Soft Aerial Manipulation

by

Joshua Fishman

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author . . . .
Department of Mechanical Engineering
May 14, 2021

Certified by . . . .
Luca Carlone
Assistant Professor, Aeronautics and Astronautics
Thesis Supervisor

Certified by . . . .
Alberto Rodriguez
Associate Professor, Mechanical Engineering
Thesis Reader

Accepted by . . . .
Nicolas G. Hadjiconstantinou
Chairman, Committee for Graduate Students

Soft Aerial Manipulation

by

Joshua Fishman

Submitted to the Department of Mechanical Engineering
on May 14, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

This thesis explores the theory and implementation of a *soft drone*, consisting of a quadrotor and a tendon-actuated soft gripper, which for the first time fully exploits the advantages of softness in aerial manipulation. Manipulation and grasping with unmanned aerial vehicles (UAVs) currently require accurate positioning and are often executed at reduced speed to ensure successful grasps. This is because modern aerial manipulation platforms employ rigid manipulators with few degrees of freedom, limiting their capability to compensate for disturbances caused by the vehicle positioning errors and maintain stability despite external contact forces. Biological systems, on the other hand, exploit softness to overcome similar limitations, and leverage compliance to enable aggressive grasping. To the best of our knowledge, ours is the first work at the intersection between soft manipulation and UAV control.

We present a control and planning approach for the *soft drone* (quadrotor and soft gripper), *decoupling* the two subsystems and employing (i) a geometric controller and a minimum-snap trajectory optimization for the quadrotor (rigid) base, and (ii) a quasi-static finite element model and control-space interpolation for the soft gripper. We prove that the geometric controller asymptotically stabilizes the quadrotor velocity and attitude despite the addition of the soft load. Next, we describe our soft drone prototype, including electro-mechanical design, software infrastructure, and fabrication. Finally, we evaluate the proposed system in a realistic soft dynamics simulator (SOFA) and in real tests, and show that: (i) the geometric controller is fairly insensitive to the soft payload, (ii) in simulation, our soft drone outperforms more rigid alternatives, (iii) the platform can reliably grasp unknown objects despite inaccurate positioning and initial conditions, both in simulation and in real testing. Our soft drone can grasp at up to 2 m/s in simulation and consistently grasps at 0.2 m/s in real tests (91.7% success rate).

Video attachments:
https://youtu.be/NNpQxP0SPFk
https://youtu.be/mqbj8mEyCdk

Thesis Supervisor: Luca Carlone
Title: Assistant Professor, Aeronautics and Astronautics

Thesis Reader: Alberto Rodriguez
Title: Associate Professor, Mechanical Engineering

Acknowledgments

I am deeply grateful to my advisor, Professor Luca Carlone, to my teammates Samuel Ubellacker and Nathan Hughes, and to the whole MIT SPARK Lab – I couldn't ask for a better mentor, team or group to learn from and with. I also want to express my appreciation to Professor Alberto Rodriguez, for his generosity and thoughtfulness as a reader for this thesis. It was a pleasure and honor to work with you all.

I also really appreciate the support of my parents, my grandparents, my brother and sister, my cousins and my whole family – I know I never did a great job of explaining what I was actually doing, but that never stopped you from acting interested in it and asking great questions!

Finally, I'd like to thank my girlfriend Bethany and my roommates over the past 2 years: Rogers, Nathaniel, Martin, Jesse, Zach, Alon, Sammy and Simon. Through thick and thin, final projects and unprecedented pandemics, late nights and random board games, you've have been there. I would not have made it through the past 2 years without your constant support, and I'll never forget it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Aerial manipulation – intentional physical interaction with the world with a flying platform – is a fundamental capability for autonomous aerial systems with the potential to transform numerous applications, including: autonomous transportation and construction [29], medical goods delivery [50], agriculture and forestry (water sampling [39], forest canopy sampling [23]), infrastructure monitoring and maintenance [5], and autonomous charging via perching [37], among others [24].

Quadrotors have been researched extensively as platforms for navigation and inspection [13, 5] due to their versatility and maneuverability, but have several limitations when used for manipulation. First, small quadrotors (often called *micro aerial vehicles* [29]) have limited payload, and can therefore only carry relatively simple and lightweight manipulators. This intrinsically limits their capability to compensate for disturbances, such as those caused by vehicle positioning errors during grasp execution. Second, aerial systems are inherently fragile and imprecise [11]; unplanned external forces (due to contact, aerodynamic effects etc.) need to be mitigated or avoided in order to preserve stability [24]. Many works circumvent these issues by reducing the speed and accelerations of the quadrotor [10, 43]. This reduces the magnitude of external contact forces but has the drawback of making operation inefficient, especially considering the short flight time of small quadrotors. One exception,

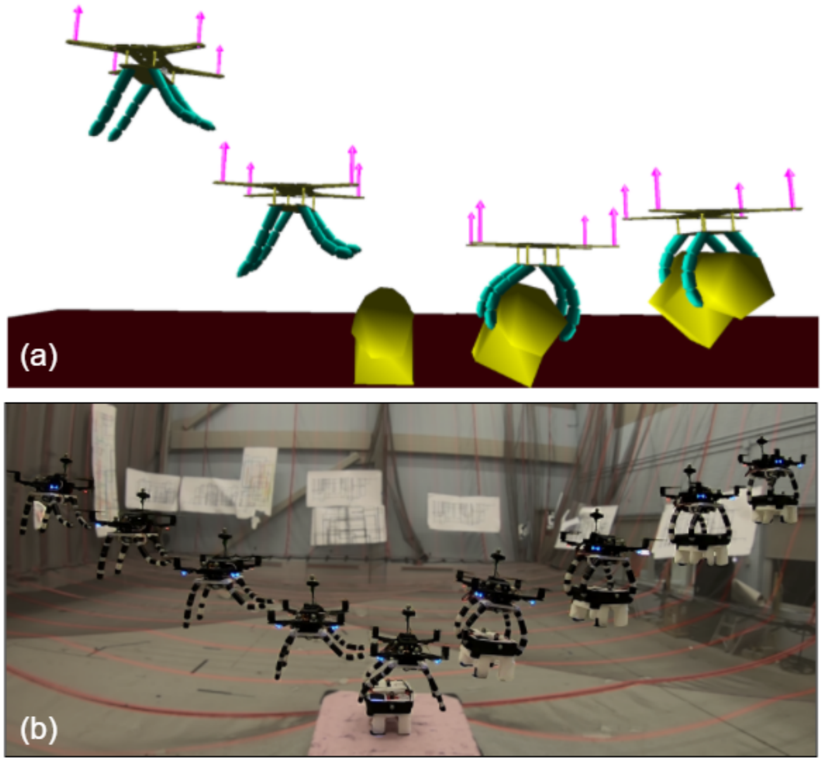


Figure 1-1: We investigate control and trajectory optimization for a *soft aerial manipulator*, consisting of a quadrotor (yellow frame with propeller thrusts in magenta) and a tendon-actuated soft gripper (cyan). The figure shows a temporal sequence leading to a successful grasp in a realistic soft dynamics simulator (a) and in real tests (b).

Thomas *et al.* [51], demonstrates aggressive aerial grasping at up to 2 m/s but the problem setup is simplified to avoid unplanned contact forces (*i.e.,* the object to grasp is extremely light and suspended rather than lying on a surface).

The recent literature bears witness to an increasing interest in compliant aerial manipulation as a means of mitigating these limitations. Compliance and underactuation are now widely exploited, to enable the manipulation of objects of varying shape and to minimize disturbances imposed by the environment [2]. However, to our knowledge this has been restricted to cases where the compliant elements either have limited degrees of freedom (often due to payload constraints) or do not exert a torque on the quadrotor (*e.g.,* cable-slung loads).

On the other hand, soft materials are ubiquitous in nature and enable the performance and robustness which so differentiate natural from artificial systems [3]. Soft manipulators passively conform to the grasped object, enabling tolerance to imprecisions and reducing the need for explicit grasp analysis; this is an example of

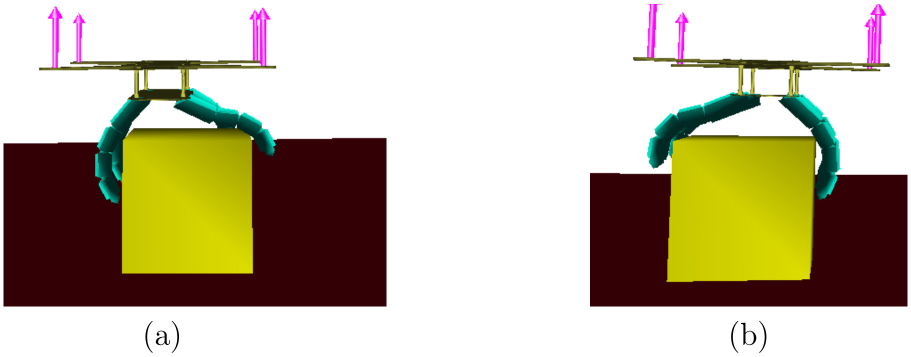


Figure 1-2: Morphological computation refers to the ability of an underactuated physical system to supplement explicit control with passive deformation. Here, we show our soft gripper achieving successful grasps despite significant variations in the quadrotor position; the gripper command is the same in (a)-(b).

morphological computation, the exploitation of passive mechanical elements to supplement explicit control [44]. Despite the potential to use soft grippers as a lightweight and compliant alternative for aerial manipulation, little attempt has yet been made to explicitly model and control continuously deformable, soft structures in an aerial context. In general, such soft elements are continuously deformable and possess theoretically infinite degrees of freedom. Therefore, they cannot be modeled in closed form and are not differentially flat, putting them at odds with typical techniques for UAV control [52]. However, modern techniques and approximations of soft robotics enable principled modeling, planning and control for soft systems. These have yet to be extended to the aerial domain.

1.2 Thesis Overview

This thesis bridges the quadrotor control and planning literature with the growing field of soft robotics. In particular, we present a soft aerial manipulator (our soft drone– Fig. 1-1), investigate control and trajectory optimization algorithms to enable aggressive grasping of an object lying on a surface and describe the real evaluation and testing of our system.

After reviewing the literature in Section 1.3, Chapter 2 presents a theoretical solution to the planning and control problem of grasping with a soft drone. Section 2.1

describes the proposed soft drone and states the grasping problem in terms of the concurrent planning and control of both the rigid quadrotor base and the soft gripper. The section also provides an overview of the proposed algorithmic approach, based on decoupling the control and planning for the rigid and soft components; this is made possible by the insight in Theorem 2 and the resilience to positioning errors afforded by the soft gripper (Fig. 1-2).

Section 2.2 describes the control law and the trajectory optimization approach for the soft gripper. We assume that the gripper remains in quasi-static equilibrium and compute forward kinematics by minimizing total energy (defined via finite element methods) using Newton's method. We then compute optimal tendon control with a gradient descent methodology and linearly interpolate these over the length of the trajectory. In defining an objective function for the gripper we assume that the quadrotor attains its nominal trajectory, relying on the inherent adaptability of the soft gripper to compensate for deviations (again, Fig. 1-2). Section 2.3 reviews a standard geometric controller for a quadrotor. The novel insight here is that by modeling the gripper as a symmetric soft payload and treating torque imposed by it as a disturbance, we can prove that the geometric controller stabilizes the quadrotor velocity and attitude irrespective of the soft gripper (Theorem 2). Moreover, we use minimum-snap polynomial optimization to plan a trajectory satisfying grasp constraints, while abstracting away the manipulation aspects (accounted for by the gripper).

Next, Chapter 3 describes our soft drone implementation, including material choice, design principles, electronics and fabrication. Section 3.1 gives an overview of the soft drone prototype: we develop a modular tendon-actuated soft gripper, consisting of silicone fingers attached to a 3-D printed base and controlled by an *Arduino Due* microcontroller over a USB connection and mounted to an *Intel Aero Ready-to-Fly* quadrotor. Section 3.2 describes the design of our soft fingers: drawing inspiration from biology, we maximize their adaptability and utility by choosing proportions modeled after a human finger. Section 3.4 reports the development of the 3-D printed gripper base, including internal tendon routing, considerations of modular attachment points for optional components and the development of a simple method to easily independently adjust tendon rest lengths. Section 3.3 describes the choice and layout of our electronics, including a microcontroller, motor drivers, motors/encoders and a voltage regulator.

Chapter 4 reports our results. Section 4.1 presents numerical experiments performed in a realistic soft dynamics simulator, *SOFA* [14]. The experiments highlight the effectiveness of the proposed system and show that: (i) the geometric controller is fairly insensitive to the soft payload, (ii) the platform can reliably grasp unknown objects at up to 2 m/s despite inaccurate positioning and starting from a variety of initial conditions, and (iii) our soft aerial manipulation platform outperforms more-rigid alternatives. Section 4.2 presents the results of grasping tests at 0.2 m/s with our real platform and discusses discrepancies from simulation. We grasp successfully on 21 out of 23 attempts, despite the presence of unmodeled aerodynamic and other effects; moreover, our 2 failures were due to catastrophic state estimation divergences unrelated to the novel attributes of the soft drone.

Finally, Chapter 5 summarizes the conclusions of this thesis while Chapter 6 (appendices) reports the detailed mathematical implementation of the soft material model we use, as well as a proof of stability for our complete system.

1.3 Related Work

Aerial Manipulation. Aerial manipulation with under-actuated and un-actuated components has been subject of extensive research; however, this has been restricted to components with limited degrees of freedom and in which the combined system is differentially flat. Cable-slung loads (Föhn *et al.* [17], Sreenath *et al.* [49]) are a well-studied under-actuated payload, but are differentially flat and are typically designed not to impose a torque on the aerial platform. The continuously deformable cable is either treated as massless or reduced to a finite number of links (Goodarzi *et al.* [20]).

Pounds *et al.* [40] develop an influential underactuated gripper with rigid links and compliant joints (the so-called *SDM Hand*) specifically for aerial manipulation.

Backus *et al.* [2] optimize tendon routing and palm size for manipulators of this type, using a mathematical model to minimize the tendon forces required to grasp. Thomas *et al.* [51] carry out aggressive aerial manipulation inspired by birds of prey at up to 2 m/s, employing a gripper based on the *SDM Hand*; their work focuses on the case of a very light suspended object (26 g), minimizing disturbances due to collisions and unmodeled aerodynamic effects. The AEROARMS project (Caballero *et al.* [8]) explored the use of a manipulator with a flexible link to minimize disturbance on the aerial platform, but incorporated a passive joint between the flexible link and the quadrotor body specifically to avoid considering the impact of compliance on the drone dynamics. Yuksel *et al.* [52] show differential flatness for aerial manipulators with arbitrary but finite numbers of rigid or compliant joints; this result does not hold for the continuously deformable case.

Morphing Drones. While not being targeted at manipulation, a recent set of papers investigates the design of UAV platforms that can mechanically change shape to tolerate collisions or fit into narrow gaps. Mintchev *et al.* [36] use insect-inspired structural compliance in a quadrotor frame to minimize impact damage, but this compliance does not affect the quadrotor dynamics during flight. Falanga *et al.* [13] develop a drone that can fold its arms to fit into narrow gaps; morphological adaptability is limited to the plane so that the resulting dynamics can be expressed in closed form. Ramon *et al.* [41] propose soft landing gear (similar in spirit to our design), but do not model the soft component nor its interaction with the quadrotor controller or control the soft landing gear beyond a binary open/close; moreover, the work focuses on landing rather than manipulation. Deng *et al.* [9] implement and control a soft-bodied multicopter in simulation; their work relies on a gray-box neural network for system identification and does not consider manipulation. Other related work includes quadrotors with tilting body or propellers (Ryll *et al.* [45, 46], Hintz *et al.* [22], Riviere *et al.* [42]), scissor-like foldable quadrotors (Zhao *et al.* [53]), quadrotors with sprung-hinge-based foldable arms (Bucki and Mueller [7]), and small-winged drones with morphing wing design (Di Luca *et al.* [31]). Unlike these works we consider a quadrotor carrying a soft gripper, explicitly considering the impact of softness on the stability and dynamics of our platform and specifically targeting manipulation (with our related publications [15, 16]).

