# GRASPING, PERCHING, AND VISUAL SERVOING FOR MICRO AERIAL VEHICLES

Justin R. Thomas

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2017

---

Vijay Kumar, Supervisor of Dissertation
Nemirovsky Family Dean of Penn Engineering and Professor of Mechanical Engineering and Applied Mechanics

---

Kevin Turner, Graduate Group Chairperson
Professor of Mechanical Engineering and Applied Mechanics

Dissertation Committee

Camillo J. Taylor, Professor of Computer and Information Science

Vijay Kumar, Nemirovsky Family Dean of Penn Engineering and Professor of Mechanical Engineering and Applied Mechanics

Kostas Daniilidis, Professor of Computer and Information Science

Koushil Sreenath, Assistant Professor of Mechanical Engineering

*Dedicated to my loving family*

# Acknowledgments

I would like to express my utmost gratitude to my advisor, Vijay Kumar, for his support and guidance. His confidence in me has been inspirational, and he is truly a great role model both personally and professionally. I value the involvement from my committee members, whose feedback has been instrumental in my progress through the years. Many thanks goes to Giuseppe, my research buddy and close friend, for our lively discussions and global escapades. It is with pride that I claim Denise Wong as my academic twin. Many other lab members and staff deserve appreciation for their guidance and support on various projects.

I have been blessed to advise many fantastic undergraduate students. They will continue to be an encouragement and blessing to many. I have also enjoyed teaming up with excellent collaborators from Stanford University, the University of Maryland, and the Czech Technical University.

My family has been incredibly supportive, encouraging me at a young age and ensuring that I had access to the resources needed to succeed. My lovely girlfriend, Cori, has been patient despite late nights and learning how to maintain a healthy work/life balance.

My research would not have been possible without the financial support of the Ashton scholarship, ARL grant W911NF-08-2-0004, ONR grants N00014-07-1-0829, N00014-14-1-0510, N00014-09-1-1051, N00014-09-1-103, and NSF grants IIP-1113830, IIS-1426840, and IIS-1138847. Thus, I am grateful for an institution and a nation dedicated to pushing the boundaries of knowledge.

# ABSTRACT

GRASPING, PERCHING, AND VISUAL SERVOING FOR MICRO AERIAL VEHICLES

Justin R. Thomas

Vijay Kumar

Micro Aerial Vehicles (MAVs) have seen a dramatic growth in the consumer market because of their ability to provide new vantage points for aerial photography and videography. However, there is little consideration for physical interaction with the environment surrounding them. Onboard manipulators are absent, and onboard perception, if existent, is used to avoid obstacles and maintain a minimum distance from them. There are many applications, however, which would benefit greatly from aerial manipulation or flight in close proximity to structures. This work is focused on facilitating these types of close interactions between quadrotors and surrounding objects. We first explore high-speed grasping, enabling a quadrotor to quickly grasp an object while moving at a high relative velocity. Next, we discuss planning and control strategies, empowering a quadrotor to perch on vertical surfaces using a downward-facing gripper. Then, we demonstrate that such interactions can be achieved using only onboard sensors by incorporating vision-based control and vision-based planning. In particular, we show how a quadrotor can use a single camera and an Inertial Measurement Unit (IMU) to perch on a cylinder. Finally, we generalize our approach to consider objects in motion, and we present relative pose estimation and planning, enabling tracking of a moving sphere using only an onboard camera and IMU.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The capabilities of Micro Aerial Vehicles (MAVs) are rapidly expanding to include surveillance [89], construction [71], manipulation of slung loads [118], collaborative transportation [90, 117], and mapping of unknown environments [110, 138]. However, aerial vehicles currently lack the ability to physically interact with their surroundings in real-world scenarios. Indeed, the ability to maneuver precisely relative to specific targets and to interact with the environment (*i.e.,* aerial manipulation) would create many opportunities for MAVs and could have significant benefits to society. For example, precise relative positioning would allow for close inspections of bridges, cell towers, rooftops, or water towers. Aerial manipulation could be useful for precision farming, construction, repairing structures, transporting objects, automated recharging, automated battery replacement, environmental sampling, or perching to turn off motors and reduce power consumption. This relative positioning would also be necessary for cooperative behavior using swarms for tasks such as cooperative aerial manipulation or maintaining formations of diverse, heterogeneous teams.

The prevalence of commercially-available MAVs is rising quickly, but platforms are currently limited to observation only. Indeed, there are many manufacturers producing aerial robots equipped with cameras, however, none are able to physically interact with objects (*e.g.,* grasp, manipulate, perch, etc.). Even more limiting is the fact that most are not able to maneuver relative to specific objects without using relative GPS coordinates since the

relative pose between an object and the robot is rarely a given in real-world scenarios. The fundamental limitations of this are numerous. First, it is assumed that the GPS position of the target is known. In practice, this typically means that the target must have a GPS receiver and be in communication with the robot, which is certainly not ideal for low-cost payloads, inspections, or already inaccessible targets. It also requires that GPS is available, which is not the case in indoor structures such as buildings or covered stadiums, and that the signal is reliable, which is not a given in the proximity of large vertical obstacles such as tall buildings or structures. Secondly, the distance between objects must be kept sufficiently large to prevent collisions resulting from pose estimate errors, which can be on the order of meters. Therefore, GPS alone is not sufficient for object retrieval, especially for commercial applications. Finally, GPS does not provide obstacle avoidance, as it can only determine an absolute position, not observations of the surroundings. Thus, there is a need for solutions empowering aerial robots to interact, manipulate, and closely track specific objects of interest.

In this dissertation, we take several strides toward the end goal of comprehensive aerial manipulation. First, we discuss preliminaries of aerial vehicles in Chapter 2, including dynamic models, control, and our notation. In Chapter 3, we present a quadrotor equipped with an actuated onboard manipulator, and we explain how to plan trajectories for the coupled system and how to compensate for the dynamics of the gripper to enable high-speed grasping. Next, in Chapter 4, we demonstrate perching on vertical surfaces using a downward-facing gripper, enabling a robot to save energy or maintain a precise position using an applicible real-world gripper that could also be used to transport payloads. Then, we take a leap toward applications in unstructured environments by leveraging vision-based control and planning to enable perching on cylindrical objects such as tree branches, telephone poles, and railings (Chapter 5). Finally, we extend the objects available for vison-based relative control to include spheres, and we consider a moving target in Chapter 6.

## 1.1 Vehicle Choice

There are many possible choices of aerial vehicles, each with their own set of pros and cons [29]. Lighter-than-air (LTA) vehicles, such as blimps, often have the longest flight time, but they must be very large to carry even a reasonable payload. Fixed wing vehicles are more agile than LTA vehicles, and they can stay airborne for a decent amount of time. However, they require forward flight and still have limited maneuverability. Rotorcraft are appealing because they can hover in place and can carry a larger payload for the same footprint as a fixed-wing vehicle. However, traditional designs of rotorcraft are mechanically complex and cumbersome to maintain.

A specific class of rotorcraft consisting of four co-planar rotors is particularly appealing because of their mechanical simplicity, agility, payload to footprint ratio, and their ability to hover [91]. These vehicles are called *quadrotors* and will be the platform considered in this work. Each rotor is directly attached to a motor, making the mechanical complexity minimal as there are no linkages, swashplates, or gears. A sample illustration of a quadrotor is provided in Figure 1.1. Further modeling of this vehicle will be discussed in Chapter 2.

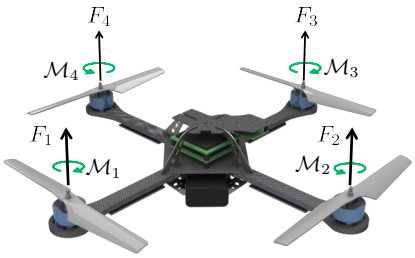



Figure 1.1: An illustration of a quadrotor. A quadrotor has four rotors, each of which generates a force, $F_i$, and a moment, $\mathcal{M}_i$. Adjacent rotors spin the opposite direction so that the net moment about the vertical axis can be controlled by varying the relative speed of the pairs of rotors.

## 1.2 Grasping and Perching

There are many challenges associated with physically interacting with objects. The shape, size, texture, and rigidity of an object are all properties that affect the success of a grasp, and there there are many different designs of grippers intended to address these variations. Dollar *et al.* demonstrated that approaches using underactuation can be quite effective [22–

24], especially when trying to cage an object. In these, flexible fingers can passively conform to a wide range of object shapes. Grippers that can apply pulling contact forces are appealing because they do not need to be large enough to cage the object. For example, directional adhesives can be used to adhere to smooth surfaces [46, 47], and they can be controllably released. These adhesives use the same underlying physics as the feet of a gecko, where many tiny fibers are placed in shear, creating a larger contact area, and adhering to a surface using van der Waals forces. Another possibility is the use of suction, which is appealing since it can accommodate a wider range of surface curvatures and textures. For example, the work in [62] uses an array of self-sealing suction cups to grasp a wide variety of objects. When a cup is not engaged, it seals shut so that only one vacuum pump and line is needed for the entire array.

### 1.2.1 Aerial Grasping

Achieving such grasping on aerial robotics introduces a significant constraint. That is, aerial vehicles have a limited payload, requiring any onboard manipulators and actuators to be sufficiently lightweight. Despite this restriction, a servo-driven claw was used on an aerial vehicle to transport plastic beams for construction of cubic structures [71, 72]. Similarly, Augugliaro *et al.* used a set of three servos in a gripper designed to transport building blocks to construct a tower [3], and Mellinger *et al.* utilized a gripper with hooks to pierce its targets, which enabled grasping of various objects [86]. A self-sealing suction cup gripper was optimized to be carried on an aerial platform [63], also enabling grasping of various objects. Stability during a grasp has also been considered [104, 105].

Despite variations in method and application, these grippers suffer from a common constraint; the vehicle must make an approach perpendicular to the plane of the target, and the approach velocity must be nearly zero when grasping. The relative position constraint was relaxed with a 2-DOF arm, but motions were still quasi-static [121]. Dynamic motions with a slung load were considered in [119] and later extended to enable dynamic object retrieval and deployment [122], but the gripper was a simple electromagnet and the relative position was difficult to control precisely, as it was a slung load.

Inspired by the dynamic grasping capabilities of birds of prey [124], we developed a gripper with underactuated digits that could be used for high-speed grasping [131]. In particular, the gripper incorporates the same strategy as birds of prey to increase the duration that the end effector has to grasp the object. That is, the gripper is located at the end of an actuated, swinging arm that swings opposite the direction of motion, reducing the relative velocity between the gripper and the target. The design was massive enough that the inertia of the gripper needed to be considered in the dynamics and also compensated in the control. Ultimately, the gripper enabled grasping objects while the robot was moving at 3 m/s. Details of this work will be provided in Chapter 3.

Manipulation using multiple aerial robots has also been explored [34, 90] with extensions considering planning and control [117]. Mutli-robot manipulation is appealing because it allows for larger payloads than that of a single-robot. However, there is the added challenge to ensure that robots are not competing against each other when carrying a rigid payload.

### 1.2.2 Perching

Perching using fixed wing vehicles was presented with a focus on control and planning [13] as well as with a focus on the design of the gripper [66, 76]. Sequential controllers were used to achieve perching on vertical surfaces using quadrotors, but the approach leveraged iterative learing to determine the relative start position, and the robot required Velcro to adhere to the perch location [87, 88]. We have also seen perching on smooth surfaces using methods that leverage tunable adhesives, however, a gripper on the side of a quadrotor is not ideal for the dual use of transporting payloads [56]. An MAV was able to perch on a door using suction, rotate the handle using a pneumatic actuator, and propel the door open using the onboard thrust [135]. Passively actuated mechanisms have been used to facilitate perching of rotorcraft, but possible perch locations are limited by the size of the gripper, which is limited by the payload capacity of the robot [5, 26, 27]. Additionally, these passive mechanisms are either not capable or are not ideal for grasping objects for transport.

In this work, we use a centered, downward-facing, tunable adhesive, enabling a robot to transport objects as well as perch on vertical surfaces [128, 132]. The proposed planning

method ensures that the robot does not saturate actuators or sensors, and it enables perching on smooth surfaces at various angles without requiring iterative learning methods. Details are explored more thoroughly in Chapter 4.

## 1.3 Vision-based Methods

So far, all of the perching and grasping methods presented have assumed that the state of the robot is known. This is certainly a crippling assumption, so we seek solutions that enable the state to be estimated entirely using onboard sensors. Vision-based techniques are particularly appealing because cameras provide rich data, are low-cost, and are lightweight. Such approaches typically incorporate one or more cameras for autonomous navigation and environment interaction, and tasks can be solved using either a single or multiple cooperating vehicles. The camera sensor data can be used to navigate in the environment or to enable the vehicle to interact with surrounding objects. In the first case, the vehicle is controlled with respect to a fixed reference frame, generally identified with the robot's starting position. In the second case, the control problem should be approached so that the vehicle is controlled relative to a specific object. For this reason, it is convenient to consider the fixed frame coincident with the object frame.

### 1.3.1 Vision-based Navigation

There are a number of Simultaneous Localization and Mapping (SLAM) approaches for MAVs. Successes have been achieved using monocular cameras and Inertial Measurement Units (IMUs) [111, 138], stereo camera configurations [112, 134], and RGB-D sensor systems [92, 110]. In [110] and [92], a Kinect and the vehicle's onboard sensors are used to perform state estimation using an Extended Kalman Filter (EKF), while in [138], a similar filter is used to combine monocular visual information with inertial sensor data to solve the scale factor problem. However, these algorithms typically only work without limitation on laptop or desktop computers. Further, RGB-D sensors generally have low quality cameras and suffer when exposed to direct sunlight. The minimal sensor suite for autonomous localization consists of two inexpensive, lightweight, and widely available sensors: a single

camera and an IMU [49, 55, 60, 81]. Theoretical observability analysis for aerial navigation is shown in [138] and [111], however, most of these platforms are heavy (over 1 kg in mass). Additionally, there is still a gap between the use of complex algorithms in the research field and its use for everyday applications. This problem was solved for the first time in [74], which presented the first fully autonomous quadrotor that was able to navigate using a Commercial Off-The-Shelf (COTS) smartphone, a Samsung Galaxy S5. The planning, control, estimation, and mapping tasks occurred in a single application embedded in the phone using only 50% of the total CPU. The work was later extended to multiple vehicles in [75].

### 1.3.2 Visual Servoing

The vision-based navigation techniques discussed in the previous subsection do not provide the ability to control the vehicle with respect to a specific object, and therefore, they are not directly applicable to scenarios such as tracking or interacting with a target. Thus, it is necessary to leverage different approaches. Specifically, we are concerned with the direct control of a robot relative to a target object, not with the control of absolute position and orientation. One approach to solve this problem is called visual servoing, which leverages visual feedback to control a robot relative to a target object.

There is a foundational body of literature covering control for monocular visual servoing applications, which discusses the differences between Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS) [6, 8, 50]. The key difference between these approaches is that with PBVS, the relative pose of the robot is estimated based on the image features, and the control law is expressed in the 3D Cartesian space. On the other hand, with IBVS, the control law is computed based on errors of features observed in the image [50]. Each has its benefits; for example, PBVS systems can use common filters in the MAV literature for the pose estimate while IBVS is considered to be more robust to camera calibration errors, making it appealing for low cost and low quality systems.

In the IBVS literature, most research analyzes and proves stability for first order, fully-actuated systems. Interaction matrices for various geometric shapes as well as simulation and experimental results are presented in [30]. Issues with singularities, local minima, ill-

conditioned Jacobians, and comparisons between nominal and estimated Jacobians were explored in [7]. Robustness to calibration errors was discussed and demonstrated in [123]. Gravity compensation of a robot arm was considered within a visual servoing framework [61]. Stereo image based visual servoing was introduced in [59], relaxing the requirement that the scale of the target was known, but losing the benefits of a simple monocular system. If a velocity estimate is known, perhaps from visual odometry, active vision approaches (see, for example, [116]) could also be used to estimate the scale of the target.

Applying such techniques to MAVs presents new difficulties. Specifically, there are two key challenges with applying visual servoing techniques to aerial vehicles; first, the systems are high-order systems, and secondly, the systems are typically underactuated. Higher order considerations were proposed in [11] and [12], which argue for feed-forward control inputs and present visual servoing as a "steering" problem. Second order systems were also explored in [32] and [140]. Control of first and second order, fully-actuated systems was demonstrated in [14] where navigation functions were used to guarantee occlusion-free trajectories. The work was later extended in [15] to consider servoing of a dynamic system relative to a sphere. Port-Hamiltonian methods were used to formulate a general IBVS approach [78], however an improvement in accuracy over traditional IBVS approaches was not demonstrated. There has also been some consideration of underactuated systems. For example, [42–44] and [51] used a backstepping approach for control of an MAV, while [127, 129] used a geometric-based control approach.

## 1.4 Roadmap

In the rest of this dissertation, we explore various aspects of grasping, perching, and visual servoing. Additionally, we will consider interaction with several geometric shapes, specifically planes, cylinders, and spheres. We first present conventions, notation, and preliminaries in Chapter 2.

Then, we explore the ability of an aerial robot to grasp objects while moving at high speeds (Chapter 3). In this work, the shape of the object is abstracted as it is simply

assumed that the object can be grasped. However, the dynamics of an actuated gripper are considered and compensated to enable sufficiently accurate tracking of the end effector.

In Chapter 4, we explore interaction with smooth, flat surfaces (*e.g.,* windows) regardless of their inclination. An appropriate gripper, its landing requirements, and a planning strategy will be explored. The planning strategy will focus on enabling aggressive maneuvers while satisfying the dynamic, sensor, and actuator constraints of the vehicle.

We incorporate vision in Chapter 5 and specifically consider cylindrical objects. A monocular, vision-based geometric model is developed to determine a relative pose. Then, we present a control law and prove stability, we propose a planning approach, and we demonstrate successful perching from various initial conditions while using only onboard sensing and computation.

Finally, we consider maneuvering relative to a sphere in Chapter 6, providing a formulation for relative pose estimation and planning. The object is no longer assumed to be stationary, and its motion is considered in the planning strategy. Additionally, the robot's dynamic, actuator, and sensor constraints, including the field of view, are considered in planning.

We conclude in Chapter 7, summarizing our results and proposing avenues for continued research.

# Chapter 2

# Quadrotor Preliminaries

This chapter will describe preliminaries related to quadrotors such as the aerodynamic assumptions, kinematics, dynamics, and control. We will also present conventions and notations which will be used throughout this dissertation.

## 2.1 Quadrotor Model

In this section, we will present our aerodynamic assumptions, the control inputs of our system, and the dynamics of the robot.

### 2.1.1 Aerodynamics

The vehicle considered in this work is a quadrotor, a multirotor aerial vehicle having four rotors as depicted in Figure 1.1. Each rotor can generate a thrust, $F_i = k_f \omega_i^2$, proportional to the square of $\omega_i$, the propeller speed [91]. Similarly, each rotor applies a moment $\mathcal{M}_i = k_m \omega_i^2$ to the robot that opposes the direction of the propeller.

### 2.1.2 Control Inputs

The speed of the rotors, $\omega_i$, can be mapped uniquely to a simpler set of equivalent control inputs (as displayed in Figure 2.1 and outlined in [85]) using the following invertible

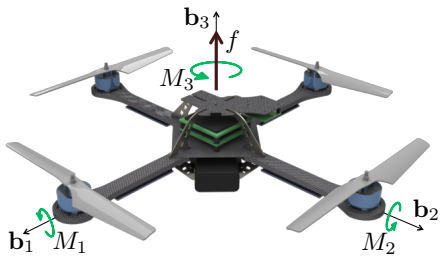



Figure 2.1: The control inputs of a quadrotor can be considered to be a net force, $f$, and moments, $M_i$, about each of the principal axes. This representation is related to Figure 1.1 through the relationship given in (2.1).

relationship

$$\begin{bmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ 0 & lk_f & 0 & -lk_f \\ -lk_f & 0 & lk_f & 0 \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{2.1}$$

where $k_f > 0$ and $k_m > 0$ are the thrust and moment coefficients of the rotors, respectively, and the distance between the motors and the center of mass is $l$. The net thrust is $f = \sum_{i=1}^{4} k_f \omega_i^2$ and the moment about the $i^{th}$ body frame axis is given by $M_i$. Using this relationship, the system can be considered to have the control inputs pictured in Figure 2.1.

### 2.1.3 Dynamics

From Newton's equations of motion, the translational dynamics of the vehicle are

$$m\ddot{\mathbf{x}} = fR\mathbf{e}_3 - mg\mathbf{e}_3 \tag{2.2}$$

where $m \in \mathbb{R}$ is the mass and $\mathbf{x} \in \mathbb{R}^3$ is the position of the robot in the world frame, $\mathcal{W}$. The rotation from the robot's frame ($\mathcal{R}$) to the world frame ($\mathcal{W}$) is given by $R \in \mathrm{SO}(3)$, $g$ is gravitational acceleration, and $\mathbf{e}_3$ is the third standard basis vector, $\mathbf{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. The angular dynamics of the system are given by

$$\mathcal{I}\dot{\boldsymbol{\Omega}} = \mathbf{M} - \boldsymbol{\Omega} \times \mathcal{I}\boldsymbol{\Omega} \tag{2.3}$$

where $\mathcal{I} \in \mathbb{R}^{3\times3}$ is the inertia tensor aligned with $\mathcal{R}$, $\mathbf{\Omega} \in \mathbb{R}^3$ is the angular velocity of the vehicle expressed in $\mathcal{R}$, and $\mathbf{M} = \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix}^T$ contains the control moments. Equations (2.2) and (2.3) are related through the orientation of the robot, namely,

$$\dot{R} = R\,\hat{\mathbf{\Omega}} \tag{2.4}$$

where $\hat{\cdot} : \mathbb{R}^3 \mapsto \mathfrak{so}(3)$ is the "hat" map and is defined such that if $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, $\mathbf{a} \times \mathbf{b} = \hat{\mathbf{a}}\mathbf{b}$. Thus, we can write the complete dynamics of the system as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \frac{f}{m} R\mathbf{e}_3 - g\mathbf{e}_3 \\ \dot{R} &= R\,\mathbf{\Omega} \\ \dot{\mathbf{\Omega}} &= \mathcal{I}^{-1}\left(\mathbf{M} - \mathbf{\Omega} \times \mathcal{I}\mathbf{\Omega}\right) \end{aligned} \tag{2.5}$$

with $\mathbf{v} \in \mathbb{R}^3$ representing the velocity and with $f$ and $\mathbf{M}$ as the control inputs to the system.

## 2.2 Control

There are many available approaches to control quadrotors. For example, a linearized controller was presented in [91]. More computationally complex approaches such as model predictive control (MPC) have also been considered [4]. In general, we will base our developments on the control law developed in [69], which guarantees exponential stability if the geodesic attitude error is less than 90° and exhibits almost global exponential attractiveness (the only exception is when the geodesic attitude error is 180°). Let the thrust be

$$f = (-k_x\mathbf{e}_x - k_v\mathbf{e}_v + mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_d) \cdot R\mathbf{e}_3 \equiv \mathbf{f}_{des} \cdot R\mathbf{e}_3 \tag{2.6}$$

where $k_x$ and $k_v$ are positive gains, $\mathbf{e}_x = \mathbf{x} - \mathbf{x}_d$ and $\mathbf{e}_v = \dot{\mathbf{x}} - \dot{\mathbf{x}}_d$ are position and velocity errors, respectively, and $\ddot{\mathbf{x}}_d$ is the nominal acceleration. The commanded moments are given

by

$$\mathbf{M} = -k_R \mathbf{e}_R - k_\Omega \mathbf{e}_\Omega + \boldsymbol{\Omega} \times \mathcal{I}\boldsymbol{\Omega} - \mathcal{I}\left(\hat{\boldsymbol{\Omega}} R^T R_c \boldsymbol{\Omega}_c - R^T R_c \dot{\boldsymbol{\Omega}}_c\right) \tag{2.7}$$

where $k_R$ and $k_\Omega$ are positive gains and

$$\mathbf{e}_R = \frac{1}{2}\left(R_c^T R - R^T R_c\right)^\vee \quad \text{and} \quad \mathbf{e}_\Omega = \boldsymbol{\Omega} - R^T R_c \boldsymbol{\Omega}_c$$

are the angular position and rate errors with $\cdot^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$ being the opposite of the "hat" map. In this case, $\boldsymbol{\Omega}_c \in \mathbb{R}^3$ is the "commanded" or nominal angular rate and $R_c$ is the "commanded" attitude. For more details, we refer the reader to [69] or the implementation in [85].

## 2.3 Conventions and Nomenclature

### 2.3.1 Frame Conventions

When ambiguous, we will use the following conventions. A superscript may be used to denote the frame the quantity is expressed in (see Table 2.1). A single subscript is used to

| **Frame** | **Superscript** |
|---|---|
| Camera | $c$ |
| Robot | $\mathcal{R}$ |
| World | $\mathcal{W}$ |

Table 2.1: Frame conventions

denote the position of the origin of the subscripted frame relative to superscripted frame. A subscript with two characters is used to denote a vector from the first frame to the second frame expressed in the coordinates of the superscripted frame. For examples, see Table 2.2. Rotation matrices may also be used with subscripts and superscripts; for example,

| **Vector** | **Convention** |
|---|---|
| World to Robot, expressed in the World frame | $\mathbf{x}_{\mathcal{R}}^{\mathcal{W}}$ |
| Camera to Object, expressed in the Camera frame | $\mathbf{x}_{\mathcal{O}}^{\mathcal{C}}$ |
| Robot to Object, expressed in the World frame | $\mathbf{x}_{\mathcal{RO}}^{\mathcal{W}}$ |

Table 2.2: Vector convention examples.

$R_{\mathcal{R}}^{\mathcal{W}} \in \mathrm{SO}(3)$ would rotate vectors from the robot frame to the world frame. If no script is provided, the default convention will be $R \equiv R_{\mathcal{R}}^{\mathcal{W}}$.

### 2.3.2 Symbols and Nomenclature

Unless specified otherwise, we will use the symbols and nomenclature outlined in Table 2.3.

Table 2.3: General Nomenclature

| | |
|---|---|
| $\mathbf{e}_i \in \mathbb{R}^3$ | The $i^{th}$ standard basis vector, *e.g.,* $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ |
| $\mathbf{I}_i \in \mathbb{R}^{i \times i}$ | The identity matrix |
| $\mathbf{x}, \mathbf{r} \in \mathbb{R}^3$ | Position |
| $\mathbf{b}_i \in \mathbb{R}^3$ | The $i^{th}$ body-fixed basis vector |
| $R \in \mathrm{SO}(3)$ | Orientation, can also be expressed as $R = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}$ |
| $\boldsymbol{\Omega} \in \mathbb{R}^3$ | Angular velocity of the quadrotor, expressed in the body-fixed frame |
| $\Omega_i \in \mathbb{R}$ | Angular velocity of the quadrotor about $\mathbf{b}_i$, equivalent to $\mathbf{e}_i^T \boldsymbol{\Omega}$ |
| $m \in \mathbb{R}$ | Mass |
| $\mathcal{I} \in \mathbb{R}$ or $\mathbb{R}^{3\times3}$ | Inertia scalar or matrix |
| $\mathcal{J} \in \mathbb{R}$ | Cost Function |
| $f \in \mathbb{R}$ | Net thrust along the $\mathbf{b}_3$ axis |
| $\mathbf{M} \in \mathbb{R}^3$ | Moments about the body frame axes |
| $M_i \in \mathbb{R}$ | Moment about the $\mathbf{b}_i$ axis, equivalent to $\mathbf{e}_i^T \mathbf{M}$ |
| $J \in \mathbb{R}^{n \times n}$ | A Jacobian matrix |

# Chapter 3

# High-Speed Grasping

Predatory birds have the ability to swiftly swoop down from great heights and grasp prey, with high rates of success, from the ground, water, and air while flying at high speeds [137]. Although recent years have seen improvement in the capabilities of Micro Aerial Vehicles (MAVs) [67, 88], such dynamic aerial manipulation, common in nature, has not been achieved using MAVs. The present state of the art in aerial manipulation ranges from using grippers for construction [71] to cable-suspended loads for dynamic transportation [119]. Acquiring, transporting and deploying payloads while maintaining a significant velocity are important since they would save time and energy by minimizing required flight time. For example, high-speed grasping could be used in rescue operations where speed and time are critical, and in operations requiring an MAV to quickly swoop down and pick up an object of interest. Additionally, the dynamic grasping functionality could be extended to achieve perching, which could be used to quickly escape high winds, achieve immediate silence in stealth operations, and improve mission duration by reducing hover time.

With the recent increase in such applications, there has also been a rising need for articulated appendages capable of interacting with the environment. Dollar *et al.* developed fingers that passively conformed to a wide range of object shapes [22], Doyle *et al.* developed a passively actuated gripper to facilitate perching [26], Lindsey *et al.* designed a servo-driven claw to transport plastic construction beams [71], and Mellinger *et al.* utilized a

Figure 3.1: Video frames of a red kite grasping food from the ground [124].

gripper with fish hooks to pierce its targets [86]. Though these grippers vary in method and application, they suffer from a common limitation; in order to be effective, the vehicle must make an approach perpendicular to the plane of the target and the approach velocity must be close to zero when grasping. An ingressive gripper enabled perching with more aggressive trajectories by triggering a spring-loaded claw that would engage upon contact, but still needed to contact the target surface with a normal velocity [89].

Video analysis of birds of prey, such as the red kite shown in Figure 3.1, reveal that raptors sweep their legs and claws backwards during grasping, thereby reducing the relative velocity between the claws of the predator and the prey [124]. This allows the raptor, without slowing down, to have a near-zero relative velocity of the claw while grasping its prey. This strategy provides a high rate of success and inspires the development of our method to enable high-speed aerial grasping and manipulation for MAVs.

The rest of this chapter[1] is organized as follows. We first present a novel gripper design capable of reducing the velocity of the claw relative to the target. Next, we present the dynamic model of the quadrotor MAV equipped with the gripper, and we demonstrate that the system is differentially flat. Following this, we present a trajectory generation method based on the flat outputs and provide an overview of the controllers. The next section presents experimental results of high-speed grasping at 2 m/s and 3 m/s. In addition, we provide a nondimensional comparison between an MAV claw trajectory and a sample avian claw trajectory. Finally, we present concluding remarks with thoughts for future work.

[1]Much of the content in this chapter originally appeared in [131] and [127].

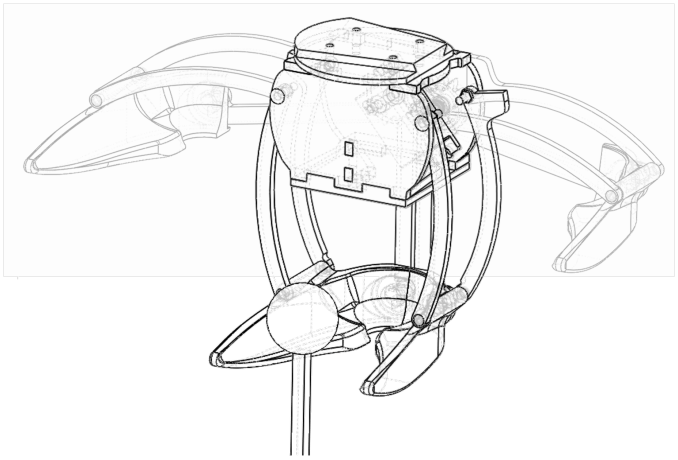


Figure 3.2: An early prototype gripper that could be used with spherical targets. The gripper is shown in its open (shaded) and closed (solid) configurations.

## 3.1 Design of an Articulated Gripper

A well designed gripper is critical for successful high-speed aerial manipulation. One primary goal of a gripper is to enable MAVs to acquire payloads while moving at significant relative velocities. A secondary goal is to enable the ability to perch by compliantly grasping objects of arbitrary shape or features such as tree branches, light poles, or rooftops.

An initial gripper design resembled a two-pronged fork that interfaces with a sphere fixed above the payload, as seen in Figure 3.2. The fork was 3D printed and is designed to guide the ball into a spherical recess where it remains secure. A quadrotor equipped with this claw was able to acquire payloads at relative speeds up to 1 m/s. To release the payload, the fork could be separated by a mini servo motor. However, this claw requires specialized fixtures on the payload and is incapable of grasping objects at higher speeds due to large relative velocities between the claw and the target.

Birds of prey possess claws that meet both of our stated objectives for a gripper. Of particular interest is the raptor's ability to maintain stable, high-velocity flight when capturing prey. Analysis of video footage (see sequence in Figure 3.1 ) reveals that raptors sweep their legs backward during grasping, reducing the relative velocity between the claws and the target.

Mechanisms with underactuated digits can allow for flexibility despite using a single actuator [22, 23]. Similarly, our design (see Figure 3.3) is underactuated, and as a result, three fingers can conform to an object while collectively being driven by a single servo motor. The digits are constructed from laser-cut ABS and covered with Dycem, a high-friction rubber that is used to improve grip. Although underactuated digits facilitates grasping of arbitrary object shapes, the fingers alone cannot close fast enough to capture payloads while the quadrotor is in motion.

To reduce the relative speed between the gripper and the target, we draw inspiration from the way a raptor sweeps its legs backwards while grasping, as can be seen in Figure 3.1. In our design, this translates to mounting the gripper on a rotating arm. The arm, composed of laser-cut ABS, pivots directly below the quadrotor's center of mass and is actuated by a servo motor. When the arm rotates, the end effector's velocity in the body frame reduces the relative velocity between itself and the payload during acquisition. See Figure 3.3 for a time-lapse visualization of the motion.

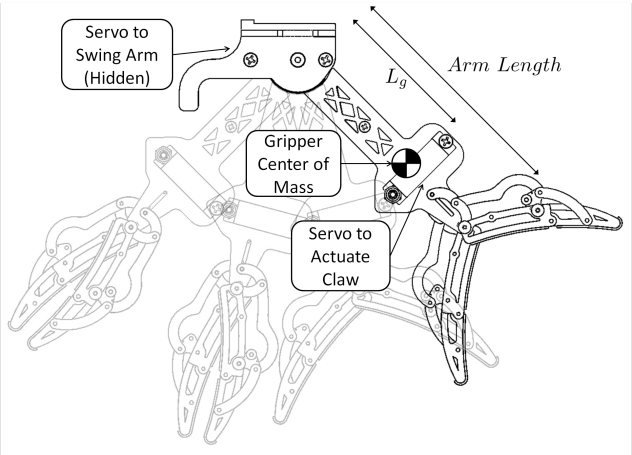



Figure 3.3: The gripper arm in motion as the claw is grasping. The shaded projections demonstrate the motion as the arm swings about the axis pointed into the page ($\mathbf{b}_2$).

