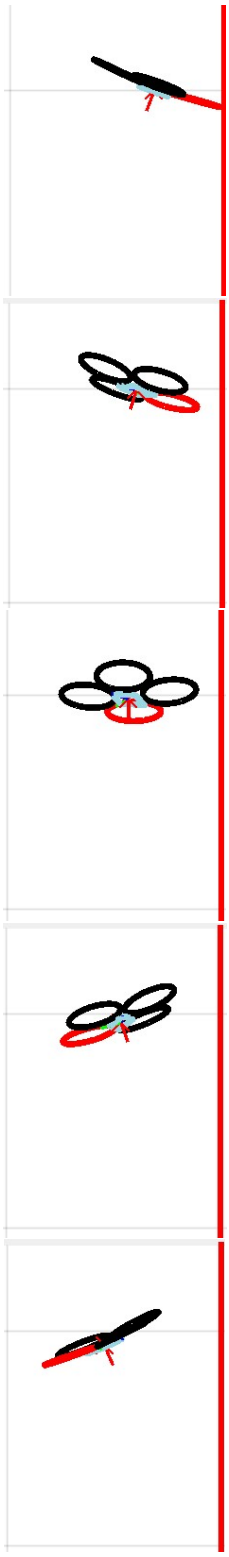


Figure 5.6: Snapshots of an example simulation trial

To understand the successful recoveries at the high velocity and high inclination cases that were predicted by the batch simulation results in Figure 5.4, we investigated several of these trials. In these cases, it appears that the vehicle is able to move away from the wall and recover its attitude; however, the response that takes place and the recovery path taken by the vehicle are not likely to lead to a recovery in a real-life scenario. Figure 5.7 illustrates several snapshots of these false-positive recoveries. Upon impact, due to the steep angle of the vehicle and high velocity at impact, the vehicle becomes nearly flush against the wall in a vertical orientation, causing at least one other bumper to impact the wall. As a result, the vehicle bounces off the wall and is able to recover. In a real-life scenario, this may not be possible since the second impact may result in the second propeller damage and actuator failure, which are not modelled in our simulation nor accounted for in the recovery controller. Therefore we will consider these recoveries as false positives and not count them as successful recoveries.

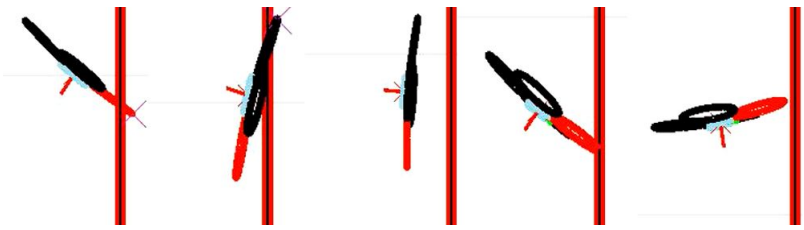


Figure 5.7: Snapshots of the false-positive recovery

5.3.2 Height Loss and Horizontal Drift

Height loss and horizontal drift of the vehicle were investigated for the successful recoveries. Height loss was not significant since it usually occurs when the vehicle becomes flush against the wall, and the propeller thrusts are pointing horizontally, so that no force is countering gravity which causes the vehicle to lose altitude. All these cases led to failures which therefore did not affect the height

loss results for the successful recoveries. All the successful recoveries ended with a slight altitude gain during recovery and then the same offset of 0.2 m from the desired altitude.