Soft Robotics. Continuously deformable, entirely compliant robots represent the extremum of the trend towards compliance and under-actuation. However, traditional rigid-body modeling and control techniques fall short when confronted with infinite degrees of freedom. The emerging discipline of soft robotics has developed principled approaches to allow control of these systems, opening a new frontier in manipulation. Rus and Tolley [44] and Thuruthel *et al.* [18] provide a comprehensive review of soft robotics and soft manipulation. King *et al.* [25], Manti *et al.* [32], and Hassan *et al.* [21] design bio-inspired tendon-actuated soft grippers. Marchese *et al.* [33, 34] implement kinematics and trajectory optimization for hydraulic soft manipulators based on piecewise-constant-curvature approximations. Bern *et al.* [4, 3] and Duriez *et al.* [12] model the kinematics and dynamics of tendon-actuated soft robots using finite element methods. These works have not been applied in an aerial context, where the soft manipulator impacts the UAV dynamics and offers unique opportunities in aerial manipulation.

Chapter 2

Control and Trajectory Optimization for Soft Aerial Manipulation

In this chapter we present a theoretical solution to the planning and control problem of grasping with a soft drone. Our proposed algorithmic approach is based on decoupling the control and planning for the rigid and soft components; this is made possible by the stability of a standard quadrotor controller despite the addition of the soft gripper (Theorem 2) and the resilience to positioning errors afforded by the soft gripper.

2.1 Soft Aerial Manipulation: Problem Statement and Decoupled Approach

2.1.1 System Overview and Problem Statement

Our soft aerial manipulator (Fig. 1-2) comprises the frame of a standard quadrotor with the (rigid and heavy) landing gear replaced by a soft gripper. The gripper consists of four soft fingers and is based on the design by Hassan *et al.* [21]. Each finger is attached to the quadrotor base and actuated with two pairs of tendons on opposite sides (Fig. 2-1).

Each finger is actuated by setting a desired length (later called the *rest length*) at the tendons. Similarly to [3], actuation of a pair of tendons (lying on the same

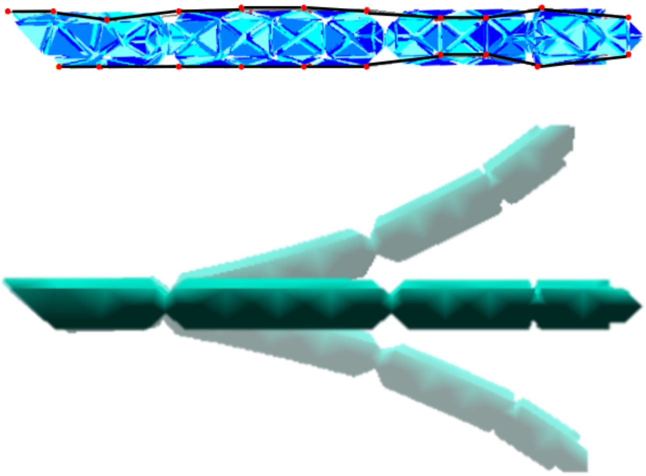


Figure 2-1: Soft finger with two pairs of tendons on opposite sides. Tendons (black, top) are modeled as passing through a set of nodes (red, top) in a tetrahedral mesh discretization of the finger (blue, top). Pulling a tendon causes a contraction of the finger (bottom).

side of the finger) is coupled to prevent finger twist. The quadrotor base uses four motors and propellers for actuation, as usual. In summary, the system uses 12 control variables (four motor speeds for the quadrotor, and one for each pair of tendons on the two sides of the fingers) to control an infinite-dimensional state (including the finite-dimensional quadrotor state and the infinite-dimensional state describing the configuration of the soft gripper). The soft gripper model is described in Section 2.2, while the quadrotor model is given in Section 2.3.

Our manipulator is tasked with grasping an object of unknown shape lying over an unknown surface: the system is only provided with the centroid of the object. In particular, we are interested in (i) computing a trajectory of state variables over time, and (ii) developing a control law that can track the computed trajectory to ensure a successful grasp in the face of external disturbances. We assume we can measure the full state of the quadrotor (its 3D pose and linear and angular velocities, typically observable using a motion capture system [35] or visual-inertial state estimation [30]), while we operate the soft gripper in open loop (*i.e.,* our approach does not need to measure the state of the gripper). While our goal is to simultaneously obtain trajectories and controllers for the soft gripper and the rigid quadrotor, in the following we propose a decoupled approach that implements separate planners/controllers for both subsystems.

2.1.2 Decoupled Control and Trajectory Optimization

Let us call $\boldsymbol{X}(t)$ the state of the quadrotor base (*i.e.,* a 3D pose and its derivatives) at time t, and $\boldsymbol{Y}(t)$ the infinite-dimensional matrix describing the 3D position of every point of the soft gripper. Moreover, call $\boldsymbol{f}(t)$ the quadrotor propeller thrust forces at time t, and $\boldsymbol{l}(t)$ the tendon rest lengths that actuate the fingers. To simplify the notation, below we omit the dependence on time t when possible.

The *soft aerial grasping problem* considered in this thesis can be formulated as an optimal control problem:

$$\begin{aligned}(\boldsymbol{X}^\star, \boldsymbol{Y}^\star, \boldsymbol{f}^\star, \boldsymbol{l}^\star) = \quad \underset{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{f},\boldsymbol{l}}{\arg\min} \quad & \textstyle\int_0^{t_f} \mathcal{J}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{f}, \boldsymbol{l})\mathrm{d}t \\ \text{subject to} \quad & \mathcal{D}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{f}, \boldsymbol{l}) = 0 \\ & \boldsymbol{X}(0) = \bar{\boldsymbol{X}}_0, \quad \boldsymbol{Y}(0) = \bar{\boldsymbol{Y}}_0 \\ & \boldsymbol{X}(t_f) = \bar{\boldsymbol{X}}_{t_f}, \ \boldsymbol{Y}(t_f) = \bar{\boldsymbol{Y}}_{t_f} \\ & \boldsymbol{Y}(t_g) = \bar{\boldsymbol{Y}}_{t_g}\end{aligned} \tag{2.1}$$

where $\mathcal{J}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{f}, \boldsymbol{l})$ is the cost functional that, for instance, penalizes control usage or encourages smooth state changes, the constraint $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{f}, \boldsymbol{l}) = 0$ ensures that the solution satisfies the platform dynamics, $(\bar{\boldsymbol{X}}_0, \bar{\boldsymbol{Y}}_0)$ is the given initial state of the soft aerial manipulator at the initial time $t = 0$, $(\bar{\boldsymbol{X}}_{t_f}, \bar{\boldsymbol{Y}}_{t_f})$ is the desired state at the final time t_f (say, the end of the execution), and $\bar{\boldsymbol{Y}}_{t_g}$ is the desired state of the soft gripper at the time of grasp $t_g \in [0, t_f]$. In words, Problem (2.1) looks for minimum-cost controls such that the platform moves from an initial to a final state, and the soft gripper is in a suitable configuration during grasp.

While in principle one would like to obtain a *control policy* that computes a suitable control $(\boldsymbol{f}, \boldsymbol{l})$ for every possible state, doing so is hard even without a soft gripper. Therefore, related work solves problems akin to (2.1) by first performing *trajectory optimization*, *i.e.,* computing an open loop state trajectory and then designing a controller that tracks such a trajectory [27]. We follow the same approach and decouple the optimal control problem into trajectory optimization and tracking control. However, we are still left with the complexity that our soft aerial manipulator is not differentially flat, which is a key requirement for tractable trajectory optimization in related work [35]. To circumvent this issue, we further decouple trajectory optimization and control for the quadrotor base and the soft gripper as follows.

We split Problem (2.1) into the cascade of two problems. First, we solve the **drone control subproblem**, where we look for an optimal control action for the drone propeller forces $\boldsymbol{f}$ while treating the soft payload as an unknown disturbance. This can be formulated as follows:

$$\begin{aligned} (\boldsymbol{X}^\star, \boldsymbol{f}^\star) = \quad & \arg\min_{\boldsymbol{X},\boldsymbol{f}} \quad \textstyle\int_0^{t_f} \mathcal{J}_q(\boldsymbol{X}, \boldsymbol{f}) \mathrm{d}t \\ & \text{subject to} \quad \mathcal{D}_q(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{f}) = 0 \\ & \qquad \boldsymbol{X}(0) = \bar{\boldsymbol{X}}_0, \quad \boldsymbol{X}(t_f) = \bar{\boldsymbol{X}}_{t_f}, \\ & \qquad \boldsymbol{X}(t_g) = \bar{\boldsymbol{X}}_{t_g} \end{aligned} \tag{2.2}$$

where $\mathcal{J}_q$ and $\mathcal{D}_q$ now only involve the quadrotor state and dynamics, and where we relaxed the grasp condition $\boldsymbol{Y}(t_g) = \bar{\boldsymbol{Y}}_{t_g}$ in (2.1), with a condition on the state of the quadrotor during the grasp $\boldsymbol{X}(t_g) = \bar{\boldsymbol{X}}_{t_g}$. Intuitively, the drone has to ensure it is close enough to the object at time t_g to enable the soft gripper to grasp, but without worrying about the specific configuration of the gripper. Note how the drone dynamics are a function of the soft gripper configuration $\boldsymbol{Y}$, which is treated as an unknown disturbance, hence will not used to solve (2.2) (see Section 2.3).

After solving (2.2) and obtaining the nominal (open loop) quadrotor trajectory $\boldsymbol{X}^\star$, we solve the **soft-gripper control subproblem**, where we look for an optimal control action for the tendons rest lengths $\boldsymbol{l}$:

$$\begin{aligned} (\boldsymbol{Y}^\star, \boldsymbol{l}^\star) = \quad & \arg\min_{\boldsymbol{l}} \quad \textstyle\int_0^{t_f} \mathcal{J}_s(\boldsymbol{X}^\star, \boldsymbol{Y}, \boldsymbol{l}) \mathrm{d}t \\ & \text{subject to} \quad \mathcal{D}_s(\boldsymbol{X}^\star, \boldsymbol{Y}, \boldsymbol{l}) = 0 \\ & \qquad \boldsymbol{Y}(0) = \bar{\boldsymbol{Y}}_0, \quad \boldsymbol{Y}(t_f) = \bar{\boldsymbol{Y}}_{t_f}(\boldsymbol{X}^\star), \\ & \qquad \boldsymbol{Y}(t_g) = \bar{\boldsymbol{Y}}_{t_g}(\boldsymbol{X}^\star) \end{aligned} \tag{2.3}$$

where now the soft-gripper dynamics $\mathcal{D}_s(\boldsymbol{X}^\star, \boldsymbol{Y}, \boldsymbol{l})$, the grasp configuration $\bar{\boldsymbol{Y}}_{t_g}(\boldsymbol{X}^\star)$, and the terminal state $\bar{\boldsymbol{Y}}_{t_f}(\boldsymbol{X}^\star)$ depend on the (fixed) nominal drone trajectory $\boldsymbol{X}^\star$.

In practice the object to be grasped has unknown shape and the soft gripper has an infinite number of points, hence it is unrealistic to enforce the condition $\boldsymbol{Y}(t_g) = \bar{\boldsymbol{Y}}_{t_g}(\boldsymbol{X}^\star)$; in Section 2.2.1 we will replace such a condition with a more realistic one involving only the positions of the 4 fingertips and the object centroid.

In the following sections, we describe our choice of cost functions and discuss how to attack problems (2.2) and (2.3), using tools from quadrotor control [27] and soft robotics [4].

2.2 Open-loop Control and Trajectory Optimization for a Soft Gripper

This section describes how to solve the soft-gripper subproblem (2.3) for a specific choice of cost function. We make the following key assumption.

Assumption 1 (Quasi-static approximation) *The soft gripper is* quasi-static*,* i.e., *there is an instantaneous relation between rest lengths and gripper configuration.*

This allows simplifying the model by neglecting the soft gripper dynamics and is fairly common in soft robotics [4].

2.2.1 Objectives for Aggressive Soft Grasping

This section discusses our choice of objective function in eq. (2.3). As anticipated in Section 2.1, rather than matching a desired configuration $\bar{\boldsymbol{Y}}_{t_g}$ (which would be hard to compute without knowing the shape of the object to grasp), we prefer finding a control which optimizes the positions of the fingertips with respect to the centroid of the object we want to grasp. Mathematically, we drop the constraint $\boldsymbol{Y}(t_g) = \bar{\boldsymbol{Y}}_{t_g}(\boldsymbol{X}^\star)$ in eq. (2.3) and optimize an objective which is only function of the fingertip positions $\boldsymbol{y}_{tip_i}$ (each being a point in $\boldsymbol{Y}$) and the centroid $\boldsymbol{o}$ of the object we want to grasp. This is an exemplary instantiation of the idea of *morphological computation*: rather than planning the full configuration of the soft gripper, we only plan for the fingertips and have the softness of the fingers adjust to the shape of the object we want to grasp.

Specifically, we plan grasps in two phases, “approach” and “grasp”. Intuitively, the “approach” phase involves opening the gripper as the quadrotor approaches the target location so as to allow the fingertips to surround the grasped object, while the “grasp” phase involves closing the gripper, contracting the fingertips to achieve an enveloping grasp. Mathematically, this reduces to assuming the following form for the objective function in eq. (2.3):

$$\begin{aligned}\int_0^{t_f} \mathcal{J}_s(\boldsymbol{X}^\star, \boldsymbol{Y}, \boldsymbol{l})\mathrm{d}t &= \int_0^{t_f} \mathcal{J}_{\text{tendon}}(\boldsymbol{X}^\star, \boldsymbol{Y}, \boldsymbol{l})\mathrm{d}t \\ &+ \mathcal{C}_{\text{approach}}(\boldsymbol{Y}(t_a)) \\ &+ \mathcal{C}_{\text{grasp}}(\boldsymbol{Y}(t_g))\end{aligned} \tag{2.4}$$

where $\mathcal{J}_{\text{tendon}}$ will be chosen to penalize the control effort, and $\mathcal{C}_{\text{approach}}$ and $\mathcal{C}_{\text{grasp}}$ can be understood as terminal costs rewarding fingertip configurations at specific times t_a (right before grasping) and t_g (time of grasping) which are assumed to be given.

We approximately solve (2.4) by first finding tendon lengths optimizing $\mathcal{C}_{\text{approach}}(\boldsymbol{Y}(t_a))$ and $\mathcal{C}_{\text{grasp}}(\boldsymbol{Y}(t_g))$ then solve for intermediate lengths minimizing $\int_0^{t_f} \mathcal{J}_{\text{tendon}}(\boldsymbol{X}^\star, \boldsymbol{Y}, \boldsymbol{l})$.

Choice of $\mathcal{J}_{\text{tendon}}$. We choose the term $\mathcal{J}_{\text{tendon}}$ in the objective function (2.4) as:

$$\mathcal{J}_{\text{tendon}}(\boldsymbol{X}^\star, \boldsymbol{Y}, \boldsymbol{l}) = \max_{0 \le t \le t_f} \left\| \frac{d\boldsymbol{l}}{dt} \right\|_\infty \tag{2.5}$$

which penalizes the maximum rate of change of the rest lengths $\boldsymbol{l}$ (our control inputs) in the interval $[0, t_f]$. This choice is motivated by the fact that rapid changes in tendon lengths result in large forces on the tendon attachment points. These are undesirable, since large forces on specific points (i) create significant localized deformations which risk damaging the soft gripper, and (ii) cause large local accelerations that violate the quasi-static assumption.

Choice of $\mathcal{C}_{\text{grasp}}$. We choose the term $\mathcal{C}_{\text{grasp}}$ in the objective function (2.4) as:

$$\mathcal{C}_{\text{grasp}}(\boldsymbol{Y}) = \textstyle\sum_{i=1}^{4} \|\boldsymbol{y}_{tip_i} - \boldsymbol{o}\|_2^2 \tag{2.6}$$

which encourages the four fingertips $\boldsymbol{y}_{tip_i} \subset \boldsymbol{Y}$ to be as close as possible to the target centroid $\boldsymbol{o}$ at the time of grasping.

Choice of $\mathcal{C}_{\textbf{approach}}$. We present two potential choices for the term $\mathcal{C}_{\text{approach}}$ in the objective function (2.4). The simplest choice is to set $\mathcal{C}_{\text{approach}}$ as:

$$\mathcal{C}_{1,\text{approach}}(\boldsymbol{Y}) = -\textstyle\sum_{i=1}^{4} \|\boldsymbol{y}_{tip_i} - \boldsymbol{o}\|_2^2 \tag{2.7}$$

which, recalling that minimizing a cost function f is the same as maximizing $-f$, simply maximizes the distance between the fingertips $\boldsymbol{y}_{tip_i}$ and the target centroid $\boldsymbol{o}$ at the time of approach. In the following, we denote this choice as $\mathcal{C}_1$. While optimizing (2.7) intuitively leads to "opening" the fingers of the gripper as much as possible right before grasping (hence maximizing the distance to the target centroid, see Fig. 2-2a), such cost may induce suboptimal behaviors, in particular when performing aggressive grasping in which the target is asymmetrically-located with respect to the fingers (i.e. the drone is not directly above the target, see Fig. 2-2). Therefore, below we consider an alternative cost that rewards the fingertips for surrounding the target so to assure that the target remains between the fingers.