As we will show later, this gripper design satisfies our goals of compliantly grasping arbitrarily shaped objects and enabling high-speed grasping. The gripper could be further improved by leveraging shape deposition manufacturing (SDM) for fabricating lightweight fingers [25], which would decrease inertia and permit acquisition at even larger velocities.

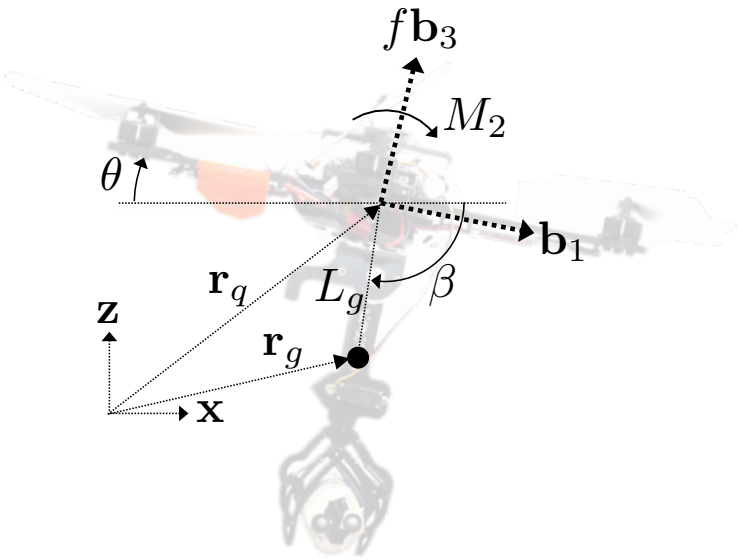



Figure 3.4: The quadrotor has control inputs $f$ in the $\mathbf{b}_3$ direction and $M_2$ as a moment about the axis into the page ($\mathbf{b}_2$). The gripper forms the angle $\beta$ with the horizontal and its center of mass is located a distance $L_g$ away from the quadrotor's center of mass.

## 3.2 Dynamic Model and Differential Flatness

The quadrotor model has been outlined in Section 2.1 with the dynamics presented in Section 2.1.3. In this paper, we adopt a planar version of this model for two reasons. First, in most examples of avian grasping and perching, the significant movements are limited to the sagittal plane of the bird. Indeed, most examples of claws and feet seen in nature have an axis of symmetry. Secondly, it is difficult to achieve high speed grasping without specifying a plane of approach. Note that most previous work requires the approach to be restricted to a single dimension (vertical), instead of two. Thus, we develop a simplified dynamic model, in which we only consider the motion in the $x-z$ plane with two control inputs, $f$ and $M_2$. See Figure 3.4 for a visualization.

The angle of the gripper relative to the horizontal ($x$-axis) is defined as $\beta$, as displayed in Figure 3.4, and the attitude of the vehicle is defined by $\theta$ such that the angle between the quadrotor and the gripper is $\gamma = \beta - \theta$. Further, the masses of the quadrotor and gripper are defined as $m_q$ and $m_g$, respectively, while the moments of inertia about the center of mass of the planar quadrotor and gripper are defined as $\mathcal{I}_q$ and $\mathcal{I}_g$, respectively. Because

of the planar assumption, $\mathcal{I}_q, \mathcal{I}_g \in \mathbb{R}$. The axis of rotation for the gripper is assumed to be at the quadrotor's center of mass so that the fixed distance $L_g$ denotes the length from the quadrotor's center of mass to the gripper's center of mass. We express the position vector of the quadrotor and gripper as $\mathbf{r}_q = \begin{bmatrix} x_q & 0 & z_q \end{bmatrix}^T$ and $\mathbf{r}_g = \begin{bmatrix} x_g & 0 & z_g \end{bmatrix}^T$, respectively. See Table 3.1 for a summary of the nomenclature.

Table 3.1: Nomenclature

| | |
|---|---|
| $\blacksquare_q$ | Subscript "q" denotes the Quadrotor |
| $\blacksquare_g$ | Subscript "g" denotes the Gripper |
| $\blacksquare_s$ | Subscript "s" denotes the coupled system |
| $\blacksquare_c$ | Subscript "c" denotes a commanded signal |
| $\blacksquare^d$ | Superscript "d" denotes desired or feed-forward |
| $\mathbf{q} \in \mathbb{R}^4$ | Vector of generalized coordinates |
| $\mathbf{y} \in \mathbb{R}^3$ | Flat Outputs |
| $\theta \in \mathbb{R}$ | Pitch of the Quadrotor |
| $\beta \in \mathbb{R}$ | Angle of the Gripper relative to the horizonatal |
| $\mathcal{I}_\blacksquare \in \mathbb{R}$ | Angular inertia of a body |
| $\tau \in \mathbb{R}$ | Arm actuator torque along the $\mathbf{b}_2$ axis |
| $L_g \in \mathbb{R}$ | Length between the quadrotor and gripper's centers of mass |
| $x^\star \in \mathbb{R}$ | Non-dimensionalized $x$ position of the claw |
| $z^\star \in \mathbb{R}$ | Non-dimensionalized $z$ position of the claw |
| $t^\star \in \mathbb{R}$ | Non-dimensionalized time |

The position of the gripper is entirely determined from the position of the quadrotor and the angle of the gripper through

$$\mathbf{r}_g = \mathbf{r}_q + L_g \begin{bmatrix} \cos(\beta) \\ 0 \\ -\sin(\beta) \end{bmatrix}. \tag{3.1}$$

Furthermore, higher-order time-derivatives of the gripper position can be expressed as functions of the position of the quadrotor, the angle of the gripper, and their derivatives.

### 3.2.1 Dynamics

The dynamics of the coupled system are determined using Lagrangian mechanics where the potential energy is

$$V = m_q g z_q + m_g g z_g, \tag{3.2}$$

and the kinetic energy is

$$T = \frac{1}{2}\left(m_g \left\|\dot{\mathbf{r}}_g\right\|_2^2 + m_q \left\|\dot{\mathbf{r}}_q\right\|_2^2 + \mathcal{I}_g \omega_g^2 + \mathcal{I}_q \omega_q^2\right). \tag{3.3}$$

Then, $\mathbf{q} = \begin{bmatrix} x_q & z_q & \theta & \beta \end{bmatrix}^T$ is the vector of generalized coordinates so that the corresponding vector of generalized forces and moments is

$$\mathbf{F} = \begin{bmatrix} f\sin(\theta) \\ f\cos(\theta) \\ M_2 - \tau \\ \tau \end{bmatrix}, \tag{3.4}$$

where $\tau$ is the actuator torque on the gripper arm. The dynamics are determined using the Euler-Lagrange equations so that

$$\ddot{\mathbf{q}} = D^{-1}\left(\mathbf{F} - C\dot{\mathbf{q}} - G\right) \tag{3.5}$$

where the matrices $D$, $C$, and $G$ are

$$D = \begin{bmatrix} m_g + m_q & 0 & 0 & -L_g m_g \sin(\beta) \\ 0 & m_g + m_q & 0 & -L_g m_g \cos(\beta) \\ 0 & 0 & \mathcal{I}_q & 0 \\ -L_g m_g \sin(\beta) & -L_g m_g \cos(\beta) & 0 & \mathcal{I}_g + L_g^2 m_g \end{bmatrix}, \tag{3.6}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & -L_g m_g \cos(\beta)\dot{\beta} \\ 0 & 0 & 0 & L_g m_g \sin(\beta)\dot{\beta} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.7}$$

$$G = \begin{bmatrix} 0 \\ g(m_g + m_q) \\ 0 \\ -gL_g m_g \cos(\beta) \end{bmatrix}. \tag{3.8}$$

### 3.2.2 Path Planning Using Differential Flatness

For this application, the controller presented in Section 2.2 alone will not be sufficient. Performance can be further improved by varying desired setpoints in time and by also considering the torque required to actuate the gripper. For a dynamic system, it is expected that a smooth transition between states (*e.g.,* not a step input) is required, but the exact requirements may not be obvious. Adding to the complexity is the fact that the quadrotor is a trivially underactuated system, as it has only four actuators and six degrees of freedom. Further, we conclude that the system has a relative degree of 4 by observing that the thrust can only directly affect the linear acceleration in the $\mathbf{b}_3$ (also $R\mathbf{e}_3$) direction and that the moments can only directly affect the angular acceleration (which impacts $\mathbf{b}_3$). Because of this, planning dynamically feasible trajectories is a significant challenge, which is only exacerbated with the consideration of adding an actuated appendage.

The authors in [36] and [98] demonstrated a method for planning dynamically feasible trajectories for certain types of underactuated systems by leveraging a change of coordinates to express the system as a *differentially flat* system in a space of *flat outputs*. Specifically, a system is said to be *differentially flat* if there exists a change of coordinates which allows the state, $(\mathbf{q}, \dot{\mathbf{q}})$, and control inputs, $\mathbf{u}$, to be written as functions of the flat outputs and their derivatives $(y_i, \dot{y}_i, \ddot{y}_i, ...)$. An additional requirement is that the flat outputs are functions of the state and the control inputs. If the change of coordinates is a diffeomorphism, trajectories

can be planned using the flat outputs and their derivatives in the flat space since there is a unique mapping to the full state space of the dynamic system. In the MAV literature, [16] and [85] demonstrated that a set of *flat outputs* exists for a quadrotor, enabling application of the method in [36] and [98]. Thus, we can plan in the simplified space of *flat outputs* and their derivatives, in which dynamic feasibility can be guaranteed by satisfying certain smoothness constraints. Trajectories in the flat space can then be mapped uniquely to feasible trajectories in the configuration space, making the challenge of planning a feasible trajectory for a quadrotor a tractable problem.

**Proposition 1.** *The coupled system comprising of the quadrotor and the actuated gripper, whose dynamics is given by* (3.5)*, is differentially flat with a set of flat outputs given by*

$$\mathbf{y} = \begin{bmatrix} x_q & z_q & \beta \end{bmatrix}^T. \tag{3.9}$$

*Proof.* Let us first define $m_s = m_q + m_g$ so that the center of mass of the coupled system is

$$\mathbf{r}_s = \frac{m_q \mathbf{r}_q + m_g \mathbf{r}_g}{m_s}. \tag{3.10}$$

We recall from (3.1) that $\mathbf{r}_q = \mathbf{r}_q(x_q, z_q)$ and $\mathbf{r}_g = \mathbf{r}_g(x_q, z_q, \beta)$. Thus, $\mathbf{r}_s = \mathbf{r}_s(x_q, z_q, \beta)$ and $\ddot{\mathbf{r}}_s$ is fully defined using the proposed flat outputs and their derivatives. In addition, this motivates the choice of $\beta$ instead of $\gamma$ defining the angle of the gripper. If the gripper angle was defined relative to the quadrotor attitude, then $\theta$ would appear in $\mathbf{r}_s$ and we would ultimately see that $\gamma$ is not a flat output. Further, it makes sense to plan using $\beta$ because at pickup, we have strict position constraints on the end effector which should be invariant to the attitude of the quadrotor. Defining $\mathbf{e}_3$ as the third standard basis vector, the Newton-Euler equations of motion provide

$$m_s \ddot{\mathbf{r}}_s = f\mathbf{b}_3 - m_s g \mathbf{e}_3 \tag{3.11}$$

revealing that

$$f = m_s \left\| \ddot{\mathbf{r}}_s + g\mathbf{e}_3 \right\| \tag{3.12}$$

and

$$\mathbf{b}_3 = \frac{\ddot{\mathbf{r}}_s + g\mathbf{e}_3}{\left\| \ddot{\mathbf{r}}_s + g\mathbf{e}_3 \right\|} \tag{3.13}$$

from which $\theta$ can be determined. In addition, (3.13) requires that $\left\| \ddot{\mathbf{r}}_s + g\mathbf{e}_3 \right\| > 0$ or that $f > 0$. Since the system is restricted to the planar case, $\mathbf{b}_2 = \mathbf{e}_2$ and $\mathbf{b}_1 = \mathbf{b}_2 \times \mathbf{b}_3$. Next, we differentiate (3.11) to obtain

$$m_s \dddot{\mathbf{r}}_s = \dot{f}\mathbf{b}_3 + \boldsymbol{\Omega} \times f\mathbf{b}_3 \tag{3.14}$$

where we know that $\boldsymbol{\Omega} = \dot{\theta}\mathbf{b}_2$. The projection onto $\mathbf{b}_3$ reveals

$$\dot{f} = \mathbf{b}_3 \cdot m_s \dddot{\mathbf{r}}_s \tag{3.15}$$

and, using this relationship,

$$\boldsymbol{\Omega} \times \mathbf{b}_3 = \frac{m_s}{f} \left( \dddot{\mathbf{r}}_s - (\mathbf{b}_3 \cdot \dddot{\mathbf{r}}_3)\mathbf{b}_3 \right). \tag{3.16}$$

We notice that this is purely in the $\mathbf{b}_1 - \mathbf{b}_2$ plane and, more specifically, that $\boldsymbol{\Omega} \times \mathbf{b}_3 = \dot{\theta}\mathbf{b}_1$. Thus,

$$\dot{\theta} = \frac{m_s}{f} \left( \mathbf{b}_1 \cdot \dddot{\mathbf{r}}_s \right). \tag{3.17}$$

Taking the second derivative of (3.11), we obtain

$$m_s \mathbf{r}_s^{(4)} = \dot{f}\dot{\theta}\mathbf{b}_2 \times \mathbf{b}_3 + \ddot{f}\mathbf{b}_3 + \dot{\theta}\mathbf{b}_2 \times \dot{f}\mathbf{b}_3 + \dot{\theta}\mathbf{b}_2 \times \left( \dot{\theta}\mathbf{b}_2 \times f\mathbf{b}_3 \right) + \ddot{\theta}\mathbf{b}_2 \times f\mathbf{b}_3. \tag{3.18}$$

Collecting terms and simplifying cross products,

$$m_s \mathbf{r}_s^{(4)} = \left( 2\dot{f}\dot{\theta} + \ddot{\theta}f \right) \mathbf{b}_1 + \left( \ddot{f} - \dot{\theta}^2 f \right) \mathbf{b}_3. \tag{3.19}$$

The projections onto $\mathbf{b}_1$ and $\mathbf{b}_3$ reveal

$$\ddot{\theta} = \frac{1}{f}\left(m_s \mathbf{b}_1 \cdot \mathbf{r}_s^{(4)} - 2\dot{f}\dot{\theta}\right) \tag{3.20}$$

and

$$\ddot{f} = \mathbf{b}_3 \cdot \left(m_s \mathbf{r}_s^{(4)}\right) + \dot{\theta}^2 f. \tag{3.21}$$

Next, we let $F_x$ and $F_z$ be reaction forces at the attachment point of the gripper so that the translational and angular equations of motion of the gripper are

$$m_g \ddot{x}_g = F_x \tag{3.22}$$

$$m_g \ddot{z}_g = F_z - m_g g \tag{3.23}$$

$$\mathcal{I}_g \ddot{\beta} = \tau + F_x L_g \sin(\beta) + F_z L_g \cos(\beta). \tag{3.24}$$

Solving for the gripper arm actuator torque, $\tau$,

$$\tau = \mathcal{I}_g \ddot{\beta} - L_g m_g \left(\ddot{x}_g \sin(\beta) + (\ddot{z}_g + g)\cos(\beta)\right). \tag{3.25}$$

Finally, we know that

$$M_2 = \ddot{\theta}\mathcal{I}_q + \tau. \tag{3.26}$$

Now we have demonstrated that state and inputs of the coupled system are related through a diffeomorphic map to the flat outputs and their derivatives. Therefore, the system is differentially flat. □

From the previous analysis, we can conclude that any sufficiently smooth trajectory in the space of flat outputs is automatically guaranteed to satisfy the equations of motion. Specifically, we see that the control inputs to the system are functions of the snap $\left(\mathbf{y}^{(4)}\right)$ of the trajectories. Thus, we require that trajectories planned in the flat space be smooth in position $(\mathbf{y})$, velocity $(\dot{\mathbf{y}})$, acceleration $(\ddot{\mathbf{y}})$, and jerk $(\dddot{\mathbf{y}})$, which is the same as requiring each flat output to be of class $\mathcal{C}^4$.

## 3.3 Trajectory Generation and Control Design

From the previous section, further examination of the control inputs reveals that the snap of the position of the quadrotor appears in the $M_2$ term through $\ddot{\theta}$. In addition, $\beta^{(4)}$ appears in $M_2$ through the $\mathbf{r}_s^{(4)}$ term in $\ddot{\theta}$. Then, to minimize the input vector in the flat space, we minimize the following cost functional constructed from the snap of the trajectory.

$$\mathcal{J}_i = \int_{t_0}^{t_f} \left\| y_i^{(4)}(t) \right\|^2 dt \quad \text{for} \quad i = 1, 2, 3 \tag{3.27}$$

Accordingly, we consider minimum-snap trajectories, which can be formulated as a Quadratic Program (QP) and solved numerically, as in [85]. However, we have now added a dimension for $\beta$.

Although we have a method to generate trajectories for the quadrotor, we do not have a definitive way to determine the constraints for the trajectory. For this, we take inspiration from nature and analyze video footage of raptors grasping prey. The extracted trajectory will be compared later in a following section.

The trajectories used for experimentation are constrained by position at the start and finish where the higher derivatives are zero at the desired start and goal locations. The position at pickup is constrained such that the gripper is oriented vertically when grasping the target, but the velocity, acceleration, and jerk of the quadrotor are free and required to be continuous. A fully-defined trajectory was planned for the $x$ and $z$ positions of the quadrotor. In order to ensure a successful grasp, the gripper must be pointed directly at the target during approach. To achieve this, position constraints were placed on the $\beta$ trajectory prior and up to the pickup time so that the gripper would be pointed directly at the target. See Figure 3.5 and Figure 3.6 for a desired and experimental trajectory of the position of the quadrotor and the gripper angle, respectively.

Next, we briefly present the controller that drives the quadrotor and gripper system along the desired trajectory. The quadrotor controller has an outer position control loop

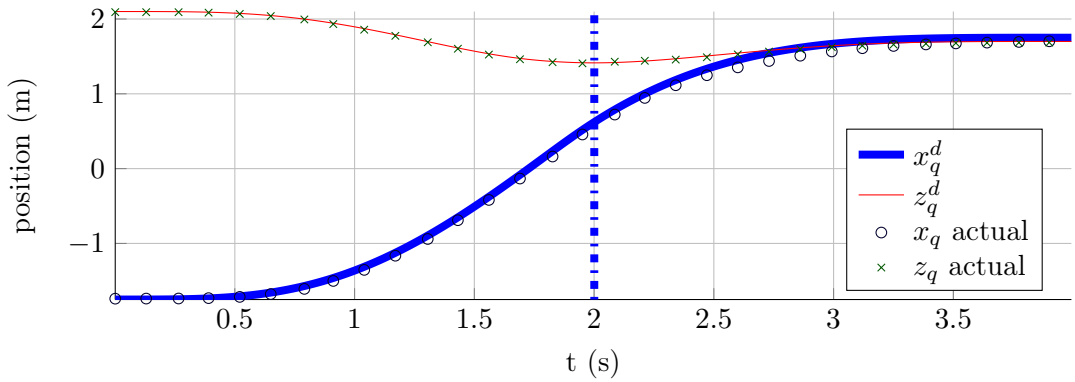



Figure 3.5: Desired quadrotor position trajectories overlayed with experimental results. The planned pickup time is $t = 2s$.

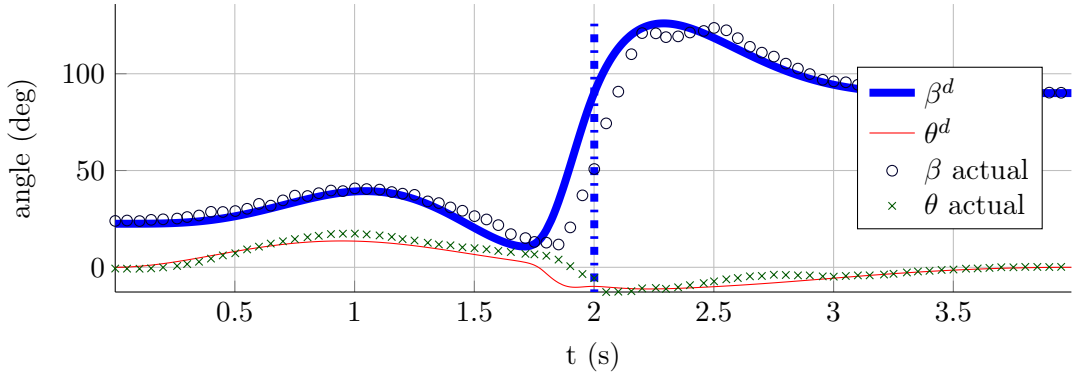



Figure 3.6: Desired $\beta$ and $\theta$ trajectories overlayed with experimenatl results. The planned pickup time is $t = 2s$.

running at 100 Hz which generates desired attitudes and feedforward control inputs. The commanded thrust is

$$f_c = k_{pz}\left(z_q^d - z_q\right) + k_{dz}\left(\dot{z}_q^d - \dot{z}_q\right) + f^d \tag{3.28}$$

where $k_{pz}$ and $k_{dz}$ are proportional and derivative gains, respectively. The desired values of various variables, denoted with a superscript "d", are computed using the flatness property. A 1 kHz inner-loop attitude controller onboard the quadrotor is used to achieve the desired attitude. The commanded moment is

$$M_{2c} = k_{p\theta}\left(\theta_c - \theta\right) + k_{d\theta}\left(\dot{\theta}^d - \dot{\theta}\right) + M_2^d \tag{3.29}$$

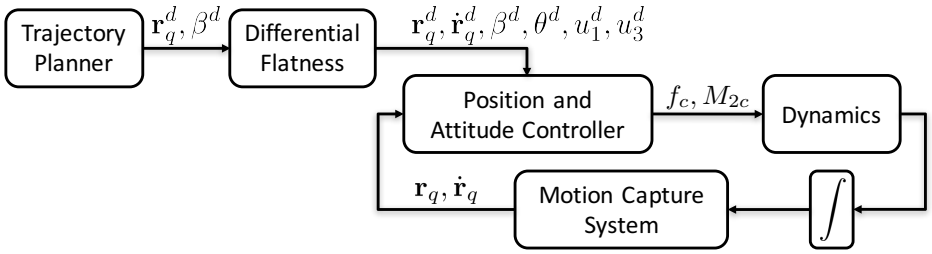



Figure 3.7: A block diagram of the controller used for experiments. A superscript "d" denotes a desired or nominal value (computed using the flatness property).

where $\dot{\theta}^d$ is the nominal angular rate, $M_2^d$ is the feed-forward moment, $k_{p\theta}$ is a proportional gain, and $k_{d\theta}$ is a derivative gain. A commanded attitude, $\theta_c$, is determined by the outer loop by

$$\theta_c = \sin^{-1}\left(k_{px}\left(x_q^d - x_q\right) + k_{dx}\left(\dot{x}_q^d - \dot{x}_q\right)\right) + \theta^d \tag{3.30}$$

where $k_{px}$ is a proportional gain and $k_{dx}$ is a derivative gain. The control design is similar to the quadrotor hover controller in [88], but with added feedforward control inputs to compensate for the motion of the gripper. The feedforward control inputs are supplied to the control loops as displayed in Figure 3.7.

## 3.4 Results

We demonstrate experimental results on a 500 g Asctec Hummingbird quadrotor [1] equipped with a 158 g gripper. The experiments leverage a motion capture system to accurately determine the state of the quadrotor [2] and to track the 27 g cylindrical target.

The controller combines feedforward control inputs and a feedback controller for the quadrotor. To compensate for some delays, the gripper claw was commanded to close slightly before the pickup time. With this setup, the quadrotor grasped the target while moving at 2 m/s with a success rate of 100% out of 5 attempts. Position errors for those trajectories are presented in Figure 3.8. The quadrotor was able to successfully grasp the target at speeds up to 3 m/s, or 9 body lengths / second (Figure 3.9).

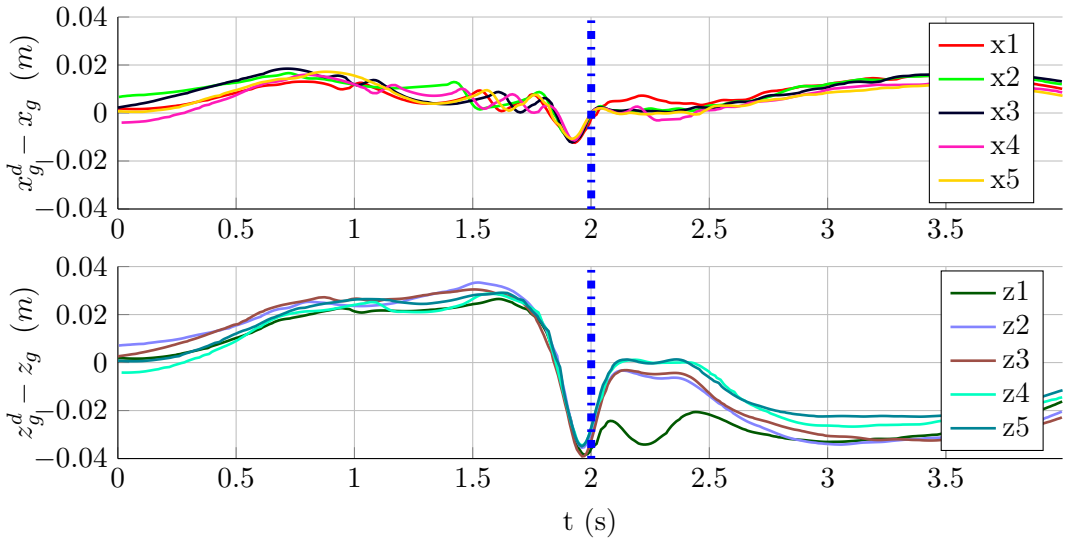



Figure 3.8: The $x$ and $z$ components of the normal error between the experimental and desired gripper positions for 5 consecutive trials. The actual pickup time is represented by a vertical dashed line.

### 3.4.1 Avian Comparison

In assessing the success of our results, it is appropriate to use the raptor's performance as a standard of comparison. We desire the same result that the raptor achieves, and therefore expect to match the bird closely.

We use video footage of an eagle grasping a fish, however, it slowed by an unknown factor resulting in an unknown time scale. The length scale is also impossible to extract accurately. However, it is still meaningful to compare the two nondimensionalized sets of trajectories. We nondimensionalize the trajectories using the following relationships

$$x^\star = \frac{x}{L}, \quad z^\star = \frac{z}{L}, \quad t^\star = \frac{tv_p}{L} \tag{3.31}$$

where $v_p$ is the body velocity at pickup and $L$ is the length from the axis of rotation to the gripping surface of the claw. The units are detailed in Table 3.2. Results using this approach are presented in Figure 3.10 with comparisons in Figure 3.11 and Figure 3.12. It can be seen that the $x$-position of the gripper followed closely to that of the eagle's claws. The significant deviation in $z$-position following pick-up can be attributed to the limited

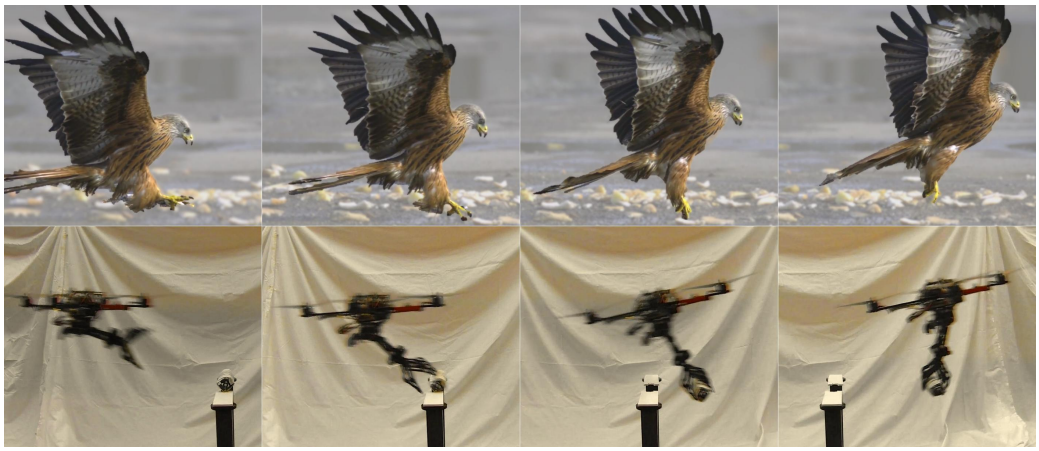


Figure 3.9: A still image comparison between a raptor (red kite, extracted from [124]) and the quadrotor for a trajectory with the quadrotor moving at 3 m/s (9 body lengths / second) at pickup. See [130] for a video of the grasping.

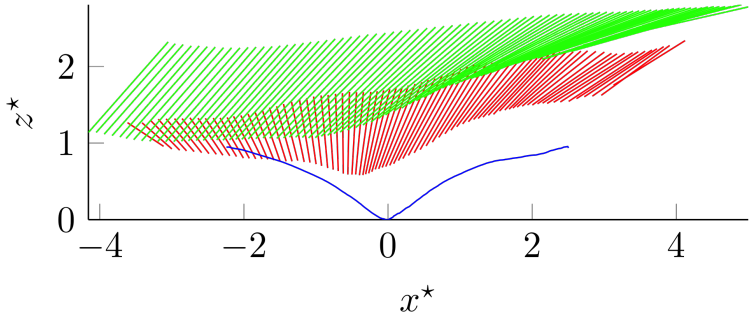
Table 3.2: Units of Non-Dimensionalization Factors

| | Eagle trajectory from video footage | Robot trajectory estimated using VICON |
|---|---|---|
| $L_g$ | pixels | meters |
| $v_p$ | pixels / frame | meters / second |

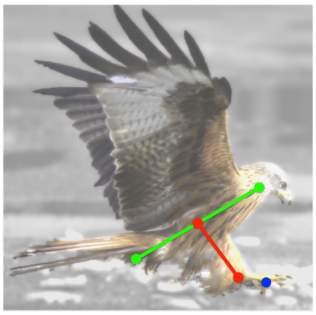
range of motion of the arm of the quadrotor compared to that of the eagle. Furthermore, the nondimensionalized length of the quadrotor's arm is slightly less than that of the fully extended eagle's leg. If the body length was used as the characteristic length, the gripper arm has a nondimensionalized length of 0.34 compared to the eagle's leg at 0.37.

## 3.5 Conclusion

In this chapter, we explored the challenges of high-speed aerial grasping using a quadrotor MAV equipped with a gripper. A novel appendage design, inspired by the articulation of a raptor's legs and claws, was shown to enable a high rate of success while grasping objects at high velocities. The dynamic model of the quadrotor and gripper system was shown to be differentially flat, and dynamically feasible trajectories were generated for the underactuated system. Experimental results were presented for quadrotor velocities of 2 m/s and 3 m/s (6 - 9 body lengths / second). Finally, a comparison of a nondimensionalized quadrotor trajectory with a sample avian trajectory was presented.




(a) Lines are sampled from the video with an even spacing in time. The motion is from left to right, and a key is provided in the image to the right.



(b) A key illustrating the linkages extracted from our video analysis.

Figure 3.10: A non-dimensionalized trajectory of an eagle grasping a fish.

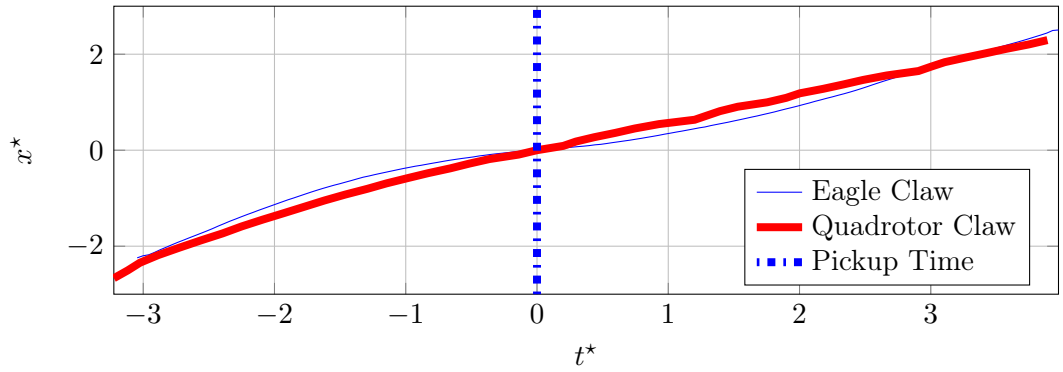



Figure 3.11: A comparison of the nondimensionalized $x$ positions of the quadrotor claw and the raptor claw. The lowest point on $z$ is considered to be the pickup point and is denoted by the vertical line. The raptor's claw has a slower relative velocity at pickup than our claw when the quadrotor body speed is 2 m/s.

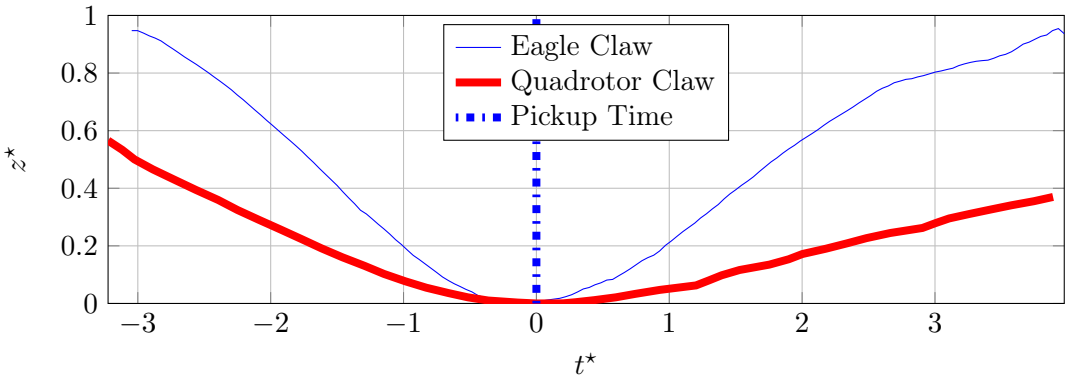



Figure 3.12: A comparison of the nondimensionalized $z$ positions of the quadrotor and the raptor claw. The vertical line indicates the pickup time.