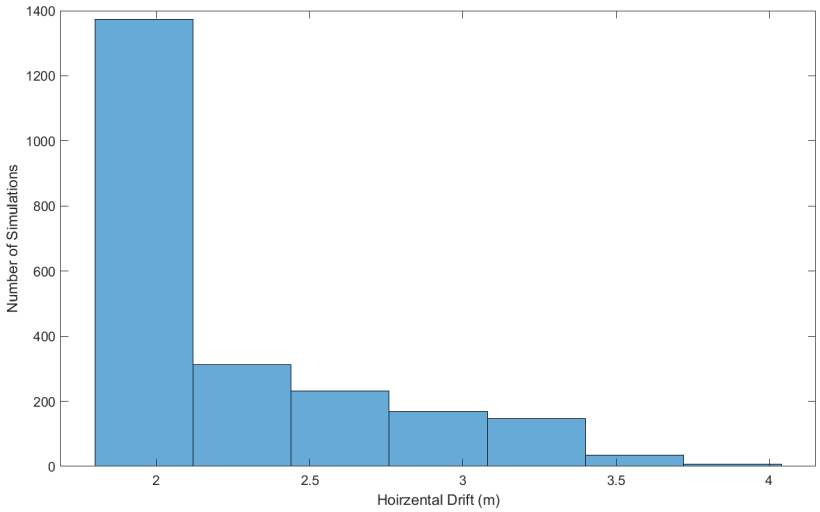
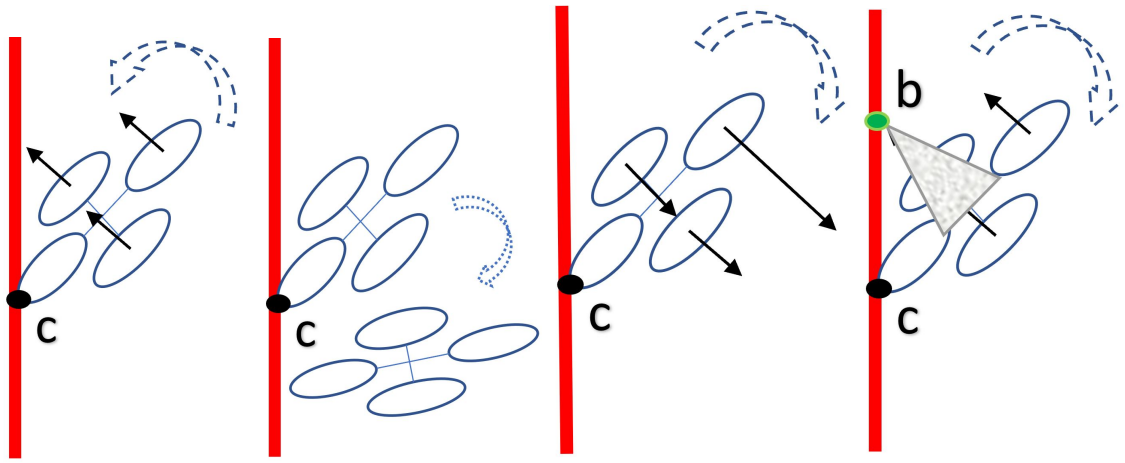


Figure 5.8: Horizontal drift for successful recovery simulations

The deviation in horizontal position or horizontal drift off the straight line path to the desired position is shown in Figure 5.8. The average horizontal drift was 2.2 m with the majority of the trials requiring less than 2.5 m to recover. Several options could be implemented to reduce this drift. One would be to more aggressively track the goal position by changing the tuning parameters in the position controller (damping ratio and frequency). However, the priority should always be to move the vehicle away from the wall. Therefore, a switching or an adaptive controller could be implemented where immediately after impact the goal is to move the vehicle away from the wall, and after the vehicle is at a satisfactory clearance from the wall, the controller can be switched to optimizing the position tracking.

5.3.3 Failed Recoveries

All the failed recoveries were due to the vehicle collapsing into the wall in a vertical orientation. This is similar to the scenario for the false positives discussed earlier, but in these cases, the vehicle does not bounce away from the wall, but rather, tends to flip towards the wall instead of away from it. Usually, if the vehicle is stuck in the vertical orientation flush against the wall, it is impossible to recover using uni-directional motors since the propeller thrusts are pointing directly into the wall. To solve this mode of failure, several solutions were thought of that could be implemented to improve the response.