A better choice for the objective for the approach phase of an aggressive grasp is the norm of the cross product of the vectors connecting fingertips and target centroid (Fig. 2-2(b)):

$$\mathcal{C}_{2,\text{approach}}(\boldsymbol{Y}) = -\textstyle\sum_{i=1,3} \|(\boldsymbol{y}_{tip_i} - \boldsymbol{o}) \times (\boldsymbol{y}_{tip_{i+1}} - \boldsymbol{o})\|_2^2 \tag{2.8}$$

which for simplicity we denote as $\mathcal{C}_2$ in the rest of this thesis. Unlike the prior objective, $\mathcal{C}_2$ rewards both fingertip distance and angle. More rigorously, the cross product denotes the area of the tetrahedron (or triangle in 2D) formed by fingertips and target centroid. All else being equal, the odds of a successful grasp improve with the volume of the target object enveloped by the gripper. Without knowing the target shape, maximization of the volume between fingertips and gripper then implies maximization of the potential target area to be enveloped. If we approximate (given finger kinematics and quadrotor motion) that, once the "grasp" phase begins,

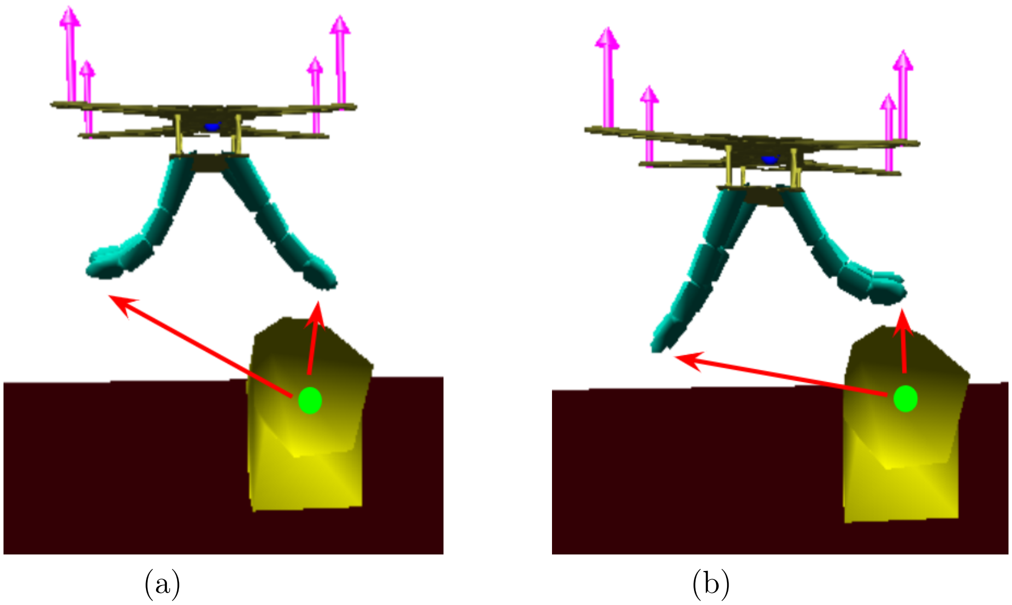


Figure 2-2: Result of optimizing for two different objectives during the approach phase, based on the vectors (red) from target centroid (green) to fingertips: (a) maximizes (locally) the fingertip distance $\mathcal{C}_1$ from the target centroid, and (b) maximizes the norm $\mathcal{C}_2$ of the cross product between the vectors connecting the fingertip to the centroid. Since the quadrotor is moving aggressively towards the target, (b) is more likely to yield an enveloping grasp.

the fingertips will move in a straight line towards the target centroid, then any part of the target object in the area between fingertips and gripper will ultimately be enveloped by the gripper and contribute to a successful grasp.

2.2.2 Modeling of a Tendon-Actuated Soft Gripper

This section shows how to compute the instantaneous rest length $\boldsymbol{l}^\star(t_g)$ at time t_g such that the corresponding soft gripper configuration minimizes $\mathcal{C}_{\text{grasp}}(\boldsymbol{Y})$ (the same approach can be used to compute $\boldsymbol{l}^\star(t_a)$ to minimize $\mathcal{C}_{\text{approach}}(\boldsymbol{Y})$). Note that we can compute an instantaneous rest length thanks to Assumption 1, which assumes we can neglect the soft gripper dynamics.

FEM Model. Our approach follows Bern *et al.* [4], using a finite element approximation to compute tendon lengths minimizing an objective $\mathcal{C}(\boldsymbol{Y})$ (as we mentioned, we will use the same approach to minimize $\mathcal{C}_{\text{grasp}}(\boldsymbol{Y})$ or $\mathcal{C}_{\text{approach}}(\boldsymbol{Y})$). We contribute analytic expressions for all Jacobians (Appendix 6.1). We approximate the infinite-dimensional soft gripper configuration $\boldsymbol{Y}$ as a set of N discrete nodes, as in *finite* *element methods* (FEM). With slight abuse of notation, we still use $\boldsymbol{Y}$ to denote the discretized set of nodes:

$$\boldsymbol{Y} \doteq [\boldsymbol{y}_1 \; \boldsymbol{y}_2 \; \ldots \; \boldsymbol{y}_n] \in \mathbb{R}^{3\times N} \tag{2.9}$$

where $\boldsymbol{y}_i \in \mathbb{R}^3$ is the position of the i-th node. The nodes are arranged in a *tetrahedral mesh*, and the tendons are approximated as one-sided springs. Finally, a set of pins (also modeled as linear springs) fixes the mesh nodes to the quadrotor base. Based on this FEM model, we use a Jacobian-based iterative solution to the soft robot inverse kinematics, *i.e.,* to find the tendon rest lengths $\boldsymbol{l}$ that yield a static configuration $\boldsymbol{Y}$ minimizing the objective $\mathcal{C}(\boldsymbol{Y})$.

Inverse Kinematics Overview. We minimize $\mathcal{C}(\boldsymbol{Y})$ with respect to the rest lengths $\boldsymbol{l}$ (recall again that $\boldsymbol{Y}$ depends on $\boldsymbol{l}$) via gradient descent. The complexity in the "soft case" is that one cannot write the relation between $\boldsymbol{l}$ and $\boldsymbol{Y}$ analytically. To circumvent this issue, and following [4], we first solve the forward kinematics problem (determining the initial system state $\bar{\boldsymbol{Y}}$ for the initial actuation $\bar{\boldsymbol{l}}$) by minimizing the total system energy using Newton's method to find *quasi-static equilibrium*, a system state where net force (and acceleration) are zero. Once an equilibrium configuration is found, we obtain an analytic expression for the actuator Jacobian $\frac{d\mathbf{Y}}{dl}$[1]. Then we compute the Jacobian $\frac{d\mathcal{C}}{d\boldsymbol{Y}}$ analytically based on the definition of $\mathcal{C}$ in eqs. (2.6)-(2.8). Finally, we use $\frac{d\mathbf{Y}}{dl}$ and $\frac{d\mathcal{C}}{d\boldsymbol{Y}}$ to compute the gradient of the cost $\frac{d\mathcal{C}}{dl} = \frac{d\mathcal{C}}{d\boldsymbol{Y}}\frac{d\boldsymbol{Y}}{dl}$ with respect to the actuation $\bar{\boldsymbol{l}}$ and find a suitable gradient descent step $\Delta\boldsymbol{l}$[2]. We solve for the new *quasi-static* configuration $\boldsymbol{Y}$ associated with the new tendon rest lengths $\boldsymbol{l} = \bar{\boldsymbol{l}} + \Delta\boldsymbol{l}$[3] and iterate till convergence. While the computation of $\frac{d\mathcal{C}}{d\boldsymbol{Y}}$ is straightforward given $\mathcal{C}$, in the following we describe the expression of the energy and the Jacobian $\frac{d\mathbf{Y}}{dl}$.

[1] To keep a matrix (rather than a tensor) notation, we assume that all Jacobians involving $\boldsymbol{Y}$, *e.g.,* $\frac{d\mathbf{Y}}{dl}$, work on a vectorization of $\boldsymbol{Y}$.

[2] In practice we use parameters $\alpha = 0.5$ and $\beta = 0.1$ in computing the *Armijo-Goldstein rule* [38] in backtracking line search along the gradient, as well as enforcing a step size norm between 1 and 10 cm. Further, we choose a learning rate of 2.5 when computing the approach lengths and 0.25 when computing the grasp lengths.

[3] We use the *Newton-CG* implementation in *scipy.optimize.minimize*, with the linearization $\bar{\boldsymbol{Y}} + \frac{d\boldsymbol{Y}}{dl}\Delta\boldsymbol{l}$ as an initial guess.

2.2.3 Jacobian $\frac{d\boldsymbol{Y}}{d\boldsymbol{l}}$ via Forward Kinematics

We solve the forward kinematics $\boldsymbol{l} \mapsto \boldsymbol{Y}$ by minimizing the total energy of a configuration $\boldsymbol{Y}$ for a given choice of $\boldsymbol{l}$ (as in [4]). The total energy of the soft gripper can be written as:

$$\begin{aligned} E(\boldsymbol{Y}, \boldsymbol{l}, \boldsymbol{X}) = & E_{mesh}(\boldsymbol{Y}) + E_{tendons}(\boldsymbol{Y}, \boldsymbol{l}) \\ & + E_{pins}(\boldsymbol{Y}, \boldsymbol{X}) + E_{gravity}(\boldsymbol{Y}) \end{aligned} \tag{2.10}$$

with the equilibrium configuration $\boldsymbol{Y}_{eq}(\boldsymbol{l}, \boldsymbol{X})$ as the minimizer of the energy: $\boldsymbol{Y}_{eq}(\boldsymbol{l}, \boldsymbol{X}) = \arg\min_{\boldsymbol{Y}} E(\boldsymbol{Y}, \boldsymbol{l}, \boldsymbol{X})$. In the following, we describe each term in the energy (2.10).

Mesh Energy. The energy term E_{mesh} models the contribution to the system energy due to deformations of the soft material. We compute the energy contribution of each tetrahedral component $\boldsymbol{Y}_{ijkl} \doteq [\boldsymbol{y}_i \; \boldsymbol{y}_j \; \boldsymbol{y}_k \; \boldsymbol{y}_l]$ separately. We define the relative displacement of each node in the element:

$$\Delta\boldsymbol{Y}_{ijkl} = \begin{bmatrix} (\boldsymbol{y}_i - \boldsymbol{y}_l) & (\boldsymbol{y}_j - \boldsymbol{y}_l) & (\boldsymbol{y}_k - \boldsymbol{y}_l) \end{bmatrix}$$

which contains the relative positions of vertices i, j, k with respect to l. When no forces are applied, the mesh assumes the rest displacement $\bar{\Delta}\boldsymbol{Y}_{ijkl}$. In the presence of external forces, the mesh assumes a deformed displacement $\Delta\boldsymbol{Y}_{ijkl}$. The energy of a configuration depends on the mismatch between rest and deformed displacement. Define the *deformation gradient* $\boldsymbol{G} = \Delta\boldsymbol{Y}_{ijkl}(\bar{\Delta}\boldsymbol{Y}_{ijkl})^{-1}$, the rest volume $\bar{v} = 1/6 \det(\bar{\Delta}\boldsymbol{Y}_{ijkl})$, and the *volumetric deformation* $v_F = \det(\boldsymbol{G})$, which is the ratio of deformed to undeformed volume. We use a neo-Hookean material model which defines the mesh energy in terms of Lamé parameters μ, λ [48]:

$$E^{ijkl}_{mesh}(\boldsymbol{Y}) = \bar{v}\left[\frac{\mu}{2} tr(\boldsymbol{G}^T\boldsymbol{G} - \mathbf{I}) - \mu \ln(v_F) + \frac{\kappa}{2}\ln^2(v_F)\right] \tag{2.11}$$

The total mesh energy $E_{mesh}(\boldsymbol{Y}) = \sum_{ijkl} E^{ijkl}_{mesh}(\boldsymbol{Y})$ is the sum of the contributions of all elements $ijkl$.

Tendon Energy. The energy term E_{tendon} models the contribution of the tendons to the system. Each tendon is defined by the set of nodes in the mesh it is attached to. Let us denote with $i_1, \ldots, i_n$ the set of node indices tendon i is attached to (the so called *routing path*). Then, the tendon deformation for tendon i is defined as:

$$\gamma_i = \textstyle\sum_{k=1}^{n-1} \|\boldsymbol{y}_{i_{k+1}} - \boldsymbol{y}_{i_k}\|_2 - \boldsymbol{l}_i \tag{2.12}$$

which intuitively is the mismatch between the desired routing path length (dictated by the rest length $\boldsymbol{l}_i$) and the actual length according to the mesh nodes ($\sum_{k=1}^{n-1} \|\boldsymbol{y}_{i_{k+1}} - \boldsymbol{y}_{i_k}\|$). We can then recover the energy of tendon i by modeling the tendon as a one-sided spring with spring constant κ_{tendon}:

$$E^i_{tendon}(\boldsymbol{Y}) = \begin{cases} 0 & \text{if } \gamma < 0 \\ \kappa_{tendon}\ \gamma_i^2 & \text{otherwise} \end{cases} \tag{2.13}$$

The total tendon energy $E_{tendon} = \sum_{i=1}^{8} E^i_{tendon}(\boldsymbol{Y})$ is the sum of the contribution of all tendons.

Pin Energy. The energy term E_{pin} models the contribution of the pins (connecting the soft gripper to the quadrotor base) to the system. Each pin i is modeled as a spring with constant κ_{pin}, connecting a mesh node $\boldsymbol{y}_i$ belonging to the soft gripper, to a point $\boldsymbol{x}_i^{pin}$ belonging to the quadrotor base (for a given drone state $\boldsymbol{X}$). The energy for each pin i is:

$$E^i_{pin}(\boldsymbol{y}_i, \boldsymbol{X}) = \kappa_{pin}\|\boldsymbol{y}_i - \boldsymbol{x}_i^{pin}\|_2 \tag{2.14}$$

The total energy E_{pin} is the sum of the contribution from all pins (we use three pins per finger).

Gravitational Energy. The energy term $E_{gravity}$ models the impact on the gravity on the system's energy. We approximate the gripper mass as concentrated in the mesh nodes, and denote with m_i the mass of node i. The gravitational potential energy depends on the mass and height of the node:

$$E^i_{gravity}(\boldsymbol{Y}) = -m_i\, \boldsymbol{g}^\mathsf{T} \boldsymbol{y}_i \tag{2.15}$$

where $\boldsymbol{g} \doteq [0, 0, -9.81]^\mathsf{T}\mathrm{m/s}^2$ is the gravity vector. The total gravitational energy $E_{gravity}(\boldsymbol{Y}) = \sum_{i=1}^{N} E^i_{gravity}(\boldsymbol{Y})$ is the sum of the contribution of all nodes.

Jacobian $\frac{d\boldsymbol{Y}}{d\boldsymbol{l}}$. Given a control $\boldsymbol{l}$ (*e.g.,* a point for which we want to obtain a gradient), we compute a quasi-static configuration $\boldsymbol{Y}$ that minimizes the system energy (2.10) using Newton's method. Then, the actuator Jacobian can be computed from the Hessians of the energy, as shown in [4]:

$$\frac{d\boldsymbol{Y}}{d\boldsymbol{l}} = -\frac{d^2E}{d\boldsymbol{Y}^2}^{-1} \frac{d^2E}{d\boldsymbol{Y}d\boldsymbol{l}} \tag{2.16}$$

The analytic expressions of the terms on the right-hand-side of (2.16) is reported in Appendix 6.1. It is worth noting that these terms are readily available as a byproduct of the application of Newton's method to the minimization of (2.10).