# Chapter 4

# Perching on Inclined Surfaces

## 4.1 Introduction

Micro Aerial Vehicles (MAVs) face very limited flight times, which adversely impacts their efficacy for scenarios such as first response and disaster relief, where it might be useful to deploy MAVs for persistent radio relays, monitoring, or sampling. Many tasks, however, do not require the robot to be in motion. In fact, some tasks do not even require the robot to be airborne. For example, an MAV may be given the objective to monitor a crime scene until police arrive, to monitor the gas concentrations from a nearby gas leak, or suspend operation when not needed or during a period of unfavorable weather. In such cases, the vehicle could not only be stationary, but could also perch and turn off its motors to preserve energy for the next task. Thus, the ability to perch is instrumental for sustained operations in a wide variety of missions.

In the last chapter, we demonstrated grasping while a robot was in motion, and the extension to perching on a stationary object would be straightforward. However, the system was assumed to be planar, and possible perch locations would have been limited to horizontal surfaces or sufficiently small cylindrical objects such as branches. Since an aerial robot would most often be observing things below, it is desirable to avoid perching on level surfaces since they would interfere with downward observations. In cases such as urban environments, the best available surfaces may be vertical windows, walls, or inclined rooftops, since most level

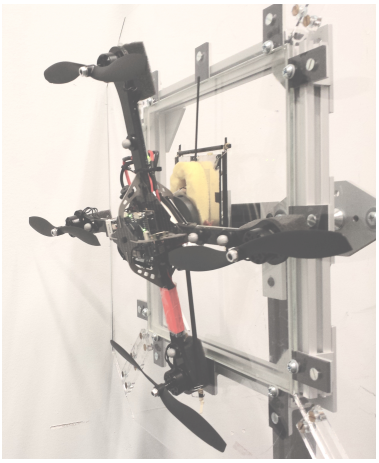


Figure 4.1: A quadrotor perched on a vertical glass surface.

surfaces are frequented by cars, pets, or pedestrians.

In this chapter[1], we predict and verify the performance of a custom gripper designed for perching on smooth surfaces. We also present control and planning algorithms, enabling an underactuated quadrotor with a downward-facing gripper to perch on smooth, inclined surfaces while satisfying constraints on actuation and sensing. Experimental results demonstrate the proposed techniques through successful perching on a glass surface at various inclinations, including vertical (see Figure 4.1).

## 4.2 Background

Fixed wing MAVs have perched on vertical walls using penetration-based grasping [66, 76, 77] and on cables such as power lines [95, 96]. However, the challenges of object detection and recognition as well as relative pose estimation using onboard sensors and computation in real-time, real-world, scenarios would make such approaches very difficult in practice for fast-moving, micro-sized, fixed-wing aircraft.

In comparison to fixed wing vehicles, quadrotors can carry larger payloads for their size (*i.e.,* more processing power, more sensors, and greater payload for the same footprint) and can provide both slow and fast, agile movements, which makes them useful in many scenarios. Their flight time is more restricted, however, which makes the motivation for perching even

[1]Much of the content in this chapter originally appeared in [128] and [132].

stronger. Passive gripping mechanisms were developed for perching in [5, 26, 27], but the mechanisms consume a large fraction of the available payload and are restricted to objects that are narrow enough for the gripper to cage. Similarly, [10] presents a gripper that requires cylindrical objects and would restrict the set of possible perch locations based on the size of the gripper. Dry adhesives are used to adhere to flat surfaces in [46], but a planning or control strategy for the robot to achieve perching is not presented. The authors in [18] and [56] use dry adhesives while [135] uses suction for perching. However, in these cases, the gripper is placed on the side of the robot, requiring extra compensation for the added moment and rendering the gripper unlikely to be useful for tasks such as transportation of objects. An opposed-grip dry adhesive is used in [54] to increase the capable load, but a launcher is used to emulate a perching maneuver.

Attempting to address the planning component, a bioinspired trajectory generation method (tau theory) was proposed for perching [141]. However, the dynamics and underactuation of the robot are not considered, and only kinematic simulations are provided. The authors of [88] and [89] present perching on vertical surfaces, but the gripper uses a hook and loop fastener, VELCRO®, for adhesion, which is not desirable for real-world scenarios since it would require preestablished perch locations and would make release from the perch location challenging. Furthermore, [88] and [89] rely on switching between linear controllers as well as iterative learning for successful perching. Their work is extended in [87] where time constants for attitude commands are incorporated, but the system still uses a state machine to toggle between a trajectory and attitude controller, which ultimately requires an iterative learning approach. All sensing and processing is executed onboard the robot in [94], but the perching is achieved using a hook and loop fastener. A correlation between scale and landing strategies is observed in nature and motivates a consideration of the tradeoffs between size, sensing, planning, dynamic flight maneuvers, and mechanical intelligence [65].

In this chapter, a passively actuated, downward-facing, dry-adhesive gripper is presented and used to adhere to smooth, inclined surfaces, and its requirements for successful perching on vertical surfaces are defined. The downward-facing gripper motivates a strategic approach

for trajectory planning since the system is underactuated such that the orientation and the position of the robot cannot be independently controlled. Thus, we present a suitable method for trajectory generation which considers actuator and sensor constraints to ensure that the planned trajectories are not only dynamically feasible, but also realizable on the physical platform. Further, we ensure that the robot gripper contacts the perching surface with an approach orientation and velocity within a required *landing envelope*. Finally, we show experiments using a quadrotor equipped with the proposed gripper. The key contributions of this chapter are threefold:

1. We model and characterize a passively-actuated, downward-facing, opposed-grip, dry adhesive gripper in terms of the conditions necessary for successful MAV perching on a vertical surface.

2. We present a method for control and planning to ensure that the robot's perch will fall within the landing envelope.

3. We experimentally demonstrate the effectiveness of the planning strategy, the controller, and the gripper.

The rest of the chapter is organized as follows: First, we provide an overview of the design of the gripper, propose a model for the landing envelope, and explore the landing envelope of the gripper experimentally. Following the gripper design, the dynamics and the controller for the robot are presented. Then, a method for planning physically realizable trajectories considering actuator and sensor constraints is developed. Finally, experimental results are demonstrated, and we conclude the chapter, offering paths for continued development.

## 4.3 A Gecko Inspired Gripper[2]

We are motivated to use a downward-facing gripper for perching, as opposed to a side-mounted gripper, since it could also be used to carry payloads. Specifically, we use a gripper

[2]Some of the content in this subsection was developed by collaborators from Stanford University, including Morgan Pope, Elliot W. Hawkes, Matthew A. Estrada, Hao Jiang, and Mark R. Cutkosky. The content was previously published in [128] and [132] and is included here for completness.

influenced by the opposed-grip dry adhesive design described in [45]. The device is "gecko-inspired" because of the dry adhesive used – a non-tacky silicone rubber with microscopic surface texturing that enables it to controllably switch between adhesive and non-adhesive states through the application of a shear force. This is inspired by the way a gecko uses micro-structures on its toes to adhere to surfaces. On dry, smooth surfaces, the adhesive can achieve up to 10 kPa in normal adhesive force [45], however, surfaces rougher than glass or acrylic will show some degradation in performance. Further, the material is stable over a wide range of temperatures, making it a viable solution for outdoor scenarios. By placing two adhesive tiles in opposition with an internal shear force, the tiles can achieve adhesion independent of surface orientation. The adhesive is lightweight (the 25 $cm^2$ of adhesive tile is only 12 g and can support a 580 g robot) and enables perching on smooth surfaces. The total mass of the gripping mechanism and suspension, including the dry-adhesive tiles, is 60 g.

To provide an adequate safety factor for the perching maneuver, two pairs of adhesive tiles were used. These are arranged in a cross shape, as illustrated in Figure 4.2. Both pairs are anchored to the same frame, a circle made out of a bent carbon fiber rod. This provides a consistent preload for both sets and keeps them in the same plane during the approach to the wall. Because of the tight tolerances maintained in this configuration, it was found to be unnecessary to use a differential mechanism (*e.g.,* pulleys or links as in [52]) to ensure even force distribution. Instead, both pairs are loaded through a single tendon attached at the center of the mechanism. In its final realization, the weaker of the two pairs averaged a failure load of 7.53 N, and the combined mechanism averaged a failure load of 14.2 N, which is 94% of the theoretical limit (calculated by doubling the limit of the weaker pads).

When perching with a quadrotor, aerodynamic surface effects tend to repel the vehicle from the target surface, and stray breezes can deflect a slow-moving vehicle just prior to impact. Additionally, impacting with too little kinetic energy makes it difficult to engage the gripper, and slow velocities are more difficult to achieve accurately, especially when perching on a vertical surface. These considerations motivate a relatively high incoming velocity so

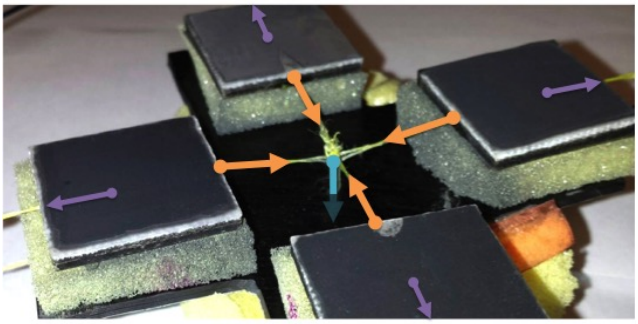


Figure 4.2: A view of the underside of the gripper. Initially, the pads are held in place by slight tension between the purple and orange arrows (opposed). Then, upon the collapse of the truss mechanism, each pad is placed in shear by pulling the pad towards the center (orange arrows) of the gripper using the tension string (blue downward arrow).

that the duration and impact of these effects is minimal. However, impact at such velocities results in high demands on the loading rate of any chosen gripper. In our case, the adhesives must be loaded within a 10 ms window when they are in active compression against the wall, or else premature failure can occur. As described in [45], this can be addressed by using a collapsing truss mechanism to guarantee an effective loading cycle, as illustrated in Figure 4.3. The truss compensates for high incoming velocities and can passively reset itself after a successful perch. A failed perch, however, requires recovery from the failure and also a manual reset of the truss mechanism. Future work will build upon the recovery strategies outlined in [17, 53], and the mechanism will be further developed to automate a reset.

### 4.3.1 Landing Envelope Modeling and Verification

The set of impact conditions which lead to success - the *landing envelope* for a perching vehicle - have previously been characterized for dry adhesive grippers in [54] using a numerical simulation and in [45] as an empirical result. Out of the range of possible state variables, two have proved crucial to understanding the performance of a given gripper on a specific robot - the velocity of the robot in the direction normal to the target surface (normal velocity) and the velocity of the robot parallel to the target surface (tangential velocity). We follow the approach used in [31] to model a ballistic perching and climbing robot, and

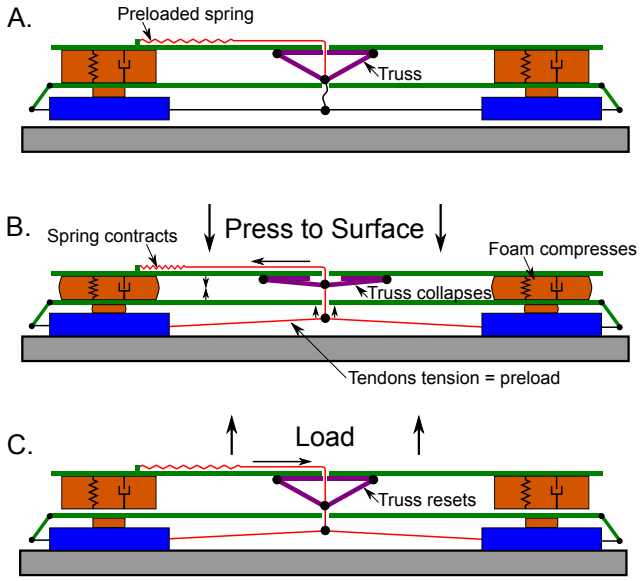



Figure 4.3: A cross-section view of the gripper (reproduced from [45]). The bistable truss mechanism is used to engage the directional adhesive pads upon contact with the surface. (A) Initially, a preloaded spring is tensioned low enough that the truss does not collapse. (B) Upon impact, the truss collapses (the magnets holding the one side together separate) and the tension in the spring is transmitted via tendons to the gripping pads to create shear. Since the mechanism is still in compression, the shear force results in an appropriate loading cycle for high speed engagement. (C) When the robot creates tension in the tendon, whether from the rebound or from static hanging, the truss mechanism resets, and the tension remains transmitted to the pads since the entire mechanism is being pulled away.

we extend it to the case of an incoming quadrotor to predict one boundary of the landing envelope. Next, we use empirical results and observations to complete the description of the safe region. Then, we briefly discuss our current understanding of the limits on other state variables, and formalize the landing envelope.

**Bounds on Rebound Spring Energy Storage**

The gripper is attached to the quadrotor by a compliant element referred to as the rebound spring, which is designed to keep the forces transmitted to the gripper below the adhesive limit of the pads until it reaches the hard stop at the end of its travel. In [31], it was shown that the potential energy stored in the rebound spring before hitting the upper limit was an effective predictor of the upper bound of incoming velocity. Note that this requires consideration of the coefficient of restitution in the direction normal to the wall to predict the rebound velocity after energy is dissipated in the suspension. The addition of foam between the mechanism and the quadrotor significantly increases damping in the normal direction. Foam is also placed underneath each rotor to soften their impact with the wall and help orient the robot consistently after perching, which also contributes to normal damping. Damping in the direction tangential to the surface is small because the adhesive pads generally adhere to the surface without translating, thus preventing the accumulation of frictional work along that boundary. With this in mind, we specify a boundary based on the energy storage of the rebound spring by integrating the spring force $F(l)$ over the spring length $l$. By defining the coefficient of restitution $c$ in the normal direction, a limit on incoming velocity $\dot{\mathbf{x}} \in \mathbb{R}^3$ can be stated as:

$$\frac{1}{2}m\left((\dot{\mathbf{x}}\cdot\mathbf{p}_1)^2+(\dot{\mathbf{x}}\cdot\mathbf{p}_2)^2+(c\,\dot{\mathbf{x}}\cdot\mathbf{p}_3)^2\right) < \int_{l_{min}}^{l_{max}} F(l)dl \tag{4.1}$$

where $m$ is the mass of the vehicle, $\mathbf{p}_3$ is a unit vector pointing into to the plane, and $\mathbf{p}_1$ and $\mathbf{p}_2$ are orthogonal unit vectors in the plane.

By launching the quadrotor against the wall with the dry adhesive surface covered, we were able to measure the coefficient of restitution in the normal direction for the current

system. The average of six trials resulted in a value of 0.30. The rebound spring applies an average force of 7.2 N over 5.0 cm of travel, resulting in 0.36 J of stored energy, which is equivalent to having an initial rebound velocity of 1.1 m/s. This limit on the energy storage results in the lower dashed curve in Figure 4.6.

**Maximum Upward Velocity and Pad Alignment**

When rotating to present the pads to a vertical surface, the quadrotor loses the ability to counteract gravity and begins to accelerate downward. Thus, in order to avoid large negative tangential velocities at impact, the robot must acquire sufficient upward velocity in the first part of the approach maneuver. Note that we use the term *tangential velocity* to mean the velocity along the plane which, in the vertical surface case, would be a velocity in the vertical direction. A desired positive (*i.e.,* upward) tangential velocity requires a greater initial vertical velocity (and also, a longer vertical acceleration), which increases the minimum height requirement for the entire maneuver. In practice, this means that target velocities that are less than or equal to zero are easier to achieve, especially when the vertical distance for perching is constrained.

With this in mind, the gripper is designed to function best when impacting the wall with negative (*i.e.,* downward) tangential velocity. As illustrated in Figure 4.4, the angle at which the pads initially contact the surface must be matched to the expected tangential velocity at time of impact. If the leading tile impacts the wall first, friction with the wall pushes the tiles closer together, creating slack in the tendon connecting them and leading to a final loading angle which is too steep to realize the full adhesive capability of the material. If the trailing tile impacts first, the tension in the connecting tendon maintains an appropriate distance between the tiles and still ensures proper loading. Therefore, the joint between the quadrotor and the gripper is designed so that the upper adhesive tile contacts the wall first. Further, the maneuver is specified so that the quadrotor impacts the wall with the upper arm inclined slightly towards the surface, which also has the benefit of requiring a smaller angle of rotation and less time to rotate. This impact orientation, however, creates problems for perch attempts with positive tangential velocity, motivating an upper limit

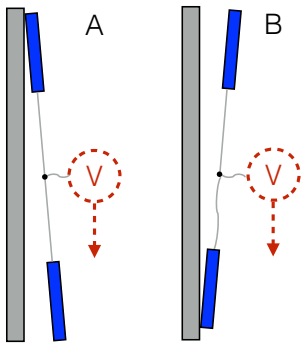



Figure 4.4: Qualitative illustration of a failure mode due to an inappropriate matching of tangential velocity and adhesive tile orientation at impact. In A), the trailing tile impacts first, maintaining tension between the two tiles and preserving a consistent final tile spacing and an appropriate final loading angle of the tendon. In B), the leading tile impacts first, causing the connecting tendon to go slack and allowing the tiles to settle on the wall closer to each other than intended. This results in inconsistent and non-optimal loading on rebound, increasing the chance of failure.

of 0 m/s, which is represented as a horizontal dashed line in Figure 4.6. Our future work will consider this effect in other directions since wind gusts could induce velocity errors in the lateral or tangential directions and could lead to non-ideal loading of the adhesive tiles unless the angle of impact is adjusted accordingly.

**Minimum Normal Velocity**

A lower bound in the normal direction is a result of velocities which are too slow to effectively align and engage the pads in compression. A statistical analysis of the failures between the energy storage limit curve and the tangential velocity limit provides an estimated lower boundary of 0.8 m/s. This limit appears as the vertical dashed line in Figure 4.6. The adhesives themselves require very little preload force; thus, it is conceivable that a design optimized for low speeds could significantly reduce this limit and will be explored in future work.

**Formalization of the Landing Envelope**

The orientation of the robot, $R \in SO(3)$, should be such that the gripper is angled slightly towards the target plane (see Figure 4.4) and with minimal angular velocity, $\boldsymbol{\Omega} \in \mathbb{R}^3$. Future work will explore the full effect of orientation, which currently has been controlled

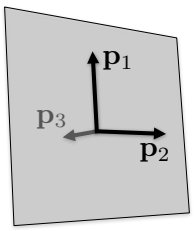



Figure 4.5: A diagram of the reference frame at the perch location. The vector $\mathbf{p}_3$ points into the plane, $\mathbf{p}_2$ is horizontal in the world and lies on the surface, and $\mathbf{p}_1$ is defined to maintain a right-handed system.

to be parallel to the plane in most tests. The lateral velocity along the surface has also been controlled to be small, but should have an effect similar to increased velocity in the tangential direction. Let the orientation of the plane be defined using the following right-handed basis such that $\mathbf{p}_3$ is a unit vector that is normal and pointed into the surface, $\mathbf{p}_2$ is a unit vector that is horizontal in the world and lies on the surface, and $\mathbf{p}_1 = \mathbf{p}_2 \times \mathbf{p}_3$, which is parallel to the surface and pointing upward in the world (angle depends on the inclination). Then, we can formalize the landing envelope in terms of the robot's position $\mathbf{x}$, velocity $\dot{\mathbf{x}}$, orientation $R$, and angular velocity $\boldsymbol{\Omega}$:

$$\left\{ \mathbf{x},\ \dot{\mathbf{x}},\ R,\ \boldsymbol{\Omega} \;\middle|\; \begin{array}{l} \mathbf{x} = \text{robot perch location} \\ \dot{\mathbf{x}} \cdot \mathbf{p}_1 < 0 \text{ m/s} \\ \dot{\mathbf{x}} \cdot \mathbf{p}_2 = 0 \text{ m/s} \\ \dot{\mathbf{x}} \cdot \mathbf{p}_3 > 0.8 \text{ m/s} \\ \frac{1}{2} m((\dot{\mathbf{x}} \cdot \mathbf{p}_1)^2 + (0.3\dot{\mathbf{x}} \cdot \mathbf{p}_3)^2) < 0.36 \text{ J} \\ R \text{ s.t. top slightly angled towards plane} \\ \boldsymbol{\Omega} = \mathbf{0} \text{ rad/s} \end{array} \right\}. \tag{4.2}$$

**Verification of Normal and Tangential Velocity Bounds**

The predicted bounds for the landing envelope were verified by launching the unpowered quadrotor against a vertical surface, recording its motion using a high-speed camera, and plotting the successes and failures as green dots and red squares, respectively, in Figure 4.6. When flying autonomous, powered perches, the maneuvers targeted velocities near the mid-

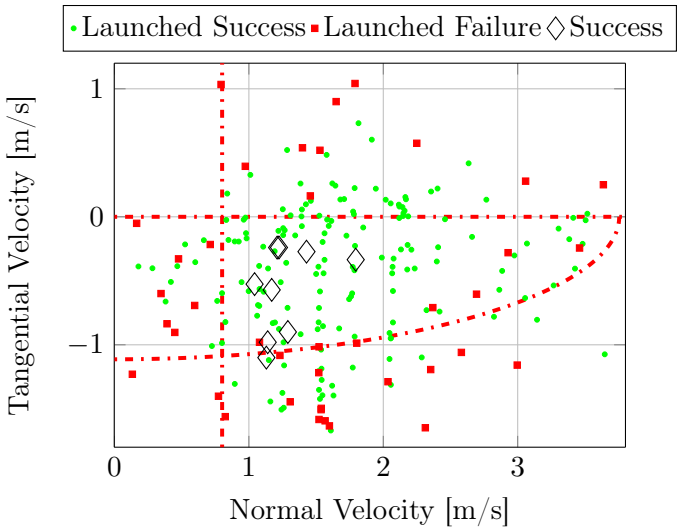



Figure 4.6: The perching envelope of the quadrotor based on impact velocities relative to a vertical surface. Successful unpowered (*i.e.,* launched) perches are shown as green dots, and unpowered failures are shown as red squares. Diamonds indicate successes for trials while the quadrotor was flying under its own power. Predicted boundaries are shown as dash-dotted red lines. The parabolic boundary is determined by calculating the kinetic energy after rebound relative to the maximum energy storage in the rebound spring. The left boundary is estimated based on the recorded low-speed failures, and the top boundary is based on the improper engagement failure observed for upward (positive) impact velocities. One can observe that a reasonable target normal velocity is 1.4 m/s with a tangential velocity of 0.4 m/s in the downward direction.

dle of the safe region. In flight, the planning and control strategies described in the rest of this chapter are able to achieve landing conditions which meet these requirements, resulting in successful perches (plotted as diamonds in Figure 4.6). In future work, the method described here for estimating the effective landing envelope should be easily extened to different quadrotors, simply by adjusting the mass, coefficient of restitution, and spring energy used in (4.1).

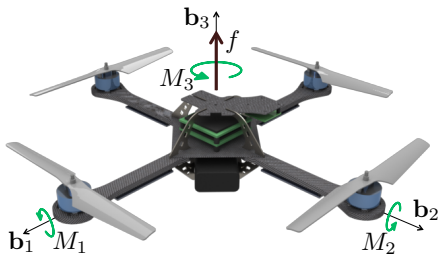



Figure 4.7: The control inputs of a quadrotor can be considered to be a net force, $f$, and moments about each of the principal axes, $M_i$.

## 4.4 Dynamics and Control

### 4.4.1 Dynamics

We refer the reader to the dynamics which have already been presented in Section 2.1.3. For reference, the results from (2.5) are reproduced here

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \frac{f}{m} R\mathbf{e}_3 - g\mathbf{e}_3 \\ \dot{R} &= R\,\boldsymbol{\Omega} \\ \dot{\boldsymbol{\Omega}} &= \mathcal{I}^{-1}\left(\mathbf{M} - \boldsymbol{\Omega} \times \mathcal{I}\boldsymbol{\Omega}\right) \end{aligned} \tag{4.3}$$

with $\mathbf{v} \in \mathbb{R}^3$ representing the velocity and with $f$ and $\mathbf{M}$ as the control inputs to the system as pictured in Figure 4.7. It will also be useful to have the translational dynamics expressed as

$$m\ddot{\mathbf{x}} = fR\mathbf{e}_3 - mg\mathbf{e}_3. \tag{4.4}$$

### 4.4.2 The Control Law

We implement the control law developed in [69], which guarantees exponential stability if the geodesic attitude error is less than 90° and exhibits almost global exponential attractiveness (the only exception is when the geodesic attitude error is 180°). Let the thrust be

$$f = (-k_x\mathbf{e}_x - k_v\mathbf{e}_v + mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_d) \cdot R\mathbf{e}_3 \equiv \mathbf{f}_{des} \cdot R\mathbf{e}_3 \tag{4.5}$$

where $k_x$ and $k_v$ are positive gains,

$$\mathbf{e}_x = \mathbf{x} - \mathbf{x}_d \quad \text{and} \quad \mathbf{e}_v = \dot{\mathbf{x}} - \dot{\mathbf{x}}_d$$

are position and velocity errors, respectively, and $\ddot{\mathbf{x}}_d$ is the nominal acceleration. The commanded moments are given by

$$\mathbf{M} = -k_R \mathbf{e}_R - k_\Omega \mathbf{e}_\Omega + \boldsymbol{\Omega} \times \mathcal{I}\boldsymbol{\Omega} - \mathcal{I}\left(\hat{\boldsymbol{\Omega}} R^T R_c \boldsymbol{\Omega}_c - R^T R_c \dot{\boldsymbol{\Omega}}_c\right) \tag{4.6}$$

where $k_R$ and $k_\Omega$ are positive gains and

$$\mathbf{e}_R = \frac{1}{2}\left(R_c^T R - R^T R_c\right)^\vee \quad \text{and} \quad \mathbf{e}_\Omega = \boldsymbol{\Omega} - R^T R_c \boldsymbol{\Omega}_c$$

are the angular position and rate errors with $\cdot^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$ being the opposite of the "hat" map. In this case, $\boldsymbol{\Omega}_c \in \mathbb{R}^3$ is the "commanded" or nominal angular rate and $R_c$ is the "commanded" attitude, which is given by

$$R_c = \begin{bmatrix} \mathbf{b}_{1_c}, & \mathbf{b}_{3_c} \times \mathbf{b}_{1_c}, & \mathbf{b}_{3_c} \end{bmatrix} \tag{4.7}$$

where

$$\mathbf{b}_{3_c} = \frac{-k_x \mathbf{e}_x - k_v \mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_d}{\|-k_x \mathbf{e}_x - k_v \mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_d\|} \tag{4.8}$$

and $\mathbf{b}_{1_c}$ is chosen such that $\mathbf{b}_{3_c} \times \mathbf{b}_{1_c}$ is well conditioned. In our case, $\mathbf{b}_{1_c}$ is defined by a combination of the planned trajectory and the desired force and will be explained further in the next section.

## 4.5 Planning with Constraints

The system is underactuated since the thrust can only directly affect the translational acceleration in the $\mathbf{b}_3$ direction, and the moments can only directly affect the angular acceleration. Since we are interested in aggressive maneuvers in which the quadrotor can perch on vertical surfaces, it is important to ensure that we can plan trajectories that are not only dynam-

ically feasible (considering the underactuation), but also physically realizable (considering actuator and sensor constraints). In the last chapter, we presented a 2D solution to the first problem and ignored the actuator and sensor constraints. Now, we will generalize the planning method to 3D and also present a way to address actuator and sensor constraints.

### 4.5.1 Planning for Dynamic Feasibility

In this subsection, we will present results similar to [85], which facilitates the computation of trajectories for the underactuated quadrotor system. We propose the following set of variables called *flat outputs*, which will be used to show that the control inputs and state of the system can be expressed in terms of this subset of variables and their derivatives:

$$\mathcal{Y} = \begin{bmatrix} \mathbf{x} \\ \psi \end{bmatrix} \tag{4.9}$$

where $\mathbf{x}$ is the position of the vehicle and $\psi$ defines the yaw angle. In particular, these results will provide insight into how smoothness constraints in the flat output space can be used to guarantee dynamic feasibility. This property of the system is called *differential flatness* and has also been shown to be useful for planning trajectories for other underactuated systems [36, 98]. For more details, refer to our discussion in Section 3.2.2.

Now, observe from (4.4) that the nominal force can be determined from the acceleration of the trajectory, $\ddot{\mathbf{x}}$, since $\|R\mathbf{e}_3\| = 1$

$$f = m \left\| \ddot{\mathbf{x}} + g\mathbf{e}_3 \right\| \tag{4.10}$$

and the orientation of the third body frame axis, $\mathbf{b}_3$ as in Figure 4.7, is

$$R\mathbf{e}_3 = \mathbf{b}_3 = \frac{\ddot{\mathbf{x}} + g\mathbf{e}_3}{\|\ddot{\mathbf{x}} + g\mathbf{e}_3\|}. \tag{4.11}$$

The rest of the rotation matrix, $R$, can be determined by defining a vector, $\mathbf{b}_1$ orthogonal to $\mathbf{b}_3$ using $\psi$ and then using $\mathbf{b}_3 \times \mathbf{b}_1$ to determine $\mathbf{b}_2$. In [85], an intermediate vector was

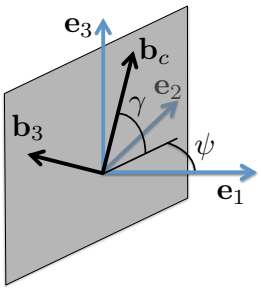



Figure 4.8: A strategy to avoid the singularity when the robot is horizontal. We define the $\mathbf{b}_c$ vector based on $\psi$ and $\mathbf{b}_3$ in order to determine $\mathbf{b}_2$ while avoiding the singularity when $\mathbf{e}_3 \cdot \mathbf{b}_3 = 0$.

defined as $\mathbf{b}_c = \begin{bmatrix} \cos\psi, & \sin\psi, & 0 \end{bmatrix}$ so that $\mathbf{b}_2$ could be determined directly by

$$\mathbf{b}_2 = \frac{\mathbf{b}_3 \times \mathbf{b}_c}{\|\mathbf{b}_3 \times \mathbf{b}_c\|}. \tag{4.12}$$

However, in our case, such an approach is dangerously close to the singularity that results when $\mathbf{b}_3$ is parallel to $\mathbf{b}_c$, which is likely to be the case when perching on a vertical surface since the thrust vector may be horizontal. To avoid this, we choose $\mathbf{b}_c$ such that the singularity is avoided by allowing $\mathbf{b}_c$ to be rotated (no more than $\pm\pi/2$) while remaining in the plane defined by $\begin{bmatrix} \cos\psi, & \sin\psi, & 0 \end{bmatrix} \times \mathbf{e}_3$. In practice, this means that we can define $\mathbf{b}_c$ as

$$\mathbf{b}_c = \begin{bmatrix} \cos\gamma\cos\psi \\ \cos\gamma\sin\psi \\ \sin\gamma \end{bmatrix}, \quad \gamma \in (-\pi, \pi) \tag{4.13}$$

where $\gamma$ is chosen based on $\mathbf{b}_3$ (see Figure 4.8). Then, $\mathbf{b}_2$ is given by (4.12),

$$\mathbf{b}_1 = \mathbf{b}_2 \times \mathbf{b}_3 \tag{4.14}$$

and

$$R = \begin{bmatrix} \mathbf{b}_1, & \mathbf{b}_2, & \mathbf{b}_3 \end{bmatrix}. \tag{4.15}$$

The next derivative of (4.4) is given by

$$\begin{aligned} m\mathbf{x}^{(3)} &= f\dot{R}\mathbf{e}_3 + \dot{f}R\mathbf{e}_3 \\ &= fR\hat{\mathbf{\Omega}}\mathbf{e}_3 + \dot{f}\mathbf{b}_3 \end{aligned} \tag{4.16}$$

and the scalar projection onto $\mathbf{b}_3$ reveals that

$$\dot{f} = \mathbf{b}_3 \cdot m\mathbf{x}^{(3)}. \tag{4.17}$$

Next, we can determine the first two terms of $\mathbf{\Omega}$ by solving (4.16) for $\hat{\mathbf{\Omega}}\mathbf{e}_3$ and independently projecting onto $-\mathbf{e}_2$ and $\mathbf{e}_1$ (note that $R^T\mathbf{b}_i = \mathbf{e}_i$ and, similarly, $\mathbf{e}_i^T R^T = \mathbf{b}_i^T$),

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix} = \frac{m}{f} \begin{bmatrix} -\mathbf{b}_2^T \\ \mathbf{b}_1^T \end{bmatrix} \mathbf{x}^{(3)}. \tag{4.18}$$

If we express $R$ as a $Z$-$Y$-$X$ rotation about $\psi$, $\theta$, $\phi$, respectively, we can determine

$$\mathbf{\Omega} = \left(R^T\dot{R}\right)^\vee \tag{4.19}$$

from which we can express $\Omega_3$ in terms of already computed parameters and $\dot{\psi}$

$$\Omega_3 = \frac{\left(1 - \left(\mathbf{e}_3^T\mathbf{b}_1\right)^2\right)\dot{\psi} - \mathbf{e}_3^T\mathbf{b}_2\Omega_2}{\mathbf{e}_3^T\mathbf{b}_3}. \tag{4.20}$$

Now we have introduced a singularity that is, to the best of our knowledge, unavoidable in the full 3-D case. Thus, any portion of the trajectory that passes near the singularity, we formulate as a vertical planar model, which results in a reduced-dimensioned flat space with $\Omega_3 = 0$ as a constant when expressed in 3-D.