(a) Stuck vehicle (b) Free fall (c) Bi-directional Propellers (d) Structural modification

Figure 5.9: Stuck vertical condition with three possible solutions

Figure 5.9 depicts the three solutions (b,c,d) from the easiest to hardest to implement. The first solution is upon detecting the impact and knowing that this scenario will result in a failure from the inclination angle and velocity at impact, all four propellers are shut off to use gravity to generate a moment to reorient the vehicle in an upright attitude. This approach was tested in simulation, and it was revealed that for gravity to generate a sufficient angular impulse to tilt the vehicle

back to hover attitude would require too long of a time period, resulting in the vehicle dropping a large distance. Even then, this may not necessarily "unlock" the vehicle from the vertical orientation facing the wall. The second proposed solution is to use bi-directional propellers instead of uni-directional propellers to generate thrust away from the wall. Incorporating bi-directional propellers into the recovery controller will likely improve the recovery rate and should result in vehicle recovery from all scenarios. Finally, a structural component can be added to the center of the vehicle to prevent it from reaching the vertical lock position by contacting the wall at point b as shown. This additional structure would help in "bouncing" the vehicle away from the wall, and in addition it may contribute to increasing the rotational drag in the yaw direction of the vehicle. This in turn would result in a lower yaw rate at equilibrium which would improve the stability of the vehicle and result in a quicker convergence to the equilibrium solution.

5.3.4 Position Control vs Velocity Control

In this section, the velocity control strategy is tested and compared to the position controller presented earlier. Overall, the failure rate matched that of the position controller and the only difference was in the response of the vehicle during the recovery. In general, the vehicle experienced a larger horizontal drift, as to be expected since the strategy does not control the position of the vehicle but its velocity and orientation. Accordingly, the final position reached by the vehicle after 8 seconds varied depending on the initial condition, but the vehicle was able to stabilize and maintain its position around the hover equilibrium solution when it was reached. The horizontal drift experienced by the vehicle velocity control compared to the goal position specified by the position controller for that case is illustrated in

Figure 5.10. On average, the difference between the two controllers was a drift of 0.7 m higher in the velocity control.

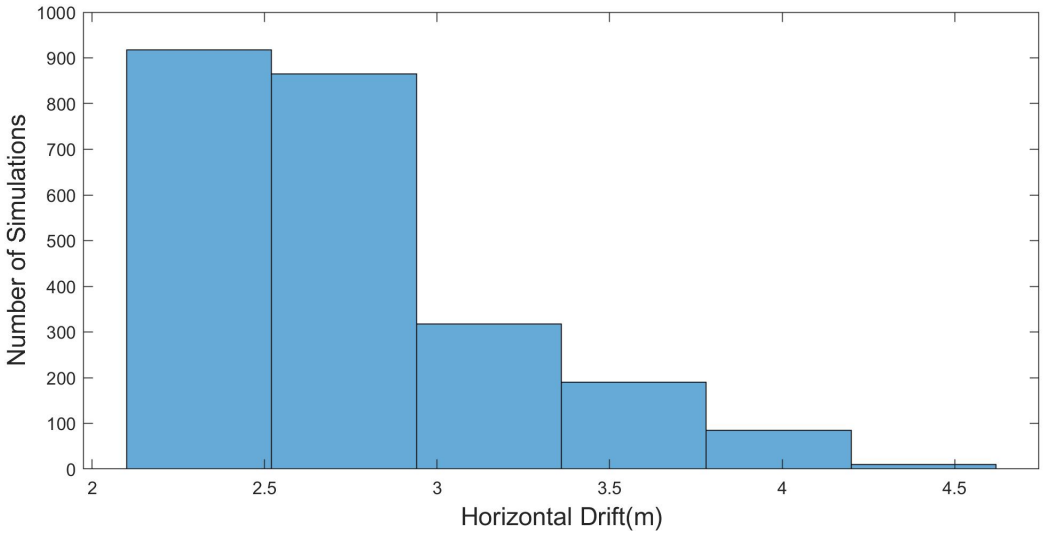


Figure 5.10: The absolute value of the change of the horizontal position of the vehicle from the desired final position found using position control for the velocity controller simulations that recovered