2.2.4 Trajectory Optimization and Open-loop Control for a Soft Gripper

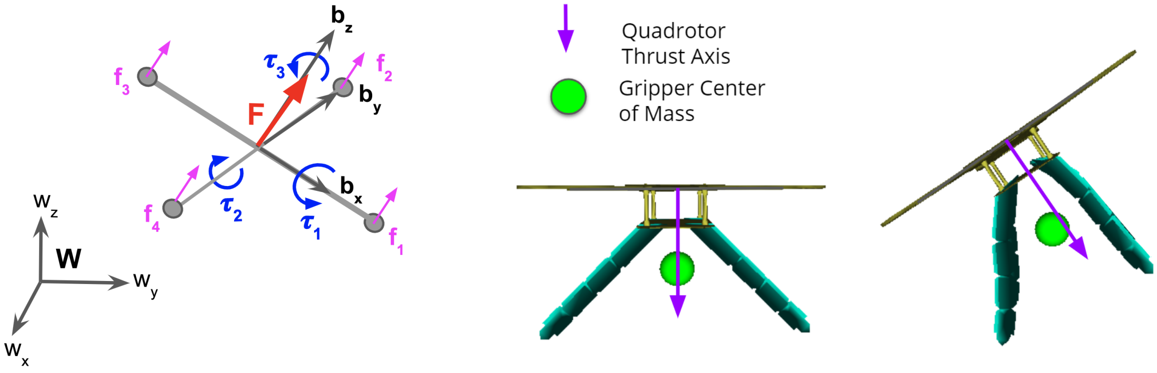
The inverse kinematics model in the previous section allows computing the tendon rest lengths $\boldsymbol{l}^\star(t_a)$ and $\boldsymbol{l}^\star(t_g)$ that ensure that the fingertips of the soft gripper are away from the target at time t_a (in the approach phase) and close to the target centroid at time t_g (*i.e.,* during the grasp). With our choice of $\mathcal{J}_{\text{tendon}}$ (Eq. (2.5)) and under Assumption 1, solving problem (2.3) reduces to (i) ensuring that $\boldsymbol{l}(t_g)$ is equal to $\boldsymbol{l}^\star(t_g)$ and that $\boldsymbol{l}(t_a)$ is equal to $\boldsymbol{l}^\star(t_a)$ and (ii) minimizing the changes of $\boldsymbol{l}$ in $[0, t_f]$. It is straightforward to see that the optimal control trajectory under this setup consists in linearly interpolating $\boldsymbol{l}$ from the initial rest length (at time 0) to the lengths $\boldsymbol{l}^\star(t_a)$ (at time t_a); then linearly interpolating between lengths $\boldsymbol{l}^\star(t_a)$ to the lengths $\boldsymbol{l}^\star(t_g)$ (at time t_g), and finally keeping them constant afterwards (until t_f). We apply the resulting control sequence $\boldsymbol{l}^\star(t)$ in open loop.

2.3 Geometric Control and Trajectory Optimization for the Quadrotor Base

This section describes how to solve the drone control subproblem (2.2). Thanks to the decoupling described in Section 2.1, problem (2.2) falls back to a standard quadrotor control formulation. Therefore, as done in related work [35], we solve it by first computing a nominal state trajectory using polynomial trajectory optimization (briefly reviewed in Section 2.3.1) and then using a geometric controller to track the nominal trajectory. The element that sets our setup apart is the presence of (the disturbance) $\boldsymbol{Y}$ in the quadrotor dynamics in (2.2). Intuitively, the soft gripper in general imposes a torque that acts to orient the quadrotor towards level. While this torque may prevent the achievement of the control goals or even destabilize the platform, in the following we show that under certain assumptions on the soft load (Fig. 2-3) and the aerodynamic forces experienced by the quadrotor, we can bound the disturbance torque such that a standard geometric controller remains asymptotically stable. In practice we empirically observe that our platform remains stable even when these assumptions are violated (see experiments in Section 4.1).

2.3.1 Minimum-Snap Trajectory Optimization

We first compute a nominal trajectory (quadrotor state and its derivatives over time) by solving (2.2) and neglecting the presence of the disturbance $\boldsymbol{Y}$. As done in related work [35], (i) we consider a cost function in (2.2) that penalizes the integral of the 4th derivative of the state (minimum snap), (ii) we assume polynomial trajectories, and (iii) we leverage differential flatness to express the optimal control problem as 4 decoupled scalar optimization problems over the flat outputs (three for the Cartesian position of the quadrotor and one for its yaw angle). Mellinger *et al.* [35] and Bry *et al.* [6] show that the resulting polynomial optimization problems can be solved efficiently via quadratic programming.



(a)

(b)

Figure 2-3: (a) Quadrotor forces, torques, local and world frames. (b) Without external perturbation, the soft gripper center of mass is aligned with the local vertical due to the symmetry of its fingers; forces that deform the gripper might cause its center of mass to be misaligned.

2.3.2 Geometric Control of a Quadrotor with a Soft Load

Given the quadrotor trajectory $\boldsymbol{X}^\star(t)$ generated according to Section 2.3.1, we are only left to design a controller that is able to track $\boldsymbol{X}^\star(t)$ in the face of external disturbances, including the torque induced by the time-varying soft payload $\boldsymbol{Y}$. Here we show that a commonly adopted solution, the geometric controller by Lee *et al.* [27], preserves asymptotic stability even in the presence of our soft gripper. We first review the basics of the geometric controller, then discuss its stability with the added soft load.

Geometric Controller. In the following, we explicitly write the quadrotor state (that we generically denoted with $\boldsymbol{X}$ so far) as $\boldsymbol{X} \doteq \{\boldsymbol{p}, \boldsymbol{R}, \dot{\boldsymbol{p}}, \boldsymbol{\Omega}\}$, including the quadrotor position $\boldsymbol{p} \in \mathbb{R}^3$, its rotation $\boldsymbol{R} \in \mathrm{SO}(3)$, the linear velocity $\dot{\boldsymbol{p}} \in \mathbb{R}^3$, and the angular velocity $\boldsymbol{\Omega} \in \mathbb{R}^3$. Using this notation, and denoting the columns of $\boldsymbol{R}$ as $\boldsymbol{R} = [\boldsymbol{b}_x \; \boldsymbol{b}_y \; \boldsymbol{b}_z]$, the quadrotor dynamics can be written as:

$$\begin{cases} m\ddot{\boldsymbol{p}} = m\boldsymbol{g} + f\boldsymbol{b}_z \\ \dot{\boldsymbol{R}} = \boldsymbol{R}\hat{\boldsymbol{\Omega}} \\ \boldsymbol{J}\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega} + \boldsymbol{\tau} + \boldsymbol{\tau}_{load} \end{cases} \tag{2.17}$$

where m is the total mass of the platform and gripper, $\boldsymbol{g}$ is the gravity vector, $\boldsymbol{J}$ is the moment of inertia, f is the scalar thrust force (applied at the quadrotor center of mass and along the local vertical direction $\boldsymbol{b}_z$) resulting from the propeller forces $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4$, $\boldsymbol{\tau} \in \mathbb{R}^3$ is the torque resulting from the propeller forces, and $\boldsymbol{\tau}_{load}$ is the torque exerted by the soft gripper (Fig. 2-3(a)). In (2.17), the symbol × is the vector cross product, the *hat* map $\hat{\cdot}$ maps a 3D vector to a 3×3 skew symmetric matrix and its inverse, the *vee* map $\cdot^{\vee}$, maps a 3×3 skew-symmetric matrix to a vector (as in [27]).

The geometric controller [27] takes as input a desired state $\boldsymbol{X}^{\star} \doteq \{\boldsymbol{p}_d, \boldsymbol{R}_d, \dot{\boldsymbol{p}}_d, \boldsymbol{\Omega}_d\}$,[4] and computes the tracking errors:

$$\begin{aligned}
\boldsymbol{e}_p &= \boldsymbol{p} - \boldsymbol{p}_d, && \text{(position error)}\\
\boldsymbol{e}_v &= \dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_d && \text{(linear velocity error)}\\
\boldsymbol{e}_r &= \tfrac{1}{2}(\boldsymbol{R}_d^{\mathsf{T}}\boldsymbol{R} - \boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}_d)^{\vee} && \text{(rotation error)}\\
\boldsymbol{e}_\Omega &= \boldsymbol{\Omega} - \boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}_d\boldsymbol{\Omega}_d && \text{(angular velocity error)}
\end{aligned} \tag{2.18}$$

Then the controller decides for suitable thrust force f and torques $\boldsymbol{\tau}$ to contrast these errors using the control law:

$$f = -\boldsymbol{b}_z^{\mathsf{T}}(k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + m\boldsymbol{g} - m\ddot{\boldsymbol{p}}_d) \tag{2.19}$$

$$\boldsymbol{\tau} = -k_r\boldsymbol{e}_r - k_\Omega\boldsymbol{e}_\Omega + \boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega} \tag{2.20}$$

$$-\boldsymbol{J}(\hat{\boldsymbol{\Omega}}\boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}_d\boldsymbol{\Omega}_d - \boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}_d\dot{\boldsymbol{\Omega}}_d) \tag{2.21}$$

where k_p, k_v, k_r, k_Ω are suitable control gains. We refer the reader to [27] for details about how to map the desired total thrust f and torque $\boldsymbol{\tau}$ to propeller forces $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4$.

Asymptotic Stability. Here we show that the geometric controller above is stable despite the presence of the soft gripper. We assume that the quadrotor is upright, i.e. $[0\ 0\ 1] \cdot \boldsymbol{b}_z \geq 0$. The key insight is that as it deforms due to gravity

[4] In [27], the desired rotation $\boldsymbol{R}_d$ and angular velocity $\boldsymbol{\Omega}_d$ are built from a desired yaw angle. We refer the reader to [27] for details.

and the quadrotor acceleration,[5] the soft gripper center of mass remains below the local vertical as long as the drone's horizontal thrust and velocity are aligned[6] (Fig. 2-3(b)); this implies that the soft load imposes a torque $\boldsymbol{\tau}_{load}$ which acts to orient the quadrotor towards level. This observation, associated with the assumption that the desired final state is level and that the soft gripper is symmetric (which ensures that no torque is asserted when level), allows proving the following result.

Theorem 2 (Stability of Velocity and Attitude Controller) *Consider a quadrotor confined to the vertical x-z plane [51], with a soft payload which is symmetric about the quadrotor's thrust axis when no external forces are applied (Fig. 2-3). Denote with m_L the load's first moment of mass along the $\boldsymbol{b}_z$ axis when the quadrotor is level. Assume that $k_x = 0$ (no position control) and $\ddot{\boldsymbol{p}}_d = \boldsymbol{0}$ (no desired acceleration, so the quadrotor is level at the desired state), and that*

$$\begin{aligned} \boldsymbol{b}_z \cdot [0,0,1] &\geq 0 \\ |\boldsymbol{\tau}_{load}| &\leq |m_L \boldsymbol{b}_z \times \boldsymbol{g}| \\ \operatorname{sgn}(\boldsymbol{\tau}_{load}) &= \operatorname{sgn}(\boldsymbol{b}_z \times \boldsymbol{g}) \end{aligned} \tag{2.22}$$

Then the geometric controller in (2.19) *asymptotically stabilizes the quadrotor velocity and attitude in* (2.17).

The proof is given in Appendix 6.2, which also provides a more extensive discussion about the assumptions in eq. (2.22). Theorem 2 proves convergence for the attitude and velocity controller. However, in Sec. 4.1.1, we additionally demonstrate experimental convergence of the position controller, as well as successful performance during agile grasping – which in practice entails relaxing the requirement that the

[5] Note that in the absence of aerodynamic drag, the torque on the gripper due to gravity is equal and opposite to the torque due to the quadrotor acceleration; gravity as a uniform body force is unable to exert a net torque on the system. In this case, the quasi-static load does not deform (remains aligned with the local vertical) and exerts no torque, and the system stability is unaffected by the load. As aerodynamic forces affect the quadrotor acceleration, the gripper deforms as we discuss.

[6] The opposite case – in which thrust and aerodynamic drag are both acting opposite the quadrotor's horizontal velocity – is of necessity transient, as both drag and thrust are acting to restore the alignment of horizontal velocity with horizontal thrust. See Appendix 6.2 for further discussion.

quadrotor thrust and velocity align and that the soft gripper remains symmetric with respect to the quadrotor vertical.

2.3.3 Summary: Grasp Planning for a Soft Aerial Manipulator

Our complete planning/control approach is as follows. First, we plan a minimum-snap polynomial trajectory towards the target with a predefined grasp time t_g and fixed velocity at the moment of grasp. Given this trajectory, we use root-finding to find the time at which the horizontal distance from quadrotor to target is small enough that the base of the leading fingertips is above the target. We choose this time as the end of the approach phase t_a and compute optimal tendon lengths given this quadrotor position; similarly, we compute optimal tendon lengths for time t_g (Section 2.2.4). Then, at execution time, the quadrotor will track the minimum-snap polynomial trajectory using a globally-stable geometric controller, which converges despite the addition of the soft gripper (Section 2.3.2). At the same time, we will execute the control actions for the soft gripper in open loop, interpolating from the initial tendon lengths to the optimal lengths at t_a and t_g. Using this methodology we consistently achieve grasps at speeds up to 2m/s.

Chapter 3

Design and Implementation of a Soft Aerial Manipulator

In this chapter we describe our soft drone implementation, including material choice, design principles, electronics and fabrication. We develop a modular tendon-actuated soft gripper based on the human hand, controlled by an *Arduino Due* microcontroller over a USB connection and mounted to an *Intel Aero Ready-to-Fly* quadrotor.

3.1 Overview of the Soft Drone Prototype

Our soft drone system consists of a standard quadrotor with the (rigid and heavy) landing gear replaced by a soft gripper (Fig. 3-1). The gripper comprises four silicone rubber fingers attached to the quadrotor base, and placed in a configuration similar to the rigid landing gear they replace; this configuration allows the fingers to achieve force closure in an enveloping grasp and support a stable landing. The gripper is actuated by 16 tendons (similar to the design by Hassan *et al.* [21]) — two on each side of each finger. However, there are only four unique tendon lengths: tendons on the same side of a finger are coupled to prevent finger twist while those on the same side of each pair of fingers oriented in the same direction are coupled to enforce planarity, simplifying planning [3]. The quadrotor base uses four motors and propellers for actuation as usual. In summary, the system uses 8 control variables: 4 motor speeds

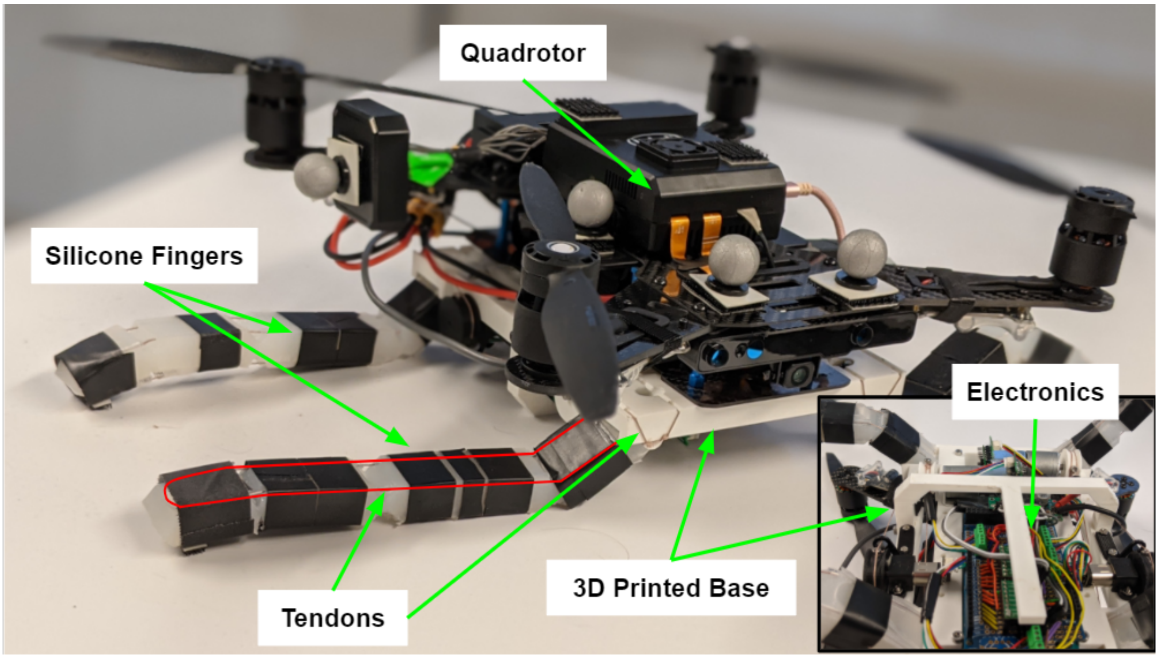


Figure 3-1: Our soft drone is a standard quadrotor (the *Intel Aero Ready To Fly*) with the landing gear replaced by a soft gripper, consisting of: four silicone rubber fingers (with braided fishing line tendons routed through embedded nylon tubing), a 3-D printed base, winches, tendon guides and other components, and an electronics module consisting of a microcontrollor, motors/drivers and related modules.

for the quadrotor, and 4 tendon lengths (later called the *rest lengths*) for the gripper. These variables control the 6-dimensional quadrotor pose and the infinite-dimensional state describing the soft gripper configuration.