Another derivative of (4.4) provides

$$m\mathbf{x}^{(4)} = fR\left(\hat{\dot{\mathbf{\Omega}}} + \hat{\mathbf{\Omega}}\hat{\mathbf{\Omega}}\right)\mathbf{e}_3 + 2\dot{f}R\hat{\mathbf{\Omega}}\mathbf{e}_3 + \ddot{f}R\mathbf{e}_3, \tag{4.21}$$

which is sufficient to determine $\dot{\Omega}_1$ and $\dot{\Omega}_2$ using the scalar projections onto $\mathbf{e}_1$ and $-\mathbf{e}_2$

$$\begin{bmatrix} \dot{\Omega}_1 \\ \dot{\Omega}_2 \end{bmatrix} = \frac{m}{f} \begin{bmatrix} -\mathbf{b}_2^T \\ \mathbf{b}_1^T \end{bmatrix} \mathbf{x}^{(4)} - 2\frac{\dot{f}}{f} \begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix} + \Omega_3 \begin{bmatrix} \Omega_2 \\ -\Omega_1 \end{bmatrix}. \tag{4.22}$$

Note that this result can also be obtained directly from the derivative of $\Omega_1$ and $\Omega_2$. The third element, $\Omega_3$, can also be determined directly and will require $\ddot{\psi}$,

$$\dot{\Omega}_3 = -\frac{\mathbf{e}_3^T\dot{\mathbf{b}}_3\Omega_3 + \mathbf{e}_3^T\dot{\mathbf{b}}_2\Omega_2 + \mathbf{e}_3^T\mathbf{b}_2\dot{\Omega}_2 + 2\mathbf{e}_3^T\mathbf{b}_1\mathbf{e}_3^T\dot{\mathbf{b}}_1\dot{\psi} + \left((\mathbf{e}_3^T\mathbf{b}_1)^2 - 1\right)\ddot{\psi}}{\mathbf{e}_3^T\mathbf{b}_3}. \tag{4.23}$$

Note that

$$\begin{bmatrix} \dot{\mathbf{b}}_1 & \dot{\mathbf{b}}_2 & \dot{\mathbf{b}}_3 \end{bmatrix} = \dot{R} = R\hat{\mathbf{\Omega}}, \tag{4.24}$$

which are already known from (4.15), (4.18), and (4.20). Having the angular acceleration, we can solve for the required moments,

$$\mathbf{M} = \mathcal{I}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathcal{I}\mathbf{\Omega}. \tag{4.25}$$

Thus, the control inputs and the state of the robot can be expressed in terms of the *flat outputs* and their derivatives. The mapping is a diffeomorphism, so we have shown that the system is *differentially flat* [98]. In particular, we see that the $4^{th}$ derivative of position and the $2^{nd}$ derivative of the yaw angle appear in the control inputs. Then, any trajectory that is sufficiently smooth in the flat outputs ($\mathbf{x}(t) \in \mathcal{C}^4$ and $\psi \in \mathcal{C}^2$) defines a trajectory which satisfies the dynamics of the system, and is therefore a dynamically-feasible trajectory.

### 4.5.2 Actuator and Sensor Constraints

In addition to dynamic feasibility, we must also consider actuator and sensor constraints, especially when planning aggressive trajectories. While the trajectories are defined in the flat space, it is likely that the physical constraints are not defined directly in the flat space. Next, we will demonstrate how these constraints can be mapped to constraints in the flat space, which enables their consideration when planning trajectories.

First, the net thrust is bounded by $f_{max}$, which can be expressed as

$$m \left\| \ddot{\mathbf{x}} + g\mathbf{e}_3 \right\| \leq f_{max}. \tag{4.26}$$

The gyros saturate at $\omega_{max}$, which can be limited by imposing a bound on the jerk expressed from (4.18) with a $\beta_1$ function as

$$\beta_1(\mathbf{x}^{(3)}, \ddot{\mathbf{x}}, \dot{\psi}) \leq \omega_{max} \tag{4.27}$$

and the maximum moment about the $i^{th}$ axis is bounded by $M_{i_{max}}$, requiring that

$$\beta_2(\mathbf{x}^{(4)}, \mathbf{x}^{(3)}, \ddot{\mathbf{x}}, \ddot{\psi}, \dot{\psi}) \leq M_{i_{max}}. \tag{4.28}$$

In practice, these constraints are coupled through (2.1). If the thrust is saturated, then all rotors are spinning at their maximum speed, and the moment inputs must be zero. Similarly, if a large moment is required, the thrust cannot simultaneously be zero. The thrust also cannot be zero because of the singularity in (4.11). For this reason, when planning, we further restrict certain constraints during portions of the trajectory that are expected to require large control inputs.

### 4.5.3 A Constrained Optimization Problem

The trajectories can be parametrized using an appropriate basis function (*e.g.,* Legendre polynomials), $\mathbf{h}(t) \in \mathbb{R}^m$, and coefficients, $\mathbf{c}_i \in \mathbb{R}^m$, such that

$$\mathcal{Y}_i(t) = \mathbf{c}_i^T \mathbf{h}(t) \;\text{ for }\; i = 1, \ldots, 4. \tag{4.29}$$

Next, an objective function is formulated to minimize the control inputs in the flat space [85]. Minimizing the integral of the square of the $n_i^{th}$ derivative of the $i^{th}$ flat output provides the

cost function,

$$\begin{aligned}\mathcal{J}_i &= \int_{t_0}^{t_f} \left\| \mathcal{Y}_i^{(n_i)}(t) \right\|^2 dt, \quad i = 1, \ldots, 4 \\ &= \mathbf{c}_i^T \left[ \int_{t_0}^{t_f} \mathbf{h}^{(n_i)}(t) \left[ \mathbf{h}^{(n_i)}(t) \right]^T dt \right] \mathbf{c}_i \\ &\equiv \mathbf{c}_i^T H_i \mathbf{c}_i\end{aligned} \tag{4.30}$$

where, in our case, $n = \begin{bmatrix} 4 & 4 & 4 & 2 \end{bmatrix}$. Then, $H_i \in \mathbb{R}^{m \times m}$ is used to formulate the problem as a Quadratic Program (QP):

$$\begin{aligned} \text{minimize} \quad & \mathbf{C}^T \mathcal{H} \mathbf{C} \\ \text{subject to} \quad & A\mathbf{C} \leq B \\ & A_{eq}\mathbf{C} = B_{eq} \end{aligned} \tag{4.31}$$

with

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_4 \end{bmatrix}, \quad \mathcal{H} = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_4 \end{bmatrix},$$

with the constraints discussed in the previous section incorporated using a series of linear approximations in $A \in \mathbb{R}^{k \times 4m}$ and $B \in \mathbb{R}^k$ where $k$ is the total number of linear constraints. The matrix $A_{eq} \in \mathbb{R}^{p \times 4m}$ and vector $B_{eq} \in \mathbb{R}^p$ can be used to impose $p$ equality constraints. For example, we can use these to specify a velocity or acceleration constraint at a desired time.

The previous formulation is satisfactory for a single trajectory segment, but in practice, more than one segment is needed to maintain a high degree of freedom without resulting in computational errors. To keep the problem well conditioned, a desired trajectory is broken into smaller segments rather than increasing the order of the polynomial. Then, the coefficients for each segment of a particular dimension can be stacked and incorporated into

the QP. Finally, it is important to incorporate equality constraints between segments such that the $i^{th}$ dimension is $\mathcal{C}^{n_i}$.

### 4.5.4 Boundary Conditions

Now, we will discuss the specific boundary conditions to enable successful perching using our quadrotor. The initial conditions can be chosen to match the current state. It is assumed that the position and orientation (orientation defined by a vector, $\mathbf{p}_3$, normal to the plane) of the perch location are known. Thus, the position at impact is defined such that the quadrotor would be perched on the window. The desired normal impact velocity can be chosen from the interior of the landing envelope. A desired acceleration vector should be defined such that $\mathbf{b}_3$ (see (4.11)) is nearly aligned with $\mathbf{p}_3$ or slightly angled towards the plate so that the plane of the quadrotor is nearly parallel to the window and so that the highest pad makes contact first. In addition, the tangential velocity is desired to be slightly downward so that the highest pad is engaged as discussed in Figure 4.4. These conditions should fall within the boundary given in Figure 4.6.

From [87], we estimate that the time to change orientation will be a maximum of 0.4 s (assuming a change of $\pi/2$ radians). Thus, during the last portion of the trajectory, we specify an acceleration such that $\mathbf{b}_3$ is nearly constant and the magnitude of the acceleration places the thrust in the middle of its range to ensure that the robot has the moment control authority to rotate in case the orientation is lagging behind the desired. Note that it is important to consider gravity when using acceleration constraints to specify an attitude. For example, if we would like the $\mathbf{b}_3$ vector to be at an angle $\theta$ and the thrust to be $f_{max}/2$, the acceleration must be expressed as

$$\ddot{\mathbf{x}}^d = \begin{bmatrix} \frac{f_{max}}{2m}\sin(\theta) \\ 0 \\ \frac{f_{max}}{2m}\cos(\theta) - g \end{bmatrix}.$$

See Figure 4.9 and Figure 4.10 for a sample planar trajectory and acceleration vectors during the trajectory. See Figure 4.11 for the components of acceleration. Notice that the

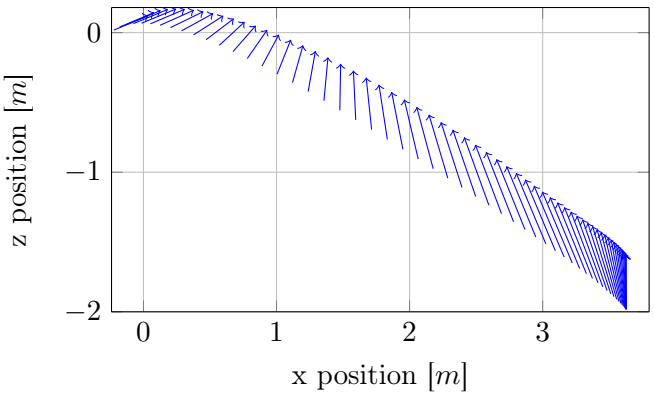



Figure 4.9: A sample trajectory with vectors denoting the acceleration direction and magnitude. The quadrotor starts on the bottom right and perches on the left at an incline of 70°. The upper left corner is presented in a higher temporal resolution in Figure 4.10.

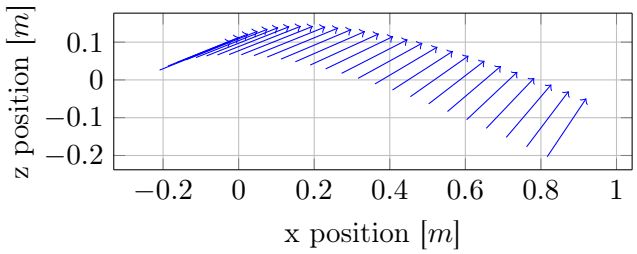



Figure 4.10: The last 40 ms of a sample perching trajectory. The arrows denote the acceleration direction (*i.e.,* the direction of $\mathbf{b}_3$) and magnitude. Notice that the direction of the vector does not change significantly towards the end of the trajectory where the acceleration is bounded during planning.

acceleration is bounded towards the end of the trajectory to ensure that, if there is lag, the robot is still able to achieve the desired orientation before impact. Finally, to minimize unnecessary kinetic energy, we require that the angular rates are zero at impact.

## 4.6 Experimental Results

In this section, we present the experimental setup and the results of the perching trials. The experiments are conducted in the GRASP lab at the University of Pennsylvania (see [91]) using a Hummingbird quadrotor from Ascending Technologies[3], and the QP is solved using Gurobi Optimizer 6.0[4] with the MATLAB® interface. A motion capture system is used for position feedback at 100 Hz. The setup is documented in Figure 4.12. Successful perches on

[3] `www.asctec.de`
[4] `www.gurobi.com`

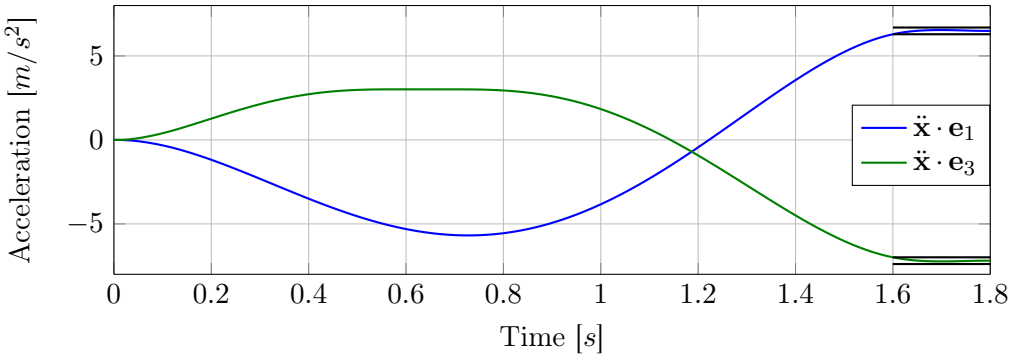



Figure 4.11: A plot of the nominal acceleration for the trajectory in Figure 4.9. Notice that during the last portion of the trajectory, the acceleration is bounded by the black lines, which dictates that the angular velocity will be nearly zero and that the robot will achieve the correct orientation before impact.

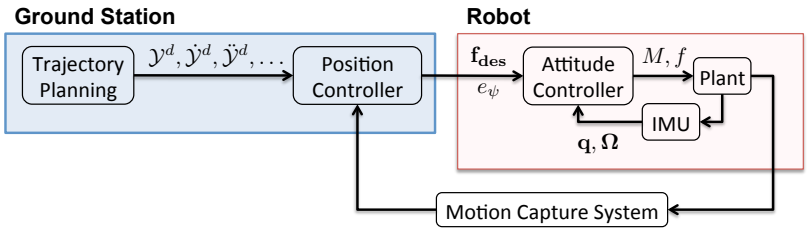



Figure 4.12: The architecture of the system. The ground station handles the trajectory planning and passes the trajectories to the position controller, which receives feedback from the motion capture system. The position controller sends a desired force, $\mathbf{f}_{des}$, a $\psi$ error, and the necessary feedforward inputs to the robot. Internally, the attitude controller runs at 1 kHz to update the commanded force and moments based on the position controller and the feedback from the IMU.

a vertical surface under various impact conditions are denoted by diamonds in Figure 4.6. These demonstrate the effectiveness of the planning algorithm, controller, and gripper as the impact conditions fall within or near the boundary of the proposed landing envelope. The angular velocities for 3 vertical surface perches are overlayed in Figure 4.13. In particular, we would like to highlight that at the time of impact, the angular velocity of the vehicle is nearly zero as desired. Further, the reader is encouraged to view an image sequence of a perching robot in Figure 4.14 as well as the online[5] multimedia material.

In addition to perching on vertical surfaces, the robot is able to perch on various inclinations without the need for iterative experimental trials. For example, perches on surfaces at 30°, 50°, 70°, and 90° are presented in Figure 4.15. Further, one of the motivations for

[5] `http://www.jtwebs.net/2016-jmr/`

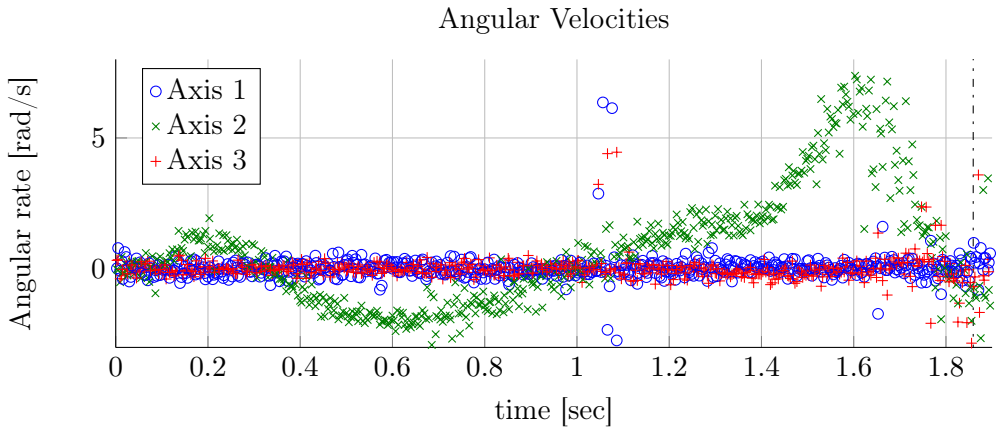



Figure 4.13: The angular velocities for 3 different perching trials on a vertical surface as estimated using the motion capture system. The vertical dash-dotted black line denotes the time of contact with the surface. As desired, the angular velocity is controlled to zero before impact.

Figure 4.14: A strobe image sequence of a perching robot. The robot starts on the right hand side outside of the field of view, accelerates towards the target, and rotates in time to achieve a successful perch within the landing envelope of the gripper.

the downward-facing gripper is that it can also be used to transport objects. For example, the robot could be used to retrieve a lost cell phone as pictured in Figure 4.16.

### Surface Effects

During these experiments, we noticed the impact of aerodynamic surface effects as the robot approached the perch plate. Even in quasi-static situations, surface effects are noticed [106]. Thus, this effect was not unexpected and will be explored further in future work.

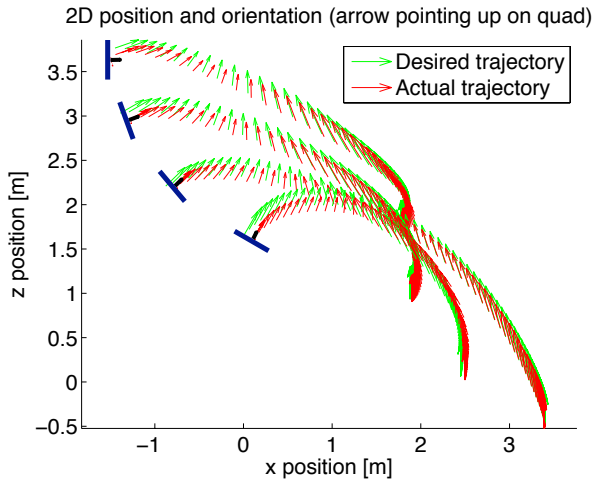



Figure 4.15: Using the proposed planning method, the angle of the surface can be changed without the need for iterative experimental trials. The root of each arrow indicates the position of the robot, and the arrow indicates the direction of the thrust (*i.e.,* the orientation of the robot).

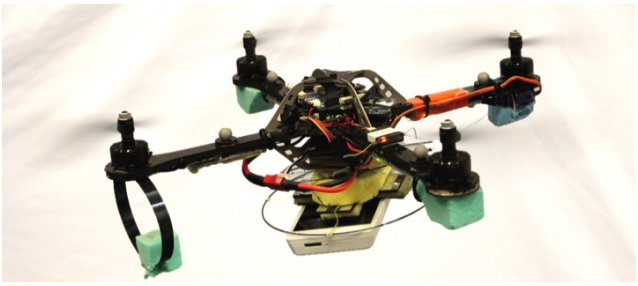


Figure 4.16: The downward-facing gripper can also be used to carry payloads such as a cell phone.

## 4.7 Conclusion and Future Work

In this chapter, we presented a strategy for planning trajectories to enable quadrotors to perch on inclined surfaces while taking into account constraints due to dynamics, actuators, sensors, and the gripper. To show the effectiveness of the proposed methodology, we demonstrated perching on inclined (up to 90°) surfaces using a quadrotor equipped with a downward-facing dry-adhesive gripper. Future work could include perching on curved surfaces, bat-like perching by hanging, and sensing failures during perching, and an exploration of how the presented methods are impacted by the scale of a vehicle. Another important future direction is the possibility to automate the gripper's detaching process using an onboard actuator, which would require balancing the need for compliance in a landing mechanism

with the desire for a well-defined perching posture to simplify the return to free flight. We also could move closer to real-world perching scenarios by exploring the impact of dusty surfaces on the gripper effectiveness and by generalizing the gripper for other surfaces (*e.g.,* brick, drywall, painted exterior metal, or marble) by using micro-spines or an appropriate amount of adhesive. Finally, in the next chapter, we will leverage vision-based techniques to enable perching independent from an external motion capture system.

# Chapter 5

# Vision-Based Perching

## 5.1 Introduction to Vision-Based Control

Micro Aerial Vehicles are becoming ubiquitous and are being employed for a wide range of tasks such as aerial photography, environmental monitoring, and disaster response. We see increasing autonomy in such areas, but the effectiveness of multi-rotor MAVs is, nevertheless, significantly limited by flight time, as discussed in Chapter 4. This is because the expensive power demands for hover and the limited energy density of batteries combine to limit flight durations [67]. However, there are many persistent monitoring operations that do not require vehicles to be actively flying once a target location has been reached. Therefore, it is appealing to develop solutions that might allow robots to perch to conserve energy and/or recharge their batteries.

The results discussed in Chapters 3 and 4 were dependent upon a structured lab environment with a motion-capture system. This requirement, however, is a significant limitation for real-world applications. In this chapter[1], we propose a novel approach to perching, independent of a motion-capture system, for an autonomous quadrotor sensing with only a single camera and an onboard Inertial Measurement Unit (IMU). We focus on the problem of vision-based control to enable grasping a cylinder with an overhead gripper, allowing a robot to hang from branches, poles (see Figure 5.1), or power lines, opening up possibilities

[1]Much of the content in this chapter originally appeared in [125].

Figure 5.1: A sample outdoor perch location for a quadrotor on a cantilevered cylindrical light post. Note that the axis of the cylinder can generally oriented.

to reduce power consumption or charge inductively. Specifically, we consider the problem of obtaining reliable pose estimates relative to a cylinder and the synthesis of a control law that allows the vehicle to converge to stable perching positions using the apparent contour of the cylinder as feedback.

In order to eliminate the dependency on a motion-capture system, robots must be able to carry onboard sensors that can provide information for localization. One sensor possibility is a camera, which has a very appealing information-to-weight ratio. In fact, cameras are sufficient to build a map of an environment and to localize a robot with only a scale ambiguity [19]. The scale can be resolved, for example, by using a second camera, a range sensor, a priori knowledge of the scene, or an IMU. Estimation using stereo cameras has proven to be effective for autonomous navigation and mapping but requires heavy processing capabilities and a pair of calibrated cameras, which increases the weight, complexity, and cost of the vehicle while decreasing the agility [112]. Similarly, a range sensor such as an ultrasonic range finder or a single beam laser adds additional weight. An IMU, however, provides sufficent information with no increase in weight (an IMU is already onboard most aerial robots) and only a slightly increased computational load. Indeed, many research groups have recognized the benefits of pairing a single camera with an IMU because the two sensors have complementary characteristics, are lightweight, and are relatively inexpensive, especially when compared to other options such as laser scanners, RGB-D sensors, or a

stereo camera rig. Recent work has shown the feasibility of using a single monocular camera and an IMU for real-time localization [49, 55, 60] and autonomous flight [39, 109, 113, 138]. In these works, information from the two sensors is fused using an Extended Kalman Filter (EKF) or an Unscented Kalman Filter (UKF) in order to provide localization that is viable for real-time control. These estimates include the positions and velocities of the robot in the inertial frame and possibly the 3-D positions of features observed by the camera. However, such techniques do not provide the ability to control the vehicle with respect to a specific object and therefore are not directly applicable to scenarios such as tracking a specific target or perching. Thus, it is necessary to leverage different approaches.

In this chapter, we are concerned with the direct, vision-based control relative to a target object, not with the control of absolute position and orientation. This paradigm of relative control is called *Visual Servoing*. There is a foundational body of literature covering control for monocular visual servoing applications which discusses the differences between Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS) [6, 8, 50]. The key difference between these approaches is that with PBVS, the pose of the robot is estimated based on the image features, and the control law is expressed in the 3D Cartesian space. On the other hand, with IBVS, the control law is computed directly from features observed in the image [50]. Each has its benefits; for example, PBVS systems can use common filters in the MAV literature for the pose estimate while IBVS is considered to be more robust to camera calibration errors, making it appealing for low cost and low quality systems.

Position-based Visual Servoing is directly related to computer vision work on recovering pose from the projection of known structures, which is widely used in 3D-tracking and augmented reality. In the context of this chapter, approaches trying to recover pose from conics, cylinders, or quadrics are closely related. An extensive survey can be found in [21], but the closest and most specialized treatment is in [99] and [100]. The projection geometry of conics and the decomposition of conic projections in intrinsic and extrinsic parameters is introduced in [57, 107], and Augmented Reality based tracking using circles and cylinder projections was accomplished in [80].

In the IBVS literature, most research analyzes first order systems [123] and proves stability for fully actuated systems [14]. One of the earlier and most well known articles in visual servoing [30] presents interaction matrices for various geometric shapes and includes simulation and experimental results. Issues with singularities, local minima, ill-conditioned Jacobians, and comparisons between nominal and estimated Jacobians were discussed in [7]. Gravity compensation of a robot arm was considered within a visual servoing framework in [61]. Higher order considerations were proposed in [11, 12], which also argues for feed-forward control inputs and presents visual servoing as a "steering" problem. Stereo image based visual servoing was introduced in [59], which relaxed the requirement that the scale of the target was known. Recently, a Port-Hamiltonian approach has been used to formulate a general IBVS approach [78], however an improvement in accuracy over traditional IBVS approaches is not demonstrated. Further, works such as [42–44, 51] use a backstepping approach for control, while [129] and [127] use a geometric-based control approach.

With regard to MAVs, a vision-based approach was used to evaluate possible flat landing-site locations, however, this makes downward observation difficult [20]. An IBVS controller was used in [68] to land a quadrotor on a moving platform, but the controller assumed that the velocity in the world frame was known. Image based cooperative manipulation was demonstrated with simulations in [84]. The authors in [116] leverage a Structure from Motion (SFM) approach to estimate the parameters of a cylinder if the velocity of the camera is known. An IBVS approach was presented for visual inspection of vertical cylinders at fixed distances, but system requires a pilot or magnetometer to control the yaw and an additional sensor to estimate the height [108]. Finally, grasping of cylindrical objects is considered in [129] and [127], where the cylinder's axis is assumed to be perpendicular to gravity and an external motion capture system is used (in addition to onboard sensing) to control the motion along the cylinder axis.

In this chapter, we develop the first fully autonomous vision-based quadrotor for perching, providing experimental results and a formulation of control and second order system dynamics in the image plane. We address the full 3D visual servoing problem of grasping an

overhead cylinder, allowing the robot to hang from branches or poles. Specifically, several key contributions are presented:

- A second order dynamic model of features in a virtual image plane is derived using a diffeomorphism between image features and robot's location in 3D space.
- An IBVS control law is proposed in which the stability is proved via Lyapunov theory.
- The system is shown to be differentially flat with respect to a set of given outputs in a virtual image space, which enables a simple method for planning dynamically feasible trajectories directly in the image space.
- The control law and trajectory planning method are demonstrated through successful perches of a quadrotor on real-world objects.

The chapter is organized as follows. In Section 5.2 and Section 5.3, the geometric and dynamic models are developed, respectively. An IBVS control law is presented in Section 5.4, and a method for trajectory planning is developed in Section 5.5. Then, experimental results are reported in Section 5.6. Finally, we conclude the chapter in Section 5.7.

## 5.2 Geometric Model

Consider a cylinder with a known radius, $r$, and an unknown axis, $\mathbf{a} \in \mathbb{S}^2$ defined in the camera frame, $\mathcal{C}$. In this section, all vectors will be defined with respect to the camera frame unless otherwise noted with a superscript. We assume that the radius of the cylinder is known (perhaps using the approach in [116]), and we focus on the problem of estimating the pose of the cylinder in a camera-fixed frame. The closest concise treatment of a relative cylinder pose can be found in [100].

### 5.2.1 Projection of Quadric

Let $\mathbf{X} \in \mathbb{R}^3$, $M \in \mathbb{R}^{3\times3}$, $\mathbf{m} \in \mathbb{R}^3$, $\mu \in \mathbb{R}$, and

$$\mathbf{X}^T M \mathbf{X} + \mathbf{m}^T \mathbf{X} + \mu = 0 \tag{5.1}$$

be the equation of a quadric surface in $\mathbb{R}^3$ expressed in the coordinate system of a camera. Its projection $\mathbf{x} = (x, y, 1)$ to the image plane $Z = 1$ of a calibrated camera can be obtained by letting $\mathbf{X} = \lambda\mathbf{x}$ be the ray of projection and requiring that this ray is tangent to the surface. This means that the equation

$$\lambda^2 \mathbf{x}^T M \mathbf{x} + \lambda \mathbf{m}^T \mathbf{x} + \mu = 0 \tag{5.2}$$

must have exactly one solution for $\lambda$. For this to happen, the discriminant must vanish

$$(\mathbf{m}^T \mathbf{x})^2 - 4\mu \mathbf{x}^T M \mathbf{x} = 0. \tag{5.3}$$

Then, for any point $\mathbf{x}$ (recall that $\lambda\mathbf{x}$ is tangent to the surface) satisfying (5.3), we can recover its depth as

$$\lambda = -\frac{\mathbf{m}^T \mathbf{x}}{2\mathbf{x}^T M \mathbf{x}}. \tag{5.4}$$

Let us find the equation of a cylinder of arbitrary pose in camera coordinates. Let the cylinder be expressed as

$$X_o^2 + Y_o^2 = r^2 \tag{5.5}$$

in the object coordinates $(X_o, Y_o, Z_o) \in \mathbb{R}^3$. Then, in the camera frame, let the axis of the cylinder be the unit vector $\mathbf{a}$, and let the direction perpendicular to the plane spanned by the origin and the axis be the unit vector $\mathbf{b}$ as displayed in Figure 5.2. Finally, let the vector from the origin to the closest point on the axis be $\beta\mathbf{c}$ with $\|\mathbf{c}\| = 1$. Choosing the closest point of the axis as the origin of the object coordinate system, the transformation from the object to the camera becomes

$$\mathbf{X} = \begin{bmatrix} \mathbf{c} & \mathbf{b} & \mathbf{a} \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} + \beta\mathbf{c} \tag{5.6}$$

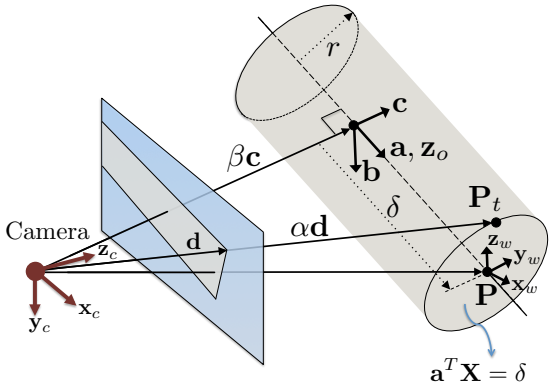



Figure 5.2: The geometry of the camera-cylinder system. The coordinate systems of the camera, cylinder object, and world are given by $(\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c)$, $(\mathbf{b}, \mathbf{c}, \mathbf{a})$, and $(\mathbf{x}_w, \mathbf{y}_w, \mathbf{z}_w)$, respectively. The axis of the cylinder $\mathbf{a}$ and the point on the axis closest to the camera, $\beta\mathbf{c}$, can be estimated directly from the lines representing the boundaries of the cylinder in the image. Further, we assume that the projection $\mathbf{d}$ of a ray passing through the camera and tangent to the cylinder at the $\mathbf{a}^T\mathbf{X} = \delta$ plane can be observed.

or, rearranging,

$$\begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} = \begin{bmatrix} \mathbf{c}^T\mathbf{X} - \beta \\ \mathbf{b}^T\mathbf{X} \\ \mathbf{a}^T\mathbf{X} \end{bmatrix}. \tag{5.7}$$

Then, the cylinder can be expressed in camera coordinates as

$$(\mathbf{c}^T\mathbf{X} - \beta)^2 + (\mathbf{b}^T\mathbf{X})^2 = r^2 \tag{5.8}$$

and following (5.3), we obtain the projection and factorize

$$0 = (\beta\mathbf{c}^T\mathbf{x})^2 - (\beta^2 - r^2)((\mathbf{c}^T\mathbf{x})^2 + (\mathbf{b}^T\mathbf{x})^2) \tag{5.9}$$

$$= (\sqrt{\beta^2 - r^2}\mathbf{b} + r\mathbf{c})^T\mathbf{x}\,(\sqrt{\beta^2 - r^2}\mathbf{b} - r\mathbf{c})^T\mathbf{x}, \tag{5.10}$$

which shows that the projection of a cylinder is the union of two lines. Let us write these two lines as

$$\mathbf{n}_1^T\mathbf{x} = 0 \quad \text{and} \quad \mathbf{n}_2^T\mathbf{x} = 0 \tag{5.11}$$

with $\mathbf{n}_{i=1,2}$ being the unit vectors

$$\mathbf{n}_{1,2} = \frac{1}{\beta}\left(\sqrt{\beta^2 - r^2}\mathbf{b} \pm r\mathbf{c}\right). \tag{5.12}$$

We see that $\mathbf{n}_{1,2}$ contains all the information about the orientation of the cylinder:

$$\mathbf{a} \sim \mathbf{n}_1 \times \mathbf{n}_2, \qquad \mathbf{b} \sim \mathbf{n}_1 + \mathbf{n}_2, \qquad \mathbf{c} \sim \mathbf{n}_1 - \mathbf{n}_2, \tag{5.13}$$

and the distance of the camera to the closest point $\beta$ can be recovered from the inner product

$$\mathbf{n}_1^T \mathbf{n}_2 = 1 - 2\frac{r^2}{\beta^2}, \tag{5.14}$$

which has a plausible geometric interpretation as $\cos\phi = 1 - 2\cos^2\frac{\phi}{2}$ where $\phi/2$ is the angle between $\mathbf{c}$ and the radial vector to the tangential point in the same plane.

**A World-Fixed Point**

The main objective is to determine the relative position of the robot, but also considering translation along the axis of the cylinder relative to some fixed point in the inertial frame. Thus, we are interested in determining the location of a fixed point, $\mathbf{P}$, that lies on the axis of the cylinder. Note that a tangent point cannot be used as the fixed point because it would move in the inertial frame based on the relative pose. We could, however, use points from features on the surface of the cylinder, but would then have to be concerned about preventing the feature from becoming occluded.

