THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MECHANICAL ENGINEERING

DESIGN AND ANALYSIS OF INVERTED LANDING MECHANISM AND BIOMIMETIC STRATEGY FOR SMALL-SCALE QUADROTORS

BRIAN DAVIS
SPRING 2021

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Mechanical Engineering
with honors in Mechanical Engineering

Reviewed and approved* by the following:

Bo Cheng
Associate Professor of Mechanical Engineering
Thesis Supervisor and Honors Advisor

Jean-Michel Mongeau
Assistant Professor of Mechanical Engineering
Faculty Reader

* Electronic approvals are on file in the Schreyer Honors College.

ABSTRACT

Insects such as flies combine sensory inputs, processing, control, and mechanical intelligence to execute complex landing behaviors. Maneuvers such as landing upside down on a ceiling require complex processing and control. A deeper understanding of these processes will provide insight into implementing sensors, controls, and mechanical design for small biomimetic quadrotors with limited weight and processing capacity.

This research built on previous studies into the behaviors of flies to examine the act of landing on a ceiling with a small quadrotor. Aspects of landing such as robustness under different impact conditions, control algorithms, and sensory inputs were examined. In this study, a landing mechanism was designed to mount on a small quadrotor with the objective of increasing its landing robustness. A simulation environment was used to evaluate the impact of the landing mechanism design on the landing success rate and the control policy. Relative retinal rate of expansion (RREV) was discussed as one of the key sensory inputs.

Control policies were derived by assuming ideal, instantaneous control and 2-D projectile motion during the rotation maneuver. These policies were evaluated in a simulated environment to provide a theoretical solution to compare with the learned behavior. The sensitivity of each control algorithm to the RREV input and the impact of error and uncertainty were examined. A method for increasing the landing mechanism performance was proposed, and a simulation environment was deployed to optimize landing mechanism geometry to increase robustness. In addition, to assist with future experiments, a launching platform was designed, fabricated, and tested to provide an initial vertical velocity to the quadrotor for testing under a wider range of conditions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This thesis is the result of the support of my faculty advisors, colleagues in the Biological & Robotic Intelligent Fluid Locomotion (BioRob-InFL) Lab, and family.

Thank you to Dr. Bo Cheng for his continuous support and guidance throughout my undergraduate career as my Honors Advisor. Dr. Cheng also served as my Thesis Supervisor, providing invaluable feedback and guidance throughout my research experience. I would also like to thank Dr. Jean-Michel Mongeau for his support of my research as my Faculty Reader. I appreciate the thoughtful feedback he provided during the writing of the thesis.

Thank you to Yixian Li and Pan Liu, who both welcomed me onto the research team from the beginning and helped me to become familiar with the inverted robotic landing research. I also had the privilege of working with Bryan Habas and Bader Al-Attar throughout the research process. Bryan and Bader were always happy to help and discuss difficult problems as they arose.

Finally, thank you to my family for their unwavering support throughout my academic career and this research project. In particular, thank you to my grandfather, Herman Maurer, who I had the pleasure of working with to fabricate the launching platform for this research. Some of my fondest memories from childhood include designing and building in his workshop with him, and it was incredible to work with him once again in that same workshop as part of this thesis.

This research was supported by the National Science Foundation (IIS-1815519 and CMMI-1554429). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Chapter 1

Background

Introduction

The study of unmanned aerial vehicles is a rapidly expanding field with vast applications across a variety of industries and disciplines. Numerous researchers have studied and documented the complex maneuvers observed in flying insects [2, 9, 10]. However, the insights garnered through this research have not been widely applied toward advancing the capabilities of small-scale aerial robotic systems that can execute similarly complex maneuvers with a limited sensor suite. The goal of this research is to advance the capabilities of small unmanned aerial vehicles by designing a robust, lightweight landing mechanism to allow for a small quadrotor to conduct complex maneuvers such as inverted landings. Additionally, this research explores the control of small robotic systems and the impact of the landing mechanism on success and failure. Design objectives and constraints were informed by research on the dynamics and perception of small flying insects. The mechanical design was guided by research on the landing mechanisms in nature and existing designs of other unmanned aerial systems that could be modified and adapted to the challenge of inverted landings for small unmanned aerial systems.

Perching and Perception

The development of robust landing and perching ability is important to the advancement of small-scale Unmanned Aerial Vehicles (UAVs) because energy efficiency during flight decreases as size decreases [9]. As UAVs become smaller, their ratio of surface area to mass increases, resulting in lower efficiency and less payload capacity [5]. The ability for small aerial robotics to perch like natural fliers would help save energy and extend mission times [9].

Research into the methods used by small flying insects to perceive the environment helps to inform the design of a robust landing mechanism. Researchers studied flies to determine the variables that trigger rotational maneuvers before impact with a surface [10]. They found that the flies used optical flow, a method that uses relative motion of light gradients to deduce motion [10]. Specifically, the flies use relative retinal expansion velocity (RREV) to perceive distance from a surface [10]. RREV works by tracking the rate of expansion of the optical flow. As a fly approaches a surface with constant velocity, the rate of expansion increases, and this information can be used to trigger a rotational maneuver to reorient the body and impact the surface with the legs [1]. Research on honeybees supports the findings that the rate of expansion in an image is used by flying insects to inform landing behavior [1]. Bees have developed a robust landing control system by holding the rate of relative retinal expansion constant, ensuring that speed is reduced “gradually and automatically” [1]. Since this method of landing does not require information on precise velocity or distance, it can be employed across a variety of small natural and robotic fliers [1]. Researchers also observed a tight correlation between the body inclination of honeybees during landing and the slope of the landing surface, indicating that bees can perceive the slope of the surface and adjust landing dynamics accordingly [4]. Methods of environmental perception are important in biomimetic robotic fliers. The on-board sensors provide key environmental information to the controller, which determines the motion of the robot. For this design process, it was assumed that the small quadrotor would have a single upward facing optical flow sensor and would use RREV to trigger rotational maneuvers.

Trajectory Generation and Touchdown

Studies on the trajectory generation and control of small biological fliers provide insight into the dynamic behaviors that could be replicated by small quadrotors while executing complex maneuvers. The landing mechanism must be able to accommodate the sequences of maneuvers utilized by flies and other biological fliers to enable the robots to mimic similar behavior. The three key landing modules observed in fruit flies are turning, deceleration, and leg extension [2].

A fly's decision to land requires a deliberate choice to maintain a collision course and override the obstacle avoidance algorithms typically used in flight [2]. This sequence of coordinated behavioral modules was studied for inverted landings of flies by recording landing sequences and using statistical analysis to determine commonalities across a variety of fly landings [6]. Some research has focused on the concept of applying separate control modules like those observed in flies and other natural fliers to larger scale UAV's [8]. Researchers were able to demonstrate the effectivity of this approach by successfully planning complex trajectories for larger UAV's and demonstrating successful perching on a slanted wall [8]. However, this research did not extend to attempting fully inverted landings [8].

Observation of touchdown patterns in flies helps to inform the design of a robust landing mechanism for small aerial robots to land in a similar manner. Flies wait to extend their legs until they are about to impact the surface [2]. This behavior of leg extensions prior to landing was consistently observed across several studies and could be mimicked in the landing mechanism of a small robotic flier. Experimental observations of fly landings found typical impact velocities of 0.39 +/- 0.13 m/s [2]. While small-scale robotic UAVs could differ significantly from the velocities observed in biological fliers, these observations of flies provide a reference when designing a mechanism to withstand the force of impact.

Robustness and Failure to Land

While observing fly behavior, researchers recorded natural fliers crashing during failed landings. This increases the importance of recovery after a failed attempt and the mechanical strength of the design to avoid damage to biomimetic robots during suboptimal landings. While small aerial robots, like their biological counterparts, will attempt to conduct every landing flawlessly, experimental data suggests that small insects frequently fail to stick their landings. In one study, researchers observed that flies crashed during 35.7% of attempted landings [2]. Most of these crashes were due to insufficient deceleration of the fly and legs that were not extended quick enough [2]. The landing mechanism can be designed to allow for successful landing for a variety of landing approaches. However, acceleration and velocity errors during landing maneuvers are largely due to perception and control challenges that are independent of mechanical design. The observation that flies' legs did not fully extend and that failed landings resulted from high velocity impacts informs the necessity of a landing mechanism design that is always extended or that consistently deploys before impact. Additionally, the landing mechanism should accommodate a variety of velocities and approach angles to improve the robustness of landing.

Advanced Biomimetic Robots

A variety of research has focused on developing and demonstrating advanced biomimetic aerial robotic capabilities. One researcher applied the landing techniques observed in birds and insects to create a small-scale robot weighing only 100 grams that used adaptive mechanical dampers and electro-adhesion to perch on a variety of structures [5]. Larger animals such as birds can rely on more complex sensing and processing for precise deceleration and control of landing orientation [5]. Insects rely partially on "mechanical intelligence," with some using their legs as soft, compliant structures to absorb the impact of landing and stick to the surface [5]. One robotic landing approach uses an ultrasonic sensor for sensing and splines to grip the surface to allow a small, fixed wing aircraft to perch on walls [3]. The splines consist of metal hooks and enable the 400-gram robot to climb and release from wall surfaces [3]. This robot uses a "virtual knee' in the landing mechanism consisting of elastic components with minimal mass to absorb shock upon landing [3]. Another small-scale biomimetic robot uses piezoelectric muscles and flapping wings to maintain steady flight [7]. The flapping robotic flier is similar in size to a penny; however, it is tethered for power supply purposes and lacks the ability for environmental perception [7]. Each of these projects has advanced the capabilities of small aerial robotics by applying nature's solutions to artificial fliers. Components of these demonstrations can be used to improve the design of a robust landing mechanism for small biomimetic quadrotors and to provide insight into experimental design, testing, and analysis.

Purpose of Research Project

The goal of this research is to build upon the previous work examining landing behaviors of small biological animals. The landing behavior was decomposed into perception, control, and mechanical design. Key factors in successful and failed landings were studied, and trends were analyzed to inform the design of a landing mechanism and controller that will achieve consistent landing. These components were examined with the goal of creating a biomimetic quadrotor that can replicate the complex landing behavior observed in flies with limited sensory inputs and processing complexity.

Chapter 2

Mechanical Design of a Landing Mechanism

Design Intent

The Crazyflie quadrotor is a small commercially available robotic flier that was chosen for use in experimental trials. To allow the robot to remain connected to the landing surface after impact, the ceiling surface was covered in Velcro. Each landing mechanism design involved using Velcro on the tips of the robot's legs to stick to this landing surface. Velcro was chosen to reduce the emphasis on the method of adhesion. This assumption could be relaxed in future studies to achieve landing on other types of surfaces. The purpose of the landing mechanism for these experiments was to serve as an interface between the quadrotor body and the landing surface and facilitate successful landings. A quality landing mechanism should increase the consistency of landing under a wide range of impact conditions. This is important because sensor noise and limited computational power will make it difficult for the controller to achieve precise and repeatable impact conditions. If the landing behavior varies due to these and other factors, the mechanical intelligence should allow for successful landings even when an optimal landing process is not achieved. The controller should learn with this landing mechanism to take advantage of the areas of robustness and weakness inherent in the mechanical design. Similarly, the landing mechanism should be designed to optimize performance in situations that are attainable by the robotic flier.

Constraints

The design of the landing mechanism has several key considerations and constraints. Minimizing weight on every part of the robot, including the landing mechanism, is critical. This is because the robot needs to accelerate significantly before and during the landing process. Without a high enough thrust to weight ratio, it is difficult or impossible for the robot to achieve a high enough vertical velocity to successfully land on the ceiling. The landing mechanism must be able to sustain a surface impact at a velocity of at least 4 m/s at any impact angle without damage. Impact with a surface could occur as the robot impacts the ceiling or when it falls back to the ground after a failed landing. The landing mechanism should have a small aerodynamic profile to minimize drag, especially as it is in the downwash from the rotors. The landing mechanism should not move significantly during flight, as this could create less consistent impact behavior and cause dynamic issues during flight. Lastly, the landing mechanism must provide a stable base for taking off and landing on the ground.

Mechanical Design and Manufacturing

The Crazyflie quadrotor comes with rigid plastic legs installed. The geometry and stiffness of these legs are not variable, but they provide a light-weight mechanism for remaining upright when the Crazyflie takes off or lands on the floor. The first iteration of improved landing mechanism replaced the default mechanism with carbon fiber rods mounted to an additively manufactured structure. These rods were variable in length and angle; however, they were stiff. The next iteration in the landing mechanism design allowed for modification of both geometry and stiffness. This design involved a mounting mechanism for each leg to be attached to the

Crazyflie body and thin additively manufactured legs made of PETG plastic. These legs were quick and simple to manufacture, light weight, and could be made in a variety of stiffnesses by varying the thickness of the leg. However, the more complex spring constant and damping of the additively manufactured legs were more difficult to accurately model in simulated environments. Nonetheless, this was the design that was chosen for testing with the Crazyflie. An early design of the additively manufactured landing mechanism is shown in Figure 1. A photo of the updated landing mechanism mounted to the Crazyflie quadrotor is shown in Figure 2.

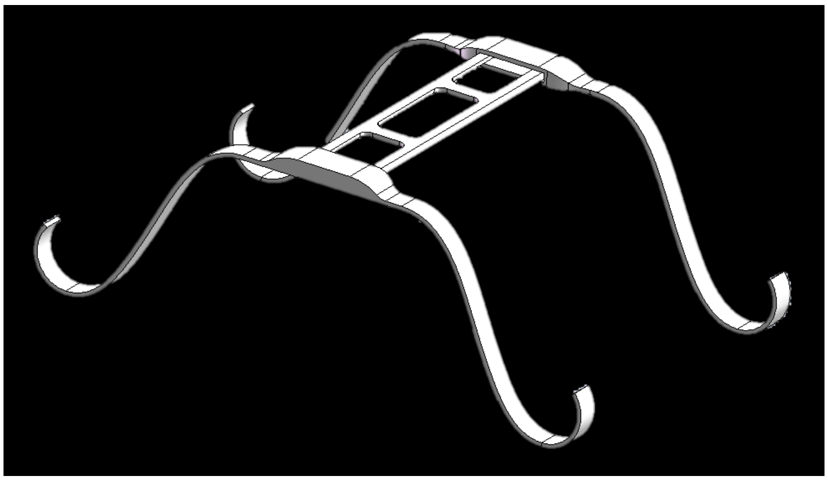

Figure 1: Landing mechanism with flexible body design.

Figure 2: Landing mechanism mounted on Crazyflie quadrotor.

Chapter 3

Simulation of Landing Mechanism

Purpose of Simulating Landing Mechanism

Simulated environments were created to explore various factors of the landing and control process. These environments allowed more trials for machine learning to occur than would have been possible in a purely experimental setup. The simulated environments were less time and cost intensive, as physical demonstrations involve significant time to set up and fix components that are damaged or malfunction during the trials. The simulated environment allowed for the rapid modification and comparison of landing mechanism designs. If it could be shown that the simulations successfully modeled the physical behavior, the number of experimental trials could be reduced. This process of exploring and learning in simulations and transferring the controls and design to a mechanical robot has wide applications outside of this experiment.

3-D Simulation Using Gazebo

One 3-D simulation environment that was used for similar fly landing research was Gazebo. Gazebo with the Robot Operating System (ROS) allowed for modeling of the quadrotor, sensors, and the controller in a realistic simulation. The geometry of the landing mechanism could be modified, and the spring constant and damping coefficient could be tuned to match the experimental trials. In this simulation, it was assumed that there was no air resistance, and the legs would stick if they touched the ceiling. This environment was more realistic than a 2-D simulation; however, it was also more computationally intensive and took longer to run each trial. A model of the Crazyflie quadrotor in the Gazebo simulation environment is shown in Figure 3. Due to the computational intensity of the 3-D simulation, it was not used for this research. Instead, a 2-D simulation environment was developed and used.

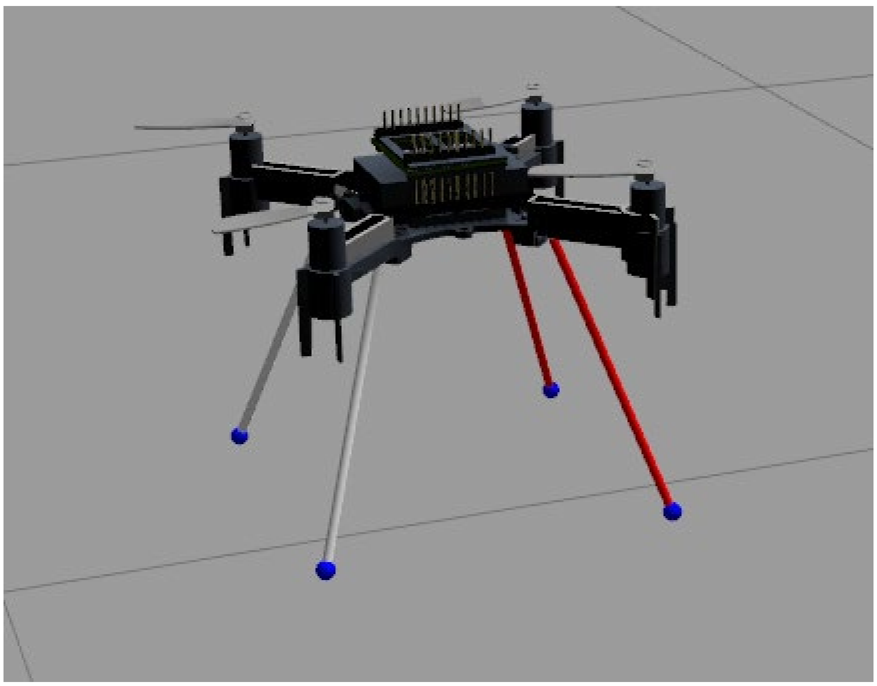

Figure 3: Crazyflie model in Gazebo simulation.

2-D Simulation Using MATLAB

A 2-D simulation environment was developed in MATLAB to provide an additional tool for analyzing landing behavior. The purpose was to provide a less computationally intensive means of exploring the landing process. Any part of the simulation could be easily modified to adjust assumptions, gather data, and investigate specific behaviors. This simulation environment helped to gain and intuitive understanding of the relationships between landing mechanism geometry and key impact parameters such as velocity and impact angle. It could provide an idealized set of trends and reveal patterns that could be further explored in the Gazebo simulation and experimental trials. This environment could be used as a tool to determine potential strategies for landing. Based on the findings, the impact of factors such as landing mechanism geometry on what impact conditions were successful and how the learned control policy would be expected to change were characterized.

There are several assumptions made in the two-dimensional model to reduce noise and simplify the calculations. The model assumed that the landing behavior could be reasonably represented in a two-dimensional space and that the entire robot, including the landing mechanism, was a rigid body. The center of mass was modeled as being in the center of the robot body, and the robot was modeled as an ideal type 3 projectile with no air resistance. Gravity was assumed to be active, although gravity can be turned off if desired. Legs are assumed to have their tips stick to the ceiling upon impact regardless of force, and an ideal hinge joint is formed between the leg tip and the ceiling. When the quadrotor first contacts the ceiling, conservation of angular momentum was applied about the contact point to determine the initial angular velocity of the body about the rotation point a moment after impact. The simplified geometric model of the quadrotor for 2-D simulation is shown in Figure 4.

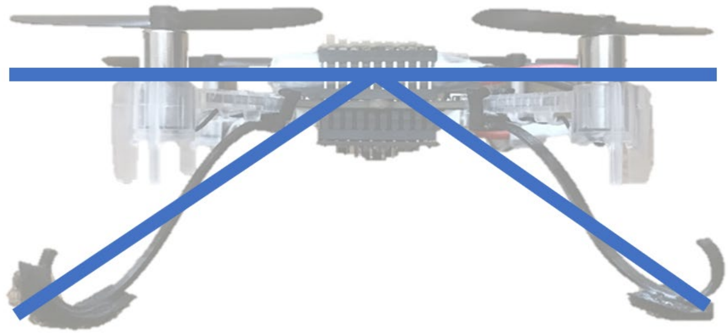

Figure 4: Simulation geometry model in 2-D.

The initial conditions of the simulation are specified by the user. These initial conditions include horizontal velocity (m/s), vertical velocity (m/s), angular rotation rate (deg/s), angular position (deg), position (assumed to be 0,0), and ceiling height (distance above initial position of vehicle). The simulation uses a numerical integration solver with constant time steps to evaluate the dynamic behavior of the system. This method made it simple to add additional parameters such as modeling rotor thrust during rotation without significant modifications to code. Some drawbacks of this method include accumulated error and deviation from true behavior if computational resources were limited and time steps were too large. In addition, this method was more computationally intensive than building a purely analytical solver. The solver tracked key aspects such as center of mass position, velocity, body orientation, and angular velocity. Based on these variables, other key points such as leg and body tips were calculated for each time step.

The simulation was created with a two-stage modeling approach. The first stage was flight before impact. This stage tracked the center of mass assuming ideal projectile motion. The orientation of the body along with translational and angular velocities were tracked. The second stage was the motion after any part of the robotic flier contacted the ceiling. To interface between the flight and rotation stages, conservation of angular momentum was applied about the leg tip that served as a pivot when the robot first touched the ceiling. This was used to determine the initial angular velocity of the body about the leg-ceiling interface pivot. This transition was triggered when any tracked point came within a defined margin of the ceiling plane. The key points were defined as the leg and body tips. Together, these four points formed a convex polygon around the robot, such that one of the four tracked points must contact a flat surface, in this case the ceiling, before any other part of the robot body contacted the ceiling. The simulation after impact and during the pivot process was subject to several termination conditions. If either body tip point touched the ceiling before the other leg tip, the landing was considered a failure. If the direction of rotation changed, indicating the vehicle did not reach the ceiling during its initial direction of swing, the landing was considered a failure. If, however, the second leg tip touched the ceiling, the landing was considered complete and successful.

The user interface allows the user to set initial conditions as well as computational and display factors. The initial distance from ceiling, orientation, horizontal and vertical velocity, and angular velocity can all be changed. Additionally, the geometry of the body and legs can be modified. The time step for computation can be adjusted, with smaller time steps leading to more accurate but also more computationally intensive calculations. The user has the option to select either a static, dynamic, or no visualization of the simulation to be shown after the computation. Figure 5 shows an example of a static visualization of the landing process.

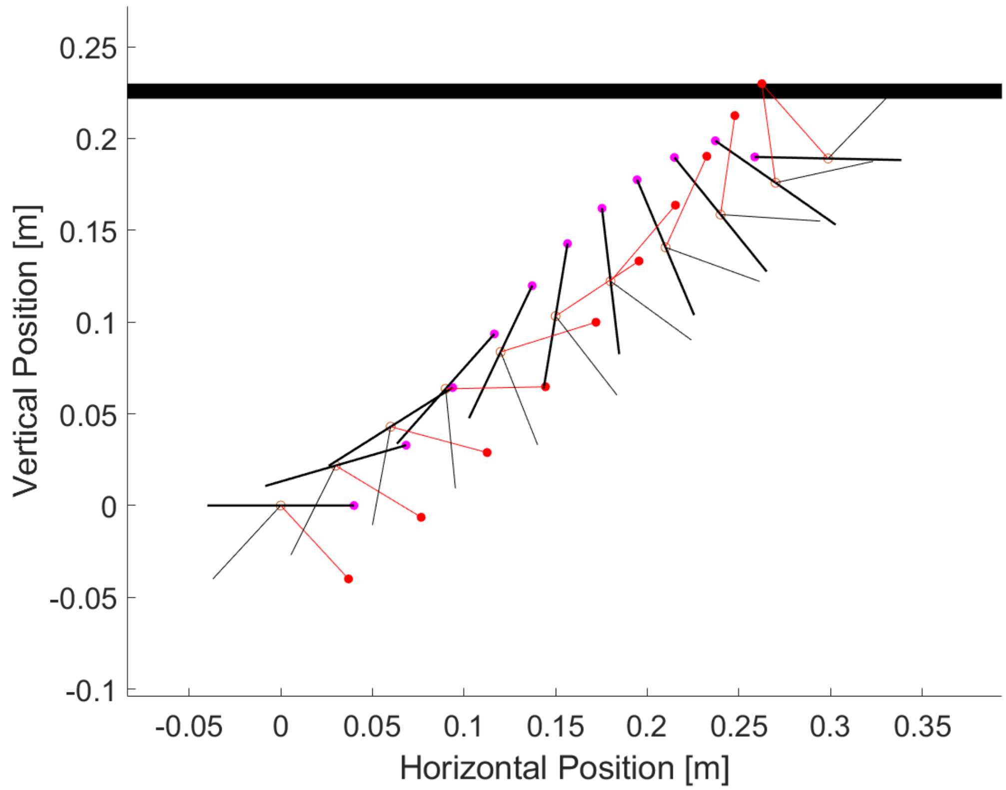

Figure 5: Landing simulation visualization with time step of 7.5 milliseconds.