3.1.1 Quadrotor

The quadrotor base of our soft drone is based on the *Intel Aero Ready To Fly* quadrotor, which is customizable and can be easily interfaced with the Robot Operating System (ROS) and standard motion capture systems (*e.g.,* Optitrack or Vicon), which are used to supplement the quadrotor's state estimation and relay the location of the target object to grasp. The quadrotor also features a relatively compact combination of an embedded real-time flight controller running the PX4 Autopilot [26] combined with a more capable companion computer (the *Intel Aero Compute Board*) dedicated to running ROS. This setup allows us to naturally relegate low-level control (running at 250Hz) to the flight controller and execute trajectory optimization on the drone (executed offline in our setup, but potentially running online at 50Hz for polynomial trajectory optimization and 1Hz for soft FEM planning – Sec. 4.1.1).

3.2 Soft Finger Design and Fabrication

3.2.1 Finger Design

Our soft fingers nominally deform continuously, but we add cutouts ("joints") to the silicone to guide the deformations; the three joints divide the palm and finger into metacarpal, proximal, medial and distal segments (Fig. 3-2). The fingers are attached to a 0.14 m-wide base ("palm") at a 45 degree angle.

We considered two different finger designs (Fig. 3-3). Our initial attempt, Design 1, had a total length of 0.18 m and was based on the design by Hassan *et al.* [21], with the latter three links (the "phalanges") of equal size. Our initial simulation and testing used this design. However, we found that it was practically ill-suited for dynamic grasping; because of the length and design of the fingers, the grasp target

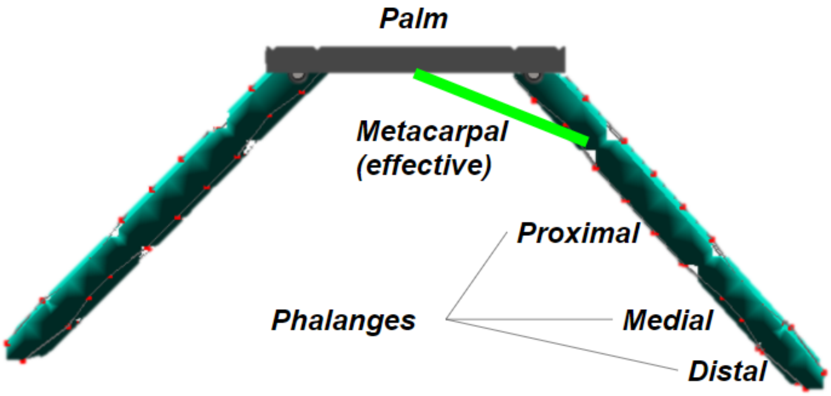


Figure 3-2: Our soft fingers are attached at a 45 degree angle to the 0.14 m palm (dark gray). Tendons (black) are attached to mesh nodes (red). Inspired by a human finger, we add cutouts to divide each finger into metacarpal, proximal, medial, and distal segments, measuring the metacarpal from the middle of the palm.

needed to be in a narrow range directly within the gripper for the grasp to succeed and high tendon forces were needed to grasp successfully.

Our second design iteration (Design 2) used a total length of 0.2 m and segment lengths inspired by a human finger (a common methodology in robotic hand design [47]). Specifically, we size the metacarpal, proximal, medial and distal segments in the same approximate 3.5:2.5:1.5:1 ratio as a human finger [1]. Further, we restrict the nominal range of motion of the cutouts to 90, 90, and 30 degrees, similar to a human finger.

To understand why these segment ratios are desirable beyond the bare fact of bio-inspiration, we observe that any choice of segment lengths represents a trade-off. As our underactuated finger moves, the fingertip follows a spiral path governed by the lengths of the proximal, medial and distal segments (Fig. 3-4). Maximizing the length of one of these segments at the expense of the others brings this spiral closer to a circle, keeping the fingertip farther from the base of the finger. This maximizes the finger's *workspace*, *i.e.,* the area enclosed by the finger as it moves (equivalent to the area swept out by the line from finger base to fingertip); if both fingertips fail to reach or surround the object to be grasped, the grasp will not succeed. However, equal proximal, medial and distal segment lengths intuitively maximize the adaptability of the finger; minimizing the maximum rigid segment length affords the finger more free-

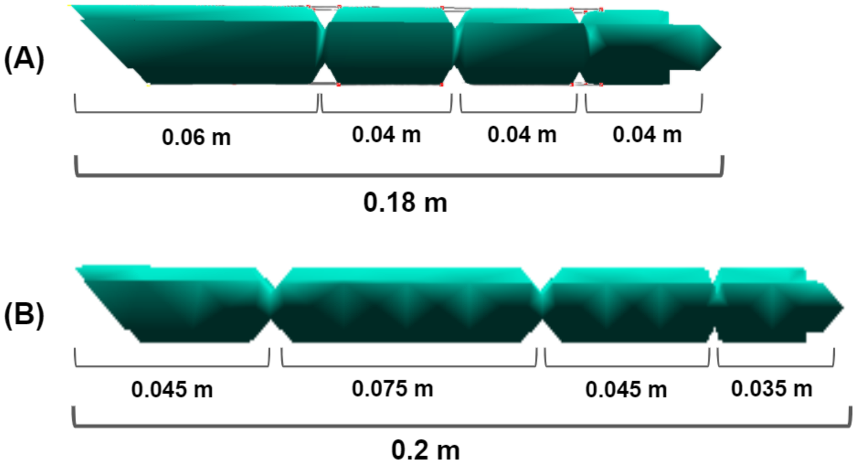


Figure 3-3: Our two candidate finger designs. Design 1 had a total length of 0.18 m and equally-sized phalanges (the last 3 segments). Design 2 is inspired by a human finger, with metacarpal, proximal, medial, and distal segments in a 3.5:2.5:1.5:1 ratio (metacarpal length is a composite of palm size and the length of the first finger segment, as in Fig. 3-2).

dom to change shape to match the object it envelops. Finally, all else being equal, maximizing the metacarpal length/palm size minimizes the tendon forces needed to grasp (shown empirically in [2]). The segment ratios of a human finger then represent an evolutionarily-selected point along this continuum, presumably optimized for everyday tasks.

We illustrate these points in Fig. 3-4, which compares a finger with evenly spaced joints with a range of motion of 60 degrees (comparable to Design 1 but of total length 0.2 m for a fair comparison) to our Design 2 [1]. Our design exhibits a greater effective workspace while still being able to grasp or envelop an object. This is visible intuitively in the first subplot, which shows Design 2 succeeding in a grasp due to its longer reach while the alternative fails. The second subplot quantifies the discrepancy, showing that our Design 2 has a longer effective reach (distance from base to fingertip) throughout its range.

[1] The underactuated finger technically has two degrees of freedom, corresponding to the two pairs of tendons. However, in computing the fingertip distance (the lower plot) we approximate its motion with a single degree of freedom by constraining all joints to rotate at the same relative rate (*i.e.,* as Joint 1 of Design 2 moves from 0-90 degrees, Joint 3 moves from 0 to 45). The upper plots show this approximation in purple, overlaid on a timelapse of a full FEM model of the finger grasping – it is clear that the 1-DOF approximation is a good one. For simplicity we also do not plot Design 2 after the first joint has exceeded 60 degrees, at which point the fingertips have contacted one another.

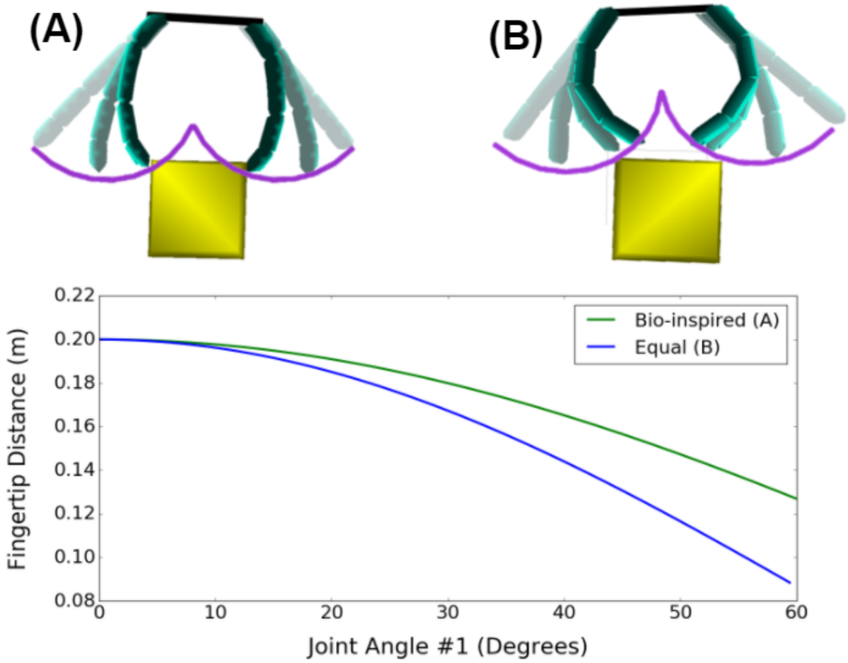


Figure 3-4: We plot the motion of (A) Design 2 and (B) a design with joints of 60, 60, 60 degrees spaced evenly along it, with the fingertip paths shown in purple. In our design, the fingertip remains farther from the finger base as the finger closes. The lower figure plots the distance to the fingertip through the range of motion of Design 2 (green) and the alternative (purple) – Design 2 has longer reach throughout (50% longer at the joint limit of 60 degrees).

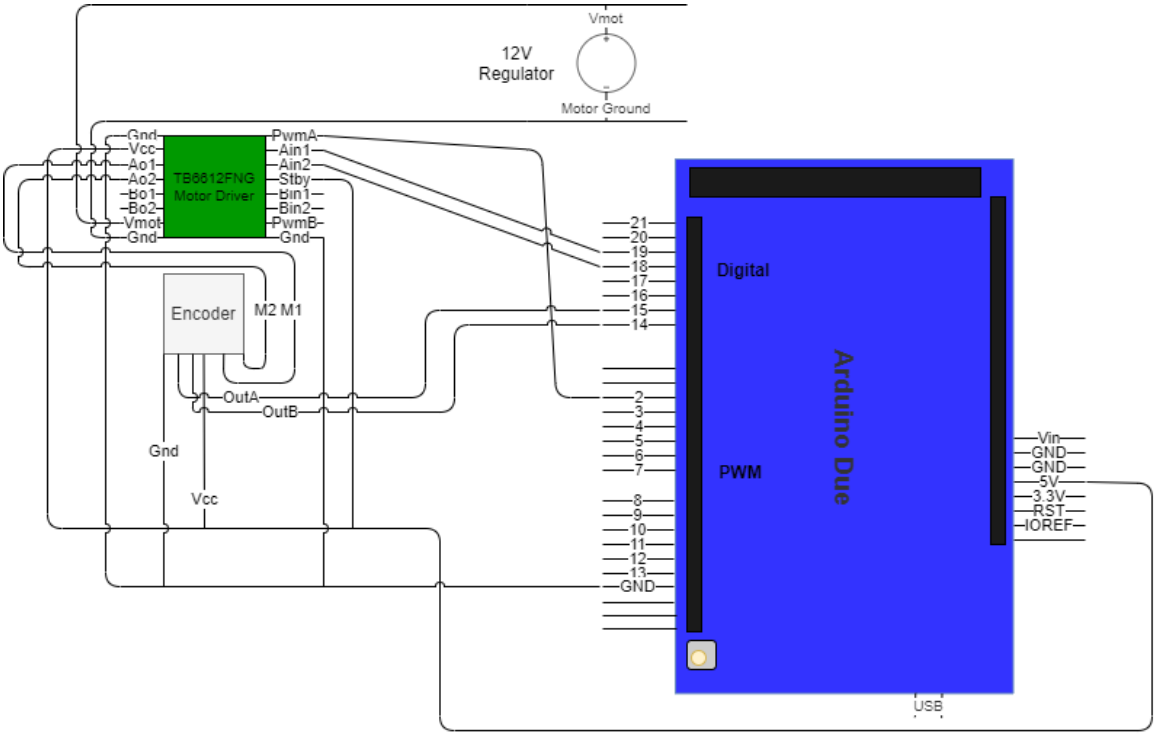


Figure 3-5: Wiring diagram for a single driver/encoder/motor. The *TB6612* driver takes in 12V power/ground from the voltage regulator and 5V signal power/ground from the microcontroller, and uses a PWM input *PwmA* and two digital inputs *Ain1, Ain2* to drive a single motor using outputs *Aout1, Aout2*. These are routed to the motor control inputs *M1, M2* on the encoder, which routs them to the motor. The encoder itself takes in 5V signal power/ground and has two digital outputs *OutA, OutB*.

3.2.2 Fabrication

We mold our fingers using *Smooth-On Dragon Skin 30* silicone rubber. To route tendons through the fingers, low-friction nylon tubing is integrated into the mold and secured with electrical tape (Fig. 3-1); initially we used silicone tubing, which bonded chemically to the silicone finger, but we found friction between tubing and tendons compromised performance unacceptably. Our tendons are 80lb-test braided fishing line (Power-Pro Super 8 Slick).

3.3 Electronics

The gripper is controlled by an *Arduino Due* microcontroller, which communicates over a serial (USB) connection with the *Intel Aero Compute Board* on the quadrotor (or any other computer). The microcontroller receives a stream of byte array messages encoding a header, footer and four tendon length setpoints[2]. It issues corresponding proportional PWM (pulse-width modulation) commands[3] to four motors via two *TB6612 Dual Motor Drivers* (with motor voltage via a *S18V20F12 12V Step-Up/Step-Down Voltage Regulator*). To control the outer tendons, we used two *298:1 Pololu Micro Metal Gearmotors*; for the inner tendons, which are under higher loads, we use two *250:1 Pololu 20D Metal Gearmotors*. Shaft angles for all motors are measured using *Magnetic Rotary Encoders*.

A wiring diagram for a single driver/encoder/motor is shown in Fig. 3-5. The *TB6612* driver takes in 12V power and ground from the voltage regulator and 5V signal power/ground from the *Due* microcontroller and can independently control two DC motors, using one PWM and two digital inputs from the microcontroller for each. The encoder has two digital outputs to the microcontroller and needs to be connected to 5V signal power/ground. Four motors/encoders therefore require 4 PWM pins and 16 digital pins on the microcontroller (out of 12 PWM and 20 digital pins available), as well as 12V motor power (from the voltage regulator) and 5V signal power. These connections are implemented on a custom (hand-soldered) shield mounted on top of the *Due*.

3.4 Gripper Base Design

The components described above are attached to a 3D-printed base (Dremel Eco-ABS – Fig. 3-6), which bolts directly to the 8 columns supporting the baseplate of the quadrotor. A removable arch over the electronic components protects them from damage in the likely eventuality of a crash.

The four fingers are mounted to 3-sided sockets in the base and attached by screws at a 45 degree angle (Fig. 3-7). Absent the socket walls around the finger, the force from the screw tended to pull the finger along the gripper base away from the desired

[2] Each encoded in a single byte, interpreted as a length from 0-256 mm.

[3] With a proportional gain of $1\,\mathrm{mm}^{-1}$, mapping to a PWM command $\in 0 \ldots 254$.