If we are given the projection $\mathbf{d}$ of a tangent point $\mathbf{P}_t$ as in Figure 5.2, we first recover the depth $\alpha$ of the point from (5.4)

$$\alpha = \frac{\beta \mathbf{c}^T \mathbf{d}}{(\mathbf{c}^T\mathbf{d})^2 + (\mathbf{b}^T\mathbf{d})^2} \tag{5.15}$$

The plane that is perpendicular to the cylinder axis and passes through $\delta\mathbf{a}$ and $\mathbf{P}_t$ can

be represented by

$$\mathbf{a}^T\mathbf{X} = \delta. \tag{5.16}$$

Since $\alpha\mathbf{d}$ is known from (5.15), we can obtain $\delta$ from

$$\delta = \mathbf{a}^T(\alpha\mathbf{d}) = \frac{(\mathbf{a}^T\mathbf{d})\beta\mathbf{c}^T\mathbf{d}}{(\mathbf{c}^T\mathbf{d})^2 + (\mathbf{b}^T\mathbf{d})^2}. \tag{5.17}$$

Knowing $\delta$, it is possible to compute $\mathbf{P}$ as

$$\mathbf{P} = \beta\mathbf{c} + \delta\mathbf{a}. \tag{5.18}$$

**An Attitude Estimate**

To control the robot, it is important to know the orientation relative to a fixed frame. We assume that we can estimate the gravity vector, $\mathbf{g}^{\mathcal{B}}$, in the robot body frame using the onboard IMU. The transformation from the robot frame to the camera frame is fixed and known by design, so we can determine the gravity vector in the camera frame, $\mathbf{g}^{\mathcal{C}}$. In addition, we have already concluded that we can estimate the axis of the cylinder, $\mathbf{a}$, also in the camera frame, which provides an anchor for the rotation about the vertical axis. If the cylinder is not vertical (*i.e.,* $\mathbf{g}^{\mathcal{C}} \times \mathbf{a} \neq 0$), then the rotation between the camera frame and the world frame is

$$R^{\mathcal{C}}_{\mathcal{W}} = \begin{bmatrix} \mathbf{x}^{\mathcal{C}}_{\mathcal{W}} & \mathbf{y}^{\mathcal{C}}_{\mathcal{W}} & \mathbf{z}^{\mathcal{C}}_{\mathcal{W}} \end{bmatrix} \tag{5.19}$$

where $R^{\mathcal{C}}_{\mathcal{W}}$ rotates vectors with coordinates in $\mathcal{W}$ to $\mathcal{C}$ and

$$\mathbf{z}^{\mathcal{C}}_{\mathcal{W}} = -\frac{\mathbf{g}^{\mathcal{C}}}{\|\mathbf{g}^{\mathcal{C}}\|}, \quad \mathbf{y}^{\mathcal{C}}_{\mathcal{W}} = \frac{\mathbf{z}^{\mathcal{C}}_{\mathcal{W}} \times \mathbf{a}}{\|\mathbf{z}^{\mathcal{C}}_{\mathcal{W}} \times \mathbf{a}\|}, \quad \mathbf{x}^{\mathcal{C}}_{\mathcal{W}} = \mathbf{y}^{\mathcal{C}}_{\mathcal{W}} \times \mathbf{z}^{\mathcal{C}}_{\mathcal{W}}. \tag{5.20}$$

### 5.2.2 Image Features in a Fixed-Orientation Virtual Frame

The observed features are in the camera frame, but we are interested in a fixed-orientation frame located at the Center of Mass (COM) in order to simplify the control and planning problems. In practice, this could be achieved using a gimbal that would keep the camera

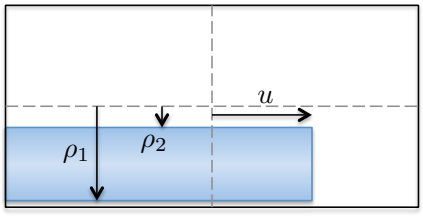



Figure 5.3: The features in the virtual image are the coordinates ($\rho_1$ and $\rho_2$) of the two lines and the coordinate, $u$, representing the projection of the $\mathbf{a}^T\mathbf{X} = \delta$ plane.

at a fixed orientation in the world frame or, as in this case, we can re-project the observed cylinder onto a virtual image based on the known orientation of the robot.

The virtual image plane is parallel to the axis of the cylinder, and its orientation can be expressed as

$$R_{\mathcal{V}}^{\mathcal{C}} = \begin{bmatrix} \mathbf{a} & \mathbf{y}_{\mathcal{W}}^{\mathcal{C}} & \mathbf{a} \times \mathbf{y}_{\mathcal{W}}^{\mathcal{C}} \end{bmatrix}. \tag{5.21}$$

In this case, we have formulated the virtual image plane as being in the $\mathbf{a}$-$\mathbf{y}_{\mathcal{W}}$ plane. However, if the user would prefer to operate beside (in contrast to above or below) the cylinder, a vertical plane would be more appropriate (*e.g.,* the $\mathbf{a}$-$\mathbf{z}_{\mathcal{W}}$ plane). The translational offset of the camera from the COM of the system is known in the camera frame, which allows us to determine

$$\mathbf{P}^{\mathcal{V}} = T_{\mathcal{C}}^{\mathcal{V}}(\mathbf{P}) \tag{5.22}$$

where $T_{\mathcal{C}}^{\mathcal{V}}$ is a known transformation. The feature vector in the virtual image (see Figure 5.3) is

$$\mathbf{s} \equiv \begin{bmatrix} \rho_1 \\ \rho_2 \\ u \end{bmatrix} = \Gamma\left(\mathbf{P}^{\mathcal{V}}, r\right) \tag{5.23}$$

where $\rho_1$ and $\rho_2$ can be computed from the lines of the cylinder and

$$u = \frac{\mathbf{e}_1^T \mathbf{P}^{\mathcal{V}}}{\mathbf{e}_3^T \mathbf{P}^{\mathcal{V}}} \tag{5.24}$$

where $\mathbf{e}_i$ denotes the $i^{th}$ standard basis vector. Now that we have computed the image features in the virtual image, we will explore their dynamics to provide insight for control.

## 5.3 Dynamic Model in the Virtual Frame

As the robot moves, the image features will also move in the virtual image. Since we are ultimately interested in driving the features to a desired goal, it is important to understand how the control inputs of the system influence the dynamics and can be used to manipulate the image features. This will enable simulations as well as provide insight for the development of an image based control law.

The velocities of the image features are (see [50])

$$\dot{\mathbf{s}} = \frac{\partial \Gamma\left(\mathbf{P}^{\mathcal{V}}\right)}{\partial \mathbf{P}^{\mathcal{V}}} \dot{\mathbf{P}}^{\mathcal{V}}. \tag{5.25}$$

Let the position of the robot in the world frame be $\mathbf{x}_q$ so that the velocity of $\mathbf{P}$ can be computed to be

$$\dot{\mathbf{P}}^{\mathcal{V}} = -R_{\mathcal{C}}^{\mathcal{V}} R_{\mathcal{W}}^{\mathcal{C}} \dot{\mathbf{x}}_q. \tag{5.26}$$

Then, we can express the image feature velocities in terms of the robot velocity and $\mathbf{P}^{\mathcal{V}}$

$$\dot{\mathbf{s}} = -\frac{\partial \Gamma\left(\mathbf{P}^{\mathcal{V}}\right)}{\partial \mathbf{P}^{\mathcal{V}}} R_{\mathcal{C}}^{\mathcal{V}} R_{\mathcal{W}}^{\mathcal{C}} \dot{\mathbf{x}}_q \equiv J \dot{\mathbf{x}}_q. \tag{5.27}$$

We assume that $R_{\mathcal{W}}^{\mathcal{V}}$ is constant (*i.e.,* the cylinder's orientation is fixed in the world) and known since $R_{\mathcal{W}}^{\mathcal{C}}$ is known from (5.19) and $R_{\mathcal{C}}^{\mathcal{V}}$ is known from (5.21). Since we have assumed that $r > 0$ and that the robot cannot pass through the cylinder, it can be shown that the Jacobian is nonsingular.

The translational dynamics of a quadrotor in the world frame can be expressed as

$$m\ddot{\mathbf{x}}_q = -mg\mathbf{e}_3 + f R_{\mathcal{B}}^{\mathcal{W}} \mathbf{e}_3 \tag{5.28}$$

where $m$ is the mass of the vehicle, $g$ is gravitational acceleration, $\mathbf{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, $f$ is the net thrust, which only acts vertically (*i.e.,* in the $\mathbf{e}_3$ direction) in the body frame of the robot, and $R_{\mathcal{B}}^{\mathcal{W}}$ is the orientation of the vehicle's body with respect to the word frame.

Further, the angular dynamics are given by

$$\dot{R}_{\mathcal{B}}^{\mathcal{W}} = R_{\mathcal{B}}^{\mathcal{W}} \hat{\mathbf{\Omega}} \tag{5.29}$$

$$\mathcal{I}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathcal{I}\mathbf{\Omega} = \mathbf{M} \tag{5.30}$$

where $\mathbf{\Omega} \in \mathbb{R}^3$ is the body-frame angular velocity of the robot, $\mathcal{I}$ is the inertial tensor, $\mathbf{M} \in \mathbb{R}^3$ is the control moments, $\hat{\cdot} : \mathbb{R}^3 \mapsto \mathfrak{so}(3)$ is the "hat" map defined such that $\hat{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$, and $\mathfrak{so}(3)$ is the Lie algebra of SO(3), the 3D rotation group.

Using (5.27), the acceleration of the robot can be expressed in terms of $J$, $J^{-1}$, $\mathbf{s}$, and their derivatives

$$m\left(J^{-1}\ddot{\mathbf{s}} + \dot{J^{-1}}\dot{\mathbf{s}}\right) = fR_{\mathcal{B}}^{\mathcal{W}}\mathbf{e}_3 - mg\mathbf{e}_3. \tag{5.31}$$

Rearranging, the dynamics of the image features are

$$\ddot{\mathbf{s}} = \frac{1}{m} J\left(fR_{\mathcal{B}}^{\mathcal{W}}\mathbf{e}_3 - mg\mathbf{e}_3 - m\dot{J^{-1}}\dot{\mathbf{s}}\right) \tag{5.32}$$

and can be used for simulation and the development of the control law in the next section.

## 5.4 An Image-Based Control Law

The goal of our controller is to drive observed image features, $\mathbf{s}$, to desired values, $\mathbf{s}_{des}$. In this section, we first show exponential stability of the closed-loop attitude dynamics. Next, we determine bounds on the translation Lyapunov candidate and its derivative. Finally, we show that the equilibrium of the full closed-loop system is asymptotically stable. For ease of notation, in the rest of this section, let $R \equiv R_{\mathcal{B}}^{\mathcal{W}}$.

### 5.4.1 Stability of the Closed-Loop Attitude Dynamics

Let the commanded moments for the attitude controller be

$$\mathbf{M} = -k_R\mathbf{e}_R - k_\Omega\mathbf{e}_\Omega + \mathbf{\Omega} \times \mathcal{I}\mathbf{\Omega} \tag{5.33}$$

where $k_R, k_\Omega \in \mathbb{R}_{>0}$ are positive gains. Also, $\mathbf{e}_R$ and $\mathbf{e}_\Omega$ represent the attitude errors

$$\mathbf{e}_R = \frac{1}{2}\left(R_{des}^T R - R^T R_{des}\right)^\vee \tag{5.34}$$

$$\mathbf{e}_\Omega = \boldsymbol{\Omega} - R^T R_{des}\boldsymbol{\Omega}_{des} \tag{5.35}$$

where $\cdot^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$ is the opposite of the hat map and the subscript "des" indicates a desired value.

**Proposition 2.** *If the initial attitude error is less than* $180°$*, the closed-loop attitude dynamics converge exponentially to the equilibrium,* $R = R_{des}$ *and* $\boldsymbol{\Omega} = \boldsymbol{\Omega}_{des}$*.*

*Proof.* We define

$$\Psi(R, R_d) = \frac{1}{2}\operatorname{tr}\left[I - R_{des}^T R\right] \tag{5.36}$$

and propose a Lyapunov candidate

$$\mathcal{V}_R = \frac{1}{2}\mathbf{e}_\Omega \cdot \mathcal{I}\mathbf{e}_\Omega + k_R\,\Psi(R, R_d) + c_2\mathbf{e}_R \cdot \mathbf{e}_\Omega, \tag{5.37}$$

with $c_2$ being a positive scalar. Let $\mathbf{z}_\theta = [\|\mathbf{e}_R\|, \|\mathbf{e}_\Omega\|]^T$. Then, following the results in [69], we can show that

$$\mathbf{z}_\theta^T M_\theta \mathbf{z}_\theta \leq \mathcal{V}_R \leq \mathbf{z}_\theta^T M_\Theta \mathbf{z}_\theta, \tag{5.38}$$

$$\dot{\mathcal{V}}_R \leq -\mathbf{z}_\theta^T W_\theta \mathbf{z}_\theta, \tag{5.39}$$

where $M_\theta, M_\Theta, W_\theta \in \mathbb{R}^{2\times 2}$ are positive definite. We can also show that the equilibrium of the closed-loop attitude dynamics is exponentially stable if the initial attitude error is less than $180°$. □

### 5.4.2 Translational Error Dynamics

Let the desired thrust vector in the world frame be

$$\mathbf{f}_{des} = m\left(g\mathbf{e}_3 + J^{-1}\left(k_x\mathbf{e}_s + k_v\dot{\mathbf{e}}_s + \ddot{\mathbf{s}}_{des}\right) + \dot{J^{-1}}\dot{\mathbf{s}}\right) \tag{5.40}$$

where

$$\mathbf{e}_s = \mathbf{s}_{des} - \mathbf{s}, \quad \dot{\mathbf{e}}_s = \dot{\mathbf{s}}_{des} - \dot{\mathbf{s}} \tag{5.41}$$

are the position and velocity errors in the image coordinates, and $k_x, k_v \in \mathbb{R}_{>0}$ are positive gains.

Note that $R$ can be computed as $R = R_{\mathcal{C}}^{\mathcal{W}} R_{\mathcal{B}}^{\mathcal{C}}$ where $R_{\mathcal{B}}^{\mathcal{C}}$ is a known, constant rotation and $R_{\mathcal{C}}^{\mathcal{W}}$ can be determined from the real image as in (5.19). Rearranging (5.31), the dynamics of the image features are

$$\ddot{\mathbf{s}} = \frac{1}{m} J \left( f R \mathbf{e}_3 - mg\mathbf{e}_3 - m\dot{J^{-1}}\dot{\mathbf{s}} \right). \tag{5.42}$$

Using (5.27), we can determine the image errors

$$\ddot{\mathbf{e}}_s = \ddot{\mathbf{s}}_{des} - \frac{1}{m} J \left( f R \mathbf{e}_3 - mg\mathbf{e}_3 - m\dot{J^{-1}}\dot{\mathbf{s}} \right), \tag{5.43}$$

so that

$$m\ddot{\mathbf{e}}_s = m\ddot{\mathbf{s}}_{des} - fJR\mathbf{e}_3 + Jmg\mathbf{e}_3 + mJ\dot{J^{-1}}\dot{\mathbf{s}}. \tag{5.44}$$

Defining

$$\mathbf{X} = J \frac{f}{\mathbf{e}_3^T R_c^T R \mathbf{e}_3} \left( R_c \mathbf{e}_3 - \left( \mathbf{e}_3^T R_c^T R \mathbf{e}_3 \right) R \mathbf{e}_3 \right), \tag{5.45}$$

the error dynamics become

$$m\ddot{\mathbf{e}}_s = m\ddot{\mathbf{s}}_{des} - J \left( \frac{f}{\mathbf{e}_3^T R_c^T R \mathbf{e}_3} R_c \mathbf{e}_3 \right) + \mathbf{X} + Jmg\mathbf{e}_3 + mJ\dot{J^{-1}}\dot{\mathbf{s}}. \tag{5.46}$$

Next, let

$$f = \mathbf{f}_{des} \cdot R\mathbf{e}_3, \tag{5.47}$$

and let the commanded attitude be defined such that

$$R_c \mathbf{e}_3 = \frac{\mathbf{f}_{des}}{\|\mathbf{f}_{des}\|}. \tag{5.48}$$

Then, from the previous two equations, we have

$$f = \|\mathbf{f}_{des}\| \, \mathbf{e}_3^T R_c^T R \mathbf{e}_3. \tag{5.49}$$

Substituting this into (5.46) and observering from (5.48) that $\mathbf{f}_{des} = \|\mathbf{f}_{des}\| \, R_c \mathbf{e}_3$, we have,

$$\begin{aligned} m\ddot{\mathbf{e}}_s =& m\ddot{\mathbf{s}}_{des} - J\left(\frac{\|\mathbf{f}_{des}\| \, \mathbf{e}_3^T R_c^T R \mathbf{e}_3}{\mathbf{e}_3^T R_c^T R \mathbf{e}_3} R_c \mathbf{e}_3\right) + \mathbf{X} + Jmg\mathbf{e}_3 + mJ\dot{J^{-1}}\dot{\mathbf{s}} && (5.50) \\ =& m\ddot{\mathbf{s}}_{des} - J\left(\|\mathbf{f}_{des}\| \, R_c \mathbf{e}_3\right) + \mathbf{X} + Jmg\mathbf{e}_3 + mJ\dot{J^{-1}}\dot{\mathbf{s}} && (5.51) \\ =& m\ddot{\mathbf{s}}_{des} - J\mathbf{f}_{des} + \mathbf{X} + Jmg\mathbf{e}_3 + mJ\dot{J^{-1}}\dot{\mathbf{s}}. && (5.52) \end{aligned}$$

Then, using (5.40), the error dynamics become

$$m\ddot{\mathbf{e}}_s = -k_x \mathbf{e}_s - k_v \dot{\mathbf{e}}_s + \mathbf{X} \tag{5.53}$$

where $\mathbf{X}$ is defined in (5.45).

### 5.4.3 Properties of the Closed-Loop Translational Dynamics

We start with the Lyapunov candidate from [70]

$$\mathcal{V}_s = \frac{1}{2} k_x \|\mathbf{e}_s\|^2 + \frac{1}{2} m \|\dot{\mathbf{e}}_s\|^2 + c_1 \mathbf{e}_s \cdot \dot{\mathbf{e}}_s \tag{5.54}$$

and define $\mathbf{z}_s = \begin{bmatrix} \|\mathbf{e}_s\| & \|\dot{\mathbf{e}}_s\| \end{bmatrix}^T$.

**Remark 1.** *Upper and lower bounds of $\mathcal{V}_s$ can be expressed as*

$$\mathbf{z}_s^T M_s \mathbf{z}_s \leq \mathcal{V}_s \leq \mathbf{z}_s^T M_S \mathbf{z}_s, \tag{5.55}$$

*where $M_s, M_S \in \mathbb{R}^{2\times 2}$ are*

$$M_s = \frac{1}{2}\begin{bmatrix} k_x & -c_1 \\ -c_1 & m \end{bmatrix}, \quad M_S = \frac{1}{2}\begin{bmatrix} k_x & c_1 \\ c_1 & m \end{bmatrix}. \tag{5.56}$$

Next, we compute

$$\dot{\mathcal{V}}_s = k_x \left(\dot{\mathbf{e}}_s \cdot \mathbf{e}_s\right) + m \left(\ddot{\mathbf{e}}_s \cdot \dot{\mathbf{e}}_s\right) + c_1 \left(\mathbf{e}_s \cdot \ddot{\mathbf{e}}_s + \dot{\mathbf{e}}_s \cdot \dot{\mathbf{e}}_s\right), \tag{5.57}$$

and incorporate the error dynamics from (5.53),

$$\dot{\mathcal{V}}_s = \frac{c_1 k_x}{m} \|\mathbf{e}_s\|^2 + (k_v - c_1) \|\dot{\mathbf{e}}_s\|^2 + c_1 \frac{k_v}{m} \left(\mathbf{e}_s \cdot \dot{\mathbf{e}}_s\right) + \mathbf{X} \cdot \left(\frac{c_1}{m} \mathbf{e}_s + \dot{\mathbf{e}}_s\right). \tag{5.58}$$

Now, we establish a bound on $\mathbf{X}$. From (5.45),

$$\mathbf{X} = J \frac{f}{\mathbf{e}_3^T R_c^T R \mathbf{e}_3} \left(\left(\mathbf{e}_2^T R_c^T R \mathbf{e}_3\right) R \mathbf{e}_3 - R_c \mathbf{e}_3\right) \tag{5.59}$$

$$\|\mathbf{X}\| \leq \|J\| \left\| \frac{\|\mathbf{f}_{des}\| \, R_c \mathbf{e}_3 \cdot R \mathbf{e}_3}{R_c \mathbf{e}_3 \cdot R \mathbf{e}_3} \right\| \|e_R\| \tag{5.60}$$

$$\leq \|J\| \, \|\mathbf{f}_{des}\| \, \|e_R\| \tag{5.61}$$

$$\leq \|J\| \, m \left\| g e_3 + J \left(k_x \mathbf{e}_s + k_v \dot{\mathbf{e}}_s + \ddot{\mathbf{s}}_{des}\right) + \dot{J^{-1}} \dot{\mathbf{s}} \right\| \|e_R\| \tag{5.62}$$

$$\leq \left(k_x' \|\mathbf{e}_s\| + k_v' \|\dot{\mathbf{e}}_s\| + B\right) \|e_R\|, \tag{5.63}$$

where $k_x', k_v', B$ are as defined as

$$k_x' = m \|J\| \, \|J\| \, k_x, \tag{5.64}$$

$$k_v' = m \left(\|J\| \, \|J\| \, k_v + \left\| \dot{J^{-1}} \right\| \right), \tag{5.65}$$

$$B = m \|J\| \left(g + \|J\| \, \|\ddot{\mathbf{s}}_{des}\| + \left\| \dot{J^{-1}} \right\| \|\dot{\mathbf{s}}_{des}\| \right), \tag{5.66}$$

and from [70], $0 \leq \|e_R\| \leq 1$.

Next we will show that there exists positive constants $\gamma_1, \gamma_2, \gamma_3$ s.t., $\|J\| \leq \gamma_1$, $\left\|J^{-1}\right\| \leq \gamma_2$, and $\left\|\dot{J^{-1}}\right\| \leq \gamma_3$.. Since $\Gamma$ is smooth (we only require $C^2$ here), $J$ is smooth on the closed set $S$. This implies $J$ is bounded on $S$, i.e., $\exists \, \gamma_1 > 0$, s.t. $\|J\| < \gamma_1$. Next, since $J$ is smooth and nonsingular on $S$, the inverse is well defined and is smooth on $S$, which implies $J^{-1}$ is bounded on $S$, i.e., $\exists \, \gamma_2 > 0$, s.t. $\left\|J^{-1}\right\| < \gamma_2$. Next, observe that $\frac{d}{dt} J^{-1}(\mathbf{x}_q) =$

$\frac{\partial}{\partial \mathbf{x}_q} J^{-1}(\mathbf{x}_q)\dot{\mathbf{x}}_q$ is a composition of smooth functions on $S$, implying that it is bounded on $S$, i.e., $\exists\, \gamma_3 > 0$, s.t. $\left\| \dot{J^{-1}} \right\| < \gamma_3$. Then, similar to [118], we can express $\dot{\mathcal{V}}_v$ as

$$\begin{aligned}
\dot{\mathcal{V}}_s &= \frac{c_1 k_x}{m} \|\mathbf{e}_s\|^2 + (k_v - c_1) \|\dot{\mathbf{e}}_s\|^2 + \frac{c_1 k_v}{m} (\mathbf{e}_s \cdot \dot{\mathbf{e}}_s) + \mathbf{X} \cdot \left( \frac{c_1}{m} \mathbf{e}_s + \dot{\mathbf{e}}_s \right) \\
&\leq - \begin{bmatrix} \|\mathbf{e}_s\| & \|\dot{\mathbf{e}}_s\| \end{bmatrix} W_{s_1} \begin{bmatrix} \|\mathbf{e}_s\| \\ \|\dot{\mathbf{e}}_s\| \end{bmatrix} + k_p' \|\mathbf{e}_s\| \|e_R\| \left( \frac{c_1}{m} \|\mathbf{e}_s\| + \|\dot{\mathbf{e}}_s\| \right) \\
&\quad + k_v' \|\dot{\mathbf{e}}_s\| \|e_R\| \left( \frac{c_1}{m} \|\mathbf{e}_s\| + \|\dot{\mathbf{e}}_s\| \right) + B \|e_R\| \left( \frac{c_1}{m} \|\mathbf{e}_s\| + \|\dot{\mathbf{e}}_s\| \right).
\end{aligned} \tag{5.67}$$

**Remark 2.** *An upper bound on $\dot{\mathcal{V}}_s$ can be written as*

$$\dot{\mathcal{V}}_s \leq -\mathbf{z}_s^T W_s \mathbf{z}_s + \mathbf{z}_s^T W_{s\theta} \mathbf{z}_\theta, \tag{5.68}$$

*where*

$$W_{s_1} = \begin{bmatrix} \frac{c_1 k_x}{m} & \frac{c_1 k_v}{2m} \\ \frac{c_1 k_v}{2m_q} & k_v - c_1 \end{bmatrix}, \quad W_{s\theta} = \begin{bmatrix} \frac{c_1}{m} B & 0 \\ B & 0 \end{bmatrix}, \tag{5.69}$$

$$W_{s_2} = \begin{bmatrix} \frac{c_1 \alpha k_x'}{m} & \frac{\alpha}{2} \left( \frac{c_1}{m} k_v' + k_x' \right) \\ \frac{\alpha}{2} \left( \frac{c_1}{m} k_v' + k_x' \right) & \alpha k_v' \end{bmatrix}, \tag{5.70}$$

$$W_s = W_{s_1} - W_{s_2}. \tag{5.71}$$

Since $W_s = (W_s)^T$ and $W_s \in \mathbb{R}^{2\times 2}$, it is sufficient to show that $\det(W_s) > 0$ and $W_s(1,1) > 0$ in order to claim that $W_s > 0$. Then, assuming $(1 - \alpha\gamma_1\gamma_2 m) > 0$, we have $w_{11} > 0$. This is reasonable since $\alpha$ is a functional on the attitude error such that $\alpha \in [0, 1]$. Thus, this assumption is simply a bound on the attitude error. The determinant can be expressed as a quadratic function of $k_v$ such that

$$\det(W_s) = \beta_0 + \beta_1 k_v + \beta_2 k_v^2, \tag{5.72}$$

and $\beta_i$ is a function of $c_1$, $k_x$, $\gamma_1$, $\gamma_2$, $\gamma_3$, and $m$. The critical point of the quadratic occurs

when

$$k_v = \frac{k_x m}{c_1} + \frac{k_x m + \alpha c_1 \gamma_1 \gamma_3}{c_1 \left(1 - \alpha \gamma_1 \gamma_2 m\right)}, \tag{5.73}$$

and has a value of

$$\det(W_s) = \frac{k_x \left(1 - \alpha \gamma_1 \gamma_2 m\right) \left(k_x m - c_1^2\right)}{m}. \tag{5.74}$$

Since we have assumed $(1 - \alpha\gamma_1\gamma_2 m) > 0$, we see that $k_v > 0$, $\det(W_s) > 0$, and therefore $W_s > 0$.

### 5.4.4 Stability of the Combined System

**Proposition 3.** *The equilibrium of the closed-loop system is asymptotically stable.*

*Proof.* We consider the combined Lyapunov candidate for the translational and angular error dynamics, $\mathcal{V} = \mathcal{V}_s + \mathcal{V}_R$. From (5.38) and (5.55), we have,

$$\mathbf{z}_s^T M_s \mathbf{z}_s + \mathbf{z}_\theta^T M_\theta \mathbf{z}_\theta \leq \mathcal{V} \leq \mathbf{z}_\theta^T M_\Theta \mathbf{z}_\theta + \mathbf{z}_s^T M_S \mathbf{z}_s. \tag{5.75}$$

Further, we see that

$$\dot{\mathcal{V}} \leq -\mathbf{z}_s^T W_s \mathbf{z}_s + \mathbf{z}_s^T W_{s\theta} \mathbf{z}_\theta - \mathbf{z}_\theta^T W_\theta \mathbf{z}_\theta, \tag{5.76}$$

$$\leq -\lambda_{min}(W_s) \left\|\mathbf{z}_s\right\|^2 + \left\|W_{s\theta}\right\| \left\|\mathbf{z}_s\right\| \left\|\mathbf{z}_\theta\right\| - \lambda_{min}(W_\theta) \left\|\mathbf{z}_\theta\right\|^2. \tag{5.77}$$

Suppose that we choose positive constants $c_1$, $k_x$, $k_v$, $k_R$ such that

$$k_x > \frac{c_1^2}{m}, \tag{5.78}$$

$$\lambda_{min}\left(W_\theta\right) > \frac{4\|W_{s\theta}\|^2}{\lambda_{min}\left(W_s\right)}. \tag{5.79}$$

Then, we have $\dot{\mathcal{V}} < 0$, ensuring the equilibrium of the closed-loop system to be asymptotically stable. □

## 5.5 Trajectory Planning

While the controller does guarantee stability, perching performance can be further improved by varying the desired setpoints in time ($\mathbf{s}_{des}(t)$) in a feasible manner. For a dynamic system, it is expected that a smooth transition between states (*e.g.,* not a step input) is required, but with the state vector represented by image features, the exact requirements are not obvious. In the rest of this section, we present a planning method that clarifies the requirements for a dynamically feasible trajectory in the image feature space and also provides a method for minimizing the inputs to the system after being expressed as a *differentially flat* system.

Let a set of proposed flat outputs be

$$\mathcal{Y} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ u \\ \psi \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ \psi \end{bmatrix} \tag{5.80}$$

where $\mathbf{s}$ is the image feature vector and $\psi$ is the yaw angle describing a rotation about the vertical axis of the inertial frame, $\mathbf{z}_w$. We will show that $\mathcal{Y}$ is a valid set of flat outputs for our system and that trajectories can be planned for $\mathcal{Y}(t)$ such that they are dynamically feasible. From a given trajectory, the thrust can be expressed considering (5.31) as

$$f = \left\| mg\mathbf{e}_3 + m\left(J^{-1}\ddot{\mathbf{s}} + \dot{J^{-1}}\dot{\mathbf{s}}\right)\right\| \tag{5.81}$$

which is dependent only on $\mathcal{Y}$ since $J$ is a function of $\mathcal{Y}$ (assuming $R_{\mathcal{W}}^{\mathcal{V}}$ is known and constant). Since the orientation of the robot can be expressed as

$$R_{\mathcal{B}}^{\mathcal{W}} = \begin{bmatrix} \mathbf{x}_{\mathcal{B}}^{\mathcal{W}} & \mathbf{y}_{\mathcal{B}}^{\mathcal{W}} & \mathbf{z}_{\mathcal{B}}^{\mathcal{W}} \end{bmatrix}, \tag{5.82}$$

the $R_{\mathcal{B}}^{\mathcal{W}}\mathbf{e}_3$ term in (5.31) is equal to $\mathbf{z}_{\mathcal{B}}^{\mathcal{W}}$ and the dynamic model can be rewritten accordingly.

Since $f \in \mathbb{R}$ and, by definition, $\left\|\mathbf{z}_{\mathcal{B}}^{\mathcal{W}}\right\| = 1$, we can solve for $\mathbf{z}_{\mathcal{B}}^{\mathcal{W}}$:

$$\mathbf{z}_{\mathcal{B}}^{\mathcal{W}} = \frac{m\left(J^{-1}\ddot{\mathbf{s}} + \dot{J^{-1}}\dot{\mathbf{s}}\right) + mg\mathbf{e}_3}{\left\|m\left(J^{-1}\ddot{\mathbf{s}} + \dot{J^{-1}}\dot{\mathbf{s}}\right) + mg\mathbf{e}_3\right\|}. \tag{5.83}$$

Defining a vector in the world, $\mathbf{x}_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \end{bmatrix}^T$, we can write the other two components of the rotation matrix $R_{\mathcal{B}}^{\mathcal{W}}$ as

$$\mathbf{y}_{\mathcal{B}}^{\mathcal{W}} = \frac{\mathbf{z}_{\mathcal{B}}^{\mathcal{W}} \times \mathbf{x}_\psi}{\left\|\mathbf{z}_{\mathcal{B}}^{\mathcal{W}} \times \mathbf{x}_\psi\right\|}, \quad \mathbf{x}_{\mathcal{B}}^{\mathcal{W}} = \mathbf{y}_{\mathcal{B}}^{\mathcal{W}} \times \mathbf{z}_{\mathcal{B}}^{\mathcal{W}}. \tag{5.84}$$

Since $\mathbf{z}_{\mathcal{B}}^{\mathcal{W}}$ and $\mathbf{x}_\psi$ are dependent on the flat outputs and the corresponding derivatives, we have shown that the body orientation is dependent on those quantities.

Next, we demonstrate that the angular velocity in the body frame is also dependent on the chosen set of flat outputs. Taking the derivative of (5.31), we obtain

$$m\left(2\dot{J^{-1}}\ddot{\mathbf{s}} + J^{-1}\mathbf{s}^{(3)} + \dot{J^{-1}}\dot{\mathbf{s}}\right) = \boldsymbol{\Omega}^{\mathcal{W}} \times f\mathbf{z}_{\mathcal{B}}^{\mathcal{W}} + \dot{f}\mathbf{z}_{\mathcal{B}}^{\mathcal{W}}, \tag{5.85}$$

and an inner product with $\mathbf{z}_B$ reveals that

$$\dot{f} = m\left(2\dot{J^{-1}}\ddot{\mathbf{s}} + J^{-1}\mathbf{s}^{(3)} + \dot{J^{-1}}\dot{\mathbf{s}}\right) \cdot \mathbf{z}_{\mathcal{B}}^{\mathcal{W}}. \tag{5.86}$$

Similarly, solving (5.85) for the $\boldsymbol{\Omega}^{\mathcal{W}} \times \mathbf{z}_{\mathcal{B}}^{\mathcal{W}}$ term and independently projecting onto $\mathbf{x}_{\mathcal{B}}^{\mathcal{W}}$ and $\mathbf{y}_{\mathcal{B}}^{\mathcal{W}}$ provides the first two terms of $\boldsymbol{\Omega}^{\mathcal{B}}$ by leveraging the circular shift property of the scalar triple product and the fact that $\boldsymbol{\Omega}^{\mathcal{W}} = R_{\mathcal{B}}^{\mathcal{W}}\boldsymbol{\Omega}^{\mathcal{B}}$. Then, the third component of $\boldsymbol{\Omega}^{\mathcal{B}}$ can be determined using an appropriate parameterization of $R_{\mathcal{B}}^{\mathcal{W}}$ and because $\dot{\psi}$ is known.