In simulations, the velocity controller was able to maintain the vehicle's position without tracking its position in the second stage of the velocity controller, although in a real life scenario this would be difficult to achieve since any external perturbations (example wind gust) would cause the vehicle to drift. Without position feedback and position control, it is recommended to land the vehicle as soon as possible since the longer the vehicle is in the air, the more it is at risk of drifting away.

5.3.5 Comparison to Recovery Control with Four Actuators

Comparing our recovery controller to that of AML recovery controller developed by Dicket et al. [11], can help to understand the limits of our recovery controller. The AML recovery controller is capable of recovering a vehicle from an

impact with a wall but without losing any propeller, at a wide range of initial conditions. Using the same initial conditions, the AML recovery results are compared to our recovery controller under a loss of actuator, as shown in Figure 5.11. Three sets of data where obtained, where the first set, color-coded blue, refers to the recoveries where both controllers were able to recover. The second set, color-coded green, which appears starting at 25° inclination angle, refer to recoveries where only the AML recovery was able to recover. Finally, the third set color-coded red refers to when both controllers fail to recover the vehicle.

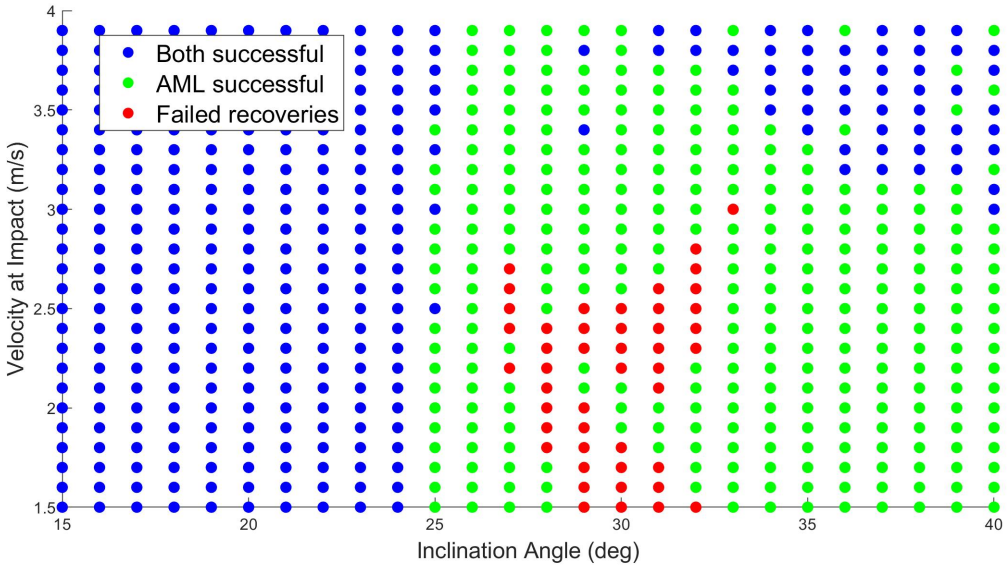


Figure 5.11: Simulated failures vs inclination and velocity of the recovery controller and the AML recovery controller

It should be noted that all the successful recoveries of the AML recovery controller past 32° inclination involve multiple bumper impacts as the one shown in Figure 5.7. As expected, the AML controller has a lower failure rate than the recovery controller under loss of actuator. In addition, the full actuator scenario does not exhibit a hard limit for recovery like what occurs for the 3 actuator scenario at 25°. Although both controllers were able to recover in the blue region, the AML

recovery controller was able to recover with less horizontal drift overall, again as expected since our controller has to recover to the periodic solution first, before moving towards the desired position.