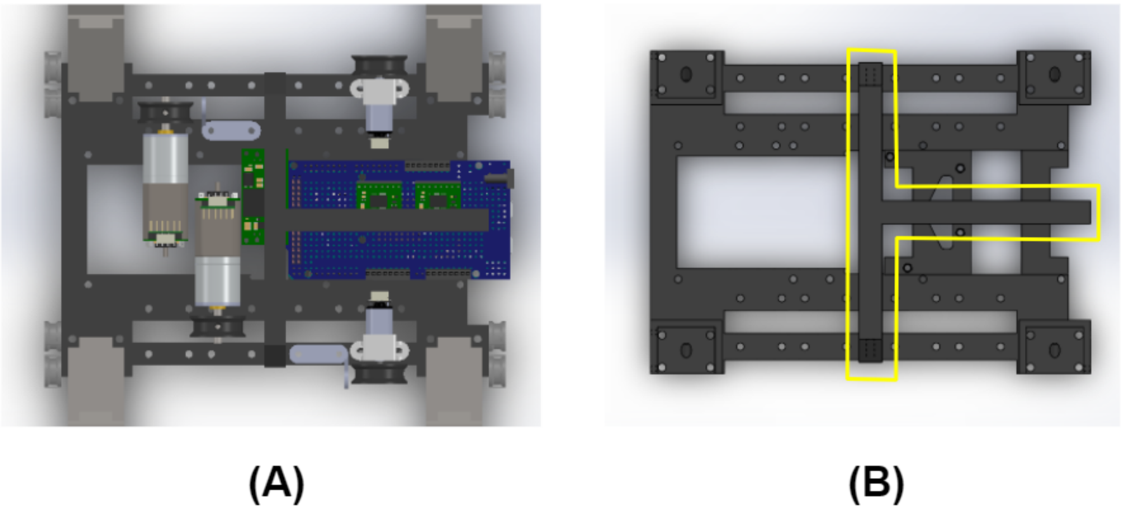


Figure 3-6: The 3-D printed gripper base, shown with components – fingers, electronics and 3-D printed *winches*, *ratchets* and *guides* – attached (A) and on its own (B), bolts directly to the baseplate of the *Aero* quadrotor. A removable arch (B, outlined in yellow) protects the electronics in the likelihood of a crash.

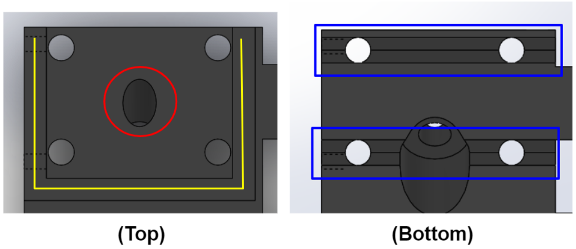
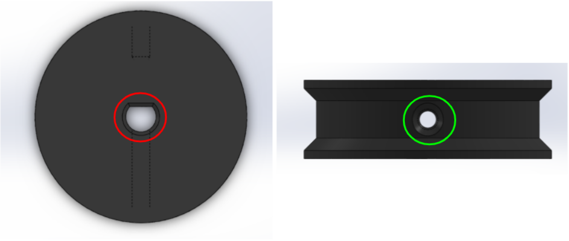


Figure 3-7: Finger socket, top and bottom views. 3 walls (traced in yellow) constrain the finger, which is screwed into the base at a 45 degree angle (red). Tendons are routed from the finger through holes in the base into channels (blue) which guide them to the motors.

mount point; adding the walls solved this issue completely. Tendons leaving the finger are routed through the base by 3-D printed channels.

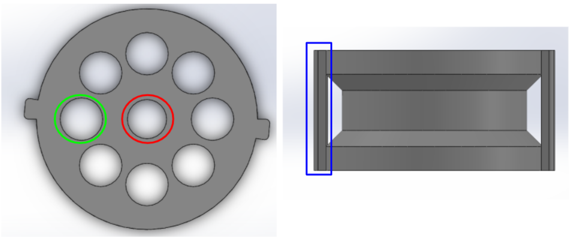
On one end, the tendons are tied to a bolt. Pairs of bolts associated with the same side of a pair of fingers (as described in Section 2.1) are inserted into a 2.5 cm-diameter *winch* (Fig. 3-8), which is mounted on the D-shaft of one of the motors. These bolts serve the dual purpose of attaching the tendons to the winch and securing the winch to the D-shaft. Additionally, a channel runs around the perimeter of each winch to secure the tendons wrapped around it.

The other end of each tendon is tied to a *ratchet* (Fig. 3-9). This allows us



(Top) (Side)

Figure 3-8: Winch, top and side views. The 2.5 cm-diameter winch mounts to a motor D-shaft (red) and has holes (green) for two bolts, which serve a dual purpose as set-screws for the shaft and tie-off points for the tendons.



(Top) (Side)

Figure 3-9: Ratchet, top and side views. The ratchet rotates freely around a central bolt-hole (red) unless constrained by a bolt in a blocking hole (green). Tendons are tied off to a bar (blue) and wrap around the ratchet as it rotates.

to easily adjust the rest length of each tendon by hand, greatly simplifying gripper setup. Ratchets rotate around a single bolt; after adjusting the initial tendon length, a second bolt is inserted through one of eight holes around the ratchet perimeter into a matching hole on the base to block the ratchet from rotating further.

Motors are bolted to mount points along the base (Fig. 3-10). A variety of mount points allow flexibility in motor placement, as well as allowing us to attach other components such as *guides*, which help ensure that tendons cleanly engage with and disengage from the winches described above.

3.4.1 Spring-Loaded Gripper

An alternative design choice involved passive tendons: replacing the motors involved in closing the gripper with a pair of springs, preloaded such that the gripper was closed when they were at rest. This saved weight (100g) and complexity by eliminating two

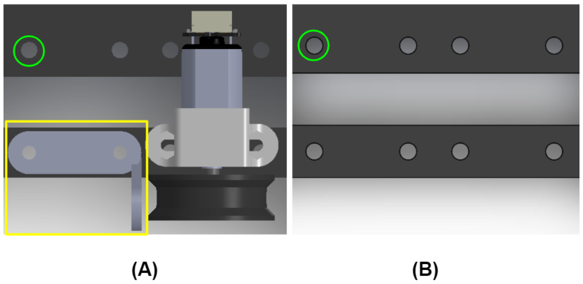


Figure 3-10: Components (shown in A) are bolted to modular mount points (green) on the base (shown alone in B). As shown, these are compatible with motors as well as 3-D printed *guides* (yellow), which help keep the tendons aligned with the winches.

motors; control and planning was nearly identical to the primary design, with the slight modification of decreasing the stiffness of the passive tendons to that of the springs they were attached to – demonstrating the flexibility of the FEM planner we use. However, although this design could grasp in hand it only succeeded in 1 out of 15 aerial tests – possibly due to plastic deformation of the springs – and was therefore abandoned.

Chapter 4

Experiments

In this chapter we report our results, including numerical experiments performed in a realistic soft dynamics simulator (*SOFA* [14]) and real grasping tests at 0.2 m/s. We show that: (i) the geometric controller is fairly insensitive to the soft payload, (ii) in simulation, our soft drone outperforms more rigid alternatives, (iii) the platform can reliably grasp unknown objects despite inaccurate positioning and initial conditions, both in simulation and in real testing.

4.1 Simulation Experiments

We validate our soft aerial manipulator design in *SOFA* [14], a popular open-source soft dynamics simulator with dedicated plugins for tendon-actuated soft manipulators [12].

We carry out two sets of experiments in simulation. The first, implemented to validate our theoretical conclusions in Section 2.1, use a set of parameters which yield physically reasonable behavior but are not directly based on our real system. Additionally, these tests used finger Design 1 (Sec. 3.2.1), contributing to an inability to grasp above 0.5 m/s. These experiments show that (i) the geometric controller converges regardless of the soft payload, (ii) the platform can reliably grasp objects of unknown shape, (iii) the decoupled controller is amenable for real-time execution, and (iv) softer systems tend to outperform more rigid analogues at dynamic grasping tasks.

Our second set of simulated experiments were implemented after prototyping the real system and attempt to directly model its behavior, using parameters as close to real life as possible. They employ finger Design 2 (Sec. 3.2.1) and an improved *adaptive controller*, enabling grasps at up to 2 m/s. Further, they show that (v) our soft drone system dramatically outperforms a fully rigid analogue at aggressive grasping tasks, succeeding a majority of the time when the rigid system fails completely.

4.1.1 Pre-Prototype Simulation Experiments

Setup

We simulate our soft aerial manipulator in SOFA [14] (see Fig. 1-1 and the video attachment). We choose a simulation timestep of 0.01 seconds. The rigid frame of the manipulator is modeled after the frame of the *Intel Ready to Fly quadrotor* (size: $0.25 \times 0.25 \times 0.04$m), while the four fingers are modeled as in Design 1 in Section 3.2.1 (each with size: $0.18 \times 0.025 \times 0.025$m). We choose quadrotor mass $m = 1\,\text{kg}$ and inertia $\boldsymbol{J} = \text{diag}\left([0.08, 0.08, 0.14]\right)\text{kg}\cdot\text{m}^2$. As material parameters, we choose Young's modulus $\mathcal{E} = 2\cdot10^4\text{N/m}^2$ (similar to a soft rubber) and Poisson's ratio $\nu = 0.25$, and derive Lamé parameters as $\mu = \frac{\mathcal{E}}{2(1+\nu)} = 8000\text{N/m}^2, \kappa = \frac{\nu\mathcal{E}}{(1+\nu)(1-2\nu)} = 6667\text{N/m}^2$ [48]. We choose a gripper density $\rho = 250\text{kg/m}^3$. The controller gains are set to $k_p = 16, k_v = 5.6, k_r = 8.81, k_\Omega = 2.54$, as in [27].

Geometric Control Evaluation

Fig. 4-1 plots the norm of the velocity, position, and rotation tracking errors defined in (2.18) for 20 runs of the geometric controller. In each run, we chose a random target location on the circle of radius 1m (similar to [51], [8]). The figure shows quick convergence to the desired state, with position error decreasing by 95% within 1.3s. The shaded area shows the 1-sigma standard deviation for the tracking errors.

Fig. 4-2 shows that convergence occurs regardless of the disturbance induced by the gripper mass. We simulate increasing gripper densities $\rho = \{10^{-2}, 10^3, 10^5\}\text{kg/m}^3$,

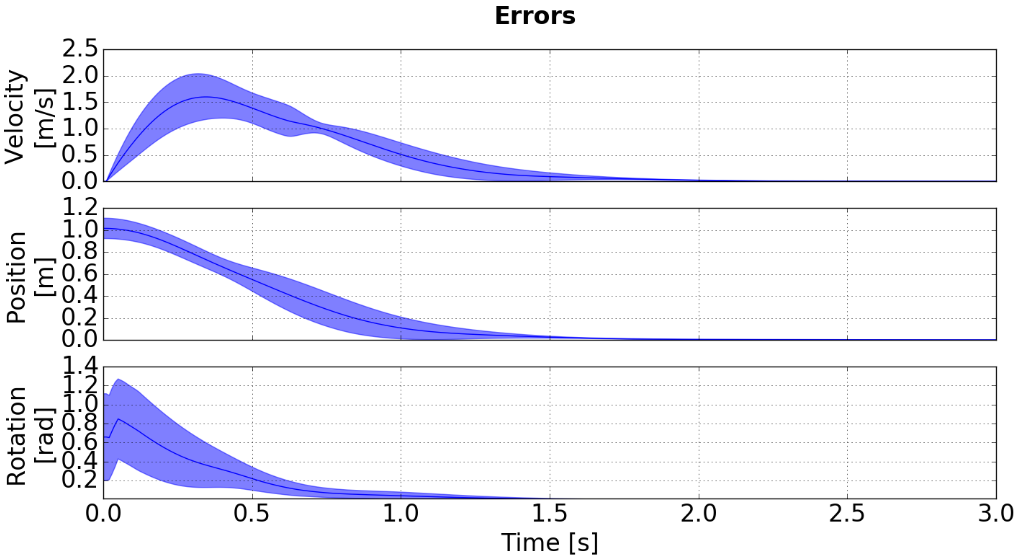


Figure 4-1: Mean and standard deviation of the tracking errors with gripper density $\rho = 250\text{kg/m}^3$. Statistics are computed over 20 runs with randomly chosen target locations on the unit circle.

ranging from a gripper ten times lighter than helium to one five times denser than lead; for each density, we repeat 20 runs and plot the tracking errors in Fig. 4-2. The figure shows that, while the increased gripper density impacts the convergence rate (in particular, larger densities induce an increased overshoot and longer convergence tails), the controller is still able to converge to the desired state within 5s.

Effects of Stiffness on Dynamic Grasping

To validate the effectiveness of the proposed soft aerial manipulator, we task it to grasp a a light object (0.05 kg) of unknown shape.

Fig. 4-3(a) shows the binary grasp outcome (success/failure) as a function of the initial quadrotor height ("z") and horizontal position ("x") with respect to the target using objective $\mathcal{C}_1$ to compute tendon lengths during the approach phase. For this objective, we require that the horizontal velocity be zero at the moment of grasp and add a trajectory setpoint 0.2m above the target. Our proposed soft gripper is able to successfully grasp for all initial conditions with $z > 0.25\,\text{m}$, corresponding to the cases where the fingertips start above the height of the target. To put things in perspective, Fig. 4-3(b) shows the same statistics for a more "rigid" design, where we chose Young's modulus to be $\mathcal{E} = 2 \cdot 10^5\text{N/m}^2$ (10 times stiffer that our design, with

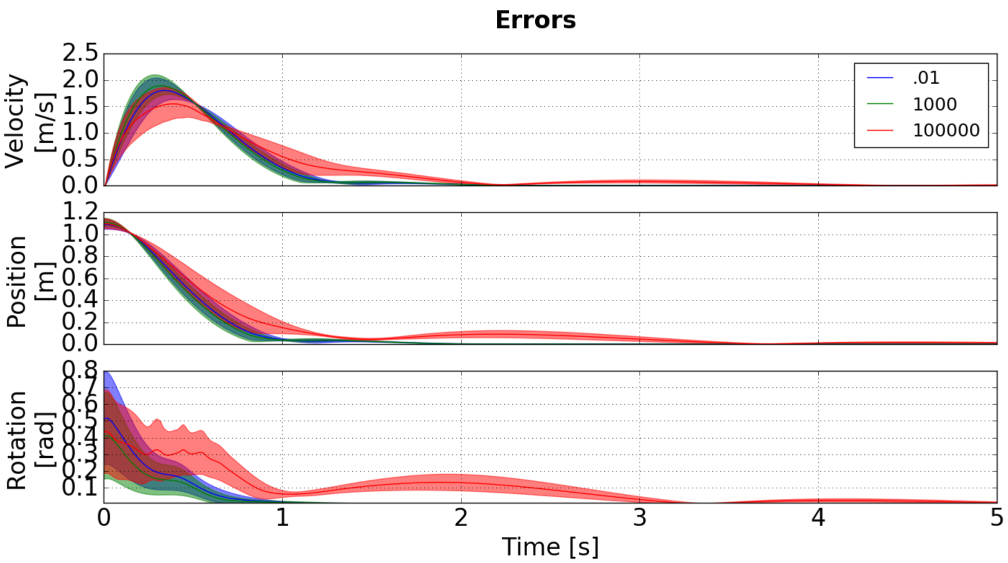


Figure 4-2: Mean and standard deviation of the tracking errors for gripper densities $\rho = \{10^{-2}, 10^3, 10^5\}\text{kg/m}^3$.

a Young's modulus analogous to a typical rubber). Comparing Fig. 4-3(a)-(b) we realize that the stiffer gripper is more likely to fail. In particular, the stiffer gripper failed in all conditions with $x > 0.3$, in which too much momentum was transferred to the target, hence preventing a successful grasp.

We carried out a similar test using the aggressive grasp objective $\mathcal{C}_2$ to compute tendon lengths during the approach phase and a fixed velocity of $[0.5, 0, -0.125]$ m/s at the moment of grasp (Fig. 4-4). As above, our proposed gripper design succeeded in all cases when the fingertips can pass above the target (which is easier when quadrotor and target are farther apart), while the stiffer gripper ($\mathcal{E} = 2 \cdot 10^5\text{N/m}^2$) imparts too much momentum to the target when starting farther away.

These tests exemplify the advantages of softness in aerial manipulation. The softer gripper adapts to deviations from the nominal quadrotor trajectory and naturally mitigates the impact of contact forces on the quadrotor and the target, thus enabling successful grasps from a wide range of initial conditions in which a more rigid solution fails.