With another derivative, the angular acceleration of the robot appears as a function of $\mathcal{Y}$ (and higher derivatives) and allows for the computation of the control moments from (5.30). Thus, the control inputs are dependent on $\mathbf{s}^{(2)}, \mathbf{s}^{(3)}, \mathbf{s}^{(4)}$, $\dot{\psi}$, and $\ddot{\psi}$, and the dynamic system could be rewritten as a chain of integrators for each of the elements of $\mathcal{Y}$ with the appropriate derivative as the input to the flat system. For this reason, we plan trajectories

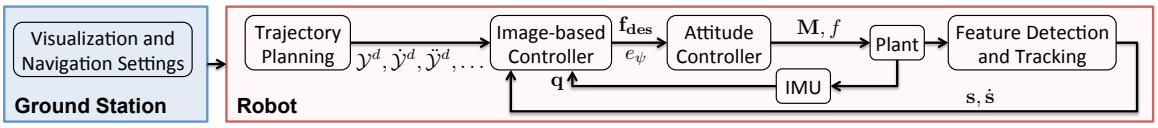



Figure 5.4: The architecture for planning, control, and estimation. The ground station is only responsible for visualization and sending high level commands. A desired trajectory $(\mathcal{Y}^d, \dot{\mathcal{Y}}^d, \ddot{\mathcal{Y}}^d, \ldots)$ is computed and sent to the IBVS controller. The controller uses IMU feedback at 200 Hz (in the form of a quaternion, $\mathbf{q}$) and image feature feedback at 75 Hz to compute a desired force $\mathbf{f}_{des}$ and a yaw error $e_\psi$, which are sent to the attitude controller. Finally, the attitude controller computes the necessary moment control inputs $\mathbf{M}$ and the thrust $f$.

that minimize the fourth derivative (*i.e.,* snap) of the image features and of the second derivative of the yaw, $\psi$.

The trajectory planning problem can be formulated as a Quadratic Program (QP) with the requirement that the trajectories are $\mathcal{C}^4$ (or $\mathcal{C}^2$ for $\psi$) and solved using the method outlined in [85], which allows for the implementation of constraints formulated in terms of the flat outputs.

## 5.6 Experimental Results

The experiments were conducted in the GRASP Lab [91] at the University of Pennsylvania using an Ascending Technologies Hummingbird quadrotor. An onboard computer (ODROID-XU3 from Hardkernel) handles all image processing and control, and the onboard sensors compose of the quadrotor's IMU and a Matrix Vision mvBlueFOX-MLC camera with a 98° horizontal and a 73° vertical field of view. The image processing and control use ROS, and the ODROID-XU3 is able to provide a rectified $752 \times 480$ image at frame-rate. A schematic and more detailed description of the system is provided in Figure 5.4. As depicted in the figure, all of the visual processing, control, estimation, and planning is done onboard without human intervention or an external motion capture system. The user can remotely change navigation and visualization settings from a base station.

We have intentionally pointed the camera upward because it more naturally keeps the cylinder within the field of view (see the discussion in Figure 5.5). Additionally, the robot is equipped with an upward-facing 1-DOF gripper, depicted in Figure 5.6, which is actuated using a small servo.

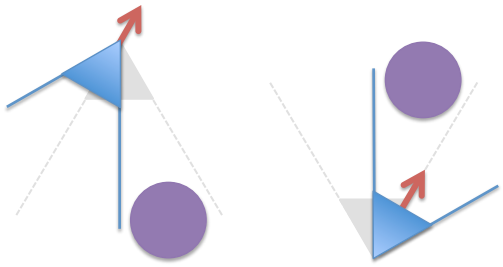


Figure 5.5: A cartoon to motivate the orientation of the camera and gripper on a perching robot. Consider the case when the goal is to keep the robot aligned with the cylinder, but the robot starts slightly to the left (gray). Then, the robot must rotate clockwise to accelerate towards the right (blue). The red arrow indicates the desired direction of the thrust and also the desired direction of the $\mathbf{z}_{\mathcal{B}}$ axis. The left hand side depicts the camera and field of view if the camera is facing downward. When the camera faces the same direction as the thrust (*i.e.,* upward), the robot more naturally keeps the cylinder in the field of view (right). In particular, this is important when maneuvering close to a cylinder such as when perching.

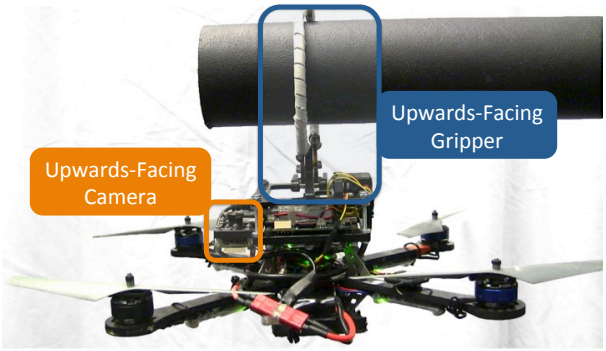



Figure 5.6: The robot has an upward-facing camera and gripper to enable perching on objects above the robot.

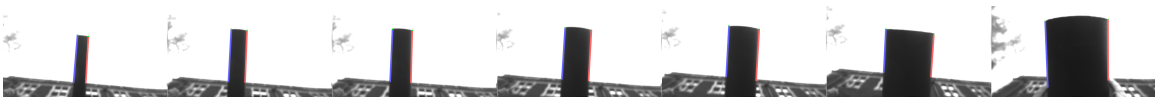


Figure 5.7: A sample image sequence from the onboard camera while perching. The time increases from left to right. As the robot gets closer to the cylinder, the cylinder becomes larger in the image. The blue (left) and red (right) lines indicate the detected lines.

### 5.6.1 Visual Processing

Efficient image processing is crucial since the algorithm must run at a sufficient rate for closed loop control. A sample of the result is given in Figure 5.7. The image processing algorithm is mainly divided into two steps: 1) Line detection and 2) Line tracking.

#### Line Detection

The goal of the first task, which is executed only at the first frame, is to reliably identify the lines representing the image of the cylinder. To accomplish this, the detection algorithm leverages the OpenCV library and a Hough transform [82] to detect lines in the image. From the set of lines detected, we identify two with similar length and orientation while enforcing a minimum distance between the two lines.

#### Line Tracking

In the second task, the two detected lines are tracked in the image. We use the Visual Servoing Platform (ViSP) [79] library to track the moving edges identified in the previous step. Each sample point along a line is tracked from one image to another along the normal to the edge passing through the point. To improve real-time image processing and control, two threads are instantiated to simultaneously track the two lines, which leverages multiple processor cores. If the tracking is lost, the two steps are repeated again. The cylinder tracking code runs onboard at 75 Hz, and the computed parameters in the virtual plane are filtered to obtain velocity estimates.

### 5.6.2 Trajectory Planning and Perching

Now, we leverage the trajectory planning in Section 5.5 to smoothly transition between states. For experimental results of a sample trajectory, see Figure 5.8. The image coordi-

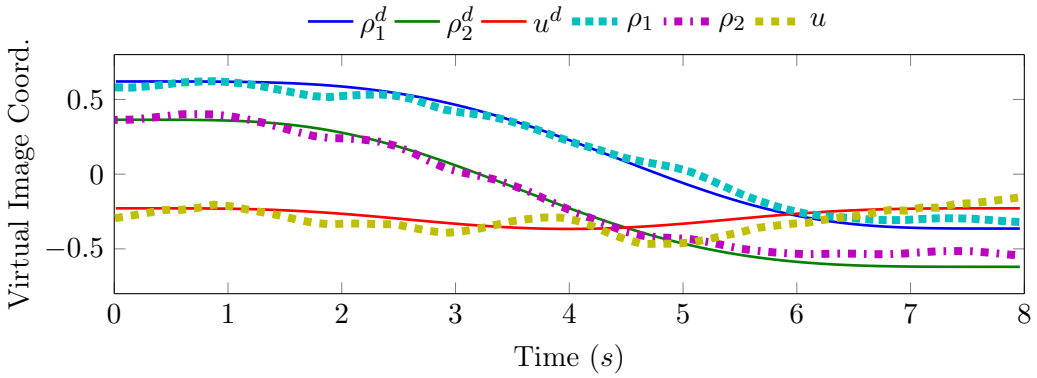



Figure 5.8: A sample trajectory starting below and to the side of the cylinder, passing directly underneath, and ending on the other side of the cylinder. The desired trajectory is denoted by a "$d$" superscript.

nates track the desired values using the control law proposed in Section 5.4. These results demonstrate the effectiveness of the proposed control law.

In many situations for aerial robots, it will beneficial for the robot to perch. Similar to the motivation for an upward-facing camera, we also use an upward-facing gripper. In this way, corrections of the robot position keep the gripper facing towards the target. Using this setup, the robot can successfully perch as displayed in Figure 5.6 using a trajectory in the first three components of $\mathcal{Y}$ as in Figure 5.9. The estimate of $\beta\mathbf{c}$ in the camera frame provides the distance of the vehicle from the cylinder and can be used to determine when to close the gripper. The estimate of $\mathbf{x}_q^{\mathcal{V}}$ throughout the trajectory is shown in Figure 5.10, and Table 5.1 shows the Root Mean Square Error (RMSE) of the vehicle with respect to the desired values. The first component is the largest because, in this maneuver, $\mathbf{x}_{\mathcal{V}}$ is closely aligned with the axis of the cylinder, and the measurement of the end of the cylinder is noisy. The average Euclidean error is 2.1 cm for the position confirming the effectiveness of the controller. Finally, a video of the experimental results is provided in the supplementary multimedia material and online (see [126]), showing that the vehicle can hover, perch from different initial configurations, and handle various cylinder orientations.

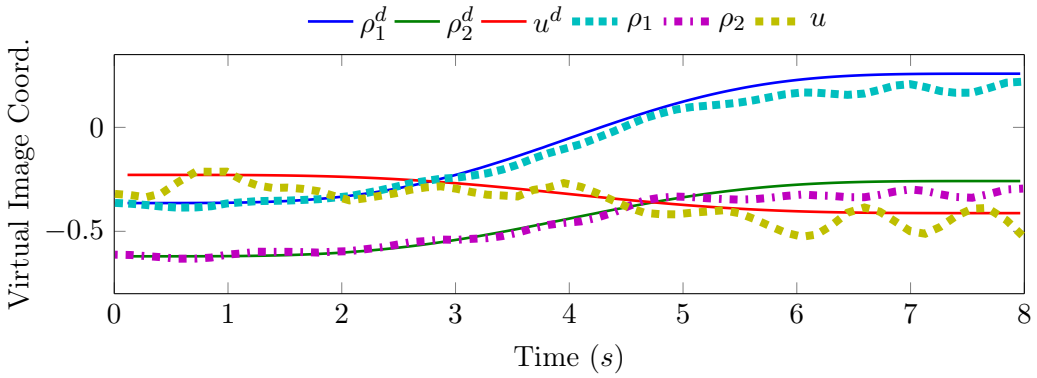



Figure 5.9: A sample trajectory starting below and to the side of the cylinder and ending in a perch. The desired trajectory is denoted by a "$d$" superscript.

| Coordinate in frame $\mathcal{V}$ | Position Root Mean Square Error (m) |
|---|---|
| $x$ | 0.020 |
| $y$ | 0.008 |
| $z$ | 0.009 |

Table 5.1: Position RMSE during a perching maneuver.

## 5.7 Conclusion

In this chapter, we presented the first vision-based control approach enabling a quadrotor, without the aid of GPS or a motion capture system, to perch on generally-oriented cylindrical objects. We developed a geometric model to determine the pose relative to a cylinder, used a virtual camera frame to simplify the image feature space, expressed the dynamics of the robot in terms of the image features, developed and proved stability of an image-based controller, and presented a method to plan dynamically feasible trajectories directly in the image plane. Finally, we provided experimental results, demonstrating the success of our methods using a fully autonomous platform with all computation and sensing onboard.

There are still opportunities for further development. Future work could include a sensitivity analysis of the control law, subject to radius uncertainty estimates, and could explore a comparison between the current IBVS approach and a PBVS approach. The gripper could accommodate perching using a smaller contact area, not by caging the cylinder, which would enable perching on larger cylinders using a smaller gripper. The cylinder detection could be improved to be more robust to objects that are cylindrical in only some sections, such

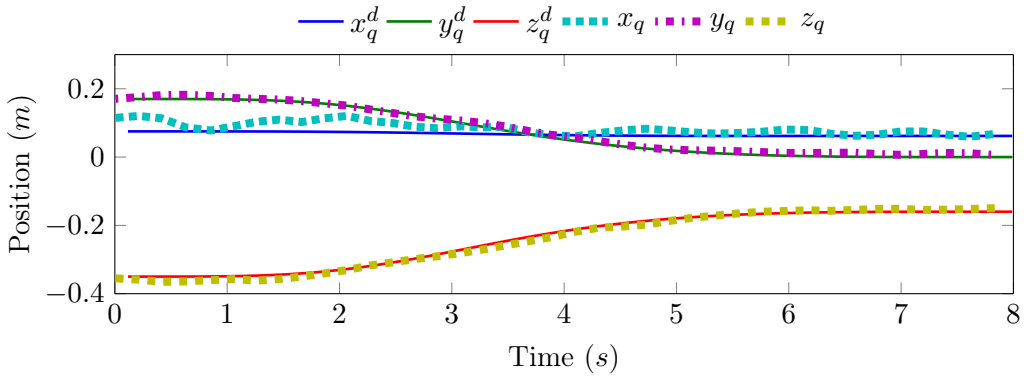



Figure 5.10: The desired (computed from $\rho_1^d$, $\rho_2^d$, and $u^d$) and computed estimate of the quadrotor's position in the virtual frame $\mathcal{V}$ for a perching maneuver.

as a crooked tree branch. Surface features of the target could also be leveraged to provide an estimate of axial motion. Finally, we could generalize this approach to consider other shapes, targets in motion, and the field of view constraints as will be discussed in the next chapter.

# Chapter 6

# Localization and Planning for Tracking Moving Targets

## 6.1 Introduction

In most previous works leveraging aerial robots for observation of another object, including the last few chapters, there is the assumption that the target is static in the world frame. However, a static object is often not a valid assumption such as when intercepting malicious aerial vehicles, tracking moving ground-based targets, or landing on a moving vehicle. Further, since localization methods such as GPS do not provide information about positioning relative to an object, researchers typically consider sensors such as cameras for tracking but often overlook the limited field-of-view (FOV) constraint. The FOV constraint has been mitigated by using creative methods such as an upward-facing camera (see, for example, Chapter 5 and [125]) or by leveraging an omnidirectional camera [64], however, aerial robots are more likely to be equipped with downward-facing sensors than upward-facing ones, and omnidirectional cameras typically require cumbersome optics, which reduces agility and payload capacity. Additionally, most of these works do not model the dynamics of the target or even predict its path, limiting them to quasi-static scenarios. Thus, the goal of this chapter is to relax the typical assumptions of a fixed target and an unconstrained field of view,

enabling a quadrotor to track a moving target while considering the robot's limited field of view. The example which will be explored is that of a robot tracking a moving sphere.[1] Our key contributions include the relative pose estimate of a spherical target as well as the planning algorithm, which considers the dynamics of the underactuated robot, the actuator limitations, and the field of view constraints. We show simulation and experimental results to demonstrate feasibility and performance, as well as robustness to abrupt variations in target motion.

As discussed in Chapter 5, a localization method such as GPS does not provide information about positioning relative to the object. Similarly, visual odometry methods focus on determining the robot's pose relative to a starting location, not relative to a specific object, making them not directly applicible to our scenario. On the other hand, visual servoing solutions focus on motion relative to a specific object, but they typically require one or more of the following assumptions:

**A1** The dynamics of the robot are first-order.

**A2** The robot is fully actuated.

**A3** The image features correspond to fixed locations on the object.

**A4** The object is fixed in the inertial frame.

**A5** The object does not leave the field of view of the camera.

In this chapter, we address these assumptions as they relate to maneuvering a quadrotor relative to a moving target. To accomplish this, we extend the planning methods used for high-speed grasping and perching from Chapters 3 and 4, respectively. As none of the previous chapters directly address **A4** or **A5**, these will require careful consideration.

The closest works regarding visual servoing relative to spheres develop the control for a first-order robotic arm (**A1** and **A2**) and assume the sphere is stationary (**A4**) [37, 38]. Circular markers have also been considered for visual servoing [28]. Our previous work

[1] Much of the content in this chapter will appear in [133].

considered servoing using a higher order underactuated system and did not require the image features to correspond to fixed locations on the target, an issue that arises when the object of interest has curvature [125]. While innovative design assisted in keeping the object in the field of view, there wasn't an explicit consideration for the field of view constraints, and there was no way to recover if the object left the image.

There have also been more general approaches for landing or maneuvering multirotors relative to a moving target (**A4**). For example, aerial grasping of moving targets has been demonstrated in a simulated environment [115]. Landing on a small carrier vehicle was accomplished by leveraging an onboard IR camera (from a Wii Remote) and IR markers on the landing pad [139]. The very limited field of view (45°) was mitigated with a pan-tilt unit. Landing on a moving target using a downward-facing camera was explored in [68], but there was no explicit consideration for the field of view. Angular dynamics were ignored but enabled landing on a vertically moving target similar to a ship deck [48]. Conveniently, this approach only relied on optical flow of the surface, making the field of view constraint less of an issue. However, the normal of the surface was assumed to be known.

The most relevant work considers tracking a moving target with a quadrotor while avoiding obstacles [9]. The trajectory of the target is estimated, and trajectories are planned to minimize the position error. In this work, we consider a similar approach with some novel extensions. Specifically, we will consider a modified objective function which is more appropriate for aggressive scenarios, and we will incorporate the field of view constraints in the robot's trajectory planner.

This work makes multiple contributions. First, we provide a solution to identify a spherical object's position with respect to the aerial platform using monocular vision. Second, a trajectory planning method enables a robot to track a moving object in real time using the single downward-facing, body-fixed camera while explicitly considering the field of view. To the best of our knowledge, this is the first time that onboard navigation techniques based on a single camera and inertial measurement unit (IMU) are used both to navigate in the world and track a moving target without the need of an external motion capture system or

additional onboard sensors.

The high-level goal to enable a robot to track or acquire a moving target will be broken down into a number of sub-tasks. First, the robot must be able to determine a relative pose, which will be discussed in Section 6.2. Since the object may be moving, we also need to estimate its motion and propagate its dyanmics in order to increase robustness to occlusions and potential failure when an object may temporarily leave the field of vew. Thus, we model the object's and robot's dynamics as well as discuss the robot's controller in Section 6.3. The planning strategy is proposed in Section 6.4. Finally, results are presented in Section 6.6, and we conclude in Section 6.7.

## 6.2 Relative Pose

In this section, we provide an approach to determine a relative pose from the image of a sphere. Specifically, we consider fitting a cone to a set of 3D points, making our approach agnostic to the camera model. Related works include fitting of ellipses and circles to points in a plane [35, 40]. Unless otherwise noted, vectors in this section will be expressed in the camera frame $\mathcal{C}$. Let a sphere be represented by

$$\|\mathbf{X} - \mathbf{C}\|^2 - r^2 = 0 \tag{6.1}$$

where $\mathbf{C} \in \mathbb{R}^3$ is the center, $r$ is the radius, $\mathbf{X} \in \mathbb{R}^3$ is any point on the surface, and $\|\cdot\|$ represents the Euclidean norm.

We define the projection operator $\pi$ which maps a point $\mathbf{X}$ in the camera frame to a point $\mathbf{x} \in \mathbb{R}^3$ on the image surface

$$\mathbf{x} = \pi(\mathbf{X}) \equiv \frac{1}{\lambda(\mathbf{X})}\mathbf{X} \tag{6.2}$$

where the choice of $\lambda : \mathbb{R}^3 \mapsto \mathbb{R}$ is dependent on the camera model. For example, we would choose $\lambda \equiv \|\mathbf{X}\|$ for a spherical camera model or $\lambda \equiv \mathbf{e}_3^T\mathbf{X}$, where $\mathbf{e}_3^T = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, for a

pinhole model. In any case, we can express $\mathbf{X} = \lambda\mathbf{x}$, allowing us to write (6.1) as

$$\|\lambda\mathbf{x} - \mathbf{C}\|^2 - r^2 = 0 \tag{6.3}$$

$$\mathbf{x}^T\mathbf{x}\lambda^2 - 2\mathbf{x}^T\mathbf{C}\lambda + \mathbf{C}^T\mathbf{C} - r^2 = 0, \tag{6.4}$$

which is quadratic in $\lambda$. Considering that points on the contour of the projection represent rays which are tangent to the sphere, we require $\lambda$ to be unique, which means that, for the contour, the discriminant of (6.4) must vanish so that

$$(2\mathbf{x}^T\mathbf{C})^2 - 4\mathbf{x}^T\mathbf{x}\left(\mathbf{C}^T\mathbf{C} - r^2\right) = 0 \tag{6.5}$$

$$\mathbf{x}^T\mathbf{C}\mathbf{C}^T\mathbf{x} - \mathbf{x}^T\mathbf{x}\left(\mathbf{C}^T\mathbf{C} - r^2\right) = 0 \tag{6.6}$$

$$\mathbf{x}^T\left[\mathbf{C}\mathbf{C}^T + \left(r^2 - \mathbf{C}^T\mathbf{C}\right)\mathbf{I}\right]\mathbf{x} = 0, \tag{6.7}$$

where $\mathbf{I}$ is the identity matrix, and we observe that (6.7) is a conic. The set of solutions for $\mathbf{x}$ is known as the *tangent cone* to the sphere from the camera origin [33]. In this case, however, $\mathbf{x}$ is an observed point, making this form not ideal for the rest of our formulation where we wish to determine $\mathbf{C}$.

### 6.2.1 A Geometric Solution

If we assume a spherical camera model for $\mathbf{x}$ so that $\mathbf{x}^T\mathbf{x} = 1$, then (6.6) simplifies to

$$\mathbf{x}^T\mathbf{C}\mathbf{C}^T\mathbf{x} - \left(\mathbf{C}^T\mathbf{C} - r^2\right) = 0 \tag{6.8}$$

$$\left(\mathbf{x}^T\mathbf{C}\right)^2 + r^2 - \mathbf{C}^T\mathbf{C} = 0. \tag{6.9}$$

Then, if we let $\mathbf{C} = \gamma\mathbf{c}$ with $\|\mathbf{c}\| = 1$ so that $\mathbf{c}$ represents the bearing to the center of the target, we have

$$\left(\gamma\mathbf{x}^T\mathbf{c}\right)^2 + r^2 - \gamma^2 = 0, \tag{6.10}$$

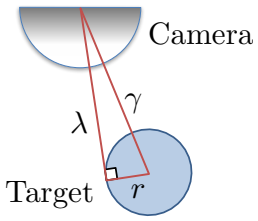



Figure 6.1: The geometry of a vector $\mathbf{X} = \lambda\mathbf{x}$ with $\|\mathbf{x}\| = 1$ in the tangent cone. The center of the sphere is given by $\mathbf{C} = \gamma\mathbf{c}$ with $\|\mathbf{c}\| = 1$. The top hemisphere represents the spherical camera model, and the circle is the cross section through a hemisphere of the sphere. Note that the Pythagorean theorem holds such that $\lambda^2 + r^2 = \gamma^2$ when a spherical camera model is used for both $\mathbf{X}$ and $\mathbf{C}$.

which is simply the Pythagorean constraint as discussed in Figure 6.1. Squared errors from this constraint can be captured in a non-linear minimization problem to estimate $\mathbf{C}$

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{n} \left[ \left(\mathbf{x}_i^T \mathbf{C}\right)^2 + r^2 - \mathbf{C}^T\mathbf{C} \right]^2 \tag{6.11}$$

given $n$ observed bearings in the tangent cone. This minimization can be seeded with the centroid as an initial guess, and a gradient descent can be used to determine the solution. With this formulation, the geometric error is minimized, however, the computational demands may be higher than is realistic for real-time implementation.

### 6.2.2 An Algebraic Solution

The conic fitting problem can also be approached algebraically, analogous to algorithms used for ellipse fitting [35]. For ease of comparison, we will use similar notation.

#### Fitting a Cone to a Set of Observations

Let a conic be defined by

$$\mathbf{x}^T A\mathbf{x} = 0 \tag{6.12}$$

with $A \in \mathbb{R}^{3\times3}$, $A = A^T$, and $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^T$. Note that $A$ can be arbitrarily scaled and represent the same conic. The constraints for $n$ observations $\mathbf{x}_i$ can be written as

$$\underbrace{\begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & 2x_1y_1 & 2x_1z_1 & 2y_1z_1 \\ x_2^2 & y_2^2 & z_2^2 & 2x_2y_2 & 2x_2z_2 & 2y_2z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & y_n^2 & z_n^2 & 2x_ny_n & 2x_nz_n & 2y_nz_n \end{bmatrix}}_{D} \begin{bmatrix} A_{11} \\ A_{22} \\ A_{33} \\ A_{12} \\ A_{13} \\ A_{23} \end{bmatrix} = \mathbf{0} \tag{6.13}$$

so that the linear system can be solved using a singular value decomposition (SVD) of $D$. From this, we can construct $A$, which represents the best algebraic-fit conic.

**Extracting the Relative Pose from a Conic**

In a frame defined such that the $z$ axis is parallel with the bearing to the target's center, our observations of the sphere boundary form a cone which can be written as

$$\mathbf{x}^T \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \mathbf{x} \equiv \mathbf{x}^T \Lambda \mathbf{x} = 0. \tag{6.14}$$

A change of basis can be applied to express the cone in camera frame coordinates with $A \in \mathbb{R}^{3\times3}$, $A = A^T$

$$\mathbf{x}^T Q \Lambda Q^{-1} \mathbf{x} \equiv \mathbf{x}^T A \mathbf{x} = 0. \tag{6.15}$$

Thus, we see that for a general conic defined by $A$, we can use an eigenvalue decomposition to determine $Q$, $\lambda_1$, $\lambda_2$, and $\lambda_3$. The axis of the cone in the camera frame is given by the eigenvector associated with the eigenvalue whose sign is the least common, which, without loss of generality, we can assume is $\lambda_3$. This is also the axis along which the bearing to the target's centroid will lie. For a circular cone, the other two eigenvalues would be equal and have a sign opposing $\lambda_3$. However, in practice, they will not be identical and can

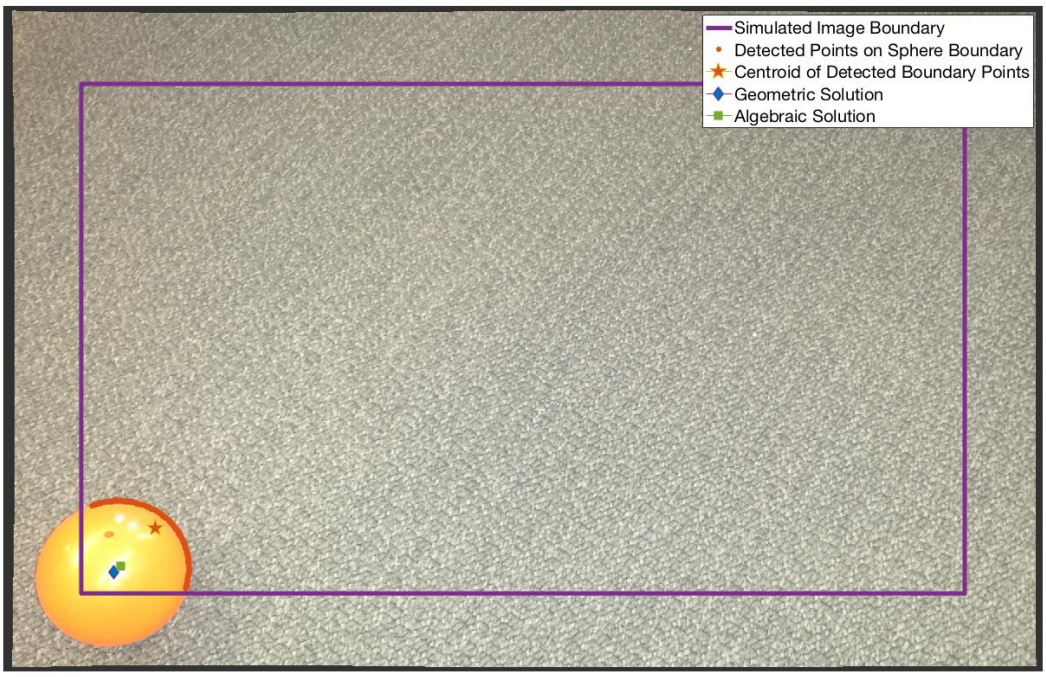



Figure 6.2: A sample comparison between sphere detection approaches. The purple frame indicates a simulated image boundary. Red circles denote detected boundary points of the sphere. In this case, more points were actually detected, but points outside of the simulated image boundary were discarded. The centroid of the remaining points is represented by the red pentagram. The computed bearing to the center of the sphere using the geometric solution is denoted by the blue diamond, and the computed bearing to the center of the sphere using the algebraic solution is denoted by the green square. In this case, the centroid does not accurately represent the bearing to the center of the sphere. The geometric solution is best, but it requires more computation time.

be approximated by their average, $\bar{\lambda}_{12}$. Then, using similar triangles, the distance to the centroid of the target can be determined to be $\gamma = r\sqrt{\left|\bar{\lambda}_{12}/\lambda_3\right| + 1}$.

### 6.2.3 Discussion

A comparison of these approaches is given in Figure 6.2. We first note that the geometric error provides the best fit but at the cost of the most computation time (about 23 ms). On the other hand, the algebraic approach is much faster (about 4 ms on the same machine) and, in most cases, provides a sufficiently accurate solution. In general, both solutions are superior to the centroid of the observed boundary points, especially when only part of the target is observed.

If we assume a planar camera model, the second approach remains valid, and we could

still recover the information needed from an ellipse in the image plane. In many cases, this may be the preferred approach as there are readily available ellipse trackers such as `vpMeEllipse` from [79], and most image processing would occur in a flat image.

With these approaches, determining the relative pose from a single image requires knowledge of the sphere's radius. A scale has been estimated online using a Structure from Motion (SfM) approach, but the relative velocity was assumed to be known [116]. In our case, the target is not stationary and the magnitude of its velocity is unknown, so determining the scale online is a more difficult problem which will be left for future work.

## 6.3 Dynamics and Control

### 6.3.1 Dynamics of the Object

One of the unique differences between this work and other visual-servoing works is that we no longer require the target object to be fixed in the world. We simply require that the object's path can be approximated and predicted over some horizon using an $n^{th}$ order polynomial in each Cartesian dimension.

### 6.3.2 Dynamics of the Robot

The dynamics of the quadrotor are assumed to be the same as outlined in Section 2.1.3 but are reproduced here for clarity. Let

$$m\ddot{\mathbf{x}} = -mg\mathbf{e}_3 + fR\mathbf{e}_3 \tag{6.16}$$

$$\dot{R} = R\hat{\boldsymbol{\Omega}} \tag{6.17}$$

$$\mathcal{I}\dot{\boldsymbol{\Omega}} = \mathbf{M} - \boldsymbol{\Omega} \times \mathcal{I}\boldsymbol{\Omega} \tag{6.18}$$

where $m$ is the mass of the robot, $\mathbf{x} \in \mathbb{R}^3$ is the position of the robot in an inertial frame, $g$ is gravity, $\mathbf{e}_3$ is the $3^{\text{rd}}$ standard basis vector, $f \in \mathbb{R}$ and $\mathbf{M} \in \mathbb{R}^3$ are the thrust and moment control inputs to the system, $R \in SO(3)$ is the orientation of the vehicle, and $\hat{\cdot} : \mathbb{R}^3 \mapsto \mathfrak{so}(3)$ is the hat map defined such that, for any two vectors, $\hat{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$. Also, $\mathcal{I}$ is the inertial matrix, and $\boldsymbol{\Omega}$ is the robot's angular velocity expressed in the robot frame.

### 6.3.3 Coupled Dynamics

It is reasonable to assume that $R$ and $\boldsymbol{\Omega}$ are observable using the IMU and magnetometer since they are necessary for general flight. This implies that $R_{\mathcal{C}}^{\mathcal{W}}$ and $\boldsymbol{\Omega}^{\mathcal{C}}$ are known since they are related through a known, fixed transformation to the robot frame. Then, to compute the dynamics of the object in the robot frame, we simply need to determine $\ddot{\mathbf{x}}$ in terms of the image features and substitute them into (6.16), which is similar to the approach in [140]. The position of the object can be expressed kinematically as

$$\mathbf{x}_{\mathcal{O}}^{\mathcal{W}} = \mathbf{x}_{\mathcal{R}}^{\mathcal{W}} + R_{\mathcal{R}}^{\mathcal{W}} \mathbf{x}_{\mathcal{R}\mathcal{O}}^{\mathcal{R}} \tag{6.19}$$

so that the velocity is

$$\dot{\mathbf{x}}_{\mathcal{O}}^{\mathcal{W}} = \dot{\mathbf{x}}_{\mathcal{R}}^{\mathcal{W}} + R_{\mathcal{R}}^{\mathcal{W}} \dot{\mathbf{x}}_{\mathcal{R}\mathcal{O}}^{\mathcal{R}} + \dot{R}_{\mathcal{R}}^{\mathcal{W}} \mathbf{x}_{\mathcal{R}\mathcal{O}}^{\mathcal{R}} \tag{6.20}$$

and the acceleration is given by

$$\ddot{\mathbf{x}}_{\mathcal{O}}^{\mathcal{W}} = \ddot{\mathbf{x}}_{\mathcal{R}}^{\mathcal{W}} + R_{\mathcal{R}}^{\mathcal{W}} \ddot{\mathbf{x}}_{\mathcal{R}\mathcal{O}}^{\mathcal{R}} + 2\dot{R}_{\mathcal{R}}^{\mathcal{W}} \dot{\mathbf{x}}_{\mathcal{R}\mathcal{O}}^{\mathcal{R}} + \ddot{R}_{\mathcal{R}}^{\mathcal{W}} \mathbf{x}_{\mathcal{R}\mathcal{O}}^{\mathcal{R}} \tag{6.21}$$

Note that these dynamics hold for any inertial frame, which means that they are valid in the frame of the target if a constant-velocity model is assumed, which would enable a straightforward transition to an image based (IBVS) approach, perhaps using the virtual fixed-orientation frame strategy from Chapter 5.