The static simulation represents the entire flipping process in a single static visual, with snapshots of the robot displayed at fixed time steps. The benefit of this visualization is that it can help to provide an intuitive visualization of the landing process in a single image. This visualization tracks key points such as leg and body tips throughout the landing process to better understand the impact of geometry on the landing process. Relative translational and angular velocities can be discerned from the visual by observing the distance between snapshots of the vehicle. As lines come closer together, this indicates that the vehicle has moved less over the same time step and is therefore moving slower. The drawbacks of this visualization include it being less intuitive to understand than a dynamic visualization and that time steps must be chosen carefully to achieve a reasonable depiction of the landing process.

The alternative visualization is a dynamic video of the landing. This method shows the entire landing process at either full or slow-motion speed. This shows the exact behavior of the robot. The entire landing process occurs in a fraction of a second, so the slow-motion option is useful for understanding why a particular landing attempt resulted in success or failure.

Summary of 2-D MATLAB Simulation Assumptions

The 2-D simulation makes several assumptions to model the physical behavior and controls of the quadrotor. The key assumptions are as follows:

Physical Modeling Assumptions:

- Ideal projectile motion during rotation and before impact
  - No air resistance
- Only torque imparted on quadrotor by propellers during rotation.
  - no net translational components of acceleration other than gravity
- Point mass at center of line representing body (rotation about this point during flight)
  - Negligible mass of legs
- Conservation of angular momentum about tip of pivot leg

Sensing Assumptions:

- RREV measured from sensor located at center of mass.
- Sensor measurements are accurate unless otherwise stated.

Control Assumptions:

- Instantaneous Control
  - Commanded angular velocity or acceleration is achieved instantaneously.

Landing Behavior Assumptions:

- Assume quadrotor will begin in a level (no angular displacement) position.
- Rigid body and landing mechanism

These assumptions were chosen to model the ideal behavior of the quadrotor. The specific assumptions were modified to match the needs of each simulation. For example, changes were made to the controller to reflect angular velocity or angular acceleration-based control depending on which control algorithm was being tested.

Chapter 4

Landing Control Strategy

Use of Relative Retinal Expansion Velocity (RREV)

One of the primary sensory inputs the robot uses for landing on the ceiling is relative retinal rate of expansion (RREV). This value is calculated based on readings from an optical flow sensor, and a similar value is believed to be used by flies to calibrate their actions to the surroundings. RREV is based on the concept that given a fixed field of view, if a viewer approaches a surface any fixed point on the surface will appear to move outward. This outward movement of all points increases as an observer gets closer to a surface and/or is traveling towards the surface at a higher speed. This value is equal to velocity divided by the distance from the surface. In the absence of acceleration after the reading, impact time is equal to 1/RREV, as 1/RREV is equal to distance/velocity. In the case of landing on the ceiling, however, this impact time calculation is more complicated because gravity results in a change of velocity after the sensor reading and before landing. This increases the impact time beyond what would be predicted by the simple 1/RREV approximation. Nonetheless, RREV provides key insights into the landing process and, through these studies, is shown to be sufficient to inform consistent impact conditions.

Investigating RREV-based Control with Theoretical Modeling and Simulation

To investigate the relationship between one of the key controller inputs, RREV, and control outputs such as rotation velocity and angular acceleration, theoretical controller algorithms were derived for the simplified 2-D landing case. Each of these models makes simplifying assumptions such as there being no air resistance and that the robot is attempting to land "flat," with all legs impacting the ceiling simultaneously after a 180-degree rotation. It is assumed that the optical flow sensor is pointing directly upwards toward the ceiling and that the optical flow sensor is located at the center of mass of the flight vehicle. The following derived control policies were intended to provide an analytical model for an idealized vehicle.

While each individual policy is computationally intensive and may not be entirely accurate given the complexities of the robot not included in the calculations, these algorithms can provide a guide of what patterns should be expected in the learned behavior. For example, these algorithms can provide insight into the relationship between RREV and rotation rate, whether linear, quadratic, or otherwise and how this relationship varies or remains constant with distance and velocity. Analytical means could be used to understand why a particular learned policy may work in certain regions, and the structure of the learning could be adjusted to fit the understanding of the landing process through theoretical models. The relationship between RREV and impact time can be understood as a function of velocity and distance from the ceiling in the presence of gravity. Additionally, the impact of sensor error and the limits of using RREV to inform the landing process were explored to form deeper understanding of the nature of the problem.

Four landing control policies were derived. Policy 1 assumed that the robot rotates with constant angular velocity. Policy 2 assumed that the robot rotates with a constant, pre-set acceleration for a calculated duration. Policy 3 assumed that the robot rotates with a constant, calculated angular acceleration with motor shutoff at a predefined angular position of 90 degrees.

Policy 4 assumed constant acceleration until a specific condition that is a function of current position and angular rate is satisfied.

Note that for Landing Control Policies 2, 3, and 4, angular acceleration was chosen as the controller output to provide a generalized solution that is independent of the robot's moment of inertia. If implemented on a specific quadrotor, however, the control output would be a moment that would achieve the targeted angular acceleration. This moment would be achieved by sending a specific rpm command to the rotors. Thus, these acceleration-based control algorithms are open loop.

Impact Time for Analytical Control Policy Development

$H_c = Vertical\ distance\ between\ center\ of\ mass\ and\ ceiling\ at\ start\ of\ rotation\ (m)$

$H_L = Vertical\ distance\ between\ leg\ tip\ and\ center\ of\ mass\ (m)$

$\Delta y = H_c - H_L: Vertical\ distance\ between\ initial\ and\ final\ position\ of\ center\ of\ mass$

$t: Impact\ time\ (seconds) from\ beginning\ of\ rotation\ to\ leg\ contact\ with\ ceiling$

$V_{y_i}: Initial\ vertical\ velocity\ (at\ beginning\ of\ rotation)\ (\frac{m}{s})$

$g: Gravitational\ acceleration\ constant\ (9.81\,\frac{m}{s^2})$

$\theta_d: Desired\ angular\ displacement\ from\ start\ of\ rotation\ to\ landing$

$\omega: Angular\ velocity\ (\frac{degrees}{s})$

$RREV: Relative\ retinal\ expansion\ velocity\ (calculated\ from\ optical\ flow\ sensor)$

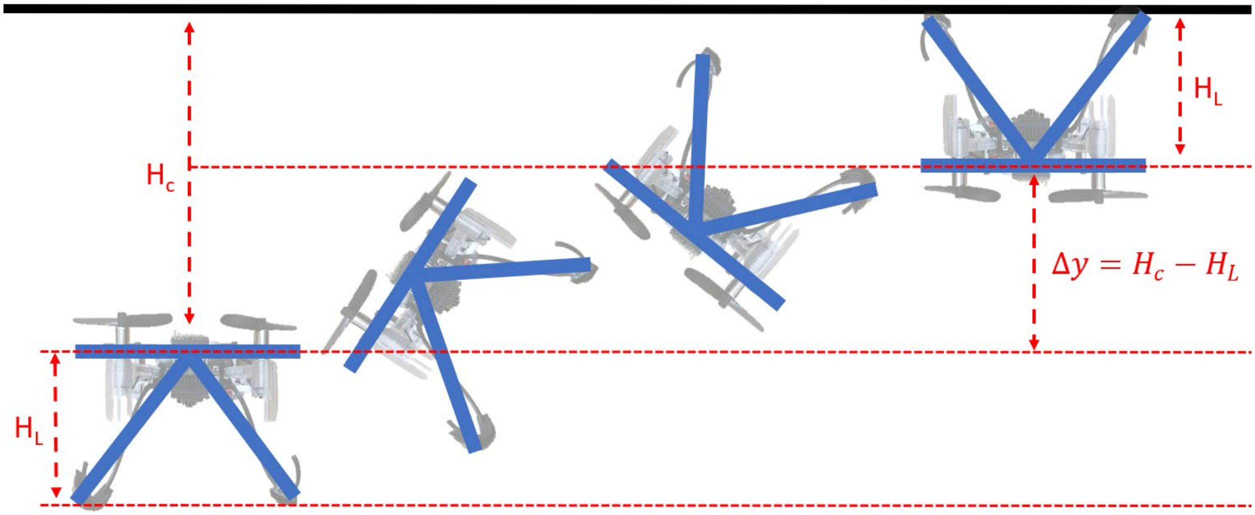

Figure 6: Key control policy dimensions.

$$\Delta y = V_{y_i} * t - \frac{1}{2} * g * t_{impact}^2 \text{ : } Projectile\ Equation$$

$$\frac{1}{2} * g * t_{impact}^2 - V_{y_i} * t_{impact} + \Delta y = 0 \text{ : } rearanged\ equation\ to\ put\ in\ quadratic\ form$$

$$t_{impact} = \left(\frac{1}{g}\right) * (V_{y_i} - \sqrt[2]{V_{y_i}^2 - 2 * g * \Delta y}\,)\text{: } calculated\ impact\ time$$

$$\boldsymbol{t_{impact} = \left(\frac{1}{g}\right) * \left(V_{y_i} - \sqrt[2]{V_{y_i}^2 - 2 * g * (H_c - H_L)}\right)} \tag{4.1}$$

Equation 4.2 and Equation 4.3 show the relationship between V_y, H_c, and RREV. If two of these three values are known or estimated, the third value can be calculated. Impact time can be calculated as a function of RREV and distance from ceiling by substituting Equation 4.2 for V_y in Equation 4.1. Similarly, impact time can be calculated as a function of RREV and vertical velocity by substituting Equation 4.3 for H_c in Equation 4.1.

$$\boldsymbol{V_{y_{derived}} = RREV * H_c} \tag{4.2}$$

$$\boldsymbol{H_{c_{derived}} = \frac{V_y}{RREV}} \tag{4.3}$$

Summary of Idealized Control Policy Assumptions

The landing control policies were derived by making several assumptions about the physical behavior and control behavior of the quadrotor. The key assumptions are as follows:

Physical Modeling Assumptions:

- Ideal projectile motion during rotation and before impact
  - No air resistance
- Only torque imparted on quadrotor by propellers during rotation.
  - no net translational components of acceleration other than gravity
- Point mass at center of line representing body (rotation about this point during flight)
  - Negligible mass of legs

Sensing Assumptions:

- RREV measured from sensor located at center of mass.

Control Assumptions:

- Instantaneous Control
  - Commanded angular velocity or acceleration is achieved instantaneously.

Landing Behavior Assumptions:

- Assume quadrotor will begin in a level (no angular displacement) position.
- Assume quadrotor seeks to land completely inverted on ceiling.
  - All legs touch simultaneously.

These assumptions were applied in Landing Control Policies 1, 2, 3, and 4. Additionally, these assumptions were consistent with the 2-D MATLAB simulation environment where these landing control policies were applied and analyzed.

Landing Control Using RREV: Policy 1- Constant Angular Velocity

Control Policy 1 assumed that the robot rotated with a constant angular velocity from the start of rotation through initial contact with the ceiling.

Derivation:

$$\theta_d = \omega * t_{impact}: estimated\ angular\ displacement\ at\ contact\ with\ ceiling$$

$$\omega = \frac{\theta_d}{t_{impact}} : commanded\ rotation\ control\ (\frac{degrees}{s})$$

$$\boldsymbol{\omega = \frac{180}{t_{impact}} \left(\frac{degrees}{s}\right) : where\ \theta_d = 180\ degrees} \tag{4.4}$$

This result shows the angular velocity that will allow the robot to land approximately flat on the ceiling as a function of the sensor reading RREV, known constants, and the distance from the ceiling or vertical velocity. The distance from ceiling and velocity are values that would not be known by the robot and therefore must be estimated to use the algorithm.

Figure 7 and Figure 8 show the calculated rotation rate as a function of RREV using Control Policy 1 shown in Equation 4.4. Figure 7 shows the differences in commanded angular velocity as a function of vertical velocity. Figure 8 shows the differences in commanded angular velocity for different distances from the ceiling. These figures show that the same RREV can result in different required angular velocities to land flat on the ceiling. Without the specific information on velocity or distance from ceiling, a controller operating on RREV alone must find a compromise policy that will work over a range of distances and velocities.

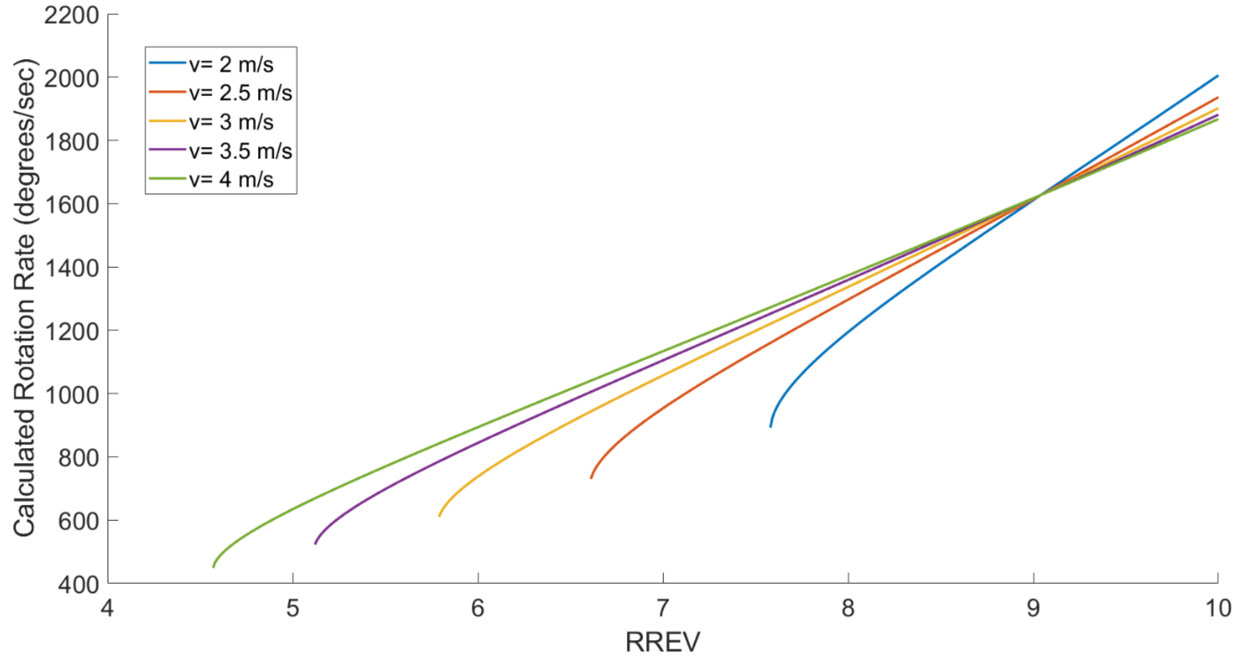

Figure 7: Rotation rate vs RREV and vertical velocity.

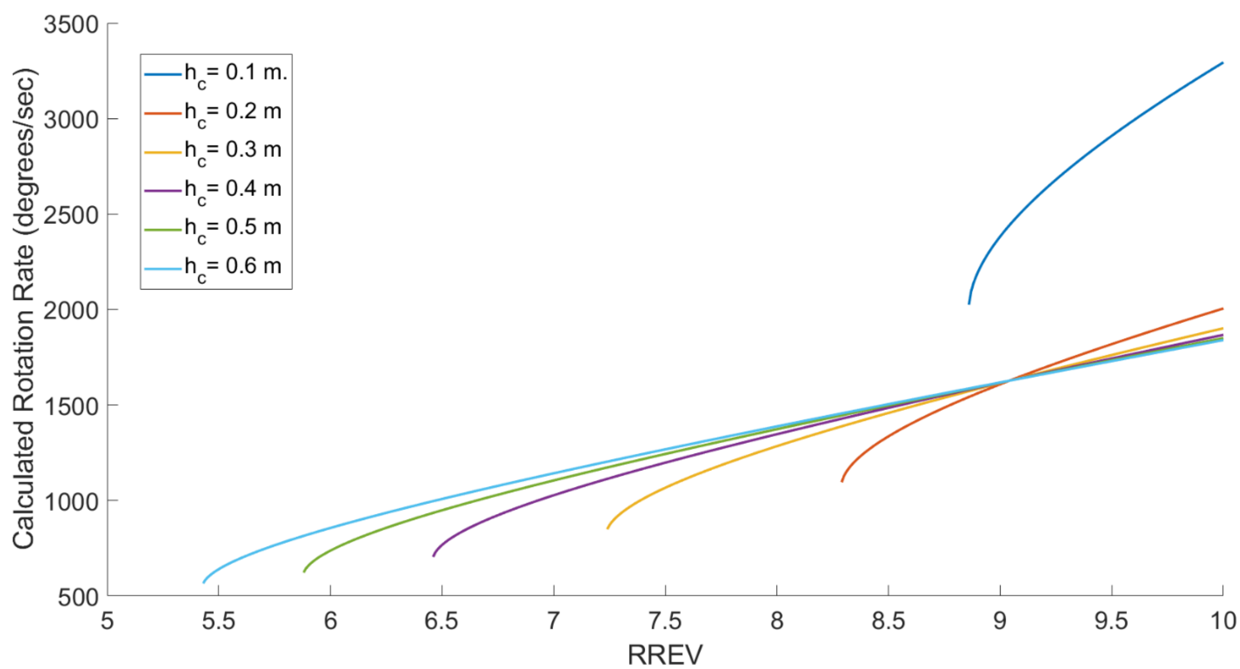

Figure 8: Rotation rate vs RREV and distance from ceiling.

Landing Control Using RREV: Policy 2- Constant Angular Acceleration

Control Policy 2 assumed that the robot accelerates at a constant, predetermined angular acceleration for a calculated duration. This constant angular acceleration could be set to the maximum, where two motors are shut off and the other two motors are at full throttle, creating the largest possible torque. Then, the motors are shut off at the calculated time and the angular velocity remains constant until impact. This method does not directly require measurement of quadrotor orientation, as the motor shutoff occurs automatically after a set duration.

The derivation of Equation 4.5 is shown in Appendix A.

t_{impact} is defined by Equation 4.1.

$$\boldsymbol{t_a = t_{impact} - \frac{\sqrt[2]{\left(\alpha * t_{impact}\right)^2 - 360 * \alpha}}{\alpha}} \tag{4.5}$$

This result shows the duration that the robot should maintain the predetermined angular acceleration and moment varies as a function of RREV, distance from ceiling, and the angular acceleration. The robot should maintain an acceleration of alpha for a duration of t_a before motor shutoff.

Figure 9 and Figure 10 show the calculated acceleration time as a function of RREV using Equation 4.5. Figure 9 shows the policies for different velocities, and Figure 10 shows the control policies for different distances from the ceiling.

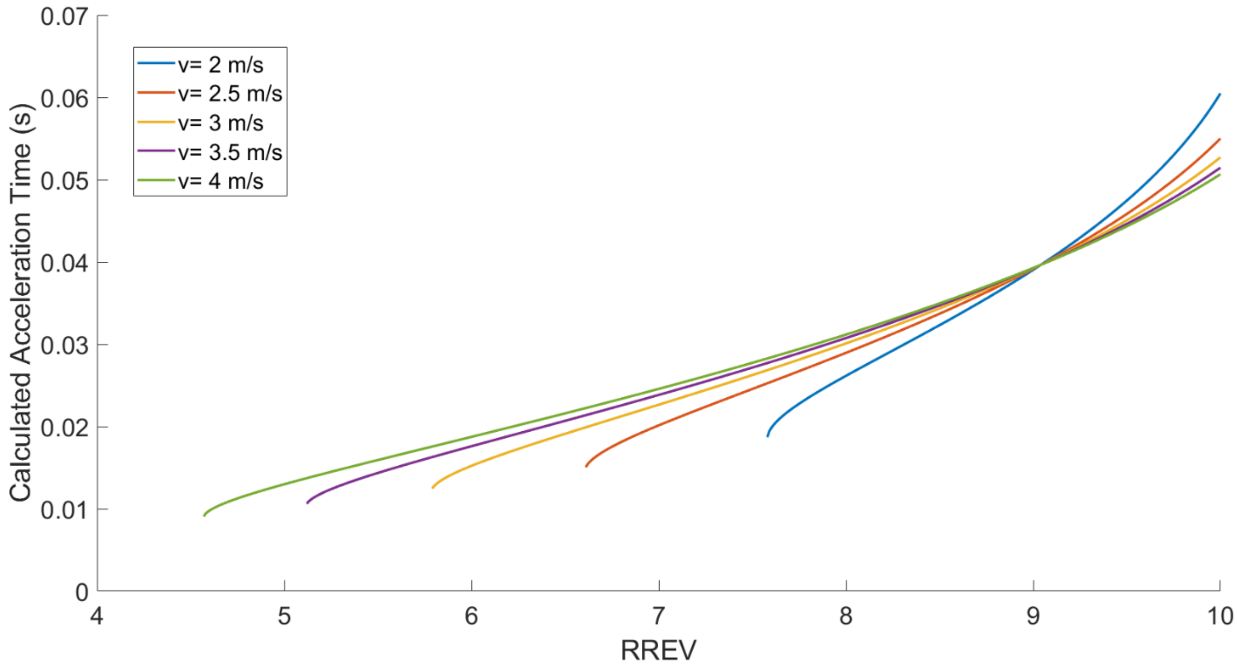

Figure 9: Acceleration time vs RREV and velocity.

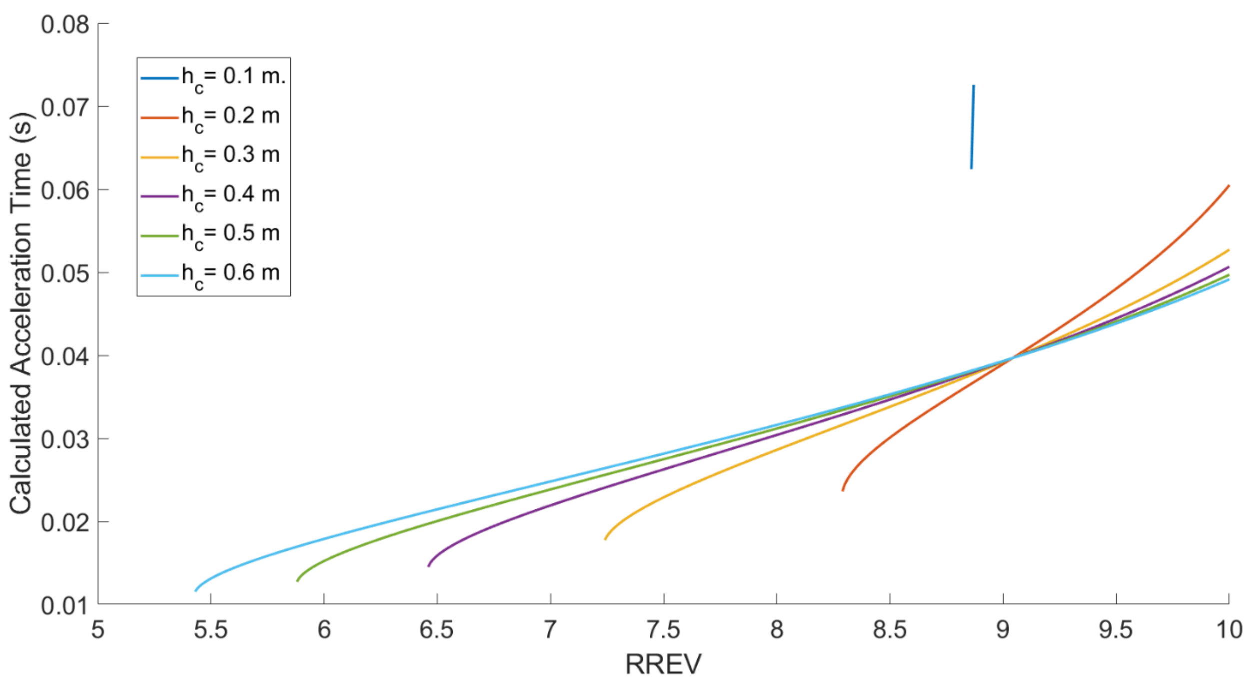

Figure 10: Acceleration time vs RREV and distance from ceiling.

Landing Control Using RREV: Policy 3- Constant Angular Acceleration with Shutoff

Control Policy 3 assumed that the robot accelerates at a calculated constant angular acceleration with motor shutoff at a predetermined angular displacement.

The derivation of Equation 4.6 is shown in Appendix B.

θ_d: *desired angular displacement from start of rotation to landing* (*degrees*)

θ_s: *motor shutoff angle* (*degrees*)

$$\boldsymbol{\alpha = \frac{405}{t_{impact}^2}}, \qquad \textbf{where } \boldsymbol{\theta_d = 180 \; and \;\; \theta_s = 90} \tag{4.6}$$

The policy is to maintain the calculated angular acceleration of alpha until the robot reaches the predetermined angular displacement, at which point the motors should be shut off. The predetermined angular displacement should occur at approximately the calculated time, t_a.