When we transitioned to heavier objects and more aggressive grasps, we chose to use a substantially stiffer material (in fact stiffer than either alternative considered above). The multiple-order-of-magnitude greater grasp forces needed to secure an

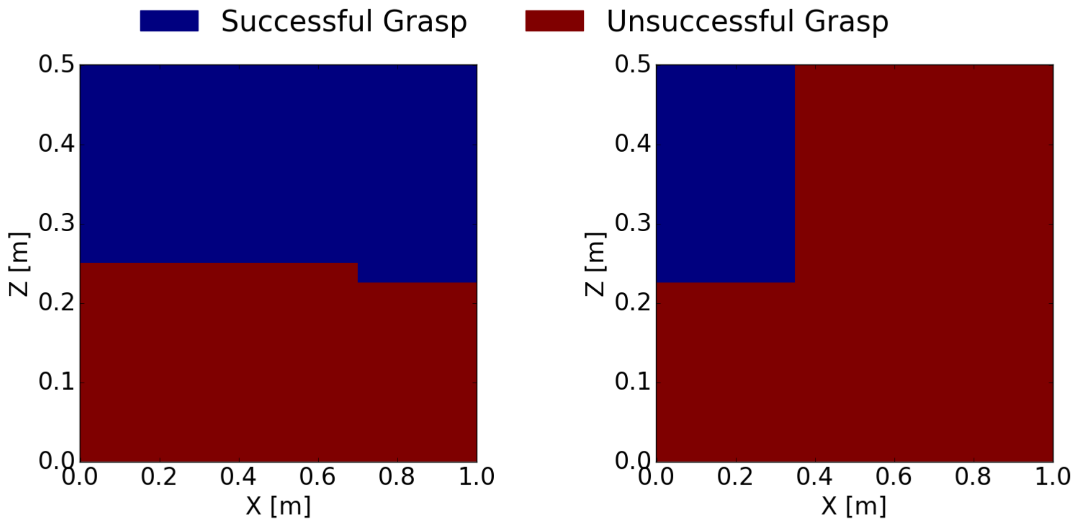


Figure 4-3: Grasp outcome as a function of the initial quadrotor height ("z") and horizontal position ("x") with respect to the target using objective $\mathcal{C}_1$ in the approach phase and with zero horizontal velocity at grasp. (a) proposed soft aerial manipulator; (b) more rigid design with higher Young's modulus ($\mathcal{E} = 2 \cdot 10^5 \mathrm{N/m}^2$).

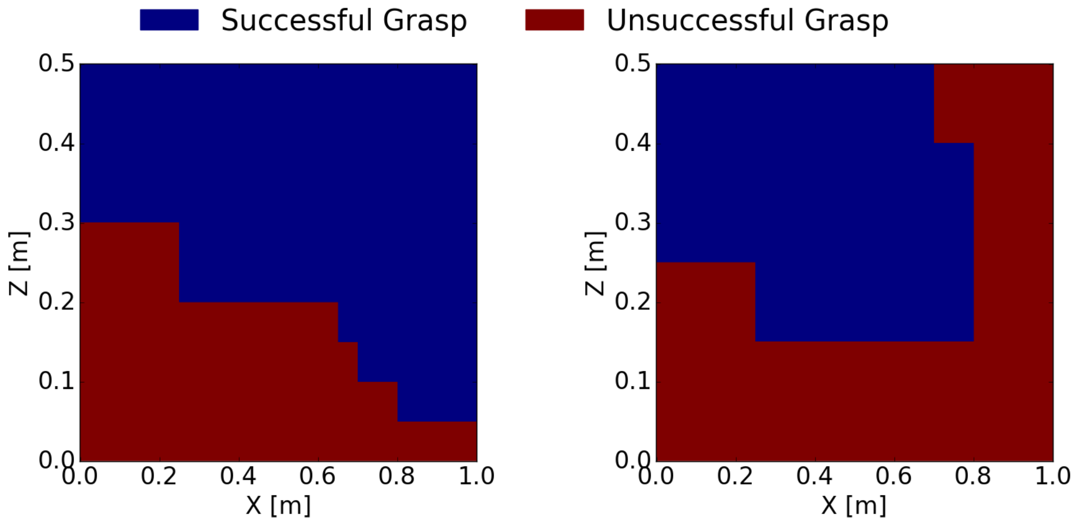


Figure 4-4: Grasp outcome as a function of the initial quadrotor height ("z") and horizontal position ("x") with respect to the target using objective $\mathcal{C}_2$ in the approach phase and with velocity at grasp of $[0.5, 0, -0.125]$ m/s. (a) proposed soft aerial manipulator; (b) more rigid design with higher Young's modulus ($\mathcal{E} = 2 \cdot 10^5 \mathrm{N/m}^2$).

object with 20x the weight at 4x the velocity (Sec. 4.2, below) necessitated order-of-magnitude increases in gripper stiffness. But the principle – that increases in stiffness correspond to decreases in adaptability, and should be avoided when possible – held true throughout our testing.

Timing

Our C++ implementation of the soft gripper trajectory optimization approach in Section 2.2 requires $\approx 1\,\mathrm{s}$ to compute a control sequence for the tendon rest lengths on an Intel Core i7-5500U CPU. Note that this can be computed offline before execution. The control is executed in open-loop, hence the computational cost to control the soft gripper during execution is negligible: interpolating the tendon actuations as discussed in Section 2.2.4 requires less than a millisecond.

For the minimum-snap quadrotor trajectory optimization, we use the *cvxopt* package in python. Our code requires $\approx 0.02\,\mathrm{s}$ to compute a minimum-snap trajectory (as before, this can be done offline). The implementation of the geometric controller is also in python and it requires $\approx 0.01\,\mathrm{s}$ to compute the instantaneous control action to be applied to the quadrotor propellers. In summary, the total computation required during execution is in the order of tens of milliseconds, and can be further reduced via an optimized multi-threaded C++ implementation.

4.1.2 Post-Prototype Simulation Experiments

Setup

For these tests, we attempt to model the measured parameters and observed behavior of our real soft drone. We choose a simulation timestep of 0.005s to prevent clipping during aggressive grasping (all controllers run at 100hz in simulation independent of the timestep). The four fingers are modeled after Design 2 in Section 3.2.1 (each with size: $0.2 \times 0.02 \times 0.025\,\mathrm{m}$). We choose quadrotor mass $m = 1.7\,\mathrm{kg}$ and inertia $\boldsymbol{J} = \mathrm{diag}\left([0.08, 0.08, 0.14]\right)\mathrm{kg}\cdot\mathrm{m}^2$ and assume aerodynamic drag on the quadrotor center of mass with body drag coefficient 0.3. For the soft components, we attempt to replicate the material properties of *Smooth-On Dragon Skin 30* in simulation: we use Young's modulus $\mathcal{E} = 1\,\mathrm{MPa}$, Poisson's ratio $\nu = 0.25$, Lamé parameters $\mu = \frac{\mathcal{E}}{2(1+\nu)} = 400000\mathrm{N/m}^2$, $\kappa = \frac{\nu\mathcal{E}}{(1+\nu)(1-2\nu)} = 333333\mathrm{N/m}^2$ [48], and density $\rho = 1000\mathrm{kg/m}^3$. Our simulated grasp target weighs 0.1 kg, approximately the same as our intended real target.

In these tests, we replace the geometric controller with the adaptive controller formulation of [19], implemented by Samuel Ubellacker. This yielded substantially more precise trajectory tracking, contributing to our greater success in aggressive grasping (details are in [16]).

Rigid vs. Soft for Aggressive Grasping

We compare our soft drone against a 2-DOF rigid gripper of identical dimensions with a pre-programmed open-close action executed in open loop.

Fig. 4-5 compares the success rates of the two candidate systems. We simulated 5 trials of grasping at 4 different desired grasp velocities (0.5 m/s, 1 m/s, 1.5 m/s, 2 m/s) and recorded the fraction of tests in which the drone remains stable and grasps the object. Our soft drone outperforms its rigid analogue at high velocities, and in particular the success of the rigid gripper drops to 40% (against 100% of the soft) at 1.5 m/s, and all "rigid" tests fail at 2 m/s. In general, the rigid gripper fails because it either makes an unexpected contact with the target or because the very large contact forces at higher velocities destabilize the controller; the magnitude of the contact forces is further exacerbated by the limitations of the SOFA simulator as discussed below. The soft gripper, on the other hand, absorbs large contact forces and consistently achieves grasping.

Limitations of Soft Simulation. We observed several limitations in our physics simulation setup. As mentioned above, clipping and penetration between objects are common in SOFA at high velocities. Moreover, rigid-to-rigid contacts often exhibit extremely large instantaneous forces; as a consequence, the stability of the high-velocity grasps (in particular when using the rigid gripper) varies dramatically depending on the simulation timestep. This is the reason why we reported the post-grasp failures

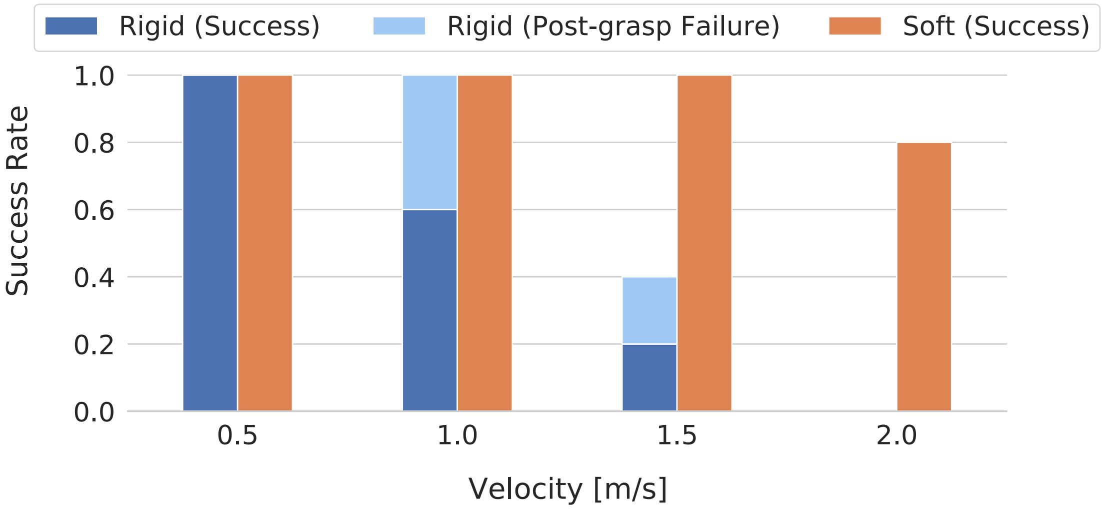


Figure 4-5: Grasp success rate for both our approach and a rigid gripper design across five trials for four different grasp velocities. "Rigid (success)" denotes that the rigid gripper achieved successful grasping and reached the desired final position, while "Rigid (Post-grasp Failure)" denotes that the system successfully grasped the target, but then the trajectory tracking controller diverged.

in Fig. 4-5 as a separate category. These issues add to the fact that our simulator does not model complex aerodynamic effects, such as the ground effect (see Sec. 4.2.2 below).

4.2 Real Experiments

In real tests, our soft drone ensures successful grasps in 21 out of 23 consecutive tests, despite unmodeled aerodynamic effects.

Real testing setup

We evaluate our real system using an Optitrack motion capture system for state estimation, communicating with the *Intel Aero* drone via ROS.

The *Aero* used a custom version of PX4 that replaced the original position and attitude controllers with the adaptive controller formulation of [19], implemented by Samuel Ubellacker and Nathan Hughes (implementation details are in [16]). The controller gains for all experiments were set to $k_p = 7.5, k_v = 6.0, k_r = 80.0, k_\Omega =$

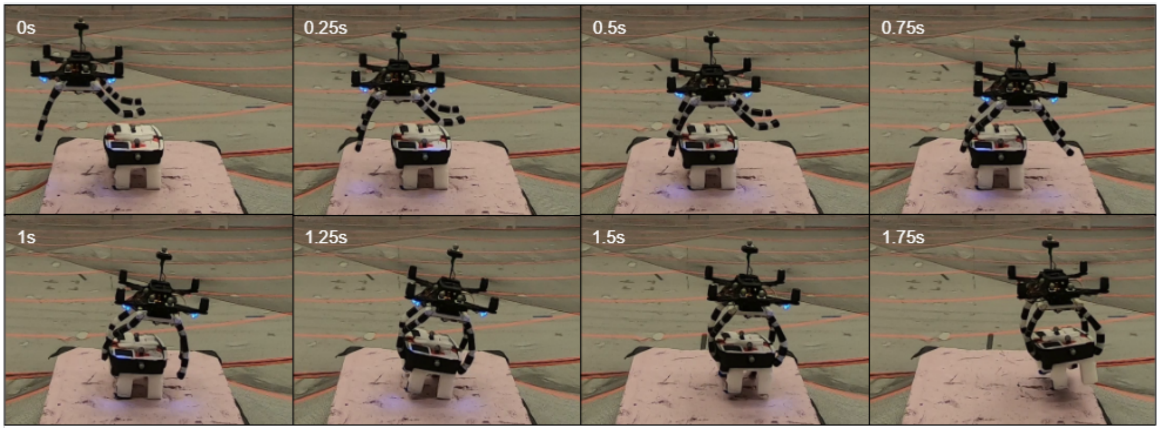


Figure 4-6: A closeup view of the grasping maneuver during a trial of the dynamic grasping tests. The gripper configuration for the "approach" phase mentioned in Section 2.2 is visible at $t = 0\,\mathrm{s}$. The drone visibly maintains consistent forward progress while grasping. The compliance of the fingers allows our *soft drone* to start contact with the grasp object before the gripper has enveloped the object fully; deformation of the back fingers is visible between the start of contact ($t = 0.5\,\mathrm{s}$) and just before the gripper fully closes ($t = 1.0\,\mathrm{s}$). The softness of the gripper also allows a more secure grasp by conforming to the geometry of the target object ($t = 1.5\,\mathrm{s}$).

$8.0, \gamma_f = 10, \gamma_\tau = 10, k_{af} = 2, k_{a\tau} = 2$. We manually calibrate the static translational offset between the drone body frame as estimated by Optitrack and the drone's center of mass to be $\{0.02, 0.03, -0.03\}$. Initial tendon rest lengths are calibrated to be $\{190, 190, 208, 208\}$ mm. We precomputed tendon lengths of $\{162, 190, 233, 208\}$ mm for approach and $\{225, 225, 190, 190\}$ mm for grasp offline, using the methodology in Section 2.2; all other computation takes place onboard. Our real grasping experiments use a foam target weighing 106 g. Our combined quadrotor and gripper weighs 1.9 kg, while the maximum load for our quadrotor is 2.0 kg. We observe good grasping performance despite the fact that we operate near the maximum payload.

4.2.1 Dynamic Grasping Results

Fig. 4-6 showcases our real system during dynamic grasping at 0.2 m/s. This maximum speed is imposed by the physical limitations of our first prototype, including thrust saturation, state estimation delays, and soft gripper latency; at the same time it enables us to provide a convincing example of dynamic grasping without the slow, precise positioning required by most existing aerial manipulators.

Total Trials	Successes	Velocity at grasp
23	21 (91.3%)	0.2 m/s

Table 4.1: Soft drone performance in real dynamic grasping.

Table 4.1 reports statistics from 23 real tests, while several examples are included in the video attachment. The 91.3% success rate confirms our soft drone can reliably perform dynamic grasping without knowledge of the target object and in the face of real-world disturbances. Moreover, our two observed failures were due to catastrophic state estimate divergence (causing the drone to crash instantly) and seem unrelated to grasping; we have observed similar errors in other tests without grasping and without the soft gripper attached. In this case they occurred at the end of the grasp trajectory on the fifth and tenth trial, causing the drone to crash while carrying the grasped object.

The 23 grasping trials took place consecutively across two days and were only interrupted to replace the batteries or reset the system after a crash. Throughout, the soft drone's performance was essentially unchanged. This consistency was despite adverse positioning errors which often resulted in at least one finger making no contact with the target (and the remaining fingers needing to adapt to compensate) – exemplifying the advantages of *morphological computing*.

4.2.2 Impact of Unmodeled Aerodynamic Effects

Despite grasping successfully, we observed several (aerodynamic) phenomena that were not captured in our simulations and which impact our ability to grasp. The airflow from the propellers reflects off the ground (or the platform our target rests on), resulting in a *ground effect* which pushes the quadrotor upward at low altitudes. Its influence is clear in Fig. 4-7, which shows the position tracking errors through a grasp trajectory with and without a target object to grasp: while the position errors in the horizontal plane (X,Y) are mostly low before grasping ($< 5\,\mathrm{s}$), the vertical error (Z) rapidly increases as the drone's target altitude decreases immediately before grasping ($t_g = 5\,\mathrm{s}$). Note that the ground effect manifests in both the grasp and no grasp cases, as the platform is present throughout.