### 6.3.4 Control

For now, we leverage a position-based visual servoing (PBVS) control strategy and make only slight modifications to the controller outlined in Section 2.2. As discussed in the last subsection, we could treat the constant velocity target as the origin of our inertial frame, and we could rewrite our control law accordingly where the errors would be defined in the object frame $\mathcal{O}$, which is also an inertial one if we assume the velocity is constant,

In either case, the position and velocity errors in the desired frame are

$$\mathbf{e}_x = \mathbf{x} - \mathbf{x}_{\text{des}} \quad \text{and} \quad \dot{\mathbf{e}}_x = \dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{des}}, \tag{6.22}$$

respectively, and the thrust is computed as

$$f = (-k_x \mathbf{e}_x - k_v \dot{\mathbf{e}}_x + mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_{\text{des}}) \cdot R\mathbf{e}_3 \tag{6.23}$$

where $k_x$ and $k_v$ are positive gains and the subscript "des" denotes a desired value. The attitude and angular velocity errors are defined as

$$\mathbf{e}_R = \frac{1}{2}\left(R_{\text{des}}^T R - R^T R_{\text{des}}\right)^\vee, \quad \mathbf{e}_\Omega = \boldsymbol{\Omega} - R^T R_{\text{des}} \boldsymbol{\Omega}_{\text{des}} \tag{6.24}$$

where $\cdot^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$ is the opposite of the hat map. The control moments are computed as

$$\mathbf{M} = -k_R \mathbf{e}_R - k_\Omega \mathbf{e}_\Omega + \boldsymbol{\Omega} \times \mathcal{I}\boldsymbol{\Omega}, \tag{6.25}$$

where $k_R$ and $k_\Omega$ are positive gains. Then, the zero-equillibrium is exponentially stable and, in general, the controller provides "almost global exponential attractiveness" [69].

## 6.4 Planning

Since we are interested in aggressive maneuvers so that the quadrotor can commence tracking a quickly moving target, it is important to not only consider dynamic feasiblity (considering the relative degree), but also to ensure that actuator and sensor constraints, including the field of view, are not violated. The incorporation of some actuator and sensor constraints was demonstrated in Chapter 4, enabling a robot to perform aggressive maneuvers to perch on vertical surfaces, but with no need to consider vision constraints. An extension to plan trajectories for image features was presented in Chapter 5, but there were no guarantees that the trajectories would satisfy the sensor and actuator constraints. The approach here will allow for the consideration of the dynamic, sensor, and actuator constraints, including

the field of view.

### 6.4.1 Representation of Trajectories

We choose to express trajectories using an $n^{th}$ order polynomial basis with terms $b_k(t)$ so that a trajectory $p(t)$ can be represented by

$$p(t) = \sum_{k=0}^{n} c_k b_k(t) \tag{6.26}$$

or with a vector of coefficients $\mathbf{c}_i \in \mathbb{R}^{n+1}$ for dimension $i$ and a basis vector $\mathbf{b}(t) : \mathbb{R} \mapsto \mathbb{R}^{n+1}$

$$p_i(t) = \mathbf{c}_i^T \mathbf{b}(t). \tag{6.27}$$

We could allow $\mathbf{b}(t)$ to be a standard power basis,

$$\mathbf{b}(t) = \begin{bmatrix} 1 & t & t^2 & \cdots & t^n \end{bmatrix}^T, \tag{6.28}$$

a Legendre Polynomial basis,

$$\mathbf{b}(t) = \begin{bmatrix} 1 & t & \frac{1}{2}\left(3t^2 - 1\right) & \frac{1}{2}\left(5t^3 - 3t\right) & \cdots \end{bmatrix}^T, \tag{6.29}$$

or any basis of the user's choice. The $r^{th}$ derivative can be computed as

$$p_i^{(r)}(t) = \mathbf{c}_i^T \mathbf{b}^{(r)}(t) \tag{6.30}$$

since $\mathbf{c}_i$ is independent of time. Let $B^{(r)}(t) : \mathbb{R} \mapsto \mathbb{R}^{d(n+1) \times d}$ and $\mathbf{c} \in \mathbb{R}^{d(n+1)}$ be defined as

$$B^{(r)}(t) = \begin{bmatrix} \mathbf{b}^{(r)}(t) & & \\ & \ddots & \\ & & \mathbf{b}^{(r)}(t) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_d \end{bmatrix} \tag{6.31}$$

for $d$ dimensions. That is, $\mathbf{c}$ is a stack of the coefficient vectors. Then, we can write the trajectory as $\mathbf{p}^{(r)}(t) : \mathbb{R} \mapsto \mathbb{R}^d$ where

$$\mathbf{p}^{(r)}(t) = \left(B^{(r)}(t)\right)^T \mathbf{c}. \tag{6.32}$$

For clarity, we note that this is equivalent to

$$\mathbf{p}^{(r)}(t) = \begin{bmatrix} \mathbf{c}_1^T \\ \mathbf{c}_2^T \\ \vdots \\ \mathbf{c}_d^T \end{bmatrix} \mathbf{b}^{(r)}(t), \tag{6.33}$$

however, the previous formulation will be useful later.

### 6.4.2 Trajectory of the Target and Robot

The trajectory of the target in dimension $i$ is defined by coefficients $\mathbf{h}_i \in \mathbb{R}^{n+1}$ so that the trajectory for all dimensions is

$$\mathbf{g}^{(r)}(t) = \left(B^{(r)}(t)\right)^T \mathbf{h} \tag{6.34}$$

where $\mathbf{h} = \begin{bmatrix} \mathbf{h}_1^T & \dots & \mathbf{h}_d^T \end{bmatrix}^T$. This allows us to fit a polynomial to the dynamic model over some horizon to predict the motion of the target [9]. We define the trajectory of the robot as

$$\mathbf{p}^{(r)}(t) = \left(B^{(r)}(t)\right)^T \mathbf{c}, \tag{6.35}$$

which is the same notation outlined previously.

### 6.4.3 General Planning Strategy

One strategy is to minimize the position error between the target and the robot's trajectories with the inclusion of smoothing on higher derivatives [9]. However, this approach is not ideal in some scenarios (see discussion in Figure 6.3) because it can produce initial transients that

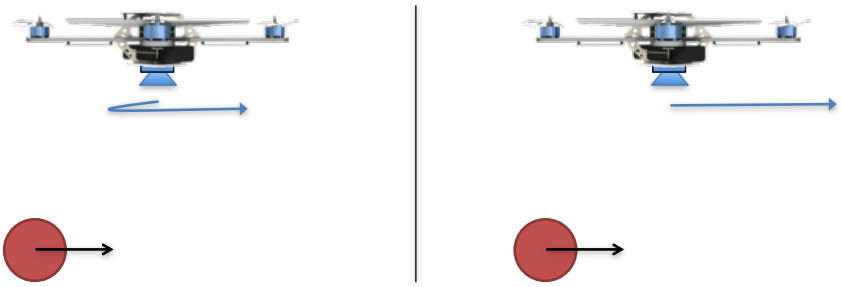


Figure 6.3: A one-dimensional example motivating the minimization of velocity error between the robot and the target. In both scenarios pictured above, the robot is assumed to start from rest as the object enters the field of view. On the left hand side, we see a possible trajectory if the position error is minimized, and on the right-hand side, we see a result if the velocity error is minimized. In the position-error case (LHS), it is obvious that the motion is not ideal for larger target velocities. Thus, we are motivated, at least during the initial transient, to minimize the velocity error. Observe that this also could aid in mitigating the field of view constraints.

are counterproductive to achieving tracking as quickly as possible.

Instead, we propose minimizing the velocity error during the initial transient. Interestingly, this results in a strategy similar to the ones taken by dragonflies [93, 101, 102], falcons [58], and human outfielders [83]. In these cases, the target is regulated to remain at a constant bearing in the field of view, and the range gap is closed using a strategy that may be captured by $\tau$ ("tau") theory [103], which doesn't require knowledge of a distance to the target. One can argue that the sensing methods of our robot and dragonflies are very similar. Even though dragonflies have two eyes, they control their motion essentially using monocular vision. Indeed, their immobile, fixed-focus, minimally-overlapping, low-resolution, compound eyes have such a small baseline that their stereo depth perception would only be accurate to a few centimeters [120]. Thus, they clearly must use other methods for range estimates. For now, we assume that the scale of our target is known a priori, allowing for the direct estimation of the range, and we leave the online range estimation, if necessary at all, for future work.

Minimizing velocity error alone, however, cannot capture the desired relative pose. Thus, we consider the planning and tracking problem as having two phases. First, there must be a phase where the robot is accelerating to match the velocity of the target. This transient phase is when the field of view constraints are most likely to be active constraints. The

objective of this phase can be expressed as a minimization of the velocity error between the robot and the target. The second phase incorporates the position error and enables planning to intercept the target or track the target from a desired relative pose.

### 6.4.4 A Multi-Objective Cost Function

We are motivated to use a multi-objective cost function to penalize both velocity errors and, when applicable, position errors. Interestingly, a similar approach was used to smoothly change formation shapes of an aerial robot team [136]. We define the error as

$$\mathbf{e}(t) = \mathbf{g}(t) - \mathbf{p}(t). \tag{6.36}$$

The frame is intentionally not specified since it could be chosen to be either the world frame, $\mathcal{W}$, or if a constant-velocity target is assumed, the object frame, $\mathcal{O}$. In the latter case, the error simplifies so that $\mathbf{g}(t) = 0$ and $\mathbf{p}(t)$ is expressed relative to the object. Also note that a translation could be included in either trajectory to achieve a constant offset.

Since we're interested in minimizing specific derivatives, we write a general objective function which computes the integrated square of the Euclidean error of the $r^{th}$ derivative

$$\mathcal{J}_r = \int_{t_o}^{t_f} \left\| \mathbf{e}^{(r)}(t) \right\|^2 dt \tag{6.37}$$

where $\| \cdot \|$ represents the Euclidean norm. Expanding, we have

$$\mathcal{J}_r = \int_{t_o}^{t_f} \left( \mathbf{e}^{(r)} \right)^T \left( \mathbf{e}^{(r)} \right) dt \tag{6.38}$$

$$= \int_{t_o}^{t_f} \left( \mathbf{p}^{(r)}(t) - \mathbf{g}^{(r)}(t) \right)^T \left( \mathbf{p}^{(r)}(t) - \mathbf{g}^{(r)}(t) \right) dt \tag{6.39}$$

$$= \mathbf{c}^T Q_r \, \mathbf{c} - 2\mathbf{h}^T Q_r \, \mathbf{c} + \mathbf{h}^T Q_r \, \mathbf{h} \tag{6.40}$$

where

$$Q_r = \int_{t_o}^{t_f} \left(B^{(r)}(t)\right) \left(B^{(r)}(t)\right)^T dt, \tag{6.41}$$

$$= \begin{bmatrix} \int_{t_o}^{t_f} \mathbf{b}^{(r)}(t) \left(\mathbf{b}^{(r)}(t)\right)^T dt & & \\ & \ddots & \\ & & \int_{t_o}^{t_f} \mathbf{b}^{(r)}(t) \left(\mathbf{b}^{(r)}(t)\right)^T dt \end{bmatrix} \tag{6.42}$$

so that $\mathcal{J}_r$ can be expressed in quadratic form as

$$\mathcal{J}_r = \mathbf{c}^T Q_r \mathbf{c} + \mathbf{f}^T \mathbf{c} + \alpha, \quad \mathbf{f} = -2Q_r^T \mathbf{h}, \quad \alpha = \mathbf{h}^T Q_r \mathbf{h}. \tag{6.43}$$

Note that a translation could be included in the object's coefficients, $\mathbf{h}$, to specify a desired relative pose.

At this point, it is interesting to consider nondimensionalizing the trajectory so that $t_o = 0$ and $t_f = 1$. This would allow us to precompute $Q_r$ for each derivative. Further, we could choose shifted, normalized Legendre polynomials

$$\bar{P}_n(t) = \sqrt{2n+1}\, P_n(2t-1), \tag{6.44}$$

where $P_n(t)$ is the $n^{th}$ Legendre polynomial and

$$\int_0^1 \bar{P}_n(t)\bar{P}_m(t)dt = \delta_{mn}. \tag{6.45}$$

The basis vector would then look like

$$\mathbf{b}(t) = \begin{bmatrix} \bar{P}_0(t) & \bar{P}_1(t) & \cdots \bar{P}_n(t) \end{bmatrix}^T \tag{6.46}$$

and, quite beautifully, $Q_0$ becomes the identity matrix. An alternative approach is to plan

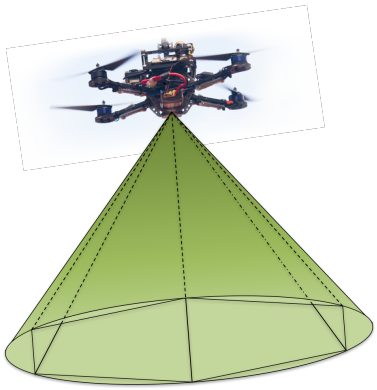


Figure 6.4: The field of view constraint of a lens. The cone's representation is not positive semidefinite, which means we cannot use it directly in a QCQP. However, the cone can be approximated with an inscribed pyramid.

all trajectories to be the same duration. In practice, we have found that numerical stability is best achieved by only non-dimensionalizing trajectories that are over 1 second in duration.

### 6.4.5 Actuator and Sensor Constraints

The field of view of a lens can be modeled as a cone in the camera (or body) frame

$$\mathbf{m}^T A \mathbf{m} \leq 0 \tag{6.47}$$

where $A = A^T$ and the solutions $\mathbf{m} \in \mathbb{R}^3$ are rays lying within the field of view. Unfortunately, this results in a non positive semidefinite constraint, which means that we can't include it as a quadratic constraint in a Quadratically Constrained Quadratic Program (QCQP). Alternatively, we could model the constraint with an inscribed pyramid similar to the approximation of a coulomb friction cone (see Figure 6.4 or [114]). Further, because of the rectangular sensor design of most cameras, the cone model may not be best. Thus, we can inscribe a convex pyramid in the field of view, which would provide a set of linear constraints representing the effective field of view.

Similar to the approach in Chapter 4, we incorporate bounds to precent actuator saturation and ensure our system can achieve the trajectory. However, we also want to consider the field of view constraints so the object does not leave the image. The simplest way to solve this problem is to prescribe a maximum attitude angle (*e.g.,* $\arccos\left(\mathbf{e}_3 \cdot R\mathbf{e}_3\right) \leq \beta_{max}$),

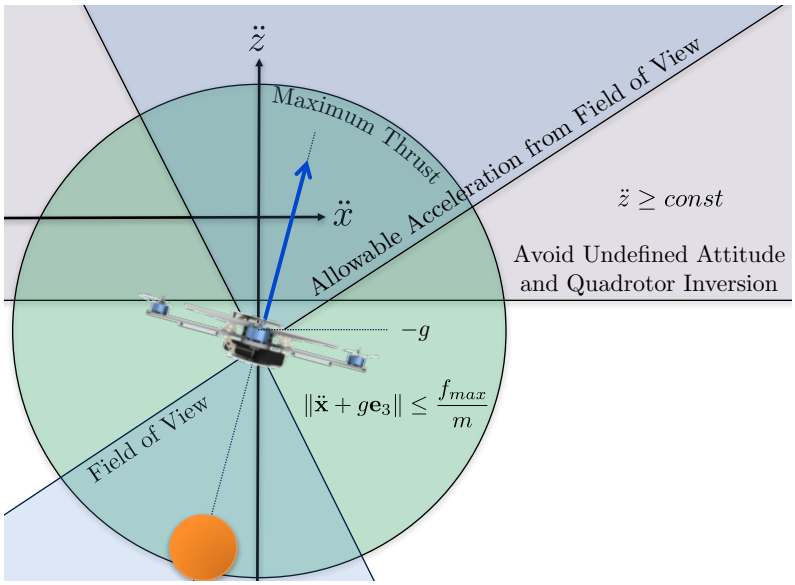



Figure 6.5: A visualization of the acceleration constraints in the $x - z$ plane. The large circle represents the maximum thrust bound and is similar to the illustration in [97]. The half plane bound keeps the quadrotor from inverting and avoids the singularity resulting when there is no thrust. Given the expected bearing between the target and the robot (inclined, dashed line), a conic (or pyramid) section represents the acceleration constraint to keep the object in the field of view.

and reduce the effective field of view accordingly. A trajectory could then be planned simultaneously using the maximum attitude constraint and the reduced field of view to constrain the relative positions. However, this approach is more conservative than desired, especially when aggressive maneuvers are necessary. Certainly, we do not want to restrict the maximum attitude.

Instead, we directly incorporate the field of view as constraints in the optimization by using the relative position to rotate the field of view and prescribe constraints on the acceleration (effectively on the attitude). We first determine a set of acceleration constraints when the object would be directly beneath the robot. Then, we rotate the axis to be coincident with the expected bearing from the object to the robot. A schematic of sample acceleration constraints at one instant is given in Figure 6.5.

### 6.4.6 The Planner

In this subsection, we describe the proposed planning algorithm. We leverage a receding horizon planning strategy to continuously update the planned trajectory based on the tar-

get's actions. We first set the default objective function to minimize the velocity error and we include a slight weighting to penalize jerk, which helps to reduce the angular velocity and has been used previously to provide smoothing [9]. Note that we could also penalize the next derivative, snap, which would most directly help to reduce the angular acceleration. Dynamic, actuator, gyro, and vision constraints are incorporated as nonlinear constraints on the coefficients of the trajectory. Then, the problem can be solved using a Sequential Quadratic Program (SQP) solver. In our case, there is the benefit that instead of the QP subproblem being an approximation of a general nonlinear cost function, it is identical to our cost function. Most convergence arguments for SQP problems require that an initial solution or "warm start" is sufficiently close to the actual solution and that the active inequality constraints at the optimal solution are the same ones that are active at the local solution. For more details, we refer the reader to the "Sequential Quadratic Programming Methods" chapter in [41].

When the relative bearing and relative velocity have a positive inner product, then there is no harm in also minimizing position error (see the example in Figure 6.3). Additionally, there may be situations where the velocity is matched before the previous condition is satisfied. In such cases, we can also incorporate the position term in the objective function once the relative velocity falls below a predefined threshold. While this planning approach does not guarantee completeness since it is dependent on a non-linear optimization, it is viable for real-time applications, and it works well in practice. The algorithm is presented in Algorithm 1, where the weighting for derivative $r$ is given by $\lambda_r$, and $\mathcal{J}_r$ is defined by (6.43).

**Algorithm 1** The Planning Algorithm

$\mathcal{J} \leftarrow \lambda_1 \mathcal{J}_1 + \lambda_3 \mathcal{J}_3$ ▷ Set the default objective function
**for** Each Horizon **do**
    update($\mathbf{g}(t)$)
    **repeat**
        $\mathbf{p}(t) \leftarrow$ iterateSQP($\mathcal{J}$, $\mathbf{g}(t)$, $\mathbf{p}(t)$)
    **until** Out of Time
    $\mathbf{e}(t) \leftarrow \mathbf{g}(t) - \mathbf{p}(t)$ ▷ Update the relative pose trajectory
    **if** ($\mathbf{e} \cdot \dot{\mathbf{e}} \geq 0$ or $\|\dot{\mathbf{e}}\| \leq thresh$) **then**
        $\mathcal{J} \leftarrow \lambda_0 \mathcal{J}_0 + \lambda_1 \mathcal{J}_1 + \lambda_3 \mathcal{J}_3$ ▷ Include position error in objective function
    **end if**
**end for**

## 6.5 Image Based Features

We may like a set of image-based features that can be directly observed from the image and still provide a unique mapping to the relative pose of the sphere. The authors in [38] propose the set $\mathbf{s} \in \mathbb{R}^3$

$$\mathbf{s} = \frac{1}{\rho}\mathbf{c} = \frac{1}{r}\mathbf{C} \tag{6.48}$$

where $\rho$ is the radius of the target sphere in an image whose optical axis lies along $\mathbf{c}$. A diagram is provided in Figure 6.6. Conveniently, this set of image-based features is uniquely related to the relative pose, $\mathbf{C}$, in the camera frame, $\mathcal{C}$. If we assume that the target moves with a constant velocity, the target can be treated as an inertial frame, and we could simply switch to the image based features using the target's radius. Since $R_{\mathcal{C}}^{\mathcal{R}}$ is a known constant, we could substitute

$$\mathbf{x} = \mathbf{x}_{\mathcal{R}}^{\mathcal{O}} = -rR_{\mathcal{R}}^{\mathcal{O}}R_{\mathcal{C}}^{\mathcal{R}}\mathbf{s}, \quad \dot{\mathbf{x}} = \dot{\mathbf{x}}_{\mathcal{R}}^{\mathcal{O}} = -rR_{\mathcal{R}}^{\mathcal{O}}R_{\mathcal{C}}^{\mathcal{R}}\dot{\mathbf{s}} - r\dot{R}_{\mathcal{R}}^{\mathcal{O}}R_{\mathcal{C}}^{\mathcal{R}}\mathbf{s} \tag{6.49}$$

to convert the proposed control law and planning approach to an IBVS one. In this case, we've assumed that $R_{\mathcal{R}}^{\mathcal{O}}$ and $\dot{R}_{\mathcal{R}}^{\mathcal{O}}$ can be estimated using the IMU, which is similar to our previous approach in Chapter 5 and could even be improved by considering visual feedback [73].

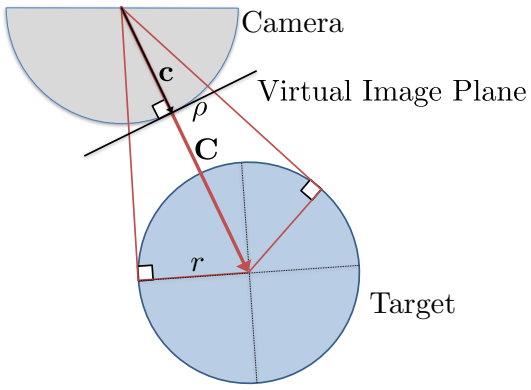



Figure 6.6: The proposed image features for an IBVS approach. The bearing to the center of the sphere is $\mathbf{c}$, the radius of the sphere is $r$, the radius of the image in a virtual plane (focal length = 1) is $\rho$, and the actual relative pose is given by $\mathbf{C}$.

## 6.6 Results

In this section, we present our simulation and experimental results. We first present a sample simulation assuming a constant-velocity target starting at

$$\mathbf{x} = \begin{bmatrix} -2.5 \\ 1 \\ 0 \end{bmatrix}, \quad \dot{\mathbf{x}} = \begin{bmatrix} 5 \\ -1 \\ 0 \end{bmatrix}$$

with the robot's initial conditions given by

$$\mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}, \quad \dot{\mathbf{x}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad R = \mathbf{I}, \quad \boldsymbol{\Omega} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The planning horizion is 1 second with a trajectory update frequency of 10 Hz. The desired relative pose is defined such that the robot is 1 m above the target, and the horizontal and vertical fields of view are both assumed to be 90°. The resultant trajectory is plotted in Figure 6.7. Each new planned trajectory's velocity is plotted in Figure 6.8. The resultant path of the object in the image is plotted in Figure 6.13. We see some very exciting results. For example, despite the fact that there is no initial velocity error in the $z$ direction, the

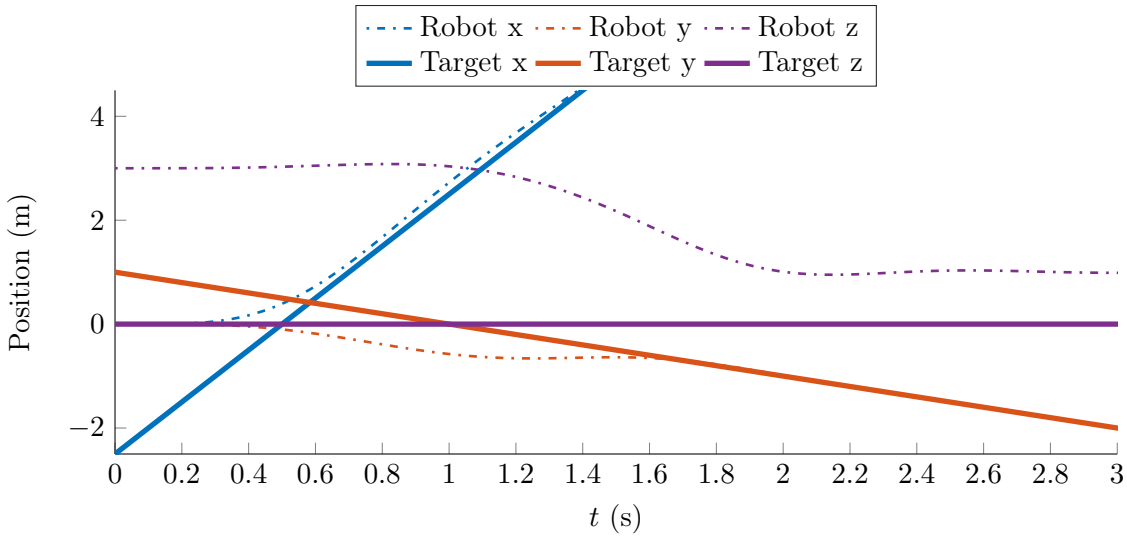



Figure 6.7: The nominal positions from the proposed planning strategy.

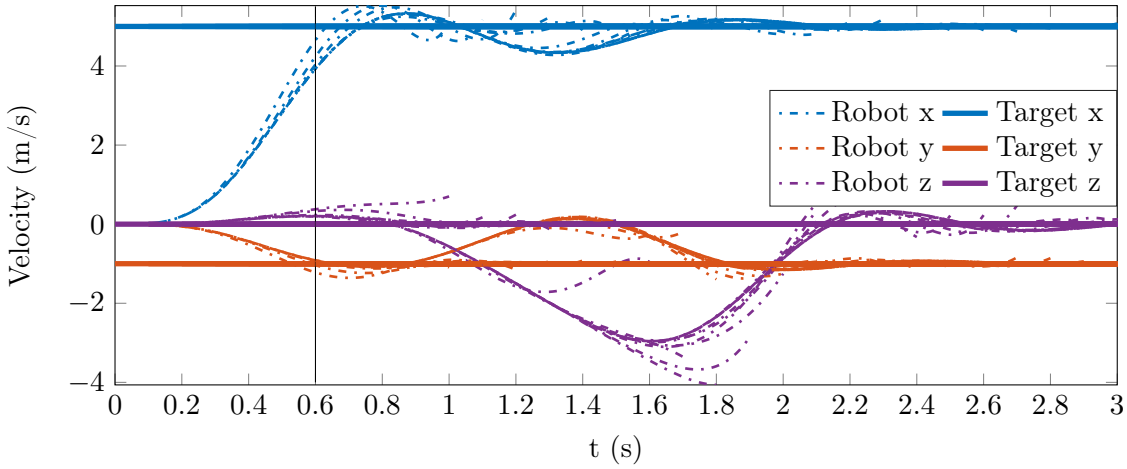



Figure 6.8: The planned velocities resulting from the receding horizion planner. All planned trajectories are shown as dash-dotted lines, but only the first 0.1 s of any trajectory is executed before the next one is planned. The planning horizon is 1 s. We observe interesting results like the initial positive velocity in $z$, which is possibly a result of the field of view constraint. At $t = 0.6$ s (represented by the partial vertical line), the position error is included in the cost function.

robot accelerates upward, helping to keep the object in the field of view (Figure 6.8). With the same initial conditions and minimizing the position error from the start, the motion is quickly dominated by the visual constraints and results in an infeasible problem.

Next we present our experimental results, which were executed in the GRASP (General Robotics Automation Sensing and Perception) lab at the University of Pennsylvania.

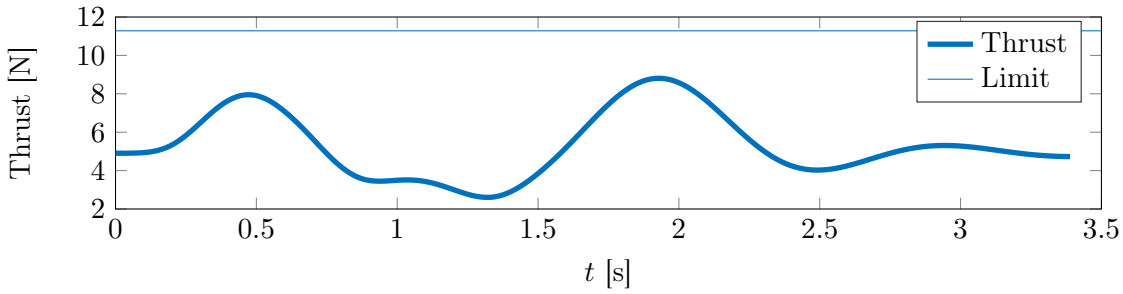



Figure 6.9: The forces required to execute the trajectory.

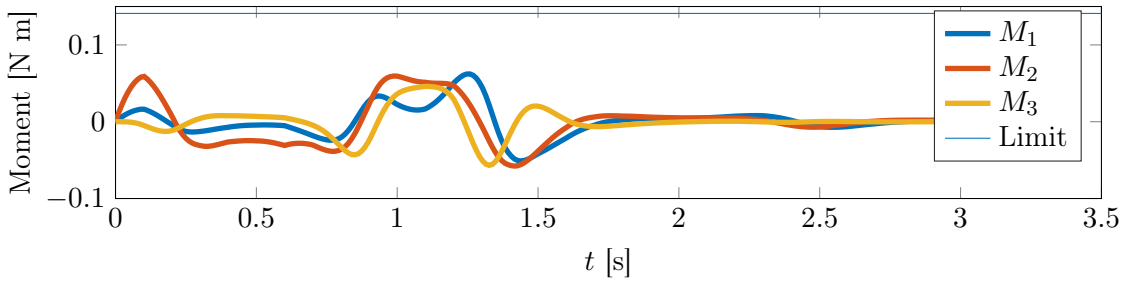



Figure 6.10: The moments required to execute the trajectory.

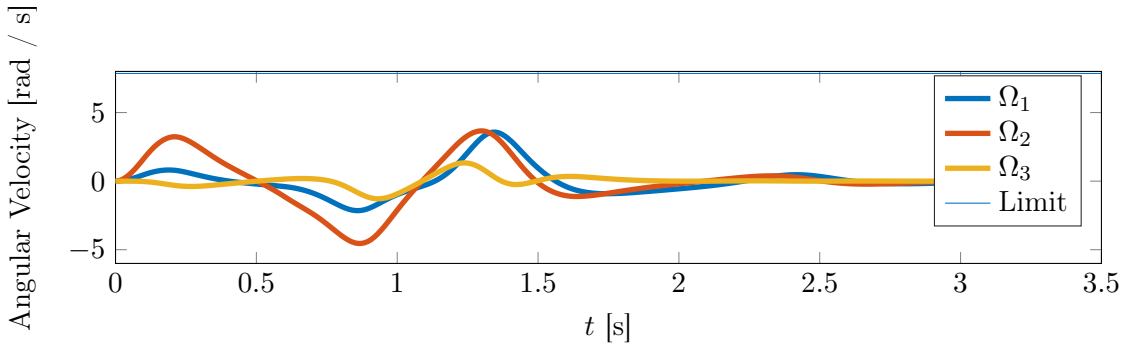



Figure 6.11: The angular velocities required to execute the trajectory.

The total flying area has a volume of $20 \times 6 \times 4$ m$^3$. We implemented the entire estimation, planning, and control pipeline, including the robot state estimator, the target's state estimator, a target trajectory predictor, and the trajectory planning algorithm on a Qualcomm Snapdragon$^{\text{TM}}$ board, featuring a Qualcomm Hexagon$^{\text{TM}}$ DSP and 802.11n Wi-Fi, all packed onto a ($58 \times 40$ mm) board based on the Snapdragon$^{\text{TM}}$ 801 processor. A Kalman filter is used for the robot's state estimation at 500 Hz with respect to a fixed reference frame. The estimation is obtained combining IMU data in the prediction phase and dis-

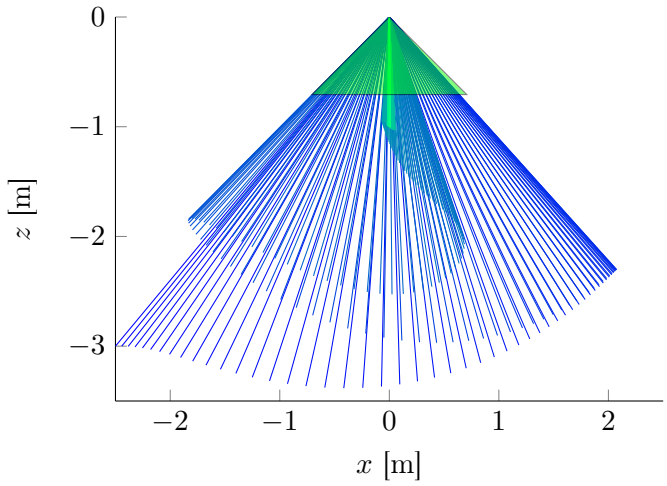



Figure 6.12: The relative positions projected in the $x$-$z$ plane of the robot frame. The camera's field of view is represented by the green triangle, and the object starts on the left (behind the robot). The relative position moves towards positive $x$ as the robot pitches forwards to accelerate.

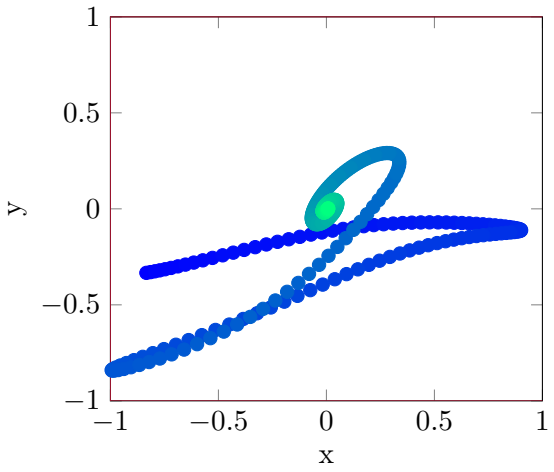



Figure 6.13: The path of the bearing to the target in the image, starting on the left and ending up centered. The image boundary is given by the solid boundaries. We observe that the field of view constraints are not violated.

tinguishable environment landmarks in the measurement update step. For more details, we refer the reader to our previous work, where reliable flights with speeds up to 5 m/s and angular rates of 800 deg/s are achieved [73]. The quadrotor has a diameter of only 15 cm, has a mass of 250 g, and uses a single downward-facing camera and IMU to perform both the state estimation of the robot and the target.