5.4 Summary

In summary, a collision failure recovery strategy was developed and tested in simulation using a contact dynamics simulator between a bumper protected vehicle and a vertical wall. A less effective velocity controller was also developed and tested to replace the position controller in the failure recovery strategy in case of a missing position estimate. Batch simulations demonstrated a high success rate of recovery but also some unrecoverable situations. The initial conditions for the batch simulations were parametrized in terms of inclination angle and incoming velocity. The recovery rate is highly influenced by the inclination angle and less so by the velocity at impact; after a critical inclination angle of approximately 25° the recovery controller fails to recover at any incoming velocity. At high velocities and inclination angles, the simulations predict that the vehicle is able to recover after a double propeller impact, where the vehicle became flush with the wall; these cases are considered a false positives. Horizontal drift during recovery was also investigated and was shown to be significant and greatest for collisions near the critical inclination angle with high incoming velocity. In general, the quadcopter fails to recover and crashes when it locks into the vertical mode of failure. This mode of failure is only preventable by incorporating substantial hardware changes to the vehicle.

Chapter 6

Conclusion

6.1 Summary of Research

A collision recovery control strategy was developed to recover a vehicle from an impact with a wall that results in a loss of one actuator and to be able to move to a safe distance away from the wall. This is a challenging scenario since the vehicle has only three actuators to recover, stabilize and move away from the wall.

As a first attempt at collision recovery with three propellers, a recovery control law from [32] was presented that is capable of flying a vehicle with only three propellers. The strategy is to make the vehicle rotate freely around an axis fixed with respect to the body. The three remaining propeller thrusts are used to rotate this axis in the inertial frame which allows the vehicle to move to the desired location in 3D space. This hover state is refereed to as reduced hover, which is characterized by pre-computed equilibrium solutions, specific to the vehicle being used. After deriving the custom equilibrium solution for our custom vehicle Navi, simulations demonstrated that the recovery control law could be used to recover a vehicle after losing a propeller, from a wide range of initial conditions and orientations, to a state of reduced hover. The motion that a quadcopter undergoes after losing a

propeller is challenging to anticipate since it is highly affected by the initial conditions, i.e., the state of the vehicle when the actuator is lost. Horizontal drift is a major concern since the vehicle drifts from the desired position in order to converge to the equilibrium solution. This recovery control law was tested in Hardware in the Loop simulation in Gazebo simulation environment but failed to recover the vehicle due to missing drag terms in the Gazebo model. No further experiments were conducted due to the high likelihood of the vehicle crashing without tuning the controller parameters in a simulated environment first.

The recovery control law investigated earlier served as the basis for the collision recovery strategy for the scenario when the vehicle looses one of its actuators as a result of collision. The collision recovery pipeline is comprised of three stages. The first stage involves detecting the impact and estimating the wall normal direction. The second stage allows for two possible options depending on whether there exists a reliable position estimate of the vehicle. If the vehicle position estimate is available, the position controller computes the desired acceleration required to move away from the wall based on the output of the first stage; otherwise, the velocity controller computes the desired acceleration based on velocity estimates. Having a reliable position estimate is challenging in urban environments where high rise buildings interfere with the GPS signal. Also, motion capture systems used indoors (like VICON) face difficulties in tracking a fast rotating body [42], which motivated the development of a velocity control independent of a position estimate. Finally, the last stage recovers the vehicle's reduced attitude while producing the desired thrust of each propeller to match the desired acceleration. The collision recovery control strategy was verified with batch simulations covering a wide range of initial conditions. Overall, the strategy was able to produce recovery of the vehicle and to move it away from the wall in the majority of the simulations. It was concluded that