Figure 11 and Figure 12 show the calculated angular acceleration as a function of RREV using Equation 4.6. Figure 11 shows the control policies for different vertical velocities. Figure 12 shows the control policy for different distances from the ceiling.

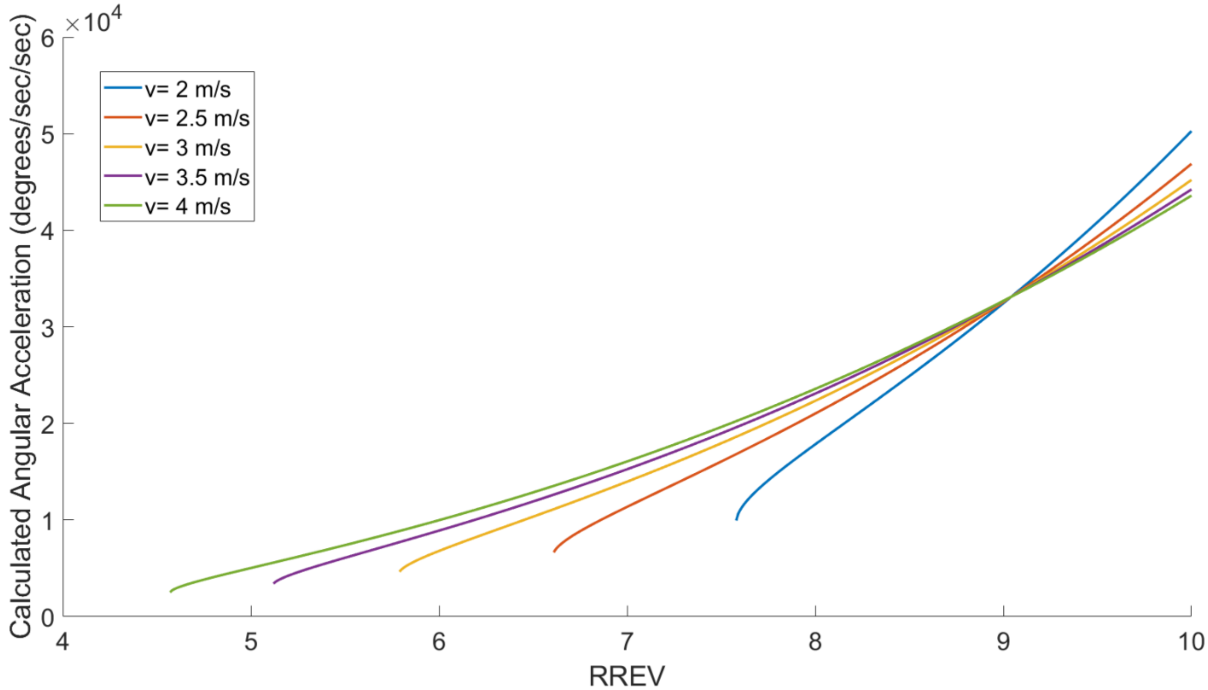

Figure 11: Angular acceleration vs RREV and velocity.

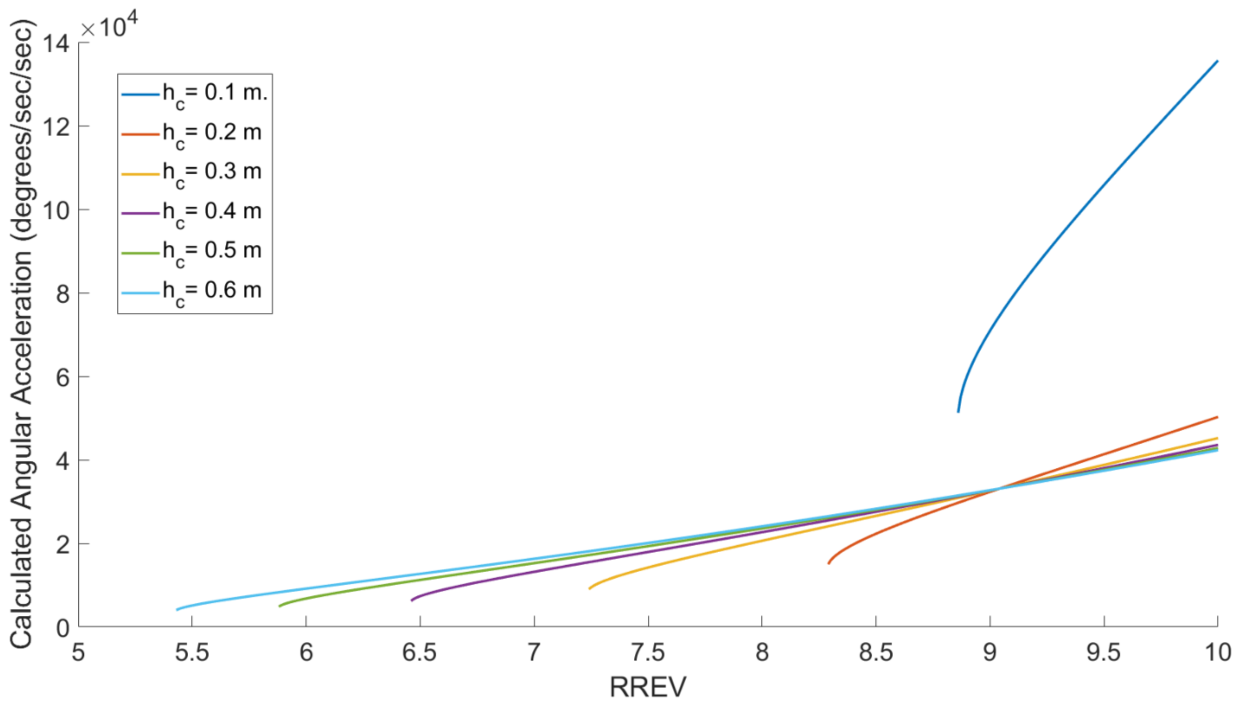

Figure 12: Angular acceleration vs RREV and distance from ceiling.

Landing Control Using RREV: Additional State Input Method

Policies 1, 2, and 3 described in equations 4.4, 4.5, and 4.6 respectively provide simple control with shutoff after a predetermined duration or at a specific predetermined body angle. In Control Policy 2, the robot accelerates for a predetermined acceleration for a calculated duration. For this variant labeled Control Policy 4, rather than accelerating for a predetermined amount of time, the robot accelerates until it reaches a calculated threshold value based on its angular position and angular velocity.

This method assumes that the robot will maintain a constant angular acceleration until motor shutoff and that after motor shutoff the angular rotation rate will remain constant until impact with the ceiling. The robot always has access to its current angular position and angular velocity through internal sensors. The angular displacement between motor shutoff and impact is equal to the angular velocity at the time of motor shutoff multiplied by the remaining time to impact. Therefore, the angular velocity at the time of motor shutoff multiplied by remaining impact time should be equal to the desired angular displacement to land at the commanded angle. The problem can be formulated as follows:

$t_{i,total}: estimated\ time\ to\ impact\ when\ rotation\ begins\ (Equation\ 4.1)(s)$

$t_{current}: time\ since\ rotation\ began\ (s)$

$t_{i,rem}: remaining\ time\ to\ impact\ (s)$

$$t_{i,rem} = t_{i,total} - t_{current}$$

$\theta_c: current\ angular\ position\ (\text{deg.})$

$\theta_f: final\ angular\ position\ at\ impact\ (target\ impact\ condition)(deg.)$

$\theta_d: required\ angular\ displacement\ motor\ shuoff\ to\ impact\ (deg.)$

$$\theta_d = \theta_f - \theta_c$$

$w_c: current\ angular\ rotation\ rate\ (\frac{deg}{s})$

The motors should shut off and stop the angular acceleration of the quadrotor when the remaining desired angular displacement before impact is equal to the current angular rotation rate multiplied by the remaining time to impact.

$$\theta_d = w_c * t_{i_{rem}}$$

Expanding this formulation to include all variables yields Equation 4.7.

$$(\boldsymbol{\theta_f} - \boldsymbol{\theta_c}) = \boldsymbol{w_c} * (\boldsymbol{t_{i_{total}}} - \boldsymbol{t_{current}}) \tag{4.7}$$

Equation 4.7 shows the condition under which the quadrotor should stop accelerating. In the simplest implementation of this algorithm, a constant moment would be applied until the left and right side of Equation 4.7 are within some defined margin of one-another. In a more complex formulation, the moment and acceleration could be decreased as the shutoff condition is approached to prevent overshoot.

Figure 13 and Figure 14 show Control Policy 4 with an angular acceleration of 18,000 degrees/s^2.

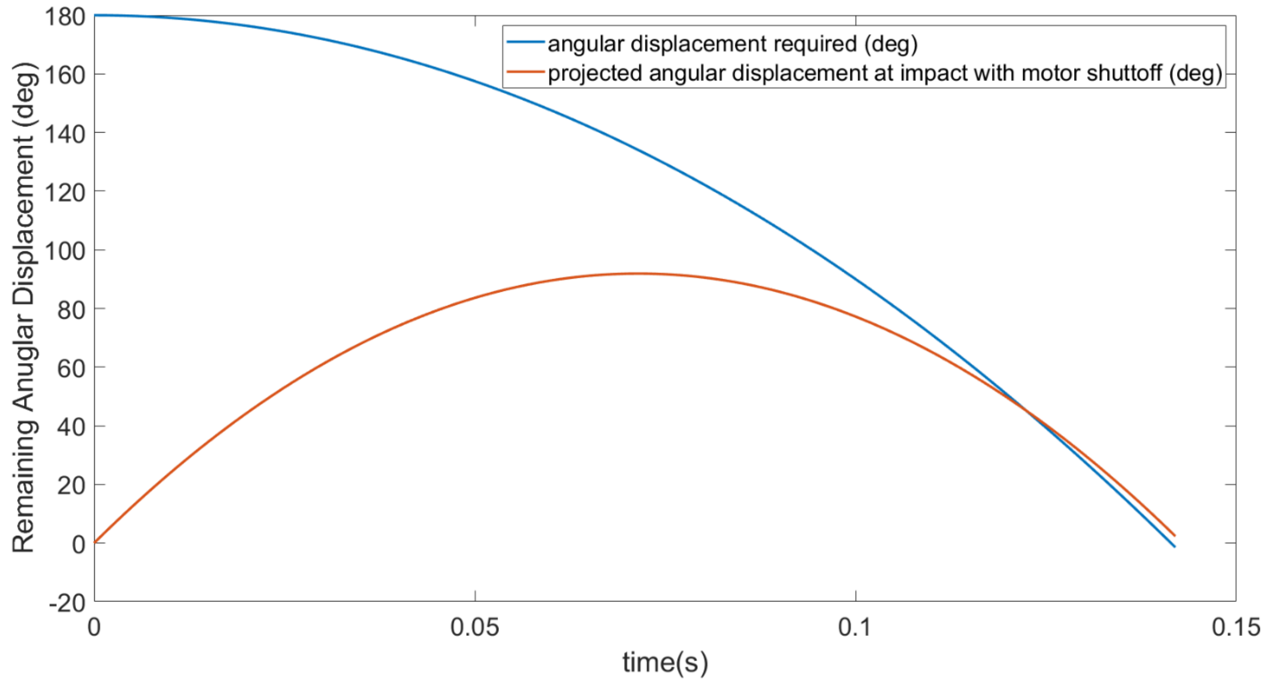

Figure 13: Control Policy 4: Angular displacement model with low acceleration.

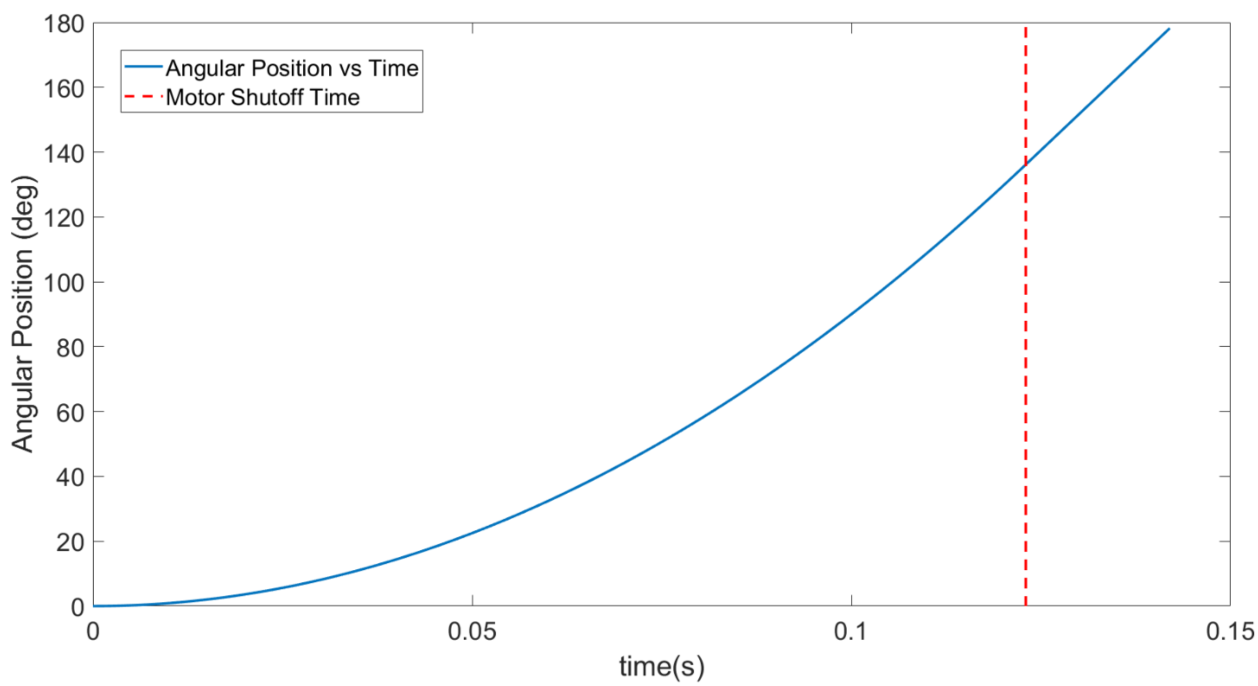

Figure 14: Control Policy 4: Angular position model with low acceleration.

Figure 15 and Figure 16 show control policy 4 with an angular acceleration of 24,000 degrees/s^2.

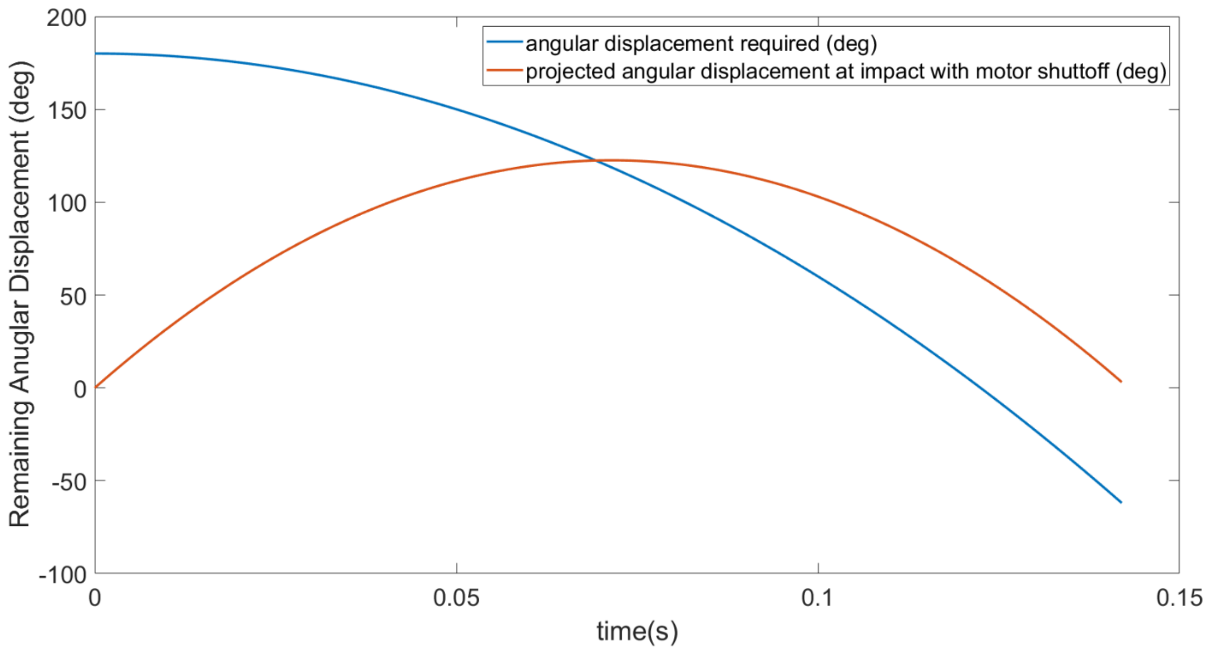

Figure 15: Control Policy 4: Angular displacement model with moderate acceleration.

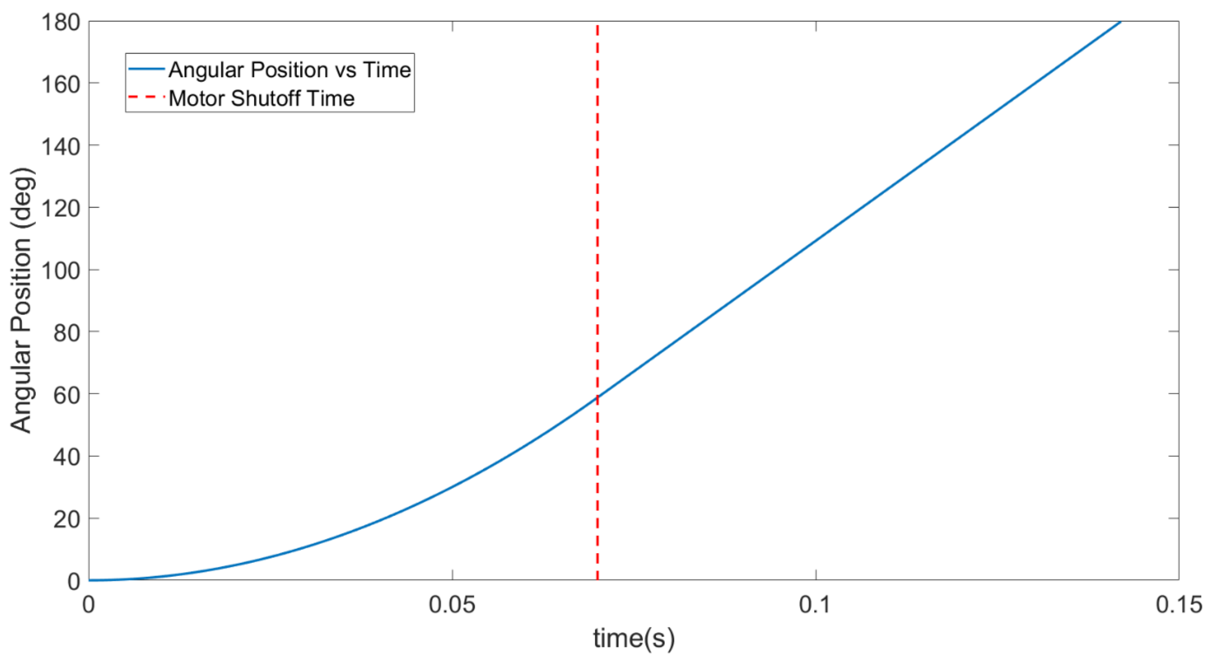

Figure 16: Control Policy 4: Angular position model with moderate acceleration.

Figure 17 and Figure 18 show Control Policy 4 with an angular acceleration of 30,000 degrees/s^2.

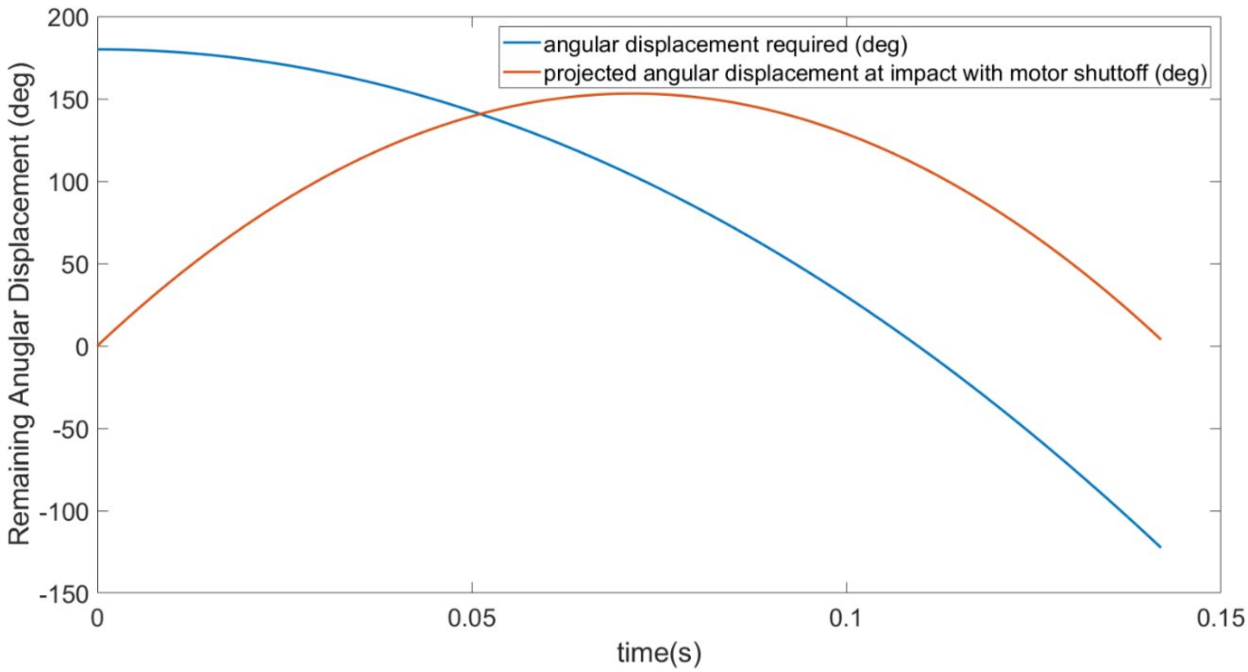

Figure 17: Control Policy 4: Angular displacement model with high acceleration.

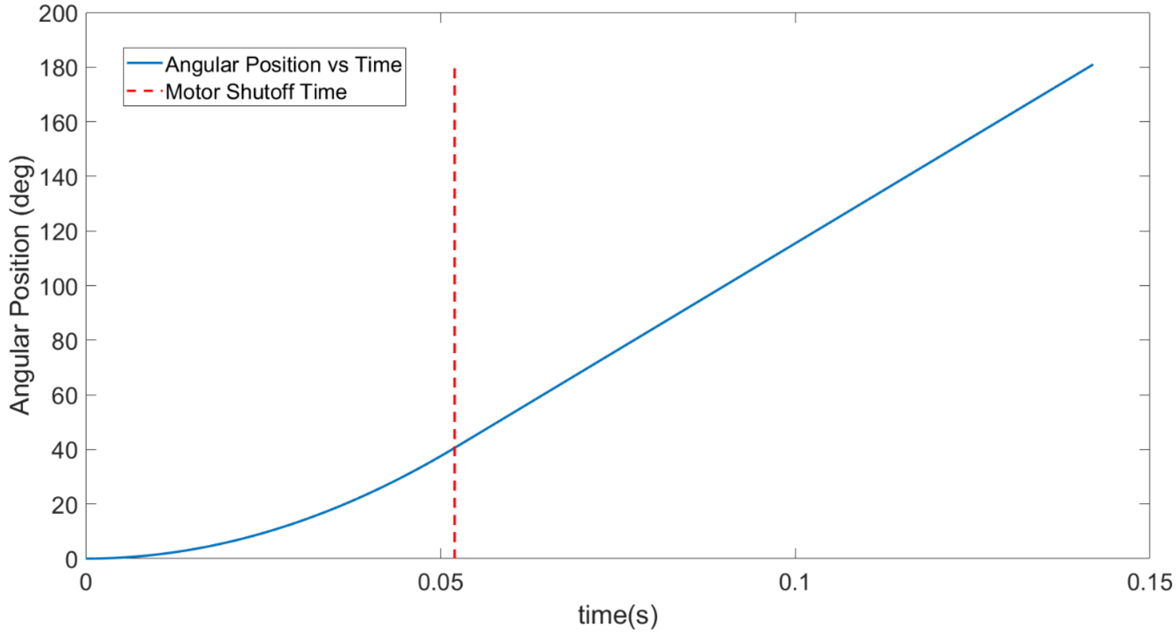

Figure 18: Control Policy 4: Angular position model with high acceleration.