The target is a 7.6 cm diameter Sphero SPRK+ spherical robot, controllable using a

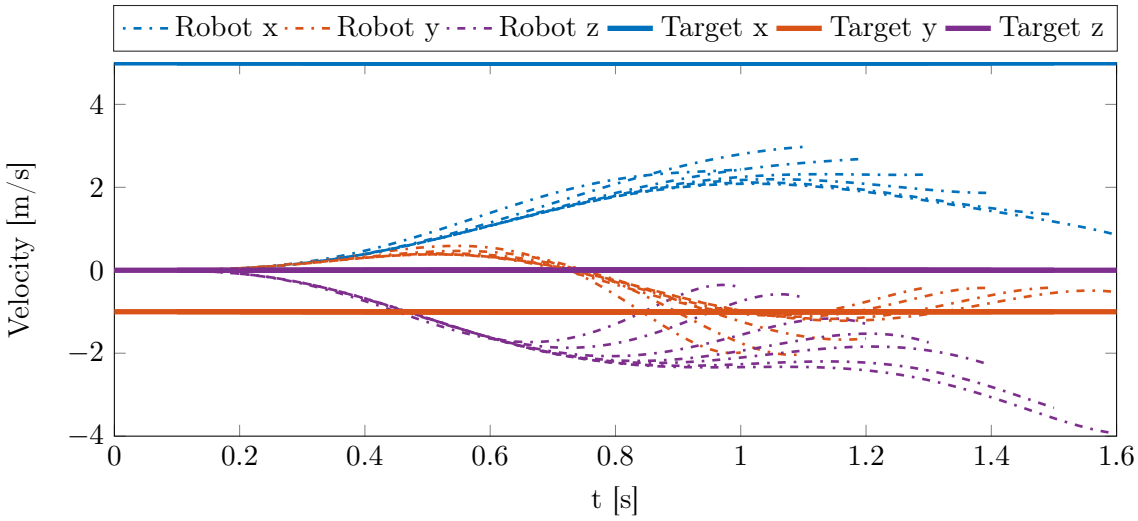



Figure 6.14: The velocities of the planned trajectories with the strategy that penalizes only position and jerk. In this case, the result quickly becomes unstable, and the motion is dominated by the field of view constraints.

smartphone via Bluetooth. A linear regression of the 10 most recent target pose measurements (at 30 Hz) is used to estimate the state of the Sphero assuming a constant-velocity model in the horizontal plane and a constant model in the $z$ direction. Then, the future trajectory of the target is predicted, providing the vector of coefficients, $\mathbf{h}(t)$, for the next horizon.

With this platform, we demonstrate successful results such as the tracking in Figure 6.16 and the acceleration to track a quickly moving target entering the field of view in Figure 6.15. In both cases, the object does not leave the field of view, and the quadrotor successfully tracks the target. For more results, including simulations of the approach applied to objects moving in 3D, we refer the reader to video located at `https://youtu.be/761z59TY54w`.

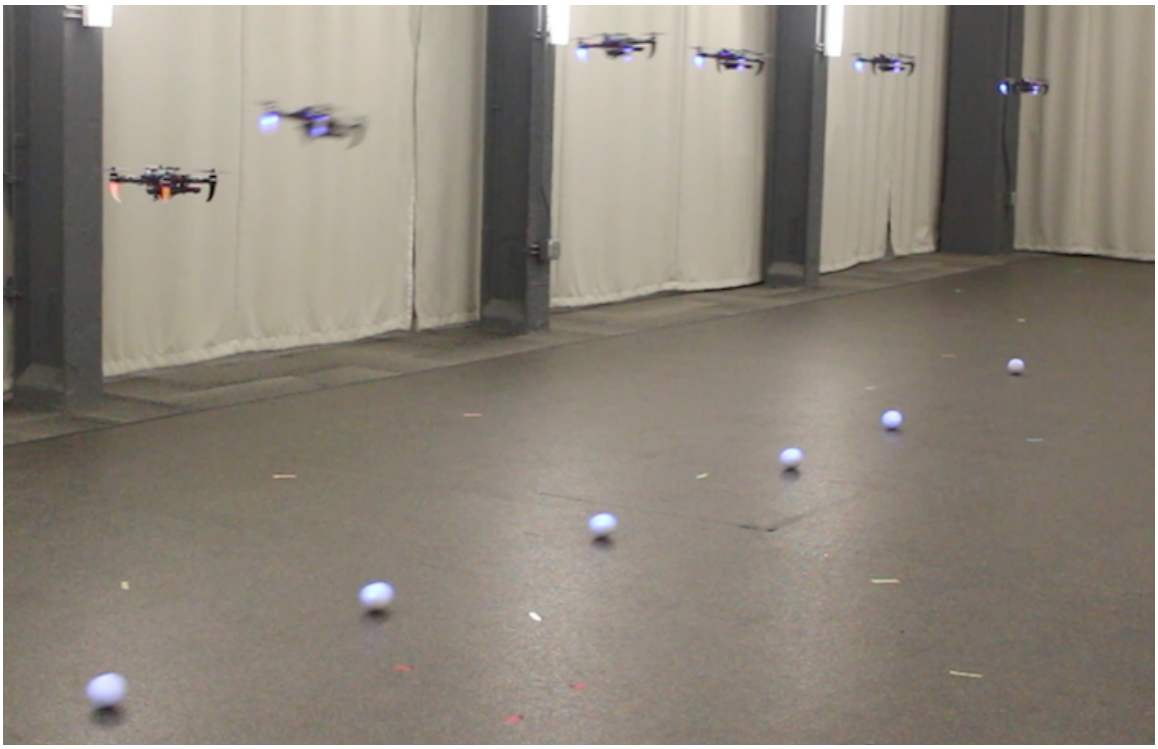


Figure 6.15: A quadrotor accelerates to track a target moving at 1.5 m/s. The motion is planned in real time and considers the estimated trajectory of the target, the dynamics of the robot, the actuator limitations, and the camera's field of view. All sensing and computation occurs onboard the robot and leverages only one downward-facing camera and an onboard inertial measurement unit.

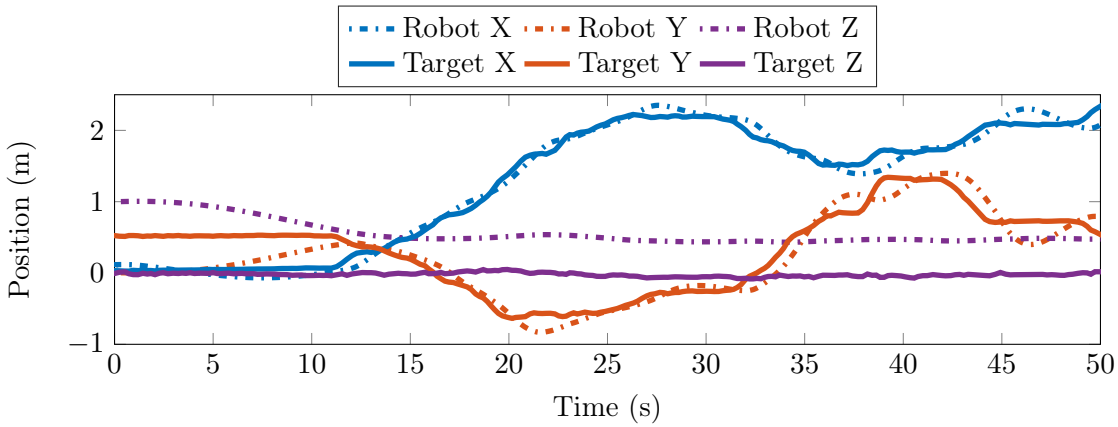



Figure 6.16: Experimental results of a robot tracking a rolling sphere for an extended period of time. The vehicle quickly locks onto the target and maintains the desired relative pose despite erratic motion in the target's path.

## 6.7 Conclusion

In this chapter, we presented a relative pose estimation and trajectory planning strategy to track a moving sphere with an underactuated micro aerial vehicle while considering the dynamic, actuator, and field of view constraints. We validated that all perception and computation can occur onboard a 250 g robot equipped with only one downward-facing camera and an inertial measurement unit. Simulation and experimental results demonstrate successful tracking of a moving target while keeping the object in the field of view.

There are many exciting research opportunities stemming from this work. For example, an online scale or range estimate is needed, which could be inspired from the strategies used by dragonflies. Optical flow (considering the parallax) between the target and background could also help improve the relative velocity estimation. Additionally, a better-informed strategy is needed to select the cost function weightings, $\lambda_i$. It would be interesting to investigate a complete planning strategy that could be plausible for real-time implementation. Even if not, a comparison exploring false negatives (predicted failures when a possible solution existed) of our approach with a complete approach would be compelling.

# Chapter 7

# Conclusion

This dissertation makes several contributions to facilitate interaction between aerial vehicles and their surroundings. First, we presented a novel solution that enabled a robot to grasp objects while moving at high speeds. This was inspired by the way birds of prey swing their legs backwards and reduce the relative velocity between their claws and the prey, giving their claws more time to grasp the target. The dynamics of the added appendage were considered in planning and compensated in the controller to demonstrate aerial grasping while moving at 9 body lengths per second.

Next, a systematic modeling and planning approach gave a quadrotor the ability to perch on inclined or vertical surfaces to help mitigate the challenge of limited flight time. The robot leveraged a downward-facing gripper, and the planner carefully considered the required landing conditions as well as the actuator and sensor constraints. As a departure from previous works, this method no longer required iterative learning to determine a successful trajectory. With this configuration, the robot was able to use the same gripper both for grasping and perching.

We next explored a vision-based strategy for perching on overhead cylindrical objects using only onboard sensors and computation. A monocular Image Based Visual Servoing (IBVS) controller was proposed and stability was proven using Lyapunov theory. Further, an image feature planner was developed, allowing for arbitrary initial conditions with only

information from the image and an Inertial Measurement Unit (IMU). With this approach, we successfully eliminated the dependence on a motion capture system.

This dissertation culminated in a comprehensive integration, enabling an aerial robot to accelerate to and track a sphere in motion using only onboard sensors for odometry and visual servoing. We extended our previous planning approaches considering the dynamic, actuator, and IMU constraints to also account for the dynamics of the target and the camera's field of view. The formulation was specifically designed to enable planning while meeting the demands of real-time scenarios using only the computation available on the 250 g aerial vehicle.

There are many opportunities for future research. The high-speed grasping strategy in Chapter 3 should be generalized to 3D and a use higher degree of freedom manipulator. Novel manufacturing methods could be explored to create appendages with lower inertia, which would increase agility in grasping and perching. Some of the results developed in the last chapter, such as the cone constraint or the receding horizon planner, may be useful to ensure the target ends up in the reachability cone of the gripper.

With regard to perching on inclined surfaces in Chapter 4, it will be necessary to explore recovery strategies for failed perch attempts. The development of multi-modal grippers could promote interaction with a wider variety of surfaces. There is also much to understand regarding the aerodynamic interactions during such perching maneuvers.

Our current strategy for vision based perching on a cylinder in Chapter 5 could be extended to consider more general surfaces of revolution or a wider variety of shapes. Surface features should also be leveraged to estimate axial motion when perching on cylindrical objects without local contour changes. For more realistic scenarios, the vehicle must autonomously detect possible perch locations. Our approach could also be reformulated and applied for perching on inclined planes. Theoretical and experimental comparisons between IBVS and PBVS methods for perching would allow for a better understanding of viable real-world approaches.

The moving-target tracking approach in Chapter 6 should be extended to consider plan-

ning and control directly in the image space and to enable tracking and grasping of arbitrary shapes. Additional image cues, such as the background optical flow, could aid in estimation of the scale or the relative velocities. Further, a sensitivity analysis would provide a better understanding of the accuracy requirements. Finally, our actuated grasping methods should be applied to achieve vision-based, high-speed grasping using a single camera for onboard estimation.

In this dissertation, we explored grasping, perching, and visual servoing using micro aerial vehicles. We presented planning strategies which respect the dynamic requirements, gyro saturations, actuator constraints, and field-of-view limitations. Further, we developed image-based methods for control and considered interaction with flat surfaces, cylinders, and spheres. Finally, we demonstrated vision-based tracking of a moving object while using only computation and sensing available onboard a micro aerial vehicle. The potential impacts are far-reaching, providing opportunities to mitigate battery life, improve aerial inspection, and engage in many facets of autonomous aerial manipulation.

# Bibliography


[1] "Ascending Technologies GmbH." [Online]. Available: http://www.asctec.de/ 28

[2] "Vicon Motion Systems, Inc." [Online]. Available: http://www.vicon.com 28

[3] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems*, vol. 34, no. 4, pp. 46–64, 2014. 4

[4] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-Based Model Predictive Control on a Quadrotor: Onboard Implementation and Experimental Results," in *IEEE Conference on Robotics and Automation*, 2012. 12

[5] M. L. Burroughs, K. Beauwen Freckleton, J. J. Abbott, and M. A. Minor, "A Sarrus-Based Passive Mechanism for Rotorcraft Perching," *Journal of Mechanisms and Robotics*, vol. 8, no. 1, p. 011010, 2015. 5, 34

[6] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, mar 2007. 7, 60

[7] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, ser. Lecture Notes in Control and Information Sciences, D. J. Kriegman, G. D. Hager, and A. S. Morse, Eds. London: Springer-Verlag, 1998, vol. 237, pp. 66–78. 8, 61

[8] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, dec 2006. 7, 60

[9] J. Chen, T. Liu, and S. Shen, "Tracking a Moving Target in Cluttered Environments Using a Quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 446–453. 86, 96, 102

[10] W. Chi, K. H. Low, K. H. Hoon, and J. Tang, "An Optimized Perching Mechanism for Autonomous Perching with a Quadrotor," in *International Conference on Robotics & Automation (ICRA)*. Hong Kong: IEEE, 2014, pp. 3109–3115. 34

[11] P. Corke and M. Good, "Dynamic effects in visual closed-loop systems," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 671–683, 1996. 8, 61

[12] P. I. Corke, "Dynamic Issues in Robot Visual-Servo Systems," in *Robotics Research*. London: Springer London, 1996, pp. 488–498. 8, 61

[13] R. Cory and R. Tedrake, "Experiments in Fixed-Wing UAV Perching," in *AIAA Guidance, Navigation and Control Conference and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, aug 2008, pp. 1–12. 5

[14] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, "Visual servoing via navigation functions," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 521–533, aug 2002. 8, 61

[15] N. Cowan and D. Chang, "Geometric visual servoing," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1128–1138, dec 2005. 8

[16] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A Prototype of an Autonomous Controller for a Quadrotor UAV," in *European Control Conference*. IEEE, 2007, pp. 4001 – 4008. 23

[17] K. L. Crandall and M. A. Minor, "UAV Fall Detection from a Dynamic Perch using Instantaneous Centers of Rotation and Inertial sensing," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 20052. IEEE, may 2015, pp. 4675–4679. 37

[18] L. Daler, A. Klaptocz, A. Briod, M. Sitti, and D. Floreano, "A perching mechanism for flying robots using a fibre-based adhesive," in *International Conference on Robotics and Automation (ICRA)*. Karlsruhe, Germany: IEEE, 2013, pp. 4433–4438. 34

[19] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, jun 2007. 59

[20] V. Desaraju, N. Michael, M. Humenberger, R. Brockers, S. Weiss, and L. Matthies, "Vision-based Landing Site Evaluation and Trajectory Generation Toward Rooftop Landing," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, 2014. 61

[21] C. Doignon, "An Introduction to Model-Based Pose Estimation and 3-D Tracking Techniques," in *Scene Reconstruction Pose Estimation and Tracking*. I-Tech Education and Publishing, jun 2007, no. June. 60

[22] A. M. Dollar and R. D. Howe, "The Highly Adaptive SDM Hand: Design and Performance Evaluation," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 585–597, feb 2010. 3, 15, 18

[23] ——, "Joint coupling design of underactuated hands for unstructured environments," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1157–1169, jun 2011. 18

[24] ——, "Towards grasping in unstructured environments: grasper compliance and configuration optimization," *Advanced Robotics*, vol. 19, no. 5, pp. 523–543, jan 2005. 4

[25] A. Dollar and R. Howe, "A robust compliant grasper via shape deposition manufacturing," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 2, pp. 154–161, apr 2006. 18

[26] C. E. Doyle, J. J. Bird, T. A. Isom, C. J. Johnson, J. C. Kallman, J. A. Simpson, R. J. King, J. J. Abbott, and M. A. Minor, "Avian-inspired passive perching mechanism for robotic rotorcraft," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2011, pp. 4975–4980. 5, 15, 34

[27] C. E. Doyle, J. J. Bird, T. A. Isom, J. C. Kallman, D. F. Bareiss, D. J. Dunlop, R. J. King, J. J. Abbott, and M. A. Minor, "An avian-inspired passive mechanism for quadrotor perching," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 506–517, 2013. 5, 34

[28] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision Based Position Control for MAVs Using One Single Circular Landmark," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 495–512, jan 2011. 85

[29] A. Elfes, S. S. Bueno, J. J. G. Ramos, E. C. de Paiva, M. Bergerman, J. R. H. Carvalho, S. M. Maeta, L. G. B. Mirisola, B. G. Faria, and J. R. Azinheira, "Modelling, Control and Perception for an Autonomous Robotic Airship," in *Control*. Springer Berlin Heidelberg, 2002, pp. 216–244. 3

[30] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, jun 1992. 7, 61

[31] M. A. Estrada, E. W. Hawkes, D. L. Christensen, and M. R. Cutkosky, "Perching and vertical climbing: Design of a multimodal robot," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2014, pp. 4215–4221. 37, 39

[32] H. Fakhry and W. Wilson, "A modified resolved acceleration controller for position-based visual servoing," *Mathematical and Computer Modelling*, vol. 24, no. 5-6, pp. 1–9, sep 1996. 8

[33] O. Faugeras, *Three-dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA, USA: MIT Press, 1993. 88

[34] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 324–334, sep 2010. 5

[35] A. Fitzgibbon and R. Fisher, "A Buyer's Guide to Conic Fitting." in *Proc. of the British Machine Vision Conference 1995*, no. February. British Machine Vision Association, 1995, pp. 51.1–51.10. 87, 89

[36] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, jun 1995. 22, 23, 46

[37] R. T. Fomena and F. Chaumette, "Visual Servoing from Spheres using a Spherical Projection Model," in *IEEE International Conference on Robotics and Automation*. IEEE, apr 2007, pp. 2080–2085. 85

[38] R. Fomena and F. Chaumette, "Improvements on Visual Servoing From Spherical Targets Using a Spherical Projection Model," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 874–886, aug 2009. 85, 103

[39] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation*. Hong Kong: IEEE, may 2014, pp. 15–22. 60

[40] W. Gander, G. H. Golub, and R. Strebel, "Least-squares fitting of circles and ellipses," *BIT*, vol. 34, no. 4, pp. 558–578, dec 1994. 87

[41] P. E. Gill and E. Wong, *Mixed Integer Nonlinear Programming*, ser. The IMA Volumes in Mathematics and its Applications, J. Lee and S. Leyffer, Eds. New York, NY: Springer New York, 2012, vol. 154. 102

[42] N. Guenard, T. Hamel, and R. Mahony, "A Practical Visual Servo Control for an Unmanned Aerial Vehicle," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, apr 2008. 8, 61

[43] T. Hamel and R. Mahony, "Visual Servoing of an Under-Actuated Dynamic Rigid-Body System: An Image-Based Approach," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 187–198, apr 2002.

[44] ——, "Image based visual servo control for a class of aerial robotic systems," *Automatica*, vol. 43, no. 11, pp. 1975–1983, nov 2007. 8, 61

[45] E. W. Hawkes, H. Jiang, and M. R. Cutkosky, "Three-dimensional dynamic surface grasping with dry adhesion," *The International Journal of Robotics Research*, pp. 1–16, jun 2015. 36, 37, 38

[46] E. W. Hawkes, D. L. Christensen, E. V. Eason, M. A. Estrada, M. Heverly, E. Hilgemann, H. Jiang, M. T. Pope, A. Parness, and M. R. Cutkosky, "Dynamic surface grasping with directional adhesion," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, nov 2013, pp. 5487–5493. 4, 34

[47] E. W. Hawkes, H. Jiang, and M. R. Cutkosky, "Three-dimensional dynamic surface grasping with dry adhesion," *The International Journal of Robotics Research*, jun 2015. 4

[48] B. Herisse, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 77–89, feb 2012. 86

[49] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, pp. 182–201, 2013. 7, 60

[50] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996. 7, 60, 68

[51] H. Jabbari, G. Oriolo, and H. Bolandi, "Dynamic IBVS control of an underactuated UAV," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, dec 2012, pp. 1158–1163. 8, 61

[52] H. Jiang, E. W. Hawkes, V. Arutyunov, J. Tims, C. Fuller, J. P. King, C. Seubert, H. L. Chang, A. Parness, and M. R. Cutkosky, "Scaling controllable adhesives to grapple floating objects in space," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 2828–2835. 36

[53] H. Jiang, M. T. Pope, M. A. Estrada, B. Edwards, M. Cuson, E. W. Hawkes, and M. R. Cutkosky, "Perching Failure Detection and Recovery with Onboard Sensing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2015, pp. 1264–1270. 37

[54] H. Jiang, M. T. Pope, E. W. Hawkes, D. L. Christensen, M. A. Estrada, A. Parlier, R. Tran, and M. R. Cutkosky, "Modeling the dynamics of perching with opposed-grip mechanisms," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2014, pp. 3102–3108. 34, 37

[55] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, jan 2011. 7, 60

[56] A. Kalantari, K. Mahajan, D. Ruffatto, and M. Spenko, "Autonomous perching and take-off on vertical walls for a quadrotor micro air vehicle," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 4669–4674. 5, 34

[57] K. Kanatani, "Computational projective geometry," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 333–348, nov 1991. 60

[58] S. A. Kane and M. Zamani, "Falcons pursue prey using visual motion cues: new perspectives from animal-borne cameras," *Journal of Experimental Biology*, vol. 217, no. 2, pp. 225–234, jan 2014. 97

[59] H. Kase, N. Maru, A. Nishikawa, S. Yamada, and F. Miyazaki, "Visual servoing of the manipulator using the stereo vision," in *IEEE Industrial Electronics*. Maui, HI: IEEE, 1993, pp. 1791–1796. 8, 61

[60] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *The International Journal of Robotics Research*, vol. 30, pp. 56–79, 2011. 7, 60

[61] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 5, no. 1, pp. 39–48, mar 2000. 8, 61

[62] C. C. Kessens and J. P. Desai, "A Self-Sealing Suction Cup Array for Grasping," *Journal of Mechanisms and Robotics*, vol. 3, no. 4, p. 045001, 2011. 4

[63] C. C. Kessens, J. Thomas, J. P. Desai, and V. Kumar, "Versatile Aerial Grasping Using Self-Sealing Suction," in *IEEE International Conference on Robotics and Automation*. Stockholm: IEEE, 2016. 4

[64] J. Kim, Y. Jung, D. Lee, and D. H. Shim, "Outdoor Autonomous Landing on a Moving Platform for Quadrotors using an Omnidirectional Camera," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. Orlando, FL: IEEE, may 2014, pp. 1243–1252. 84

[65] M. Kovac, "Learning from nature how to land aerial robots," *Science*, vol. 352, no. 6288, pp. 895–896, may 2016. 34

[66] M. Kovač, J. Germann, C. Hürzeler, R. Y. Siegwart, and D. Floreano, "A perching mechanism for micro aerial vehicles," *Journal of Micro-Nano Mechatronics*, vol. 5, no. 3-4, pp. 77–91, dec 2009. 5, 33

[67] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, aug 2012. 15, 58

[68] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2012, pp. 971–976. 61, 86

[69] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE Conference on Decision and Control*. IEEE, dec 2010, pp. 5420–5425. 12, 13, 44, 70, 94

[70] ——, "Nonlinear Robust Tracking Control of a Quadrotor UAV on SE(3)," *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, mar 2013. 72, 73

[71] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of Cubic Structures with Quadrotor Teams," in *Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, 2011. 1, 4, 15

[72] ——, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012. 4

[73] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, Control, and Planning for Aggressive Flight With a Small Quadrotor With a Single Camera and IMU," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, apr 2017. 103, 107

[74] G. Loianno, Y. Mulgaonkar, C. Brunner, D. Ahuja, A. Ramanandan, M. Chari, S. Diaz, and V. Kumar, "Smartphones power flying robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2015, pp. 1256–1263. 7

[75] G. Loianno, J. Thomas, and V. Kumar, "Cooperative Localization and Mapping of MAVs using RGB-D Sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*. Seattle: IEEE, 2015, pp. 4021–4028. 7

[76] A. Lussier Desbiens, A. T. Asbeck, and M. R. Cutkosky, "Landing, perching and taking off from vertical surfaces," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 355–370, jan 2011. 5, 33

[77] A. Lussier Desbiens and M. R. Cutkosky, "Landing and Perching on Vertical Surfaces with Microspines for Small Unmanned Air Vehicles," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 313–327, jan 2010. 33

[78] R. Mahony and S. Stramigioli, "A port-Hamiltonian approach to image-based visual servo control for dynamic systems," *The International Journal of Robotics Research*, vol. 31, pp. 1303–1319, 2012. 8, 61

[79] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, dec 2005. 80, 92

[80] É. Marchand and F. Chaumette, "Virtual Visual Servoing: a framework for real-time augmented reality," *Computer Graphics Forum*, vol. 21, no. 3, pp. 289–297, sep 2002. 60

[81] A. Martinelli, "Visual-inertial structure from motion: Observability and resolvability," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, nov 2013, pp. 4235–4242. 7

[82] J. Matas, C. Galambos, and J. Kittler, "Robust Detection of Lines Using the Progressive Probabilistic Hough Transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119–137, apr 2000. 80

[83] M. McBeath, D. Shaffer, and M. Kaiser, "How baseball outfielders determine where to run to catch fly balls," *Science*, vol. 268, no. 5210, pp. 569–573, apr 1995. 97

[84] R. Mebarki, V. Lippiello, and B. Siciliano, "Toward image-based visual servoing for cooperative aerial manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 6074–6080. 61

[85] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2011, pp. 2520–2525. 10, 13, 23, 26, 46, 50, 78

[86] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2011, pp. 2668–2673. 4, 16

[87] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 5, pp. 664–674, jan 2012. 5, 34, 52

[88] ——, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and G. Sukhatme, Eds. Springer Berlin Heidelberg, 2014, vol. 79, pp. 361–373. 5, 15, 28, 34

[89] D. Mellinger, M. Shomin, and V. Kumar, "Control of Quadrotors for Robust Perching and Landing," in *International Powered Lift Conference (IPLC)*, 2010, pp. 1–8. 1, 16, 34

[90] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *International Symposium on Distributed Autonomous Robotic Systems*, 2010. 1, 5

[91] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP Multiple Micro-UAV Testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, sep 2010. 3, 10, 12, 53, 78

[92] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, sep 2012. 6

[93] M. Mischiati, H.-T. Lin, P. Herold, E. Imler, R. Olberg, and A. Leonardo, "Internal models direct dragonfly interception steering," *Nature*, vol. 517, no. 7534, pp. 1–13, 2014. 97

[94] K. Mohta, V. Kumar, and K. Daniilidis, "Vision-based control of a quadrotor for perching on lines," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2014, pp. 3130–3136. 34

[95] J. Moore, R. Cory, and R. Tedrake, "Robust post-stall perching with a simple fixed-wing glider using LQR-Trees." *Bioinspiration & Biomimetics*, vol. 9, p. 025013, 2014. 33

[96] J. Moore and R. Tedrake, "Powerline Perching with a Fixed-Wing UAV," in *AIAA Infotech@Aerospace Conference*. Reston, Virigina: American Institute of Aeronautics and Astronautics, apr 2009, pp. 1–16. 33

[97] M. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *European Control Conference (ECC)*, 2013, pp. 1383–1389. 101

[98] R. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME International Congress and Exposition*. Citeseer, 1995, pp. 1–9. 22, 23, 46, 49

[99] N. Navab, "Canonical Representation and Three-View Geometry of Cylinders," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXIV, 2002, pp. 218–224. 60

[100] N. Navab and M. Appel, "Canonical Representation and Multi-View Geometry of Cylinders," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 133–149, 2006. 60, 62

[101] R. M. Olberg, A. H. Worthington, and K. R. Venator, "Prey pursuit and interception in dragonflies," *Journal of Comparative Physiology A: Sensory, Neural, and Behavioral Physiology*, vol. 186, no. 2, pp. 155–162, feb 2000. 97

[102] R. M. Olberg, "Visual control of prey-capture flight in dragonflies," *Current Opinion in Neurobiology*, vol. 22, no. 2, pp. 267–271, 2012. 97

[103] L. Peper, R. J. Bootsma, D. R. Mestre, and F. C. Bakker, "Catching balls: how to get the hand to the right place at the right time." *Journal of experimental psychology. Human perception and performance*, vol. 20, no. 3, pp. 591–612, jun 1994. 97

[104] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2011, pp. 2491–2498. 4

[105] P. E. I. Pounds and A. M. Dollar, "UAV rotorcraft in compliant contact: Stability analysis and simulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2011, pp. 2660–2667. 4

[106] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of Aerodynamics and Proximity Effects in Quadrotor Flight," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, J. P. Desai, G. Dudek, O. Khatib, and V. Kumar, Eds. Springer International Publishing, 2013, vol. 88, pp. 289–302. 55

[107] L. Quan, "Conic reconstruction and correspondence from two views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 151–160, 1996. 60

[108] I. Sa, S. Hrabar, and P. Corke, "Inspection of Pole-Like Structures Using a Vision-Controlled VTOL UAV and Shared Autonomy," in *IEEE International Conference on Intelligent Robots and Systems*. Chicago, IL: IEEE, 2014, pp. 4819–4826. 61

[109] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier, "Vision-Controlled Micro Flying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014. 60

[110] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3D exploration with a micro-aerial vehicle," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 9–15. 1, 6

[111] ——, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 5303–5310. 6, 7

[112] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor," in *Robotics: Science and Systems*, Berlin, Germany, 2013. 6, 59

[113] ——, "Initialization-Free Monocular Visual-Inertial State Estimation with Application to Autonomous MAVs," in *International Symposium on Experimental Robotics*, Marrakech, Morocco, 2014. 60

[114] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 100

[115] R. Spica, A. Franchi, G. Oriolo, H. H. Bulthoff, and P. R. Giordano, "Aerial grasping of a moving target with a quadrotor UAV," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2012, pp. 4985–4992. 86

[116] R. Spica, P. R. Giordano, and F. Chaumette, "Active structure from motion for spherical and cylindrical targets," in *IEEE International Conference on Robotics and Automation*. Hong Kong: IEEE, may 2014, pp. 5434–5440. 8, 61, 62, 92

[117] K. Sreenath and V. Kumar, "Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots," in *Robotics: Science and Systems (RSS)*, Berlin, Germany, 2013. 1, 5

[118] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load," in *IEEE Conference on Decision and Control (CDC)*. IEEE, dec 2013, pp. 2269–2274. 1, 74

[119] K. Sreenath, N. Michael, and V. Kumar, "Trajectory Generation and Control of a Quadrotor with a Cable-Suspended Load - A Differentially-Flat Hybrid System," in *International Conference on Robotics and Automation*, 2013. 4, 15

[120] M. V. Srinivasan, "Visual control of navigation in insects and its relevance for robotics," *Current Opinion in Neurobiology*, vol. 21, no. 4, pp. 535–543, 2011. 97

[121] Suseong Kim, Seungwon Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 4990–4995. 4

[122] S. Tang and V. Kumar, "Mixed Integer Quadratic Program Trajectory Generation for a Quadrotor with a Cable-Suspended Payload," in *International Conference on Robotics & Automation (ICRA)*, 2015, pp. 2216–2222. 4

[123] C. J. Taylor and J. P. Ostrowski, "Robust vision-based pose control," in *IEEE International Conference on Robotics and Automation*, vol. 3, no. April. IEEE, 2000, pp. 2734–2740. 8, 61

[124] The Slow Mo Guys, "Red Kites in Slow Motion." [Online]. Available: http://youtu.be/AYOx-iCMZhk 5, 16, 30

[125] J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "Visual Servoing of Quadrotors for Perching by Hanging From Cylindrical Objects," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 57–64, jan 2016. 58, 84, 86

[126] ——, "Visual Servoing of Quadrotors for Perching by Hanging from Cylindrical Objects: Supplementary Material," 2016. [Online]. Available: http://repository.upenn.edu/meam_papers/300/ 81

[127] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, "Toward autonomous avian-inspired grasping for micro aerial vehicles," *Bioinspiration & Biomimetics*, vol. 9, no. 2, p. 025010, jun 2014. 8, 16, 61

[128] J. Thomas, G. Loianno, M. Pope, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Planning and Control of Aggressive Maneuvers for Perching on Inclined and Vertical Surfaces," in *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Boston: ASME, 2015, pp. 1–10. 5, 33, 35

[129] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *IEEE International Conference on Robotics and Automation*. Hong Kong: IEEE, may 2014, pp. 2113–2118. 8, 61

[130] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-Inspired Grasping For Quadrotor Micro Aerial Vehicles," 2013. [Online]. Available: https://youtu.be/ol8c9bdp7YI 30

[131] ——, "Avian-Inspired Grasping for Quadrotor Micro UAVs," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Portland: ASME, aug 2013, p. V06AT07A014. 5, 16

[132] J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Aggressive Flight for Perching on Inclined Surfaces," *Journal of Mechanisms and Robotics*, vol. 8, no. 5, may 2016. 5, 33, 35

[133] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar, "Autonomous Flight for Detection, Localization, and Tracking of Moving Targets With a Small Quadrotor," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1762–1769, jul 2017. 85

[134] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012. 6

[135] H. Tsukagoshi, M. Watanabe, T. Hamada, D. Ashlih, and R. Iizuka, "Aerial manipulator with perching and door-opening capability," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 4663–4668. 5, 34

[136] M. Turpin, N. Michael, and V. Kumar, "Trajectory design and control for aggressive formation flight with quadrotors," *Autonomous Robots*, vol. 33, no. 1-2, pp. 143–156, aug 2012. 98

[137] N. Venable, *Birds of Prey*. West Virginia University Extension Service, 1996. 15

[138] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, nov 2011. 1, 6, 7, 60

[139] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 61, no. 1-4, pp. 221–238, 2011. 86

[140] H. Zhang and J. Ostrowski, "Visual Servoing with Dynamics: Control of an Unmanned Blimp," in *International Conference on Robotics and Automation*, vol. 1, no. May. IEEE, 1999, pp. 618–623. 8, 93

[141] Z. Zhang, P. Xie, and O. Ma, "Bio-inspired trajectory generation for UAV perching," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2013, pp. 997–1002. 34