the control law is more sensitive to high inclination angles rather than impact velocity. Past a critical inclination angle of approximately 25°, the recovery controller failed to recover the vehicle at any incoming velocity. The failed simulations all consist of the same mode of failure, where the quadcopter becomes vertically stuck to the wall. Both the position and the velocity controller led to the same failure rate with the difference being a larger horizontal drift under the velocity controller, as to be expected. Finally, a comparison was made between the collision recovery controller with three actuators to the AML collision recovery controller with all four actuators, presented previously in [11]. Overall, the AML recovery had a higher recovery rate, and it did not exhibit a hard limit for recovery like what occurs for the 3 actuator scenario. Finally, the AML controller recovered with less horizontal drift overall.

6.2 Suggestions for Future Work

There are several paths that could be pursued for future work. First, the equilibrium solution for the reduced hover could be chosen in an optimized way to improve performance. It can be chosen such that the total power consumed by the quadcopter is minimized while in the reduced hover state; another alternative would be to choose an optimal solution that minimizes the total horizontal drift of the vehicle after impact.

A strategy for the collision recovery that was not pursued is to shut off the propeller opposing the one that was damaged by the impact; this may help the vehicle to avoid becoming "stuck" to the wall. The two still operational propellers could ensure the vehicle maintains its altitude (provided high enough thrust to weight ratio). At the same time, the torque produced by the two active propellers induces

a yawing moment which helps the vehicle to reach the equilibrium yaw rate faster. Also, this would relocate the propeller that was intentionally shut off near to the wall which could help in moving the vehicle to the desired position after activating the three propeller recovery controller. This strategy may potentially improve the response of the vehicle in recovering but it requires an additional layer of complexity for the two-propeller controller that can fly the vehicle.

Another path would be adding a detection of actuator failure module into the recovery pipeline. This along with the AML recovery using four propellers would result in a complete collision recovery strategy. The literature is extensive on fault-tolerant control and fault detection of actuators in aerial vehicles [43], [44], [45], but most fault tolerant controllers are tuned to detect an impact and switch controllers within 0.5 seconds. In our scenario, after impact is detected, 0.5 seconds may not be sufficiently fast enough since the AML recovery controller executing for 0.5 seconds after the impact with only three propellers would lead to a crash of the vehicle. Therefore, there is a need for designing a custom fault tolerant strategy. A proposed strategy would, upon detecting an impact, invoke the AML recovery controller as the default; the switch to the collision recovery with three propellers would occur if within the first few milliseconds of the recovery the vehicle is not able to pitch away from the wall.

Testing the firmware in a simulated environment would be the safest before conducting experiments. Simulation-In-Hardware (SIH) is a new promising alternative to Hardware in the Loop being developed by Chiappinelli at Coriolis-g Corporation [46]. In SIH, the controller, state estimator, and the simulator are all implemented and executed on the Pixracer board. In this setup, it is possible to provide our own dynamics model and specifically modify the aerodynamic model to execute on the board itself.

One interesting path that was not investigated in this thesis is to replace the standard uni-directional propellers of a quadcopter with bi-directional propeller, that is propellers capable of producing thrust in either direction. With such propellers, the stuck condition which was a common failure mode identified in simulations would likely be recoverable because thrust can be generated away from the wall. This would require major modifications to the control law but would improve the recovery rate. Designing a control law to fly the vehicle with three propellers, capitalizing on the use of bi-directional capabilities would yield interesting results.

Bibliography

[1] Gentry, Nicholas Kristofer, Raphael Hsieh, and Luan Khai Nguyen. "Multi-use UAV docking station systems and methods." *U.S. Patent No. 9,387,928*. 12 Jul. 2016.

[2] Dorling, Kevin, et al. “Vehicle Routing Problems for Drone Delivery.” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, 2017, pp. 70–85.

[3] Scott, Judy, and Carlton Scott. “Drone Delivery Models for Healthcare.” *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*, 2017.