Deploying Landing Strategy to Investigate Assumption Sensitivity

Each ideal control strategy was applied in the MATLAB simulation environment to evaluate the sensitivity of the landing control and impact conditions to the RREV input and the uncertainty of state in the vertical dimension with using only a RREV reading. Each trial was designed to seek a more fundamental understanding of the relationship between the sensor data and control along with the impact of sensor error and uncertainty.

The simulation trials assumed there was no sensor error in determining the correct RREV, and the rotation occurs at a constant angular velocity until impact once motors are shut off. It was also assumed that the Crazyflie desires to land “flat” on the ceiling, with all legs touching almost simultaneously. RREV is equal to velocity/distance, and in the absence of gravity, 1/RREV would be equal to the impact time. However, since the robot was landing on the ceiling in the presence of gravity, the control problem with RREV as the sensor input becomes more complicated.

Assuming the RREV reading from the sensor was accurate, in the absence of any other sensor readings, the robot would be aware of the proportion between its instantaneous velocity and distance from a surface, in this case the ceiling. The controller cannot determine its complete state in the vertical dimension. The complete state is defined here as the robot knowing both its position relative to the ceiling and its velocity. While there are methods of using combinations of RREV data and other sensor data like acceleration to better estimate the complete state, that is beyond the scope of this work. This simulation explored the problem the robot faces with attempting to achieve a "flat" landing despite lack of complete information.

Each ideal control policy was not proven to be the best algorithm for achieving a flat landing with RREV as an input for a real robot. However, these algorithms represent a few possible control policies. This experiment showed the impact of uncertainty on the landing behavior using the idealized control algorithm, and other algorithms may later be developed that demonstrate more robust landing behavior.

Figure 19 and Figure 20 show the results of implementing Control Policy 1 shown as Equation 4.4 in the simulated environment. These figures were produced by simulating a quadrotor in the 2-D MATLAB simulation. The quadrotor was simulated with the initial conditions of being 0.3 m. from the ceiling and traveling upwards at a velocity of 3.5 m/s. It was assumed that the quadrotor accurately measured RREV but did not know the specific velocity or distance from ceiling. The controller assumed a distance from ceiling (h_c) and calculated the angular rotation rate based on Equation 4.4 with the RREV and assumed distance from ceiling as inputs. This calculated angular rotation rate was imparted on the quadrotor body. Figure 19 shows the commanded rotation rate as a function of the assumed distance from ceiling while Figure 20 shows the result of these commanded policies, displaying the impact angle found in the simulation when the control policy was used for each assumed distance from the ceiling.

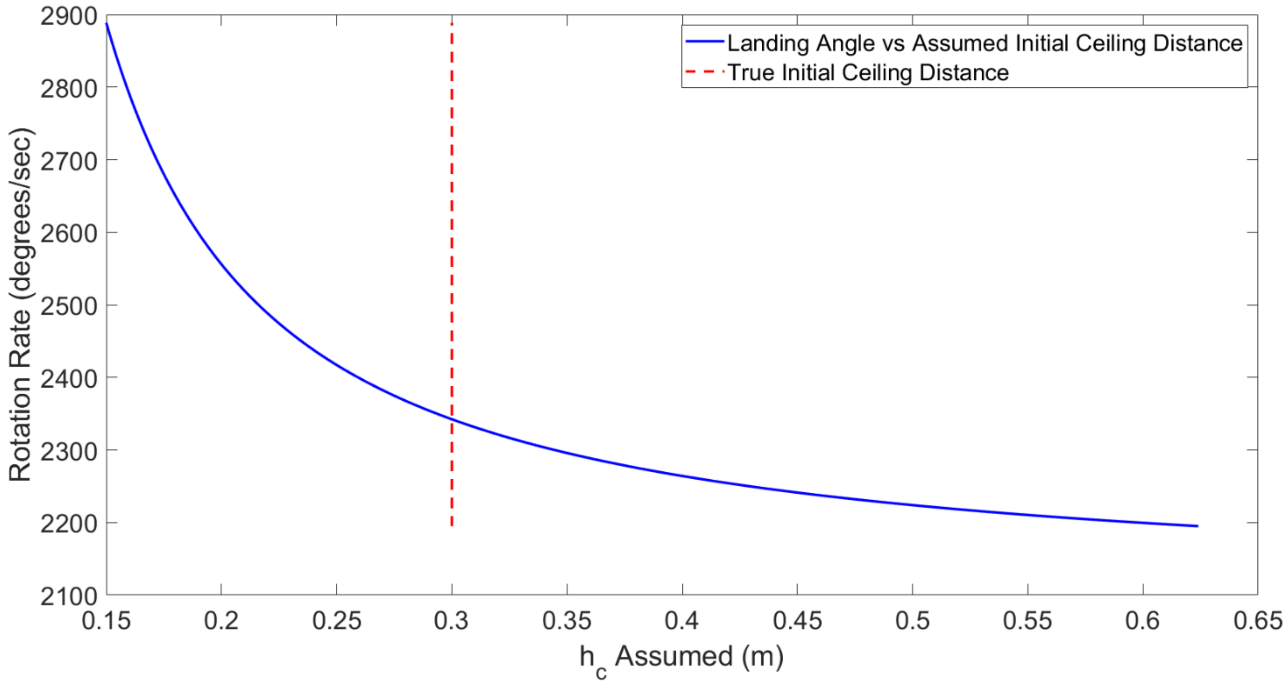

Figure 19: Sensitivity of rotation rate to assumed distance from ceiling.

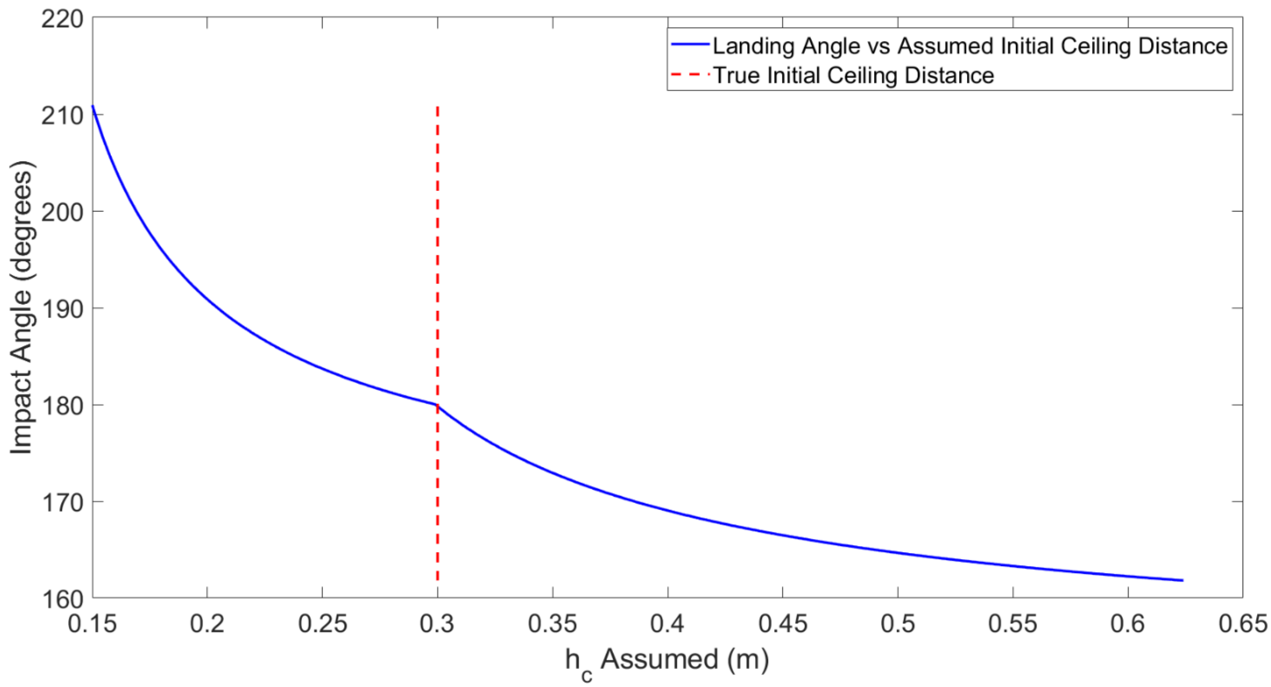

Figure 20: Sensitivity of impact angle to assumed distance from ceiling.

As can be seen in Figure 19 and Figure 20, the assumed distance from the ceiling influences both the commanded rotation rate and the impact angle with the ceiling. If the quadrotor knew its exact distance from the ceiling and RREV, the quadrotor could calculate the time to impact. The graphs show the sensitivity of the control policy and impact angle to the uncertainty of distance from ceiling. This uncertainty is a result of utilizing RREV as an input when the system experiences acceleration.

Figure 21 and Figure 22 implement the algorithm described in Equation 4.5. Measured RREV and the assumed distance from ceiling were passed into Equation 4.5. The preset angular acceleration for these trials was set at 50,000 degrees/s^2 and could be adjusted based on the limits of the quadrotor being used. The true distance from the ceiling was 0.4 m. and the true vertical velocity was 3.5 m/s. The calculated angular acceleration time is shown in Figure 21 and the result of implementing this angular acceleration time is shown in Figure 22 by graphing impact angle.

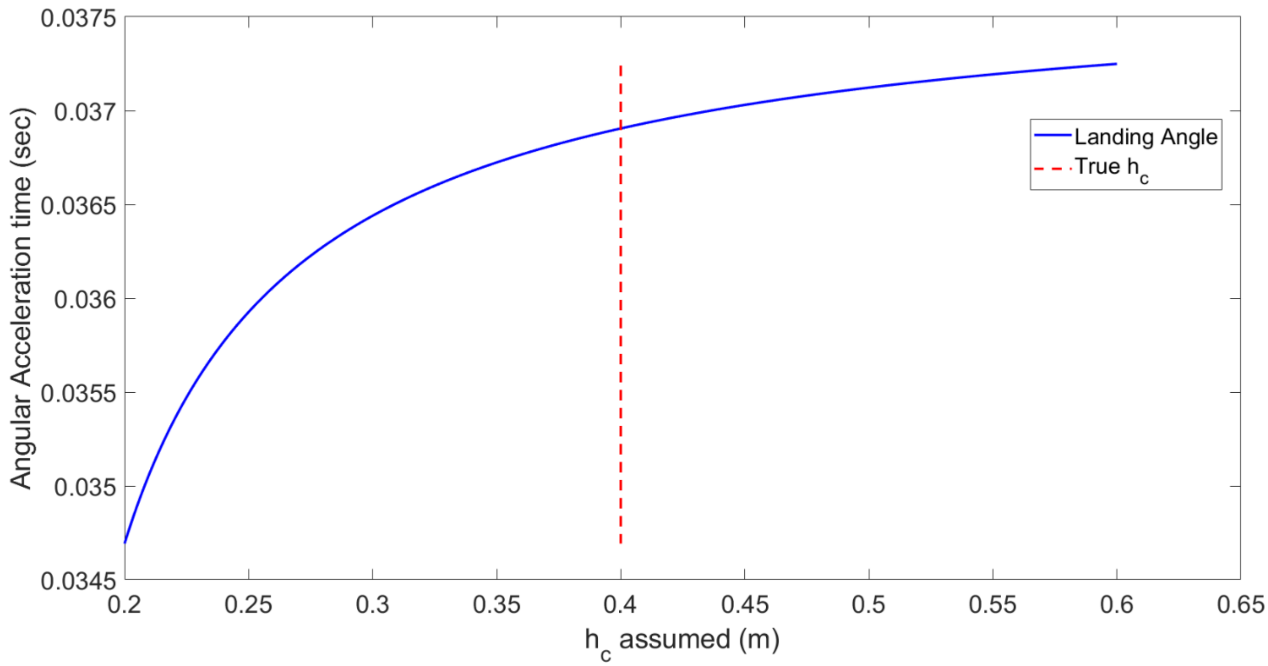

Figure 21: Angular acceleration vs assumed distance from the ceiling.

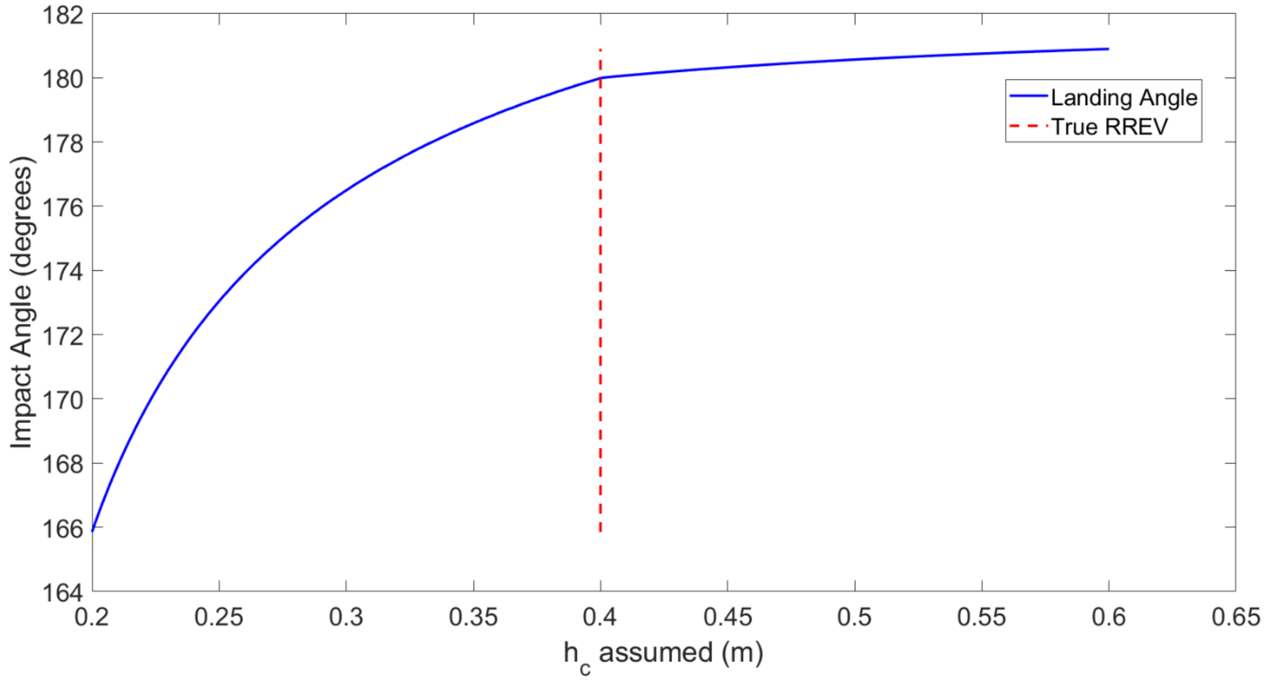

Figure 22: Impact angle vs assumed distance from the ceiling.

Figure 23 and Figure 24 show the calculated angular acceleration as a function of assumed distance from ceiling when Equation 4.6 is implemented as the control policy. The true distance from the ceiling is 0.4 m and the true vertical velocity is 3.5 m/s.

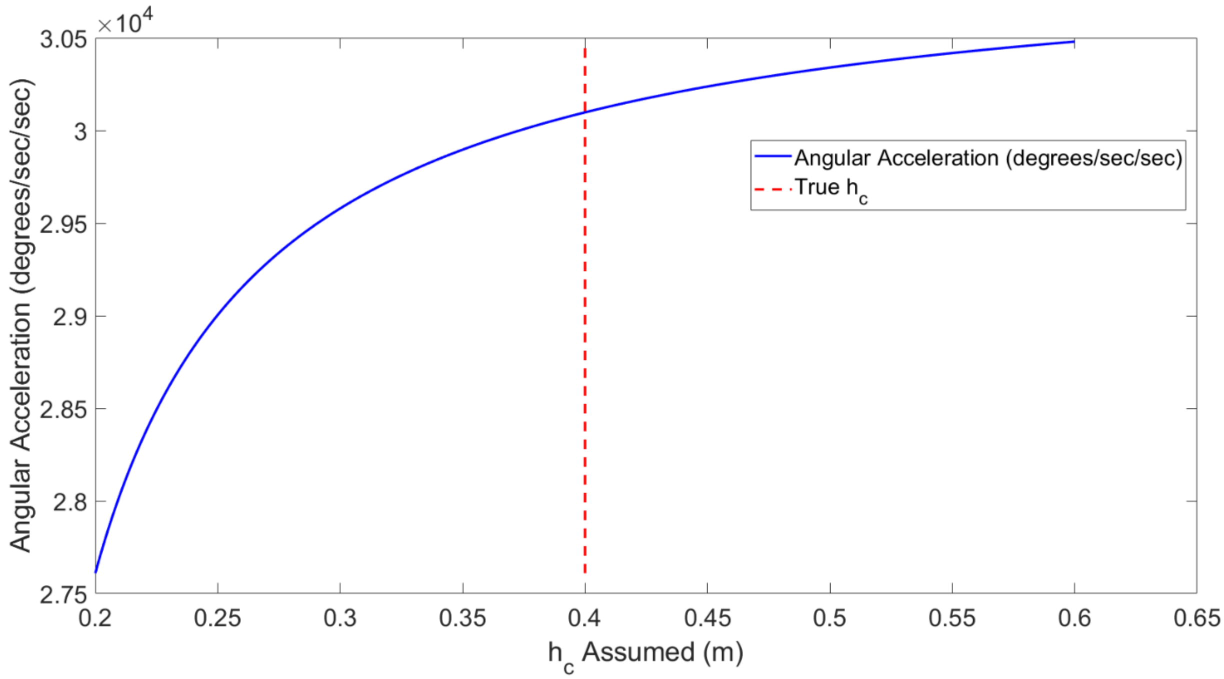

Figure 23: Angular acceleration vs assumed distance from ceiling.

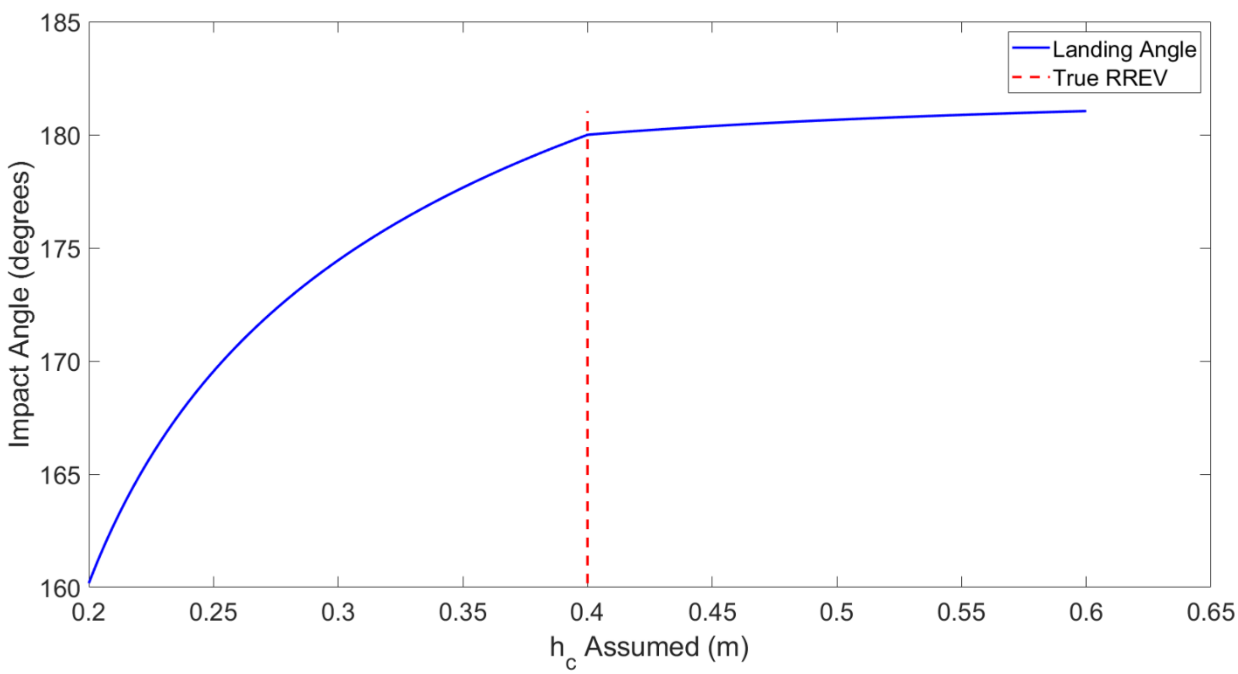

Figure 24: Impact angle vs assumed distance from ceiling.

Figure 25 and Figure 26 show the sensitivity of Control Policy 1 to the sensed RREV. The true RREV is shown as the vertical line and the intersection between the vertical red line and the blue line would be the result if there was no error in the RREV reading and the distance from ceiling was known. Figure 25 shows the sensitivity of angular rotation rate to error in the RREV reading. Figure 26 shows the sensitivity of impact angle to the RREV sensor error.

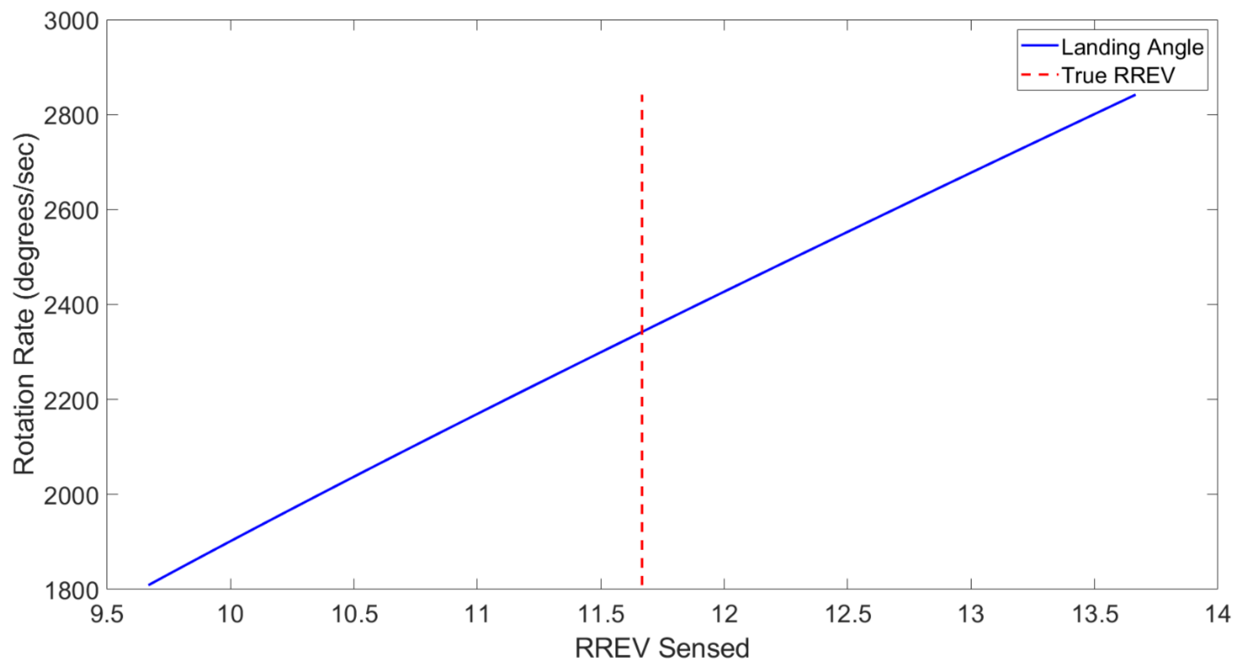

Figure 25: Control Policy 1: Angular velocity vs RREV sensed.

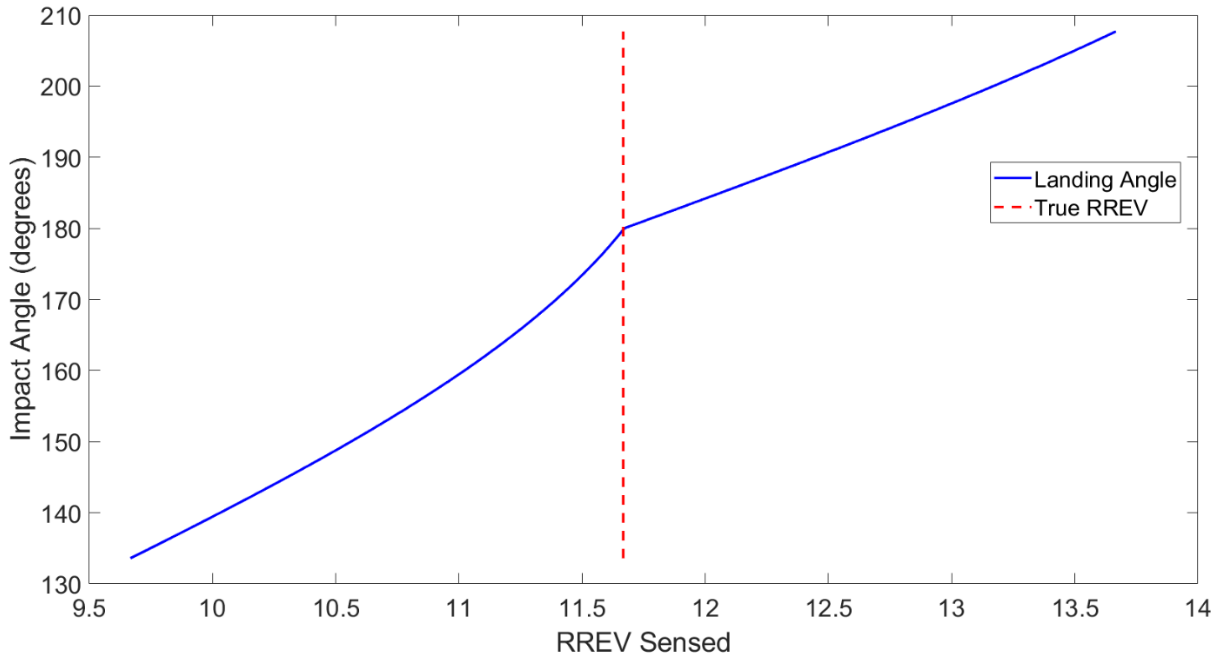

Figure 26: Control Policy 1: Impact angle vs RREV sensed.