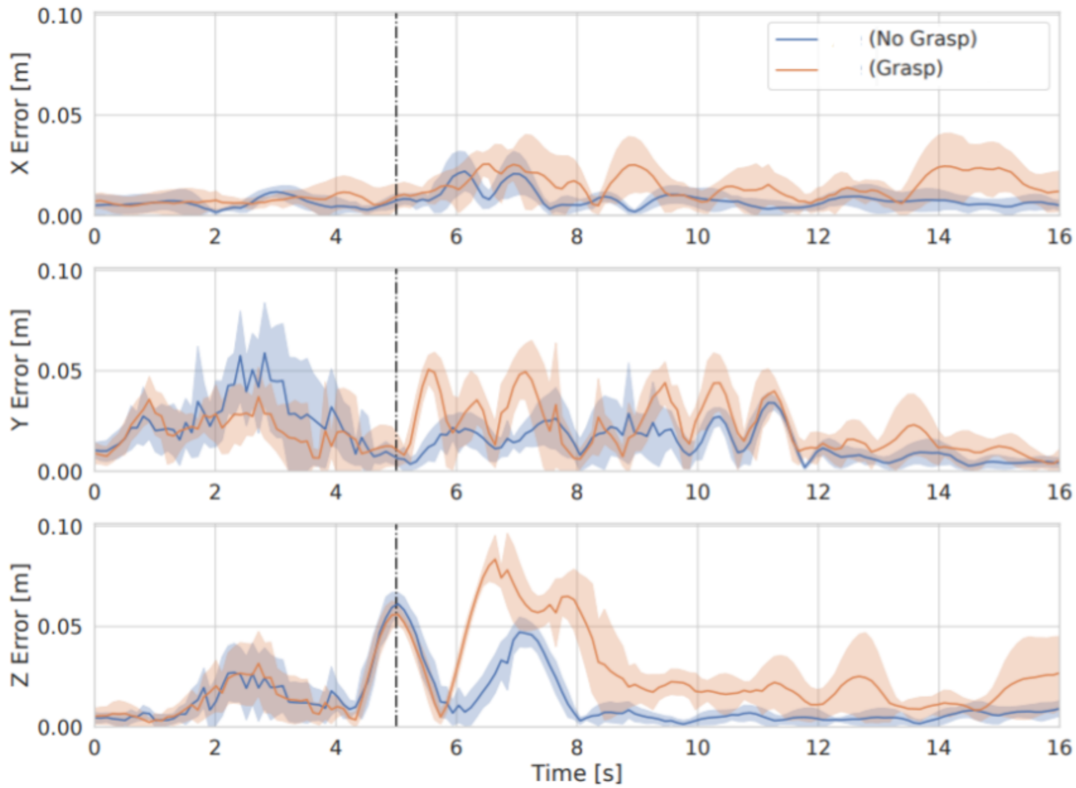


Figure 4-7: Position tracking error along the X, Y, Z axes through a grasp trajectory across 5 trials, with and without a target object to grasp ("Grasp" and "No Grasp"). The dashed black vertical line denotes time of grasp ($t_g = 5\,\mathrm{s}$). Errors likely due to unmodeled aerodynamic effects (*ground effect* and *thrust stealing*) are evident along the Z axis near t_g.

Besides the ground effect that caused poor tracking performance for the drone in the vicinity of the object, we also observe *thrust stealing* – where the propeller airflow is blocked by the grasped object – after a successful grasp. This is visible especially in the Z error after $t = 6\,\mathrm{s}$ in the grasp case, but not the no-grasp case. We can rule out the effect of the relatively small added mass of the grasped object as the sole cause of these errors: increasing the target surface area while keeping mass the same left the quadrotor effectively unable to maintain altitude after grasping, and more off-center grasps (such that the propellers are more blocked by the grasped object) led to worse post-grasp trajectory tracking performance. While these aerodynamic effects were exaggerated in our system due to our proximity to the maximum thrust of the quadrotor, we believe that they will have a substantial impact on the performance of any aerial manipulation platform and deserve serious consideration in future systems.

Chapter 5

Conclusion

We presented a soft aerial manipulator that fully exploits compliance to enable aggressive grasping of unknown objects. We proposed a decoupled control and trajectory optimization approach for the soft gripper and the rigid quadrotor base, and showed theoretically and experimentally that the quadrotor is stable despite disturbance from the soft gripper. Finally, we observed that our system achieves consistent grasp of a target object in realistic simulations and in real testing, and that it is a promising alternative to a more rigid design.

The fact that our system is capable of grasping consistently with a simple open-loop trajectory plan is, in our opinion, sufficient to demonstrate its utility compared to current paradigms of aerial manipulation. However, we believe its promise extends further. The fastest, most aggressive aerial manipulation carried out in research to our knowledge took place at 2 m/s; in simulation we can meet or exceed that mark, with larger objects, in less constrained contexts. In real life, though, we have hitherto been limited to grasping at 0.2 m/s – 10% of the speed. Pushing our real system to match its simulated promise will require substantial engineering/electro-mechanical design work, to decrease gripper latency and improve performance. However, it will likely also necessitate compensating for or at least modeling the "ground effect" and "thrust stealing" aerodynamic effects which cause our trajectory to deviate near the instant of grasp – and accurately modeling nonlinear aerodynamic phenomena of exactly this sort is among the most pressing open problems in the field of aerodynamics.

Another path forward might be incorporating vision, object detection and/or SLAM in place of the restrictive and error-prone motion-capture environment (which was associated with the only grasp failures in our tests). Other desirable future work includes investigating the proposed “soft drone” design for other applications, including aggressive landing, perching, and collision-resistant navigation. Softness has the demonstrated potential to directly address the inherent limitations of aerial manipulation – and the possibility to thereby revolutionize every context in which aerial systems currently appear.

Chapter 6

Appendices

6.1 Finite Element Methods for Soft Manipulators

This appendix provides an extended description of the finite element model (FEM) of our soft gripper, including details on how to compute the energy of the soft gripper (Sections 6.1.1-6.1.5) and how to obtain the actuator Jacobian (Section 6.1.6) required to implement the inverse kinematics of Section 2.2 in the main manuscript.

6.1.1 FEM Model and Energy

A standard approach to model a continuously deformable body is to discretize it into a finite set of nodes [48]:

$$\boldsymbol{Y} \doteq [\boldsymbol{y}_1 \; \boldsymbol{y}_2 \; \ldots \; \boldsymbol{y}_n] \in \mathbb{R}^{3\times N} \tag{6.1}$$

where $\boldsymbol{y}_i \in \mathbb{R}^3$ is the position of the i-th node.

These nodes are organized in a tetrahedral mesh, where each mesh element includes four not necessarily unique nodes. In our soft aerial manipulator (following Bern *et al.* [4]), the mechanical elements are described by this mesh (parametrized by Lamé parameters μ and κ), tendons (defined by a list of nodes the tendon passes through, rest length l and stiffness κ_{tendon}) and pins (defined by a Cartesian position and stiffness κ_{pin}).

The total energy of the soft body is the collective contribution of all of these components as well as gravity. This depends on the node positions $\boldsymbol{Y}$, tendon rest lengths $\boldsymbol{l}$ and quadrotor position $\boldsymbol{X}$. Mathematically, the total energy can be written as:

$$\begin{aligned} E(\boldsymbol{Y}, \boldsymbol{l}, \boldsymbol{X}) =& E_{mesh}(\boldsymbol{Y}) + E_{tendons}(\boldsymbol{Y}, \boldsymbol{l}) \\ & + E_{pins}(\boldsymbol{Y}, \boldsymbol{X}) + E_{gravity}(\boldsymbol{Y}) \end{aligned} \tag{6.2}$$

The *force* acting on each node is the negative gradient of the total energy with respect to the node position (a matrix of size $3 \times N$), and the *stiffness* is the Jacobian of the force or the negative Hessian of the energy; in our derivation, we vectorize $\boldsymbol{Y}$ to obtain a Hessian of size $3N \times 3N$. All Hessians or Hessian components are vectorized or referred to specifically using component-wise notation.

In the following we derive analytic expressions for all energies, forces, and stiffnesses associated with our mesh model. All of these are required to compute a stable (energy-minimizing) configuration with Newton's Method. We additionally describe an expression for the actuator Jacobian $\frac{d\boldsymbol{Y}}{d\boldsymbol{l}}$, the mapping from changes in tendon rest lengths to changes in node positions at a stable configuration (following [4]), which is computed from forces and stiffness already calculated in the process of energy minimization.

6.1.2 Mesh Energy

Our FEM mesh analysis follows [48].

The energy contribution of each tetrahedral element $\boldsymbol{Y}_{ijkl} \doteq [\boldsymbol{y}_i \;\; \boldsymbol{y}_j \;\; \boldsymbol{y}_k \;\; \boldsymbol{y}_l]$ is computed independently. We define the relative displacement of each node in the element:

$$\Delta\boldsymbol{Y}_{ijkl} = \Big[(\boldsymbol{y}_i - \boldsymbol{y}_l) \quad (\boldsymbol{y}_j - \boldsymbol{y}_l) \quad (\boldsymbol{y}_k - \boldsymbol{y}_l)\Big]$$

which contains the relative positions of vertices i, j, k with respect to vertex l.

When no force is applied, the element assumes the rest displacement $\bar{\Delta}\boldsymbol{Y}_{ijkl}$; otherwise it assumes a deformed displacement $\Delta\boldsymbol{Y}_{ijkl}$. These define the deformation gradient $\boldsymbol{G}$, which is a linearised map between rest and deformed configurations:

$$\boldsymbol{G} = \Delta\boldsymbol{Y}_{ijkl}(\bar{\Delta}\boldsymbol{Y}_{ijkl})^{-1} \tag{6.3}$$

The rest volume $\bar{v}$ of the element is:

$$\bar{v} = \frac{1}{6}|\det \bar{\Delta}\boldsymbol{Y}_{ijkl}| \tag{6.4}$$

Following [4] we use a Neo-Hookean material model, in which the strain energy density of each element is defined in terms of $\boldsymbol{G}$ and $v_F = \det(\boldsymbol{G})$, the ratio of deformed to undeformed volume of the element:

$$\psi = \frac{\mu}{2} tr(\boldsymbol{G}^{\mathsf{T}}\boldsymbol{G} - \mathbf{I}) - \mu \ln(v_F) + \frac{\kappa}{2}\ln^2(v_F) \tag{6.5}$$

and energy $E^{ijkl}_{mesh} = \psi\bar{v}$, where μ and κ are the Lamé parameters. The volumetric component of the energy $-\mu\ln(v_F) + \frac{\kappa}{2}\ln^2(v_F)$ is only quasi-convex in general; however, it is convex for $v_F < \mathrm{e}^{\frac{\mu}{\kappa}+1}$ with $\frac{\mu}{\kappa} \geq 0$, so that for any material properties convexity is guaranteed when the ratio of deformed to undeformed volume is less than e.

The nodal force, or the negative energy gradient, is defined in terms of the First Piola Stress Tensor $\mathbf{P_s}$:

$$\mathbf{P_s} = \mu(\boldsymbol{G} - \boldsymbol{G}^{-T}) + \kappa\ln(v_F)\boldsymbol{G}^{-T} \tag{6.6}$$

The force on the first three nodes in the element is:

$$\begin{bmatrix}\mathbf{F}^i_{mesh} & \mathbf{F}^j_{mesh} & \mathbf{F}^k_{mesh}\end{bmatrix} = -\bar{v}\mathbf{P_s}\bar{\Delta}\boldsymbol{Y}^{-T}_{ijkl} \tag{6.7}$$

and by conservation of momentum $\mathbf{F}^l_{mesh} = -(\mathbf{F}^i_{mesh} + \mathbf{F}^j_{mesh} + \mathbf{F}^k_{mesh})$.

The mesh stiffness is the gradient of nodal forces with respect to their positions. For the purpose of this calculation we vectorize the element $\boldsymbol{Y}_{ijkl} \in \mathbb{R}^{12}$. Let $r =$

$1, 2, \ldots, 12$ be an index and $\boldsymbol{Y}_{ijkl}^{(r)}$ be the r-th entry of $\boldsymbol{Y}_{ijkl}$. First, we calculate the gradient for the stress tensor, $\frac{d\mathbf{P_s}}{d\boldsymbol{Y}_{ijkl}}$. We define $\frac{d\Delta\boldsymbol{Y}_{ijkl}}{d\boldsymbol{Y}_{ijkl}^{(r)}}$ as the constant 3×3 matrix representing the gradient of the deformed displacement matrix with respect to coordinate $\boldsymbol{Y}_{ijkl}^{(r)}$. We first compute the gradient of the deformation gradient $\boldsymbol{G}$ with respect to coordinate $\boldsymbol{Y}_{ijkl}^{(r)}$:

$$\frac{d\boldsymbol{G}}{d\boldsymbol{Y}_{ijkl}^{(r)}} = \frac{d\Delta\boldsymbol{Y}_{ijkl}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \bar{\Delta}\boldsymbol{Y}_{ijkl}^{-1} \tag{6.8}$$

We use $\frac{d\boldsymbol{G}}{d\boldsymbol{Y}_{ijkl}^{(r)}}$ in (6.8) to compute the gradient of the stress tensor $\mathbf{P_s}$ (6.6) with respect to coordinate $\boldsymbol{Y}_{ijkl}^{(r)}$:

$$\begin{aligned} \frac{d\mathbf{P_s}}{d\boldsymbol{Y}_{ijkl}^{(r)}} = (-\mu + \kappa \ln(v_F)) &\left(-\boldsymbol{G}^{-1} \frac{d\boldsymbol{G}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \boldsymbol{G}^{-1} \right)^{\mathsf{T}} \\ + \kappa \operatorname{tr} &\left(\boldsymbol{G}^{-T} \frac{d\boldsymbol{G}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \right) \boldsymbol{G}^{-T} + \mu \frac{d\boldsymbol{G}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \end{aligned} \tag{6.9}$$

where $\frac{d\mathbf{P_s}}{d\boldsymbol{Y}_{ijkl}^{(r)}}$ is a 3×3 matrix. The relationship of nodal stiffness to the gradient of the stress tensor is the same as the relationship of the nodal forces to the stress tensor (6.7):

$$\begin{aligned} \begin{bmatrix} \frac{d\mathbf{F}_{mesh}^{i}}{d\boldsymbol{Y}_{ijkl}^{(r)}} & \frac{d\mathbf{F}_{mesh}^{j}}{d\boldsymbol{Y}_{ijkl}^{(r)}} & \frac{d\mathbf{F}_{mesh}^{k}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \end{bmatrix} &= -\bar{v} \frac{d\mathbf{P_s}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \bar{\Delta}\boldsymbol{Y}_{ijkl}^{-T} \\ -\left(\frac{d\mathbf{F}_{mesh}^{i}}{d\boldsymbol{Y}_{ijkl}^{(r)}} + \frac{d\mathbf{F}_{mesh}^{j}}{d\boldsymbol{Y}_{ijkl}^{(r)}} + \frac{d\mathbf{F}_{mesh}^{k}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \right) &= \frac{d\mathbf{F}_{mesh}^{l}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \end{aligned} \tag{6.10}$$

We vertically concatenate these four stiffnesses to form a vector $\frac{d\mathbf{F}_{mesh}^{ijkl}}{d\boldsymbol{Y}_{ijkl}^{(r)}} \in \mathbb{R}^{12}$, which is the r-th column of the 12×12 vectorized element stiffness matrix. The mesh energy E_{mesh}, force $\mathbf{F}_{mesh}$, and stiffness $\frac{d\mathbf{F}_{mesh}}{d\boldsymbol{Y}}$ are the sums of the contributions of all elements $ijkl$.

6.1.3 Tendon Energy

The *routing path* $i_1, \ldots, i_n$ is the set of node indices tendon i is attached to; each of these nodes is a *via point*. The *routing* $\boldsymbol{Y}^{t_i} = [\boldsymbol{y}_{i_1}, \ldots, \boldsymbol{y}_{i_n}]$ is the Cartesian location of each via point. The tendon length deformation γ is defined in terms of $\boldsymbol{Y}^{t_i}$ and rest length l_i:

$$\gamma_i = \sum_{k=1}^{n-1} ||\boldsymbol{Y}^{t_i}_{k+1} - \boldsymbol{Y}^{t_i}_{k}||_2 - l_i \tag{6.11}$$

Following [4], the energy of tendon i is a smooth polynomial in γ_i defined in terms of a small smoothing parameter ϵ (which we choose in practice to be zero) and tendon modulus κ_t (κ_{tendon} in the main text):

$$E^i_{tendon} = \begin{cases} 0 & \text{if } \gamma_i < -\epsilon \\ \frac{\kappa_t}{6\epsilon}\gamma_i^3 + \frac{\kappa_t}{2}\gamma_i^2 + \frac{\kappa_t \epsilon}{2}\gamma_i + \frac{\kappa_t \epsilon^2}{6} & \text{if } \gamma_i < \epsilon \\ \kappa_t \gamma_i^2 + \frac{\kappa_t \epsilon^2}{3} & \text{otherwise} \end{cases} \tag{6.12}$$

The tendon tension σ_i is the (scalar) derivative