[4] Lichtman, et al. “Humanitarian Uses of Drones and Satellite Imagery Analysis: The Promises and Perils.” *The AMA Journal of Ethic*, vol. 17, no. 10, 2015, pp. 931–937.

[5] Luppicini, Rocci, and Arthur So. "A technoethical review of commercial drone use in the context of governance, ethics, and privacy." *Technology in Society*, vol. 46, 2016, pp. 109–119.

[6] Holton, Avery E., et al. “Unmanned Aerial Vehicles:Opportunities, Barriers, and the Future of ‘Drone Journalism.’” *Journalism Practice*, vol. 9, no. 5, 2014, pp. 634–650.

[7] Stehr, Nikki J. “Drones: The Newest Technology for Precision Agriculture.” *Natural Sciences Education*, vol. 44, no. 1, 2015, p. 89.

[8] Achtelik, Michael, et al. "Design of a multi rotor MAV with regard to efficiency, dynamics and redundancy." *AIAA Guidance, Navigation, and Control Conference*, 2012.

[9] Bateman, Francois, et al. “Fault Diagnosis and Fault-Tolerant Control Strategy for the Aerosonde UAV.” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, 2011, pp. 2119–2137.

[10] Mendes, et al. “Assisted Teleoperation of Quadcopters Using Obstacle Avoidance.” *Journal of Automation Mobile Robotics and Intelligent Systems* , vol. 7, no. 1, 2013, pp. 54–58.

[11] Dicker, Gareth, Fiona Chui, and Inna Sharf. "Quadrotor collision characterization and recovery control." *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[12] Freddi, Alessandro, Alexander Lanzon, and Sauro Longhi. “A Feedback Linearization Approach to Fault Tolerance in Quadrotor Vehicles.” *IFAC Proceedings Volumes*, vol. 44, no. 1, 2011, pp. 5413–5418.

[13] Luukkonen, Teppo. "Modelling and control of quadcopter." *Independent research project in applied mathematics*, 2011.

[14] Wang, Pengcheng, et al. "Dynamics modelling and linear control of quadcopter." *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, 2016.

[15] Tayebi, A., and S. Mcgilvray. "Attitude Stabilization of a Four-Rotor Aerial Robot." *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, 2004.

[16] Fatan, Mehdi, Bahram Lavi Sefidgari, and Ali Vatankhah Barenji. "An adaptive neuro PID for controlling the altitude of quadcopter robot." *2013 18th International Conference on Methods Models in Automation Robotics (MMAR)*, 2013.

[17] Argentim, Lucas M., et al. "PID, LQR and LQR-PID on a quadcopter platform." *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 2013.

[18] Bouabdallah, Samir, Andre Noth, and Roland Siegwart. "PID vs LQ control techniques applied to an indoor micro quadrotor." *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, Vol. 3, 2004.

[19] Madani, Tarek, and Abdelaziz Benallegue. "Backstepping control for a quadrotor helicopter." *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[20] Zemalache, Kadda Meguenni, Lotfi Beji, and H. Marref. "Control of an underactuated system: application a four rotors rotorcraft." *2005 IEEE International Conference on Robotics and Biomimetics-ROBIO*, 2005.

[21] Martin, Philippe, and Erwan Salaün. "The true role of accelerometer feedback in quadrotor control." *2010 IEEE International Conference on Robotics and Automation*, 2010.

[22] Hunt and Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *J. of Appl. Mechanics*, vol. 42, no. 2, pp. 440445, 1975.

[23] Tomić, Teodor, Christian Ott, and Sami Haddadin. "External wrench estimation, collision detection, and reflex reaction for flying robots." *IEEE Transactions on Robotics*, vol. 33, no. 6, 2017, pp. 1467–1482.

[24] Gentili, Luca, Roberto Naldi, and Lorenzo Marconi. "Modeling and control of VTOL UAVs interacting with the environment." *2008 47th IEEE Conference on Decision and Control*, 2008.