Figure 27 and Figure 28 show the sensitivity of Control Policy 2 to the sensed RREV. The true RREV is shown as the vertical line and the intersection between the vertical red line and the blue line would be the result if there was no error in the RREV reading and the distance from ceiling was known. Figure 27 shows the sensitivity of angular acceleration time to error in the RREV reading. Figure 28 shows the sensitivity of impact angle to the RREV sensor error.

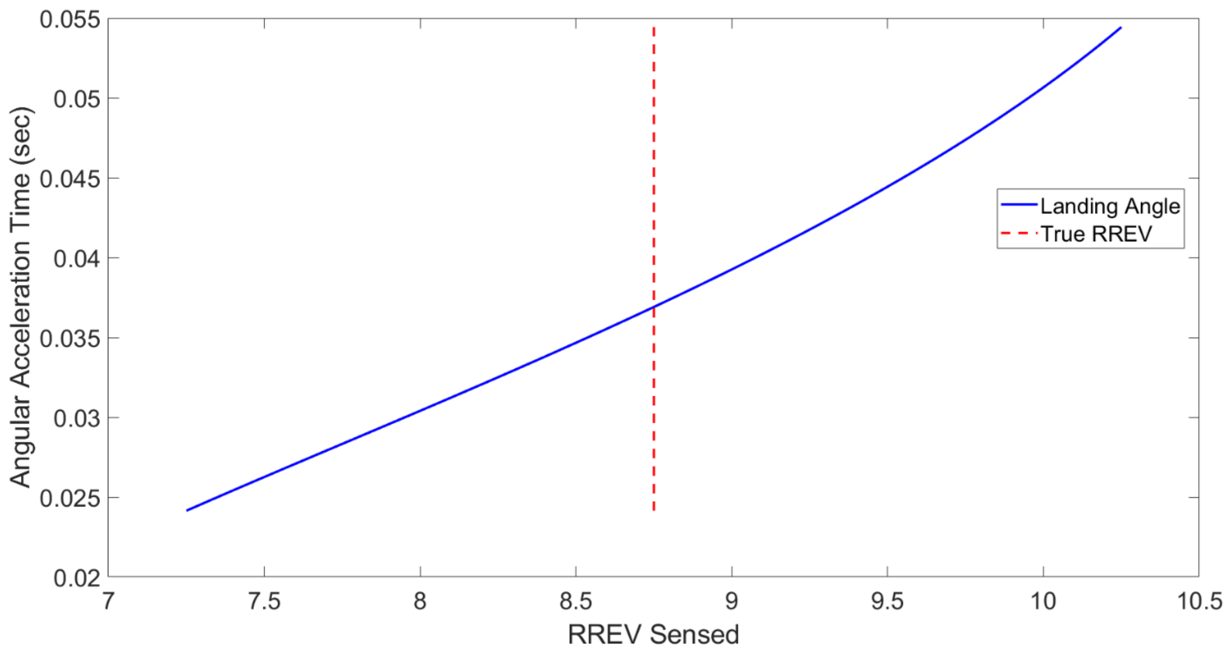

Figure 27: Control Policy 2: Angular acceleration time vs RREV sensed.

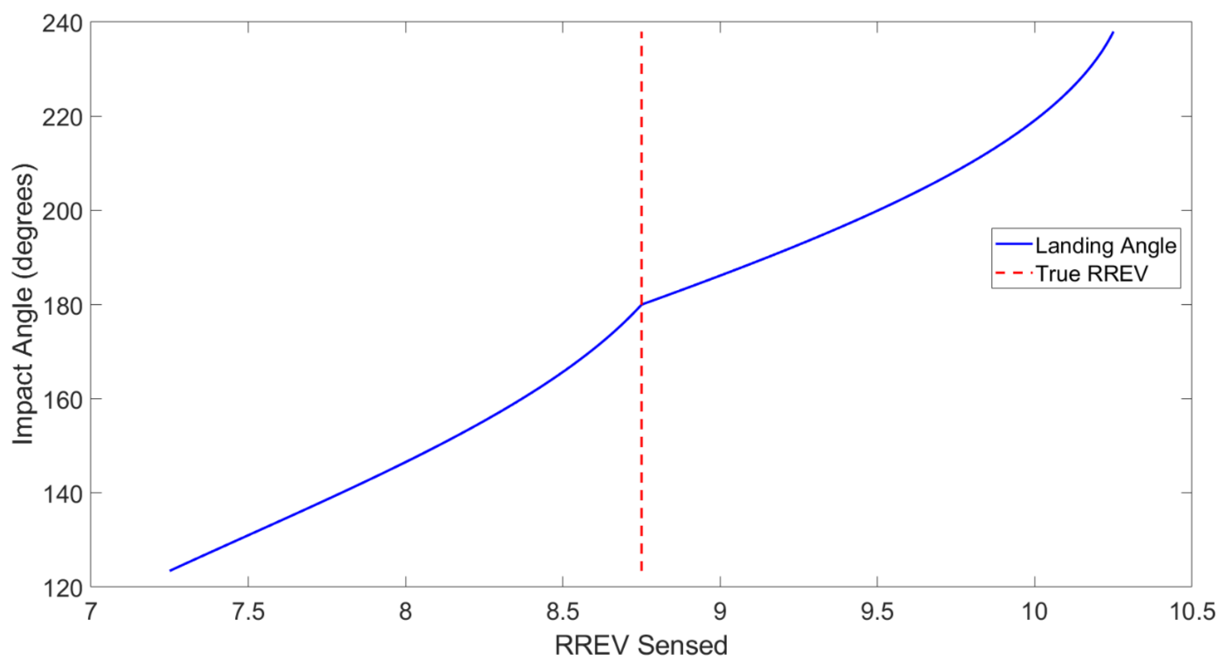

Figure 28: Control Policy 2: Impact angle vs RREV sensed.

Figure 29 and Figure 30 show the sensitivity of Control Policy 3 to the sensed RREV. The true RREV is shown as the vertical line and the intersection between the vertical red line and the blue line would be the result if there was no error in the RREV reading and the distance from ceiling was known. Figure 29 shows the sensitivity of angular acceleration to error in the RREV reading. Figure 30 shows the sensitivity of impact angle to the RREV sensor error.

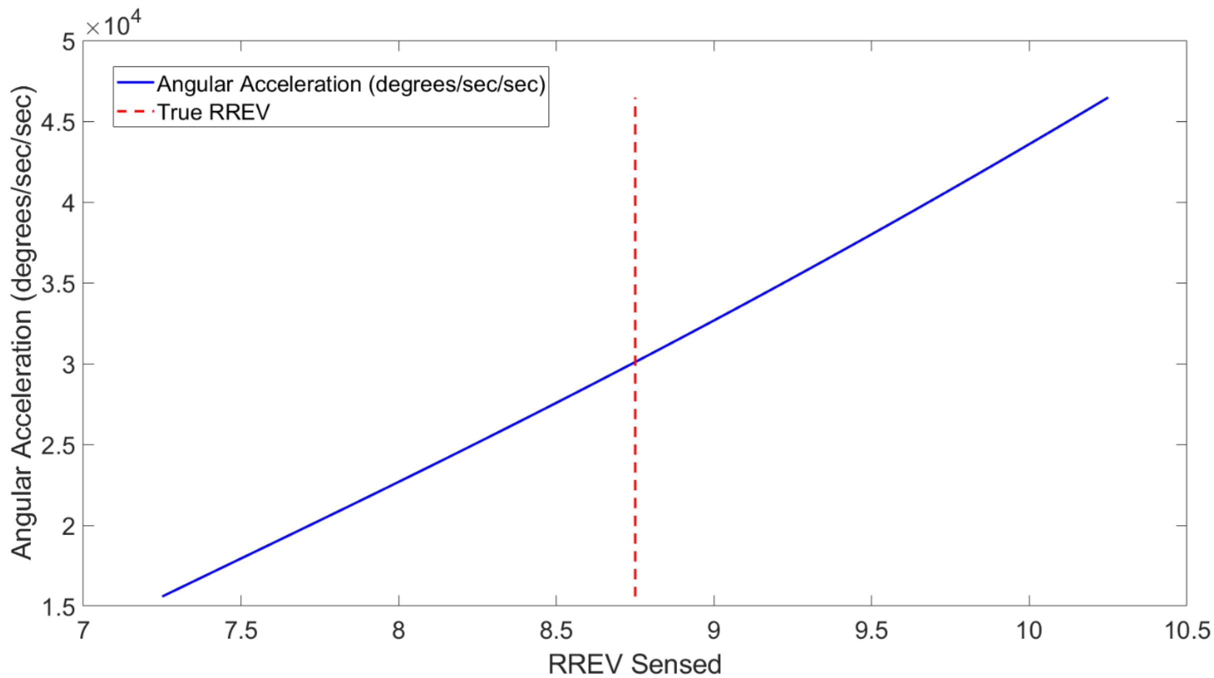

Figure 29: Control Policy 3: Angular acceleration vs RREV sensed.

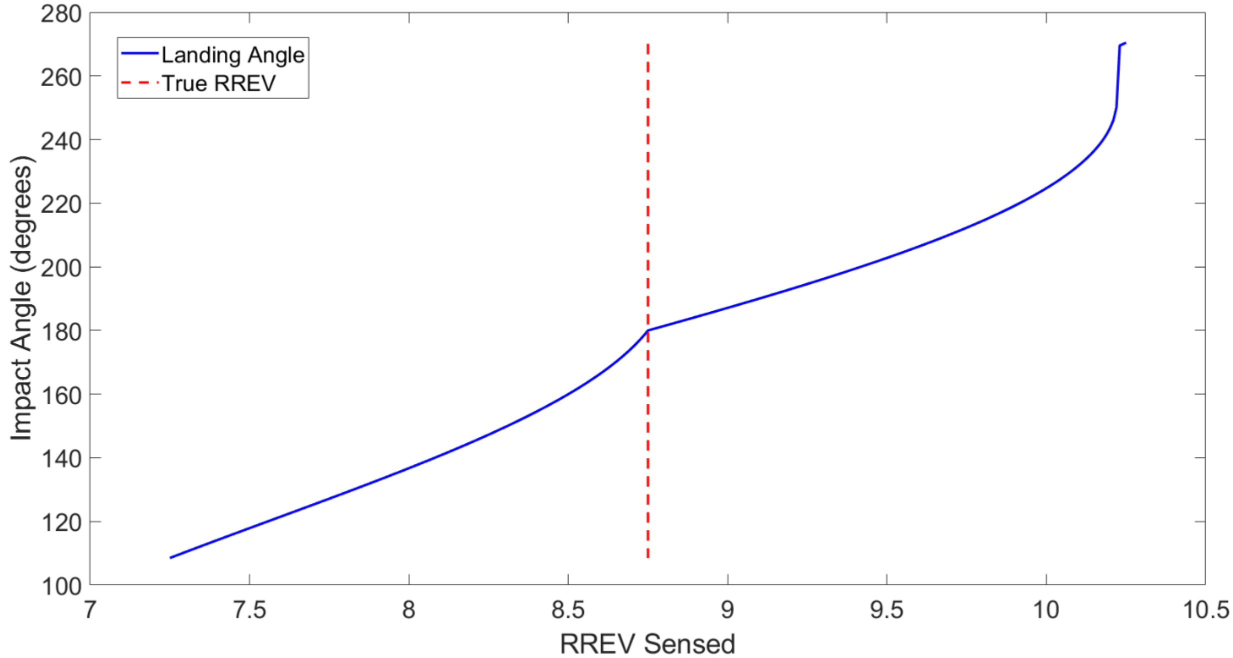

Figure 30: Control Policy 3- Impact angle vs RREV sensed.

RREV as a Control Input

An equation was derived to estimate impact time with the ceiling as a function of the initial velocity of the robot and the RREV reading. This model was created to help explain trends in controller behavior as a function of two of the key quadrotor state variables. For this model, it is assumed that the leg height is negligible compared to the distance from the ceiling and the quadrotor can be modeled by using ideal projectile motion.

The derivation of Equation 4.8 is shown in Appendix C.

$$\boldsymbol{t} = \left(\frac{\mathbf{1}}{\boldsymbol{g}}\right) * \left((\boldsymbol{V}_{y_i} - \sqrt[2]{\boldsymbol{V}_{y_i}^2 - \mathbf{2} * \boldsymbol{g} * \frac{\boldsymbol{V}_{y_i}}{\boldsymbol{RREV}}} \right) \tag{4.8}$$

Figure 31 was generated by calculating the impact time for every combination of several initial velocities and RREV readings over ranges of interest while assuming the height of the legs is negligible. If the impact time were calculated to have an imaginary component, the robot would never contact the ceiling and the trial was depicted as a red x with an impact time of zero. For the rest of the data, there is clearly a trend where for a given initial velocity, the impact time increases exponentially as RREV decreases. For a given RREV, impact time increases as initial velocity decreases.

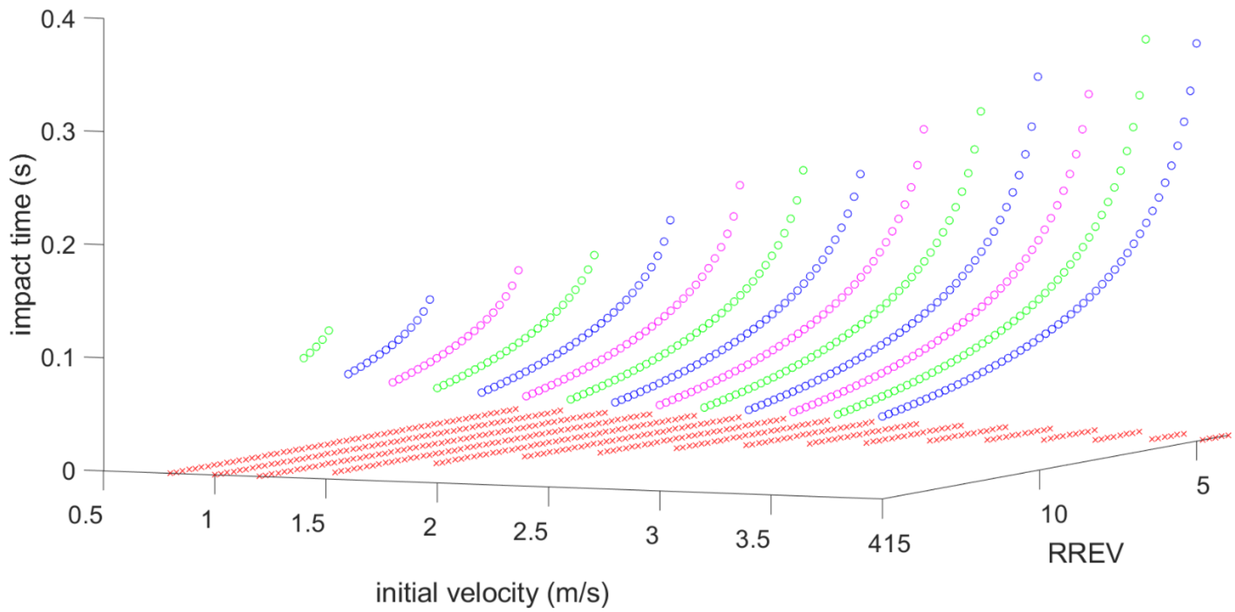

Figure 31: Impact time as a function of velocity and RREV.

To generalize these observations, it was assumed that the commanded angular acceleration or angular velocity would need to be higher to achieve the same impact angle if the impact time is lower. Therefore, if the controller has both RREV and initial velocity as inputs, higher rotation rates or angular accelerations for higher RREV values or higher initial velocities are expected. Since the controller does not have initial velocity as an input, the control algorithm will need to base its commanded angular acceleration or velocity off the RREV reading alone.

Figure 32 was generated by using the initial velocity and the impact time as calculated in Equation 4.5 to determine the impact velocity with the ceiling assuming the only vertical acceleration is due to gravity. As shown in Figure 32, the vertical impact velocity increases for a given initial velocity as RREV increases. Impact velocity increases with increasing initial velocity for a given RREV value.

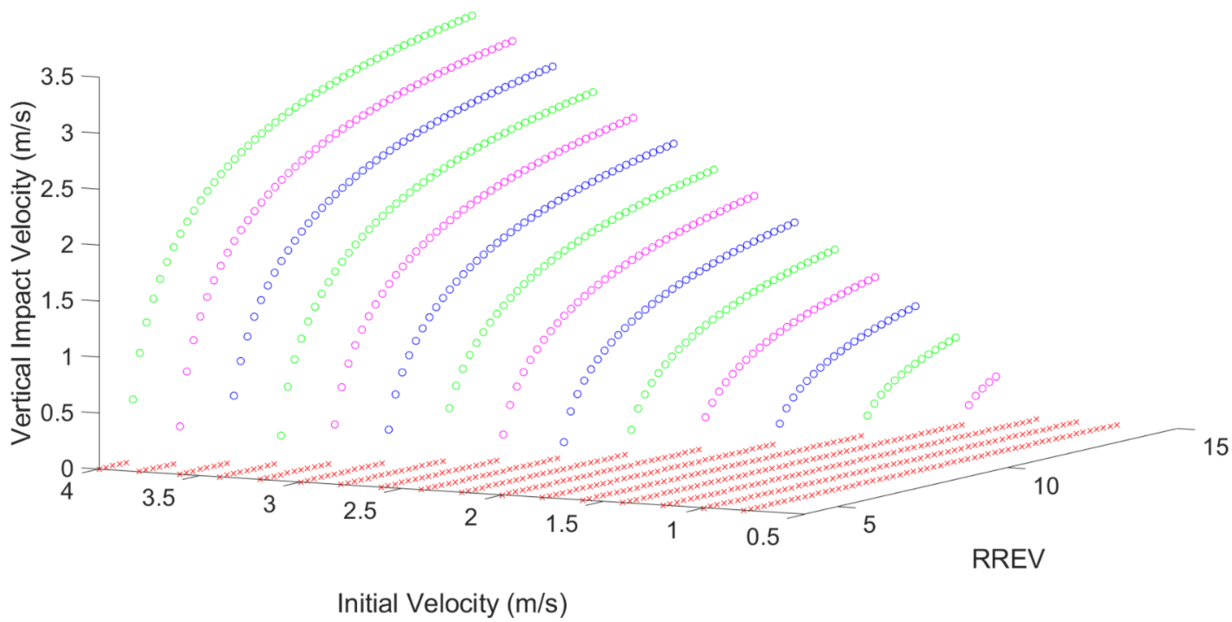

Figure 32: Impact velocity as a function of initial velocity and RREV trigger.

Choosing RREV Threshold for Landing Process

Just as flies and other flying insects must use their perception of their surroundings to trigger a landing process at some point during flight before colliding with a surface, small robotic fliers must also decide when to initiate the landing process. One method is to choose a RREV threshold such that the rotation and landing process begins when the sensed RREV value exceeds the RREV threshold. While there are a variety of RREV thresholds that can result in successful landings for a variety of velocities, the time to impact depends on both RREV and flight velocity or distance from ceiling.

If, however, the RREV threshold is chosen carefully based on the geometry of the landing mechanism, the time to impact will be approximately the same regardless of the specific velocity and distance from ceiling. This would allow for the same exact control policy, whether angular velocity or angular acceleration based, to be used in all cases. This can be seen in Figure 7 through Figure 12 where the control policies were a function of RREV. For all velocities and assumed ceiling distances, there is some RREV value where all the control output lines intersect, indicating that the impact time is the same, and the necessary control policy would also be the same.

If a 180-degree rotation before landing is assumed, there is a clear relationship between the leg height and the RREV threshold where the control policy is approximately the same for all velocities. This relationship was determined numerically by finding $RREV_{threshold}$ for leg heights varying from 0.02 m. to 0.10 m. in increments of 0.01 m. through the MATLAB simulation. An equation was fit to the data as shown in Figure 33.

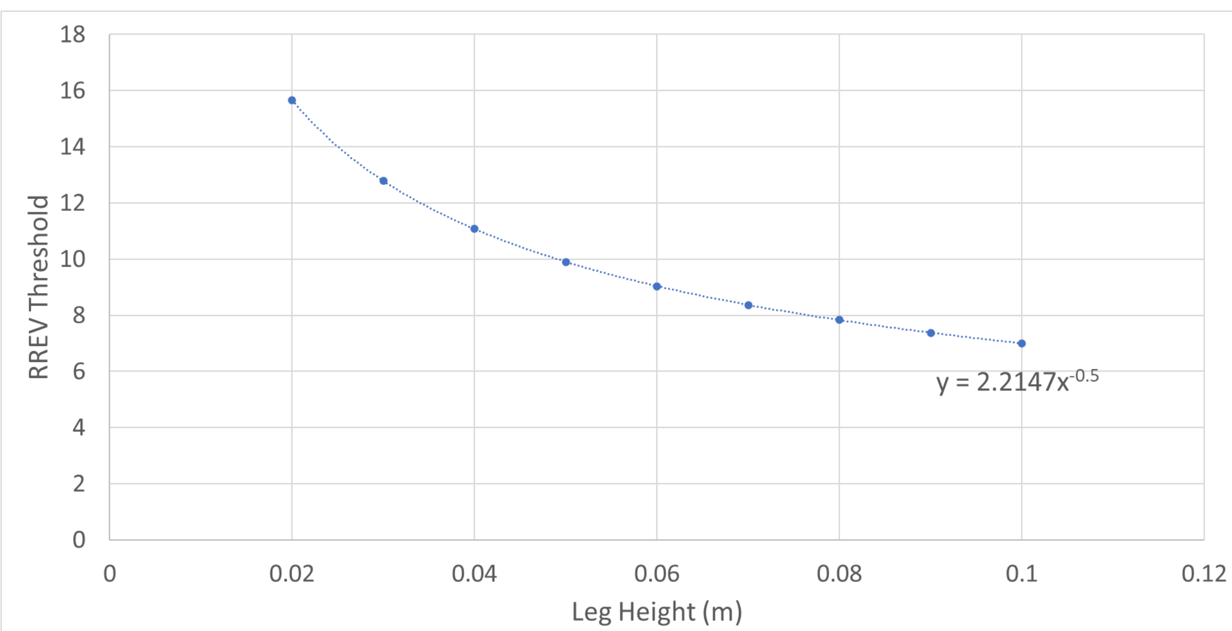

Figure 33: RREV trigger threshold vs leg height.

$$RREV_{threshold} = \frac{2.2147}{\sqrt{h_L}} \tag{4.9}$$

Figure 34 and Figure 35 show the impact time as a function of initial velocity and RREV for different leg geometries. The impact times are all shown in blue except for the impact times when the RREV is equal to the RREV threshold as calculated in Equation 4.9 which are shown in green. As shown in Figure 34 and Figure 35, the impact time is close to independent of initial velocity when RREV is equal to the value predicted by Equation 4.9. For all other RREV values, impact time varies more significantly as a function of initial velocity.

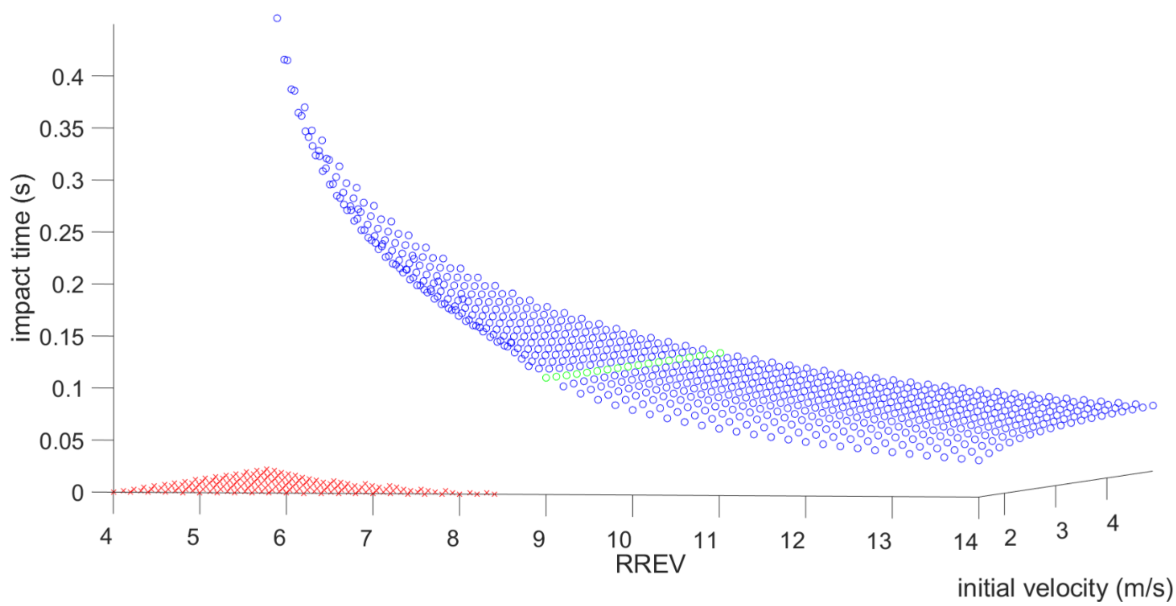

Figure 34: Impact time vs RREV and initial velocity: leg height= 0.06 m.

$$RREV_{threshold} = \frac{2.2147}{\sqrt{0.06}} = 9.041$$

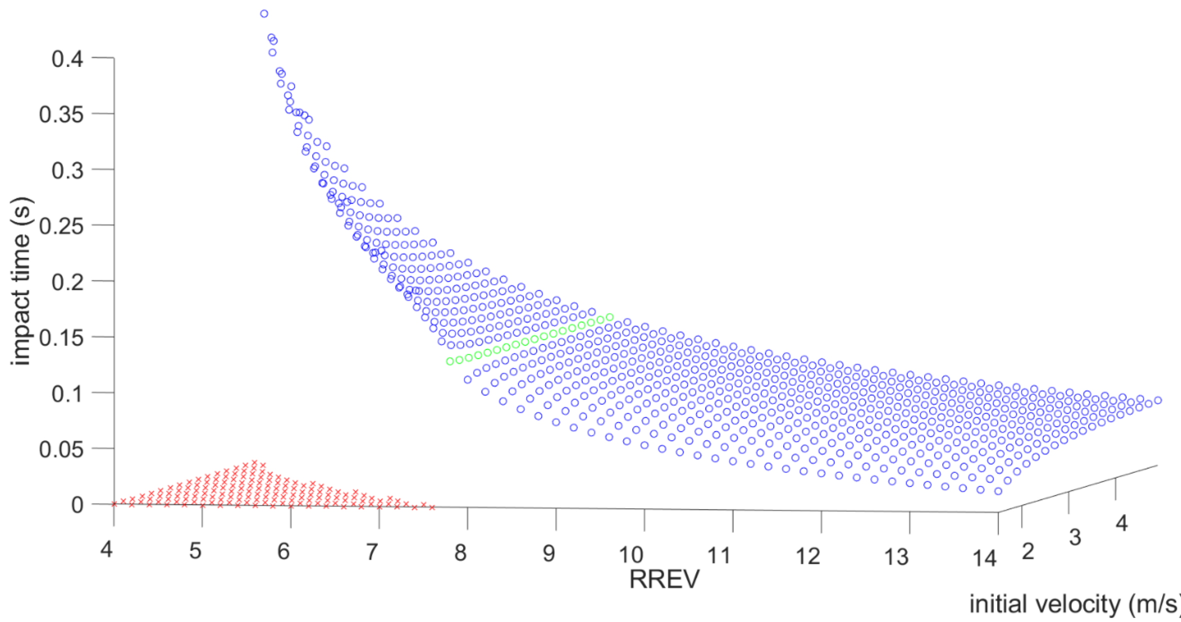

Figure 35: Impact time vs RREV and initial velocity: leg height= 0.08 m.

$$RREV_{threshold} = \frac{2.2147}{\sqrt{0.08}} = 7.830$$

Chapter 5

Simulation-based Landing Mechanism Design and Optimization

Impact of Landing Mechanism Design and Impact Conditions on Simulated Landings

Several leg design factors have been identified as contributing to the behavior of the robot after impact with the ceiling. Among these factors are the stiffness and damping of the legs, the quantity and orientation of the legs, the placement and attachment of the Velcro or adhesive, the degrees of freedom of the leg attachment to the robot body, and the geometry of the legs. Of these factors, the simulation environment was used to explore the impact of leg geometry in more detail.

For this study, the 2-D MATLAB simulation was used, and all assumptions stated for this simulation apply to the results of the impact trials. The simulation was first configured to run a series of trials with fixed leg geometry while varying the impact angle, vertical impact velocity, and horizontal impact velocity. For each set of impact conditions, the simulation recorded whether the robot successfully landed (both sets of legs touch ceiling), partially landed (only one set of legs touches ceiling), or failed landing where either no part of the robot touched the ceiling or a part of the robot other than the leg tip impacted the ceiling first. This data was visualized with the three independent variables on the axes and color denoting the success or failure of landing to provide an intuitive understanding of landing patterns for different landing mechanism geometries. Green indicated successful landing, yellow indicated partial landing, and red indicated failed landing.

Different leg geometries were tested in the simulation trials. These leg geometries were labeled based on the length and width of the leg. Figure 36 shows how the leg dimensions are defined. Table 1 shows the dimensions corresponding to each leg geometry label.

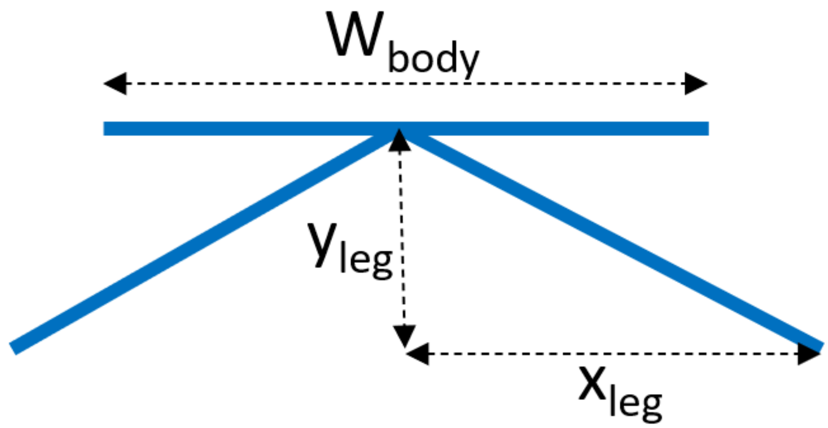

Figure 36: Leg dimensions.

Table 1: Leg geometry labels and dimensions.

Leg Design Label	x_{leg} **(m)**	y_{leg} **(m)**
Extra Narrow Short	0.017	0.040
Extra Narrow Long	0.017	0.060
Narrow Short	0.037	0.040
Narrow Long	0.037	0.060
Wide Short	0.057	0.040
Wide Long	0.057	0.060

Based on the simulation, the landing patterns could be grouped into several categories with similar characteristics. These groupings are impact angle assisted landing, horizontal velocity assisted landing, and vertical velocity assisted landing. Impact angle assisted landing is characterized by a small impact angle between the robot and the ceiling on impact. This results in a small rotation required for a successful landing. Horizontal velocity assisted landing is characterized by a high horizontal velocity and a large impact angle, where the horizontal component of momentum is the primary contributor to the body swing after impact. Vertical velocity assisted landing is characterized by a high vertical velocity, where the vertical component of momentum is the primary contributor to the body swing after impact.

Several categories of failure were also evident from the simulation data. High impact angle failure is characterized by a high impact angle between the quadrotor body and the ceiling. This often resulted in failure due to the insufficient momentum and energy transfer to complete the swing necessary for successful landing. Additionally, this can occur when the body of the quadrotor, including the tips of the rotors, contact the ceiling before either of the legs.

Figure 38 and Figure 39 show the physical maximum impact angle with the ceiling such that the leg tip and the body would contact the ceiling simultaneously. Impact angle is defined as the angle between the body and the ceiling during impact as shown in Figure 37. This maximum physical impact angle is a function of both the x and y position of the leg. For a given leg geometry, any impact angle greater than the corresponding maximum shown in Figure 39 will result in the body contacting the ceiling before the leg. This shows a physical limitation of landing angle as a function of leg geometry.

The physical maximum impact angle, as shown in Figure 38, was derived by calculating the angle between the body and the ceiling such that the leg tip and body tip would touch the ceiling simultaneously (are at the same vertical position). The origin is defined as the center of the body segment, with positive-x to the right and positive-y upwards. If the body width is defined as the variable W, the right body tip is located at position $[\frac{W}{2}\ 0]$. The right leg tip is at location [x_{leg}, y_{leg}], where x_{leg} is positive and y_{leg} is negative. The maximum impact angle can be calculated using the law of cosines on the triangle with vertexes at the following 3 points: right body tip, right leg tip, and the center of the body as shown in Figure 38. The dimensions of the legs and body of the quadrotor are shown in Figure 36.

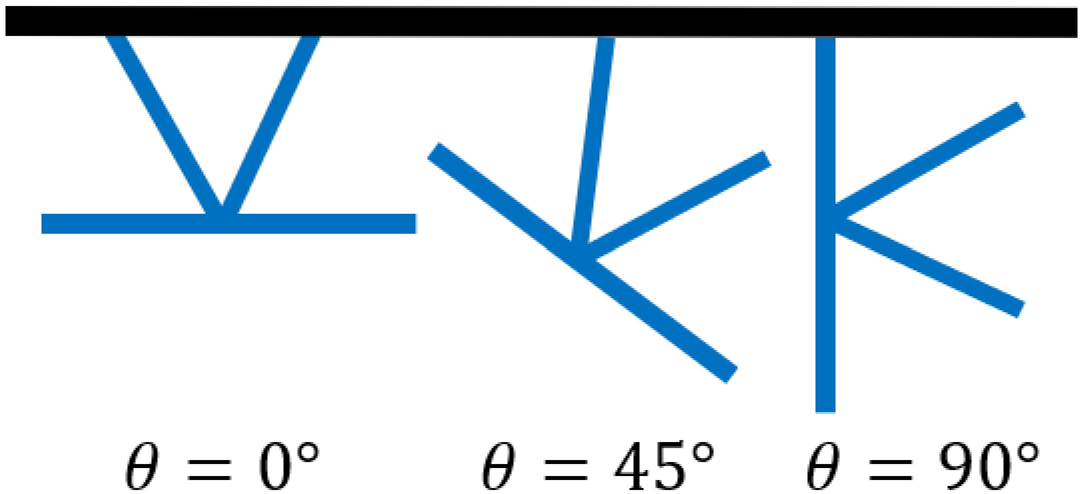

Figure 37: Impact angle between body and ceiling.

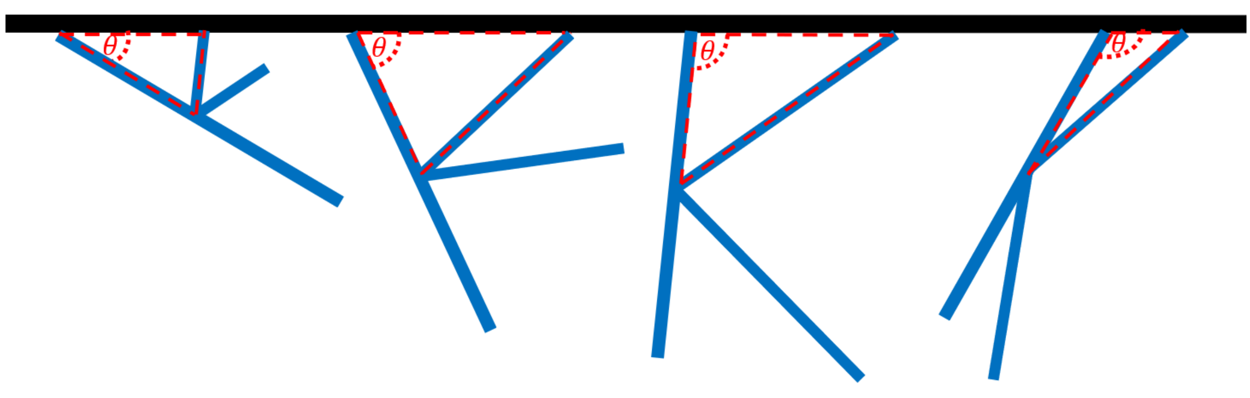

Figure 38: Physical maximum impact angle.

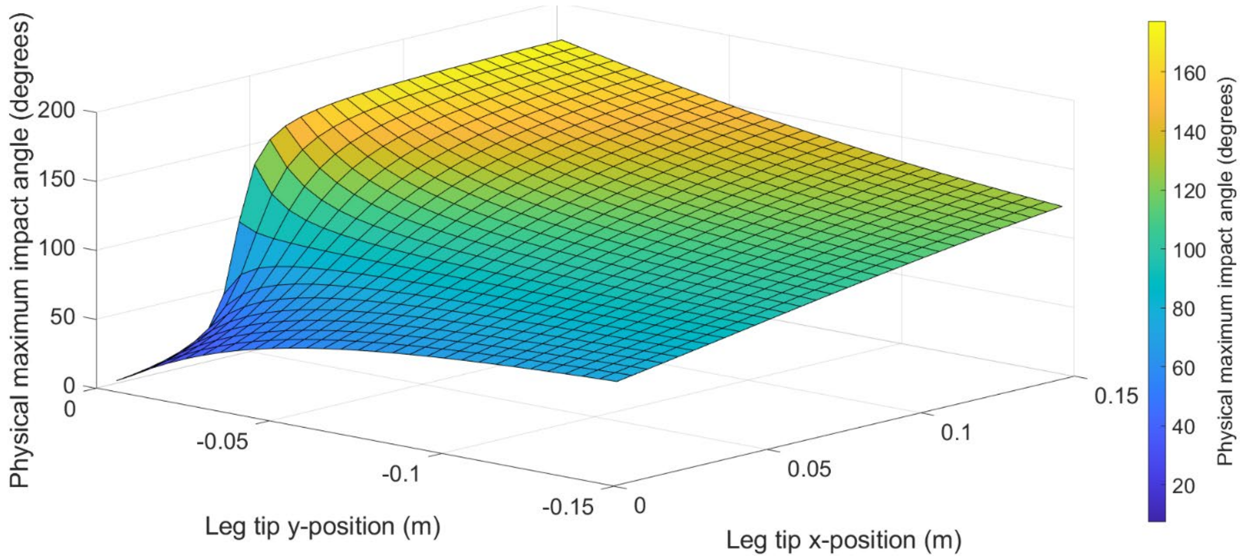

Figure 39: Physical maximum impact angle vs leg geometry.

High vertical velocity assisted failure is characterized by the vertical velocity contributing to angular momentum in the wrong direction. If the robot must rotate counterclockwise to complete a successful landing and the vertical momentum contributes towards clockwise rotation, this reduces the chances of successful landing. This category of failure can be identified by the center of mass being to the left of the impact point when the robot is flying to the right. The impact angle threshold for different leg geometries is shown in Figure 41. For any impact angle above the threshold shown in Figure 41, the vertical momentum contributes to angular momentum in the opposite direction than is necessary for successful landing. While it is possible to land despite the vertical velocity working against the successful landing rotation, successful landing becomes dependent on the angular momentum from body rotation before impact and horizontal velocity to overcome the vertical momentum.

Neutral vertical impact angle is equal to the angle between the leg and vertical as shown in Figure 40. This neutral impact angle is calculated as follows:

$$\textbf{Neutral impact angle} = \textbf{abs}\left(\textbf{atand}\left(\frac{x_{leg}}{y_{leg}}\right)\right) \tag{5.1}$$

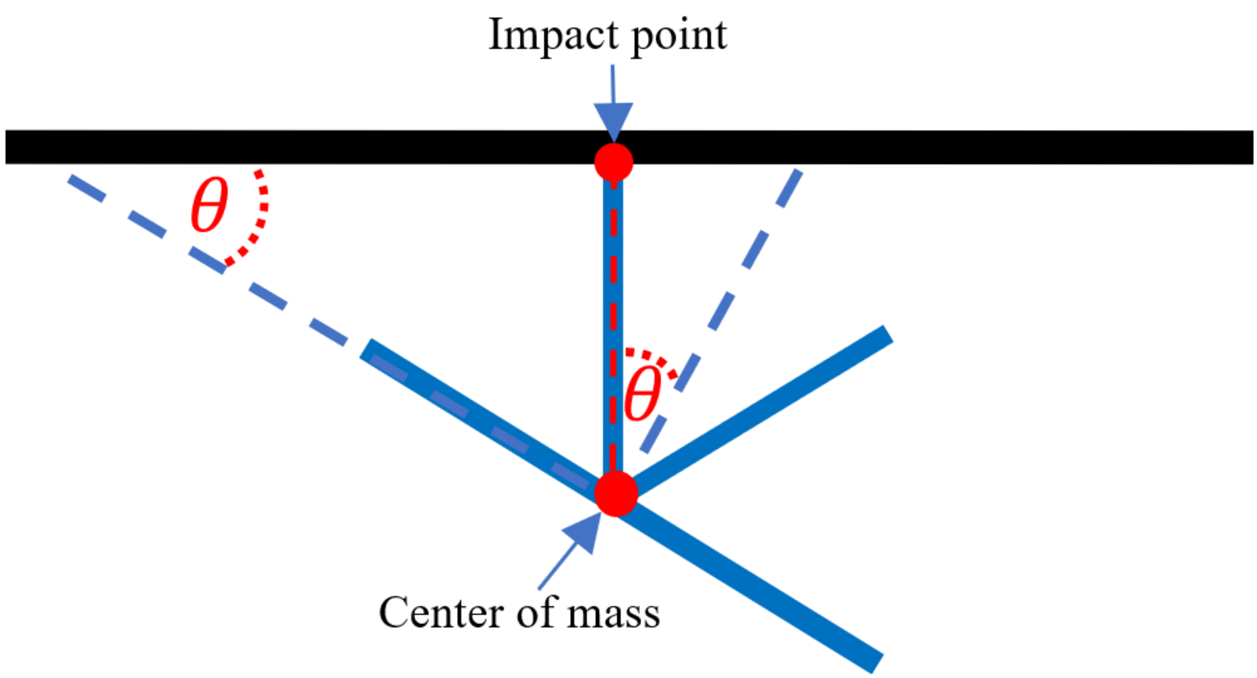

Figure 40: Neutral vertical momentum impact angle.

As shown in Figure 41, the neutral angle varies between 0 and 90 degrees. When the leg is long and narrow, there is a small range of impact angles where the vertical velocity will contribute to successful rotation and landing. When the legs are short and wide, there is a much wider range of impact angles where the vertical velocity will contribute to successful rotation and landing.

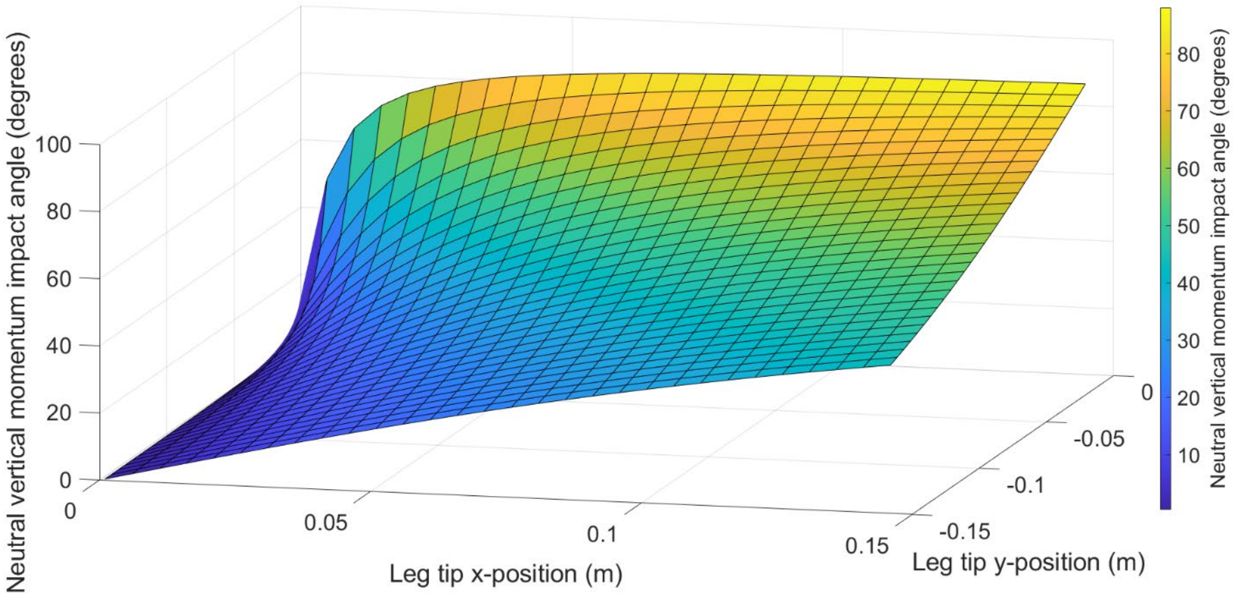

Figure 41: Neutral vertical momentum impact angle vs leg geometry.

High horizontal velocity assisted failure is characterized by the horizontal velocity contributing to angular momentum in the wrong direction after impact, lessening the momentum needed to swing the other leg to the ceiling or sometimes resulting in rotation away from the ceiling. This can occur when the robot over rotates before landing. Low velocity failure is characterized by minimal horizontal and vertical velocity upon impact. For this type of failure, there is not enough momentum and energy transferred to rotation to complete the landing.

Figure 42 shows how impact angle is defined for the impact landing trials, and Figure 43 shows different regions of successful landing.

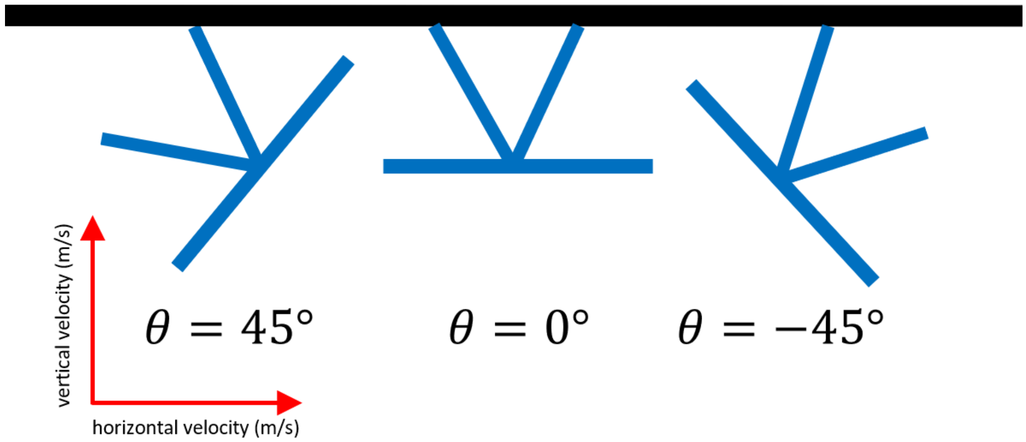

Figure 42: Landing angle notation.

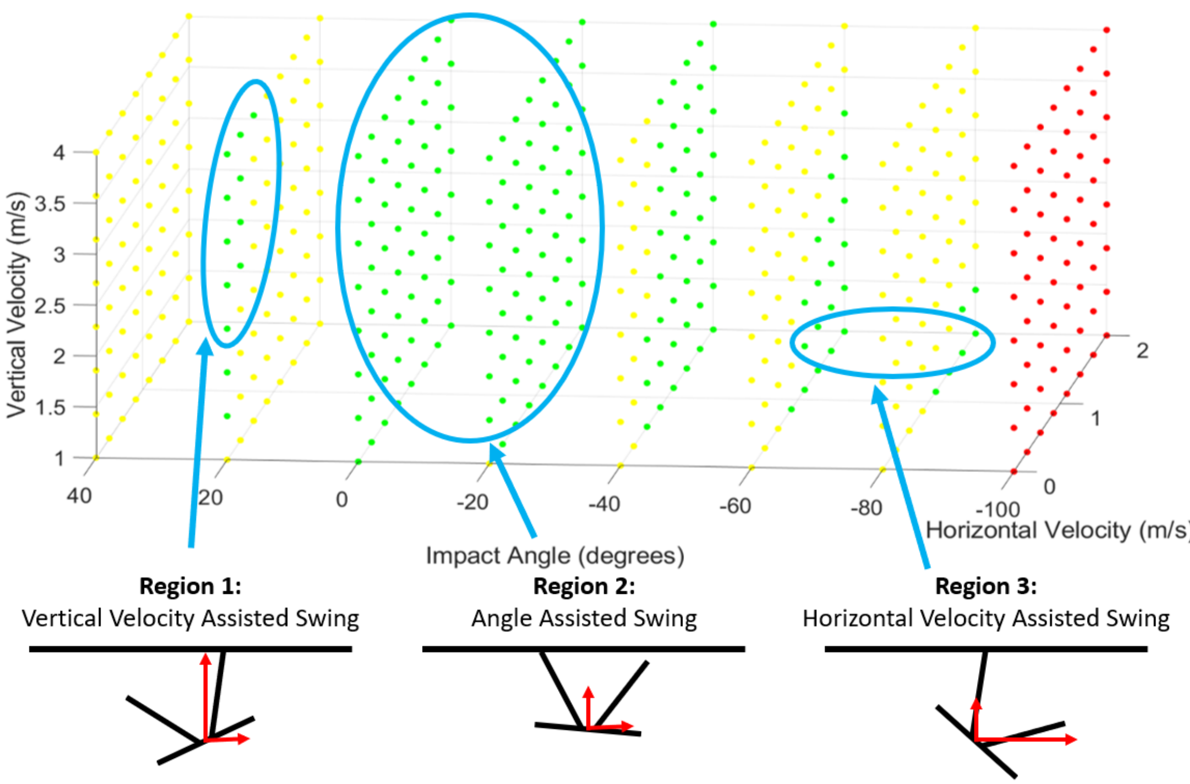

Figure 43: Landing regions- green: successful, yellow: incomplete, red: failure.