[25] Galea, Brendan, and Paul G. Kry. "Tethered flight control of a small quadrotor robot for stippling." *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[26] Vempati, Anurag Sai, et al. "A Virtual Reality Interface for an Autonomous Spray Painting UAV." *IEEE Robotics and Automation Letters*. 2019.

[27] Chui, Fiona, Gareth Dicker, and Inna Sharf. "Dynamics of a quadrotor undergoing impact with a wall." *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.

[28] Faessler, Matthias, et al. "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor." *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[29] Lanzon, Alexander, Alessandro Freddi, and Sauro Longhi. "Flight control of a quadrotor vehicle subsequent to a rotor failure." *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, 2014, pp. 580-591.

[30] Lippiello, Vincenzo, Fabio Ruggiero, and Diana Serra. "Emergency landing for a quadrotor in case of a propeller failure: A backstepping approach." *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[31] Merheb, Abdel-Razzak, Hassan Noura, and François Bateman. "Emergency Control of AR Drone Quadrotor UAV Suffering a Total Loss of One Rotor." *IEEE/ASME Transactions on Mechatronics*, vol. 22, no .2, 2017, pp. 961-971.

[32] Mueller, Mark W., and Raffaello D'Andrea. "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers." *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[33] Mueller, Mark W., and Raffaello D'Andrea. "Relaxed hover solutions for multicopters: Application to algorithmic redundancy and novel vehicles." *The International Journal of Robotics Research*, vol. 35, no. 8, 2016, pp. 873-889.

[34] Diebel, James. "Representing attitude: Euler angles, unit quaternions, and rotation vectors." *Matrix* vol. 58, no.15, 2016, pp. 1-35.

[35] Agrawal, Kanaiya, and Punit Shrivastav. "Multi-rotors: A revolution in unmanned aerial vehicle." *International Journal of Science and Research* vol. 4, no. 11, 2015, pp. 1800-1804.

[36] Chaturvedi, Nalin A., Amit K. Sanyal, and N. Harris McClamroch. "Rigid-body attitude control." *IEEE control systems magazine* vol. 31, no. 3, 2011, pp. 30-51.

[37] Kao, Wei-Wen. "Integration of GPS and dead-reckoning navigation systems." *Vehicle Navigation and Information Systems Conference*, 1991.

[38] Mattos, Philips G. "Integrated GPS and dead reckoning for low-cost vehicle navigation and tracking." *Proceedings of VNIS'94-1994 Vehicle Navigation and Information Systems Conference*, 1994.

[39] Gruen, Armin, et al. "Joint processing of UAV imagery and terrestrial mobile mapping system data for very high resolution city modeling." *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W2, 2013, pp. 175–182.

[40] Bry, Adam, Abraham Bachrach, and Nicholas Roy. "State estimation for aggressive flight in GPS-denied environments using onboard sensing." *2012 IEEE International Conference on Robotics and Automation*, 2012.

[41] Patire, Anthony D., et al. "How much GPS data do we need?." *Transportation Research Part C: Emerging Technologies*, vol. 58, 3025, pp. 325-342.

[42] Merriaux, Pierre, et al. "A study of VICON system positioning performance." *Sensors*, vol. 17, no.7, 2017, p. 1591.

[43] Saied, Majd, et al. "Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor." *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[44] Sadeghzadeh, Iman, et al. "Fault-tolerant trajectory tracking control of a quadrotor helicopter using gain-scheduled PID and model reference adaptive control." *AIAA Guidance, Navigation, and Control Conference*, 2011.

[45] Freddi, A., S. Longhi, and A. Monteriu. "A model-based fault diagnosis system for a mini-quadrotor." *7th workshop on Advanced Control and Diagnosis*, 2009.

[46] Chiappinelli, Romain. "Simulation-In-Hardware." 2019, DOI: https://github.com/romain-chiap/PX4_SIH_QuadX