Figure 44 shows the success and failure to land under a variety of impact conditions for the extra-narrow short landing mechanism. This landing mechanism geometry shows a limited range of impact angles that lead to successful landings. Successful landing is dependent on high horizontal velocity.

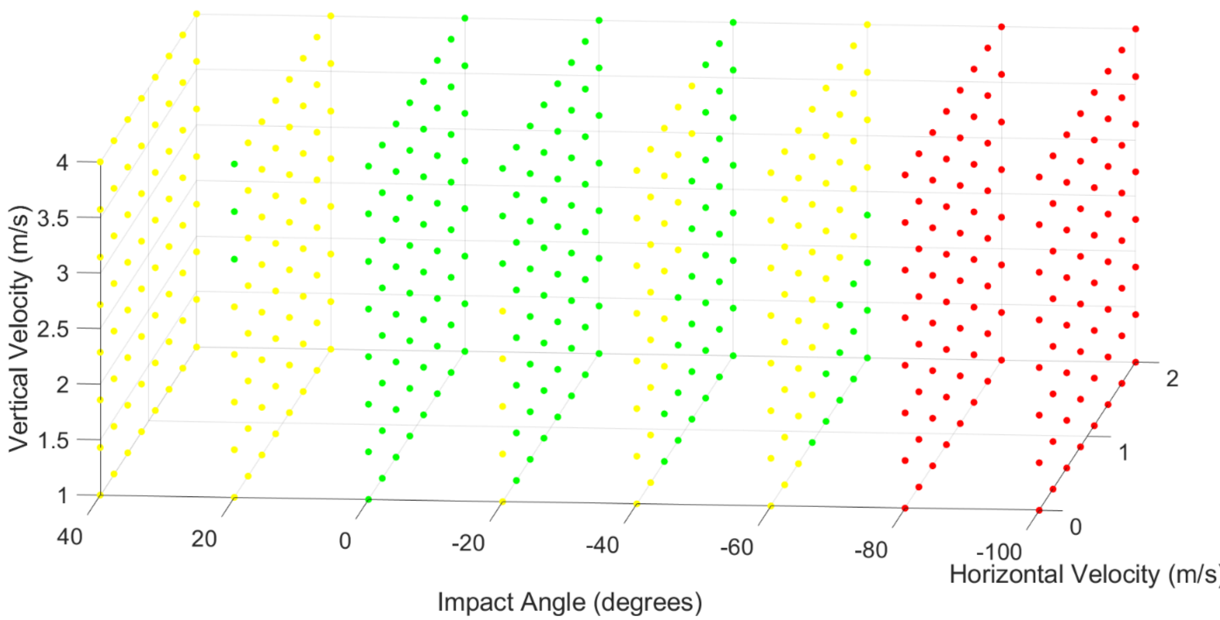

Figure 44: Extra-narrow short- green: successful, yellow: incomplete, red: failure.

Figure 45 shows landing success over a range of impact conditions for the narrow short landing mechanism geometry. This shows a wider range of impact angles that can result in successful landing than the extra-narrow-short landing mechanism; however, this geometry still depends on horizontal velocity to expand the range of impact angles that can be successful.

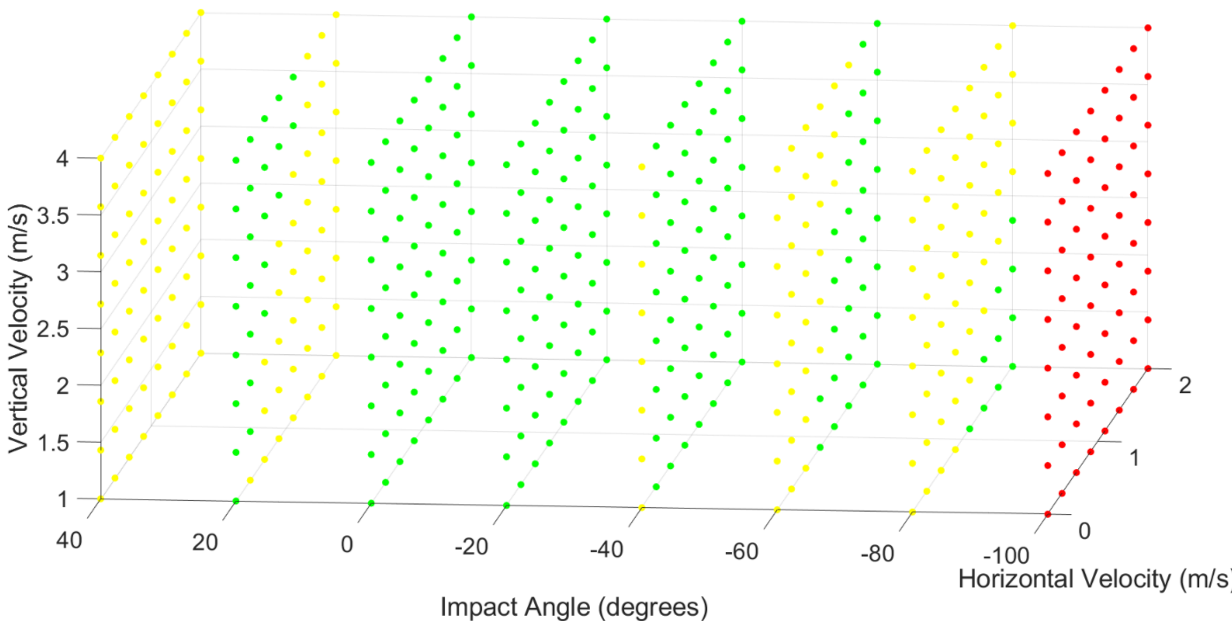

Figure 45: Narrow short- green: successful, yellow: incomplete, red: failure.

Figure 46 shows landing success over a range of impact conditions for the wide short landing mechanism geometry. This landing mechanism geometry shows robust landing over a wide range of conditions. The short wide configuration outperforms other configurations for low horizontal velocities.

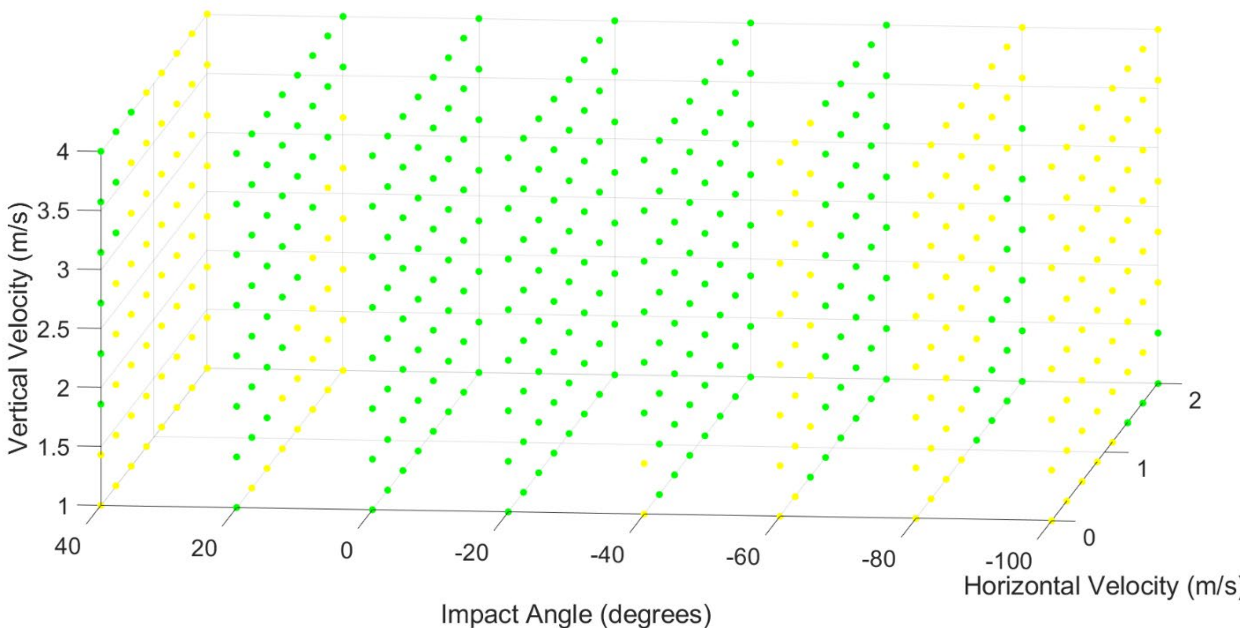

Figure 46: Wide short- green: successful, yellow: incomplete, red: failure.

Figure 47 shows landing success over a range of impact conditions for the extra-narrow long landing mechanism geometry. This landing mechanism has a narrow range of angles that will result in successful landing for low horizontal velocities.

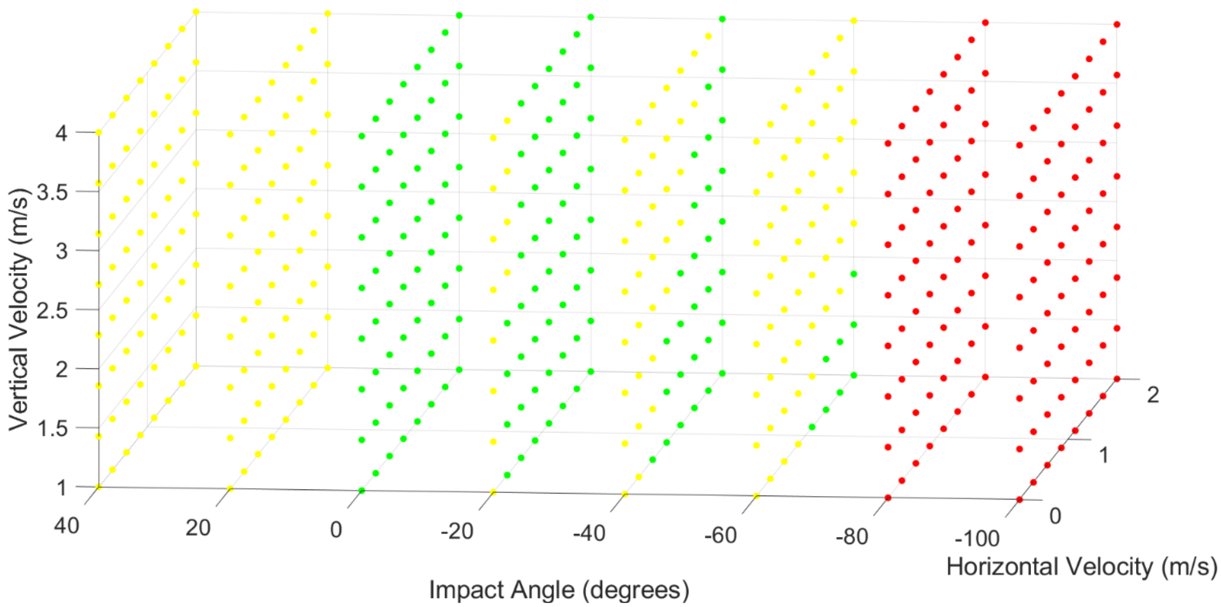

Figure 47: Extra-narrow long- green: successful, yellow: incomplete, red: failure.

Figure 48 shows landing success over a range of impact conditions for the narrow long landing mechanism geometry. This leg geometry results in high success for small impact angles but depends on high horizontal velocity to increase the range of impact angles that will result in successful landing.

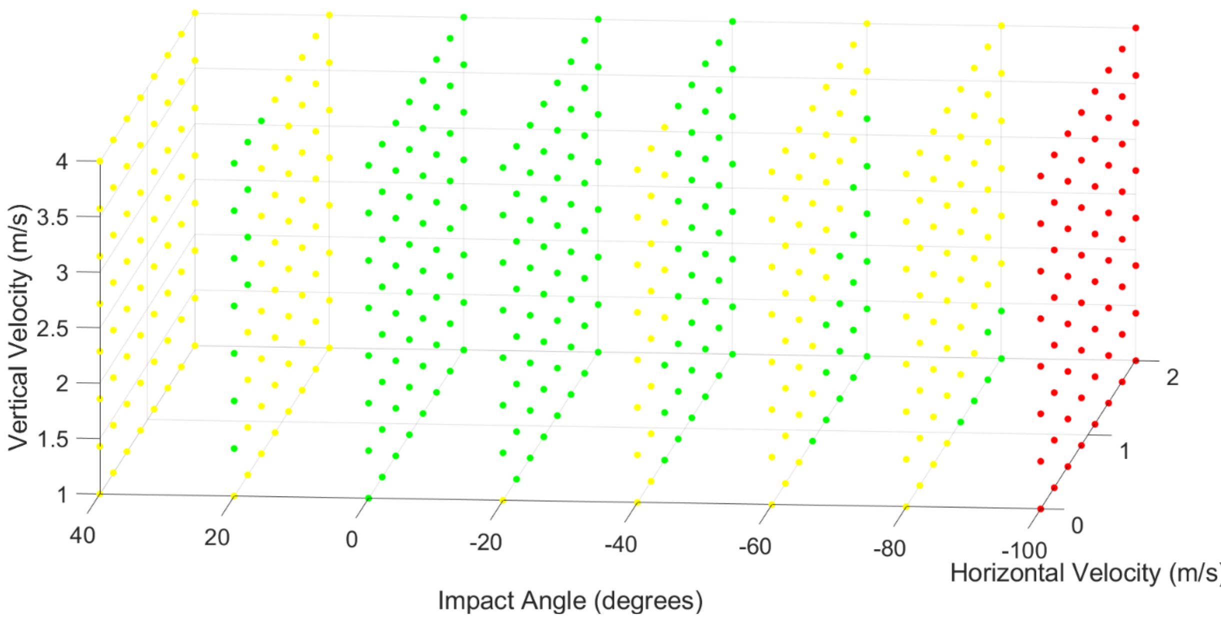

Figure 48: Narrow long- green: successful, yellow: incomplete, red: failure.

Figure 49 shows landing success over a range of impact conditions for the wide long landing mechanism geometry. This landing mechanism geometry shows high landing performance over a wide range of velocities and impact angles. Performance is better than the narrow landing mechanism geometries for low horizontal velocity trials.

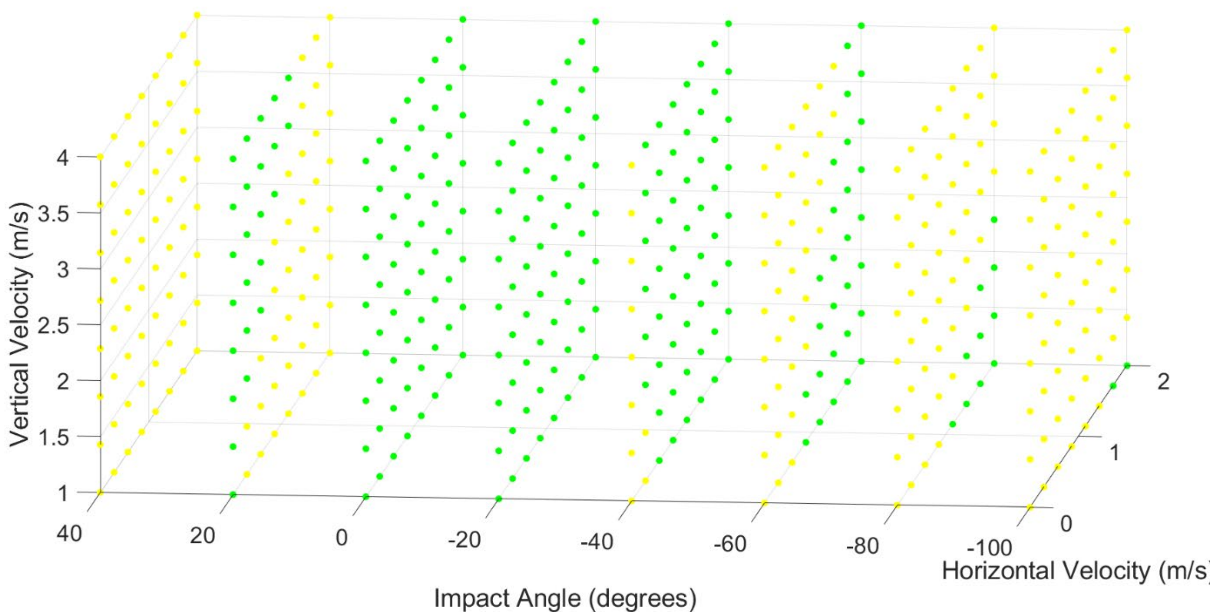

Figure 49: Wide long- green: successful, yellow: incomplete, red: failure.

Design Optimization of Landing Mechanism Geometry

The 2-D landing simulation environment was applied to the optimization of landing mechanism geometry. There are two primary areas of focus for addressing the robotic landing on the ceiling problem. The first area of focus is the robot and control of the robot. This determines how the robot senses the surroundings and how it reacts to these surroundings. The second area of focus is on the physical design of the landing mechanism, which is the interface between the robot and the ceiling from initial impact through completed landing.

The goal of the landing mechanism is to increase the robustness of the landing process. For this study, robustness of landing is defined as providing the greatest flexibility to the controller and expanding the margin of error such that successful landing can be achieved even with uncertainty and error in the sensing and control process. The controller is expected to evolve based on the design of the landing mechanism to maximize the landing success rate for a given situation with a specific landing mechanism. This presents an issue for the optimization of both the controller and the landing mechanism. The controller and the landing mechanism are inter-related, and the optimization of one is subject to the design of the other. However, attempting simultaneous evolution of both the control algorithm and the physical landing mechanism design is computationally intensive.

The optimization problem can be addressed by separating the control algorithm and landing mechanism optimization and following an iterative optimization strategy. The moment of impact provides a clear time to differentiate the purpose and responsibility of each component. Before impact, the control algorithm is responsible for maintaining stability, determining the correct moment to initiate rotation, and deciding the speed and/or angular acceleration to maintain during the rotation process. These actions by the controller must result in impact with the ceiling for a successful landing. The goal of the controller is to impact the ceiling with conditions such as vertical velocity, horizontal velocity, impact angle, and angular rotation rate such that landing will have the highest likelihood of success for a given landing mechanism. For the purposes of this optimization model, it was assumed that the motors shut off on or before impact with the ceiling. Once the motors shut off, it was assumed that the controller no longer has any impact on the events that follow. After impact, the robot is subject to rotation and motion under the influence of gravity. The purpose of the landing mechanism is to increase the probability of successful landing under a range of impact conditions.

In summary, the landing mechanism should be designed to increase the probability of successful landing under a wide range of impact conditions. The controller should learn how to react to the surroundings and control the robot such that it impacts the ceiling under conditions that are favorable to successful landing for a given landing mechanism.

Under this model, a landing mechanism that will promote successful landing under a wide range of impact conditions will allow for more flexibility in the impact conditions that the controller must seek to achieve. A controller and system that can more consistently achieve a desired impact condition under the widest range of initial conditions should be able to consistently impact the ceiling under conditions that will result in successful landing with a given landing mechanism.

The controller is determined through machine learning in the Gazebo simulation environment. In this simulation, the landing mechanism design is fixed, and the control algorithm is optimized for the robot in the configuration with that landing mechanism design. The landing mechanism geometry can be optimized separately to find the design that increases the success rate over a range of impact conditions. For this optimization process, three key impact conditions are considered for the 2-D case: horizontal component of velocity, vertical component of velocity, and impact angle.

For the first step in the optimization of the landing mechanism, a wide range of horizontal velocities, vertical velocities, and impact angles should be chosen. These ranges should be chosen such that they are physically achievable by the controller and can reasonably be expected to result in successful landing. If regions that are not attainable are chosen, the landing mechanism will be optimized to be successful under impact conditions that are not possible, potentially at the detriment of performance under impact conditions that are attainable by the robot.

Next, a series of equally spaced sample values should be chosen within each range. For example, if the vertical impact velocity range of 0-2 m/s is chosen, discrete sample values of 0, 1, and 2 m/s can be chosen. The same process should be followed for horizontal impact velocity and impact angle. Next, a complete set of impact conditions should be created by forming every combination of the three sets of impact conditions. This set of impact conditions will serve as a benchmark for a given landing mechanism. The goal of the optimization process will be to identify the landing mechanism geometry that will result in successful landing for the greatest number of these impact conditions.

The optimization of the landing mechanism under this series of impact conditions can be organized as follows. The objective function for the optimization process is the successful landing rate over the set of landing conditions. The variables to be optimized are defined as the horizontal and vertical position of the right leg tip relative to the center of mass. All legs are assumed to be symmetrical. For each iteration, a given landing mechanism is tested under each impact condition defined in the set and an objective score is calculated based on the number of landings that were successful. The leg parameters continue to be iterated until landing success rate is maximized.

After a landing mechanism geometry is optimized, the design should be implemented in the controller simulation, and the controller should be optimized for this new landing mechanism design. Based on the impact conditions that occur from the new controller, a new set of desirable impact conditions can be formed for optimization of the landing mechanism. This newly optimized landing mechanism can be implemented for another round of controller optimization. This process can be repeated until the range of landing conditions the controller achieves consistently matches the range of impact conditions that the landing mechanism has been optimized for.

To demonstrate one step of the optimization process, the landing mechanism geometry was optimized over a range of impact conditions. The chosen impact conditions were vertical velocity over the range 0.2 m/s to 1.2 m/s, horizontal velocity of 0 m/s to 1 m/s, and impact angle of 10 degrees to -90 degrees. These impact conditions represent achievable impact conditions for the Crazyflie quadrotor.

Simulations were first run using the extra-narrow short landing mechanism to evaluate performance. The results are shown in Figure 50. Next, the optimization algorithm was run with the objective of increasing the success rate under the same impact conditions. The landing mechanism x dimension was constrained to between 0.001 m. and 0.1 m., and the y dimension was constrained to be between -0.1 m. and -0.02 m. The assumed body width was 0.08 m. for both trials. The optimized landing mechanism was found to have an x-dimension of 0.0491 m. and a y-dimension of -0.0206 m. This optimized landing mechanism was tested over the range of impact conditions, and the result is shown in Figure 51. There is a clear increase in performance between the extra-narrow landing mechanism and the optimized landing mechanism. The optimized landing mechanism design was slightly wider than the body width, and the y dimension was close to the minimum bound set for optimization. This short and wide geometry allowed for the greatest rate of successful landings under the specified impact conditions. This landing design should allow for greater margin of error for the controller, thereby increasing the robustness and repeatability of the landing process.

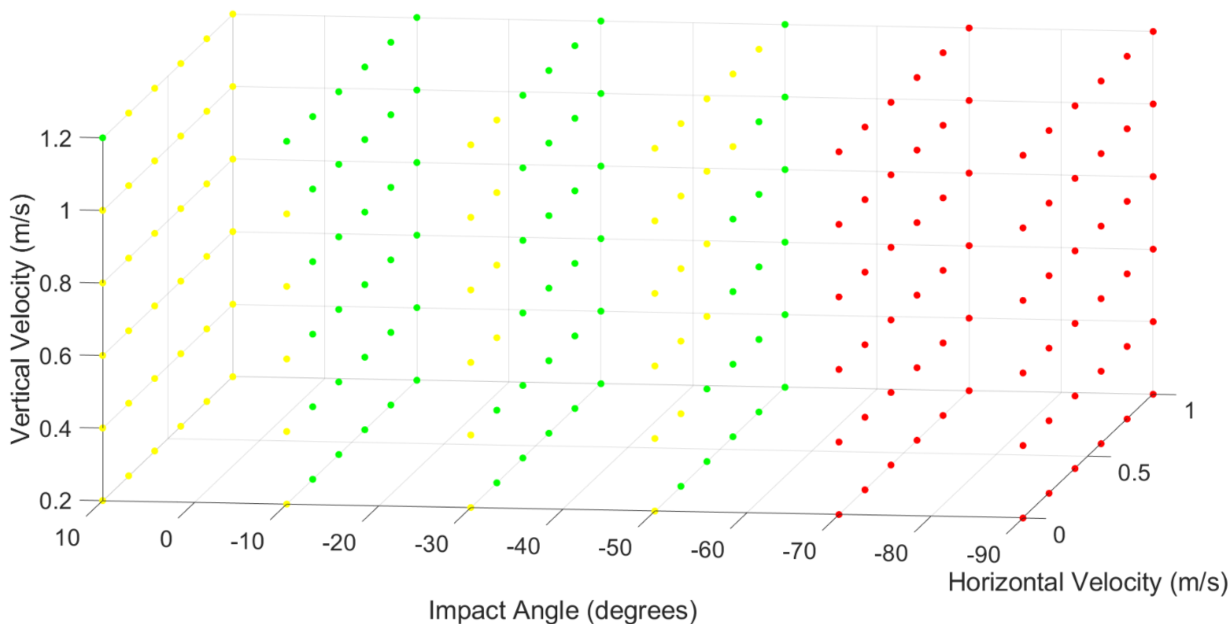

Figure 50: Extra-narrow short landing mechanism performance before optimization- green: successful, yellow: incomplete, red: failure.

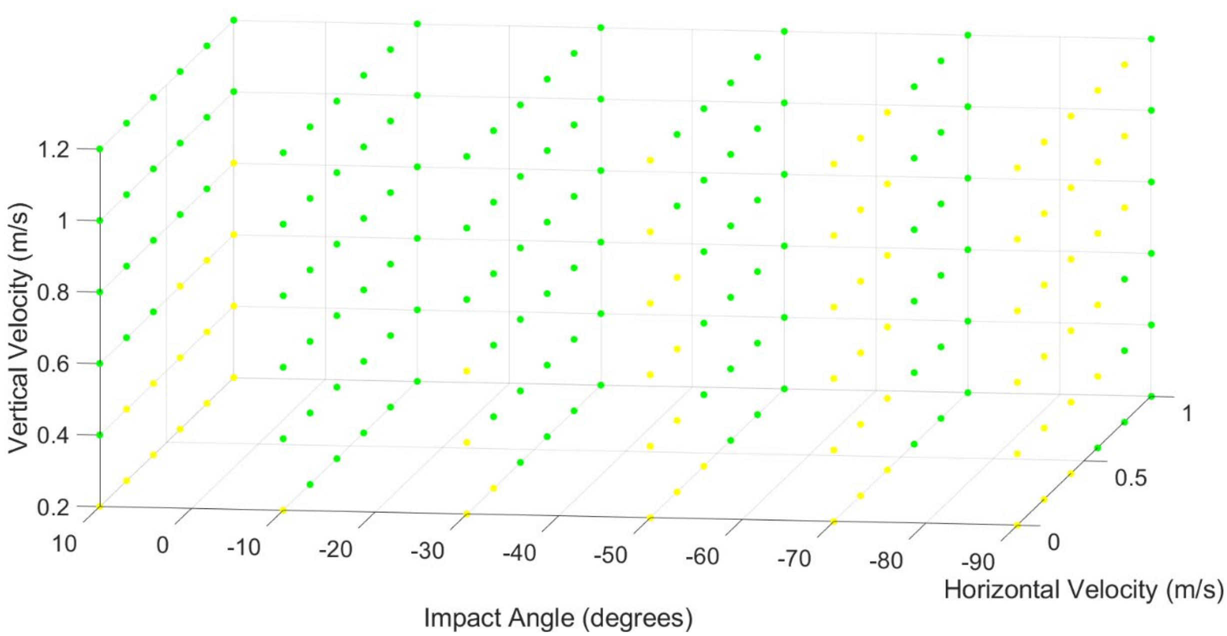

Figure 51: Landing mechanism performance after optimization- green: successful, yellow: incomplete, red: failure.

Chapter 6

Design and Testing of a Quadrotor Launching Platform

Purpose of Launcher Design

A quadrotor launching platform was designed and implemented to aid in the testing and development of the ceiling landing controller. The purpose of this launching platform was to allow the ceiling landing study to focus primarily on the landing behavior and put less focus on trying to achieve a wide range of initial conditions with the quadrotor alone. There are limits to quadrotor technology at this scale, and while the Crazyflie was modified to lower its mass, it was only able to achieve limited velocities in the constrained test space. The launcher will allow for testing and improvement of the quadrotor's landing behavior under a wider range of initial conditions than would be attainable without the launcher. As quadrotor and battery technology improve, or when the testing space is expanded, the controller will already have been trained under the higher velocity conditions.

Mechanical Design

The goal of the launching platform is to achieve an initial vertical velocity over the range of 1-3 m/s. The platform must be able to launch a 50-gram mass at the desired velocity range. The launcher has two primary means of actuation. Option 1 uses the potential energy of a dropped mass along with a class 1 lever to propel the platform and quadrotor upwards with a desired initial velocity. Method 2 uses an electromagnet at the base and permanent magnets on the launching platform for magnetic propulsion of the platform. A computer aided design model for a launching mechanism is shown in Figure 52.

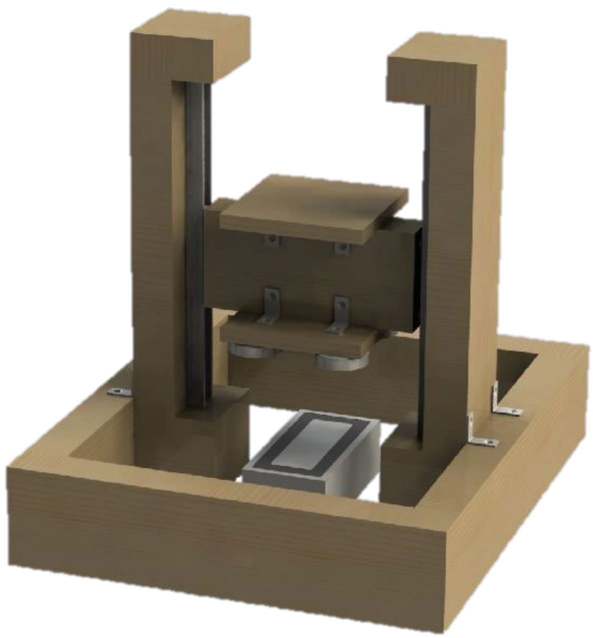

Figure 52: Launching mechanism CAD design.

Fabrication

The launcher was designed to accommodate both dropped mass and electromagnetic means of launching, but the dropped mass propulsion method was implemented due to the simplicity of operation. Later, the launcher can be converted to permit computer-controlled electromagnetic launch and/or initial launch at different angles. The design was implemented with a primarily wood structure. Using wood for construction allowed for a simple and rugged build with enough weight to prevent movement during the launching process. A linear slide, designed for use with drawers, was used to constrain the launching platform to linear motion. The fabricated launcher is shown in Figure 53.

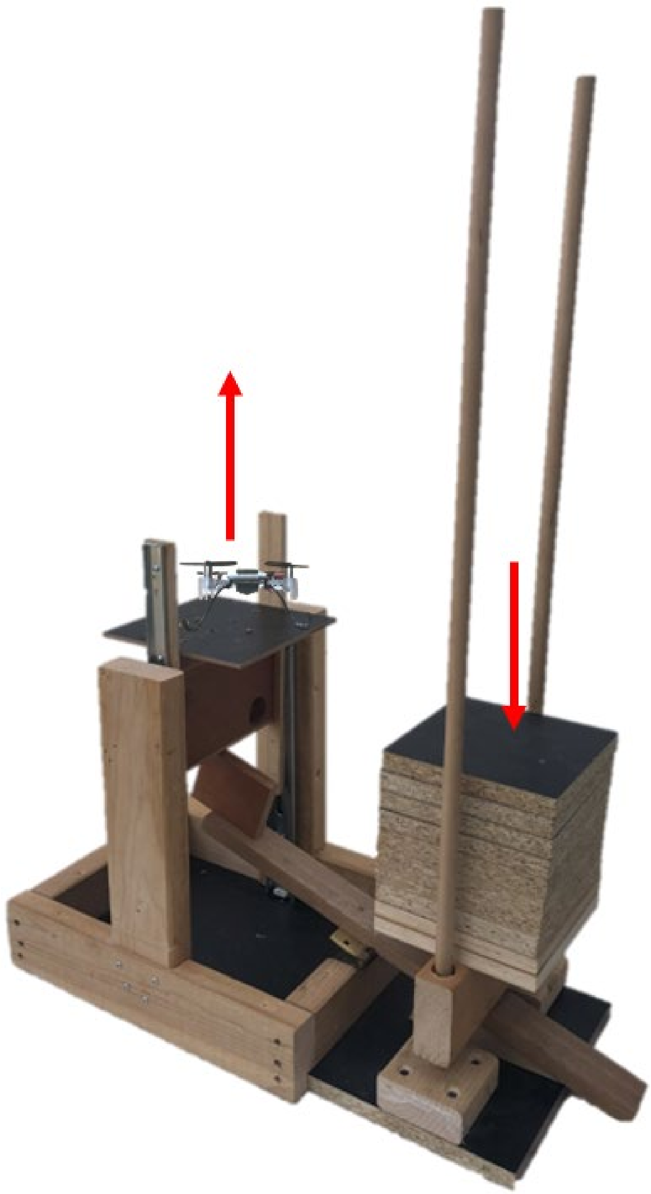

Figure 53: Fabricated launching mechanism.

Testing and Data Collection

The input-output behavior of the launcher needed to be characterized so that the desired launch velocity could be consistently attained. To accomplish this goal, an experiment was designed to collect data on the drop height of the mass and the maximum height of the projectile. This maximum height was then used to calculate the initial launch velocity of the platform for each drop height of the mass.

The experiment was conducted by selecting a series of mass drop heights. For each drop height, three trials were conducted. To record the drop height and maximum projectile height, an extended tape measure was placed in the background and a slow-motion video was captured of the launch process.

Analysis of Testing Data

The test data recorded the drop height of the mass and the maximum height of the projectile, measured as the distance above the ground. Each of these was adjusted to find the effective drop height of the block and the effective launch height. Drop height was set to zero when the base of the dropped mass was at its lowest point. Launch height was set to zero where the launch surface is when the dropped mass is at its lowest position. This position was chosen because the launched projectile will leave the platform when the platform is no longer being accelerated upwards by the lever. For the initial velocity calculation, ideal projectile motion was assumed where air resistance is negligible. Each projectile height was converted to an initial launch velocity. Next, an equation was fit to the drop height and launch velocity data as shown in Figure 54. Equation 6.1 describes the line of best fit. This allows the operator to control the launch velocity by adjusting the height that the block is dropped from.

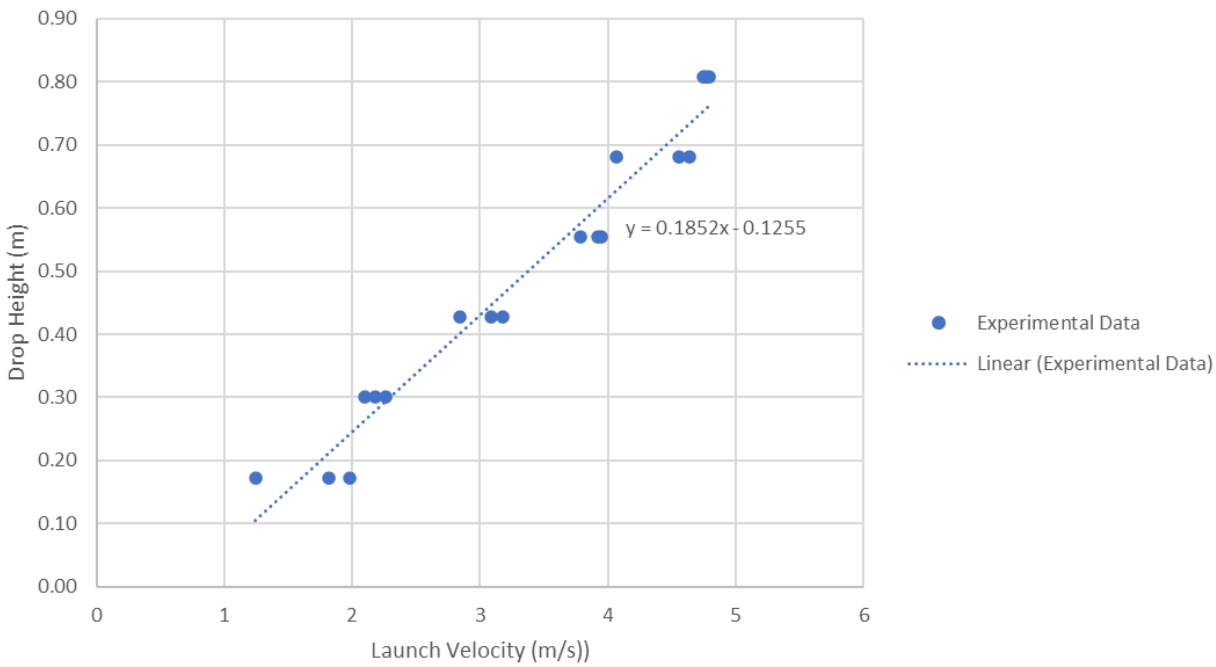

Figure 54: Launch velocity vs drop height experimental data.

Equation 6.1 shows the necessary drop height to achieve a desired launch velocity.

$$\boldsymbol{h_d = 0.1852 * v - 0.1255} \tag{6.1}$$

Chapter 7

Future Inquiry, Conclusions, and Applications

Conclusions

An additively manufactured landing mechanism was designed and mounted to the Crazyflie quadrotor to enable and assist inverted landing on the ceiling. This landing mechanism design is lightweight and can be manufactured with a range of different stiffnesses and geometries.

The effects of landing mechanism geometry were tested under a wide range of impact conditions in a simulated environment for comparison between different designs. These experiments showed a clear impact of landing mechanism geometry on the impact conditions that would lead to successful landing.

Idealized landing control strategies were derived to better understand the process of rotation and landing on the ceiling with several landing strategies such as constant angular velocity and constant angular acceleration rotation. These control policies were applied to examine the difference in the control behavior based on sensor error and uncertainty.

A method of optimization of both the controller and landing mechanism geometry was proposed to increase the success rate of inverted landing.

A launching platform was designed to increase the range of initial conditions under which the robot lands on the ceiling. This launching platform was fabricated and tested to enable the desired initial velocity to be imparted on the quadrotor during testing.

Altogether, this work provides insight into key factors in the design and analysis of a landing mechanism and a control algorithm for inverted landing. These findings will help to facilitate the development of robust inverted landing of small quadrotors as research on insects and robots in the simulated environment is transferred to physical robots.

Future Inquiry

Based on the findings, several additional questions and areas for inquiry arose. One area is to investigate how the learned control patterns for landing on the ceiling change with other landing mechanism factors such as leg stiffness, number of legs, and leg shape. This study examined the impact of leg geometry on successful landing behavior; however, introducing additional factors such as lessening the stiffness of the legs or increasing the number of legs to 6 or more could result in different learned methods of landing. Additionally, the simulation environments can be compared to physical trials, and the simulations can be refined to match the observations in the experimental trials.

Furthermore, the learned behaviors for the robotic flier can be compared to the patterns observed in flies and flying insects. Landing patterns have been observed in flies as a group, and several landing patterns have been characterized. This group behavior could be compared with the behavior of individual flies to determine if individual flies exhibit multiple landing behaviors or if an individual fly learns and repeats a single behavior. This could then be compared with the learned robotic policies. This could provide insight into the learned behavior and nature of the algorithms used by small insects.

Lastly, the impact of the adhesive or Velcro can be examined. The method of attachment of these materials could impact the behavior upon impact with a surface. Additional materials could be tested to allow for landing on any surface.

Application to Small Aerial Robotics

The use of RREV as a control mechanism is valuable for the future development of small robotic systems with limited processing and weight capacity. Sensing and reacting to the surroundings with simple sensing and processing may allow for miniaturized robotics like the Crazyflie to have complex interactions with the environment. While this study focuses specifically on a small quadrotor landing on a ceiling, this behavior can be later expanded to navigating and landing on a variety of surfaces in different orientations. Behaviors of insects will continue to serve as inspiration for implementation of small-scale sensing and control. As robotic and energy storage technology continues to improve, these studies will serve as a foundation for the sensing, control, and design of small robots to execute complex landing for a variety of applications.

Appendix A

Control Policy 2 Derivation

$t_a = acceleration\ duration$

$t_b = duration\ between\ motor\ shutoff\ and\ touchdown$

t_{impact} is defined by Equation 4.1.

$$\theta_d = 180\ degrees = \frac{1}{2}\alpha * t_a^2 + \alpha * t_a * t_b$$

$$t_{impact} = t_a + t_b$$

$$t_b = t_{impact} - t_a$$

$$180 = \frac{1}{2}\alpha * t_a^2 + \alpha * t_a * (t_{impact} - t_a)$$

$$\left(\frac{1}{2} * \alpha\right) * t_a^2 - \left(\alpha * t_{impact}\right) * t_a + 180\ = 0$$

$$t_a = \frac{\alpha * t_{impact} - \sqrt[2]{\left(\alpha * t_{impact}\right)^2 - 4 * \left(-\frac{\alpha}{2}\right) * (-180)}}{\alpha}$$

$$t_a = \frac{\alpha * t_{impact} - \sqrt[2]{\left(\alpha * t_{impact}\right)^2 - 360 * \alpha}}{\alpha}$$

$$\boldsymbol{t_a = t_{impact} - \frac{\sqrt[2]{\left(\alpha * t_{impact}\right)^2 - 360 * \alpha}}{\alpha}} \tag{4.5}$$

Appendix B

Control Policy 3 Derivation

θ_d: $desired\ angular\ displacement\ from\ start\ of\ rotation\ to\ landing\ (degrees)$

θ_s: $motor\ shutoff\ angle\ (degrees)$

t_{impact} is defined by Equation 4.1.

$$\theta_d = \frac{1}{2}\alpha * t_a^2 + \alpha * t_a * t_b$$

$$t_{impact} = t_a + t_b$$

$$t_b = t_{impact} - t_a$$

$$\theta_d = \frac{1}{2}\alpha * t_a^2 + \alpha * t_a * (t_{impact} - t_a)$$

$$\theta_d = (\frac{-1}{2} * \alpha) * t_a^2 + (\alpha * t_{impact}) * t_a$$

$$\theta_s = \frac{1}{2} * \ \alpha * t_a^2$$

$$\theta_d = \ -\theta_s + \ \alpha * t_a * t_{impact}$$

$$\theta_d + \ \theta_s = \ \alpha * t_a * t_{impact}$$

$$\alpha = \ \frac{\theta_d + \ \theta_s}{t_a * t_{impact}}$$

$$t_a = \sqrt[2]{\frac{2 * \theta_s}{\alpha}}$$

$$\left(\sqrt[2]{\frac{2 * \theta_s}{\alpha}} * t_{impact} * \ \alpha\right)^2 = (\theta_d + \theta_s)^2$$

$$2 * \theta_s * t_{impact}^2 * \alpha = \ (\theta_d + \theta_s)^2$$

$$\boldsymbol{\alpha} = \frac{(\boldsymbol{\theta_d} + \boldsymbol{\theta_s})^2}{2 * t_{impact}^2 * \boldsymbol{\theta_s}}$$

$$\boldsymbol{\alpha} = \frac{405}{t_{impact}^2}, \qquad \textbf{where } \boldsymbol{\theta_d} = 180 \; \boldsymbol{and} \;\; \boldsymbol{\theta_s} = 90 \tag{4.6}$$

Appendix C

Impact Time Using RREV Derivation

$$t: impact\ time\ (s)$$

$$\Delta y = V_{y_i} * t - \frac{1}{2} * g * t^2\ (Projectile\ Equation)$$

$$\frac{1}{2} * g * t^2 - V_{y_i} * t + \Delta y = 0\ \ (rearanged\ equation\ to\ put\ in\ quadratic\ form)$$

$$t = \left(\frac{1}{g}\right) * (V_{y_i} - \sqrt[2]{V_{y_i}^2 - 2 * g * \ \Delta y}\ ): calculated\ impact\ time$$

$$RREV\ = \frac{V_{y_i}}{\Delta y}: definition$$

$$\Delta y = \frac{V_{y_i}}{RREV}: rearrange\ previous\ equation$$

$$\boldsymbol{t = \left(\frac{1}{g}\right) * \left((V_{y_i} - \sqrt[2]{V_{y_i}^2 - 2 * g * \frac{V_{y_i}}{RREV}}\ \right)} \tag{4.8}$$

BIBLIOGRAPHY

1. Baird E, Boeddeker, N, Ibbotson, M, Srinivasan M. A universal strategy for visually guided landing. Proceedings of the National Adademy of Sciences of the United States of America; 2013; 110(46) 18686-18691; doi: 10.1073/pnas.1314311110
2. van Breugel Floris, Dickinson Michael. The visual control of landing and obstacle avoidance in the fruit fly Drosophila melangaster; The Journal of Experimental Biology; 2012; doi:10.1242/jeb.066498
3. Desbiens Alexis, Asbeck Alan, Cutkosky Mark. Landing, perching, and taking off from vertical surfaces. International Journal of Robotics Research. 2011; 30(3); doi: 10.1177/0278364910393286
4. Evangelista C, Kraft P, Dacke M, Reinhard J, Srinivasan M. The moment before touchdown: landing manoeuvres of the honeybee Apis mellifera. The Journal of Experimental Biology; 2010; 213:262-270; doi: 10.1242/jeb.037465
5. Kovac Mirko; Learning from nature how to land aerial robots. Science Magazine. 2016; 352(6288): 895-896; doi: 10.1126/science.aaf6605
6. Liu P, Sane SP, Mongeau JM, Zhao J, Cheng B. Flies land upside down on a ceiling using rapid visually mediated rotational maneuvers. Sci Adv. 2019; 5(10); doi:10.1126/sciadv.aax1877
7. Ma KY, Chirarattananon P, Fuller SB, Wood RJ. Controlled flight of a biologically inspired, insect-scale robot. Science. 2013; 340(6132): 603-607; doi:10.1126/science.1231806
8. Mellinger D, Michael N, Kumar V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. In: Springer Tracts in Advanced Robotics. ; 2014; 31(5): 664-674; doi:10.1007/978-3-642-28572-1_25
9. Roderick WRT, Cutkosky MR, Lentink D. Touchdown to take-off: At the interface of flight and surface locomotion. Interface Focus. 2017; 6(1); doi:10.1098/rsfs.2016.0094
10. Wagner H. Flow-field variables trigger landing in flies; Nature Research Journal; 1982; 297:147-148; doi: 10.1038/297147a0

Academic Vita of Brian Davis
bpd5194@psu.edu

EDUCATION

The Pennsylvania State University: Schreyer Honors College

Bachelor of Science in Mechanical Engineering

WORK EXPERIENCE

Mechanical Engineering Intern: Lockheed Martin Rotary and Mission Systems Summer 2020

- Worked on trade studies to evaluate technical feasibility for future upgrades.
- Developed preliminary designs and investigated design tradeoffs for electromechanical systems.
- Presented findings to guide future development efforts.

Mechanical Engineering Intern: Lockheed Martin Rotary and Mission Systems Summer 2019

- Performed detailed mechanical design for hardware engineering team.
- Developed CAD models, bill of materials, and drawings using Creo Parametric.
- Interpreted schematics and updated testing procedures and documentation.
- Provided manufacturing support for components in production, analyzed design intent, and delivered recommendations.

ENGINEERING EXPERIENCE

President, Chief Engineer: Penn State Unmanned Aerial Systems Club 2017- 2021

System Design and Integration:

- Lead interdisciplinary team of mechanical, electrical, and computer science students to develop autonomous UAV systems.
- Developed systems capable of conducting reconnaissance missions, navigating airspace, and capturing/processing aerial image data.
- Programmed 3-D obstacle avoidance software and integrated with vehicle.

Flight Vehicle Design and Payload Development:

- Designed and tested autonomous fixed wing, multirotor, and helicopter platforms.
- Designed and manufactured vibration isolated enclosure for imaging system.

Electromechanical Design and Control:

- Designed and tested autonomous ground vehicle.
- Prototyped and tested mechanisms for precise lowering an autonomous ground vehicle from unmanned helicopter from an altitude of 100 feet to the ground.
- Designed electrical and control systems for antenna tracking system including component selection, network integration, and control algorithm development.

Undergraduate Researcher: Penn State Biomechanics and Control Lab 2020- 2021

- Developed a small-scale biologically inspired quadrotor to enable inverted landings using limited on-board sensors.
- Analyzed and optimized design using simulated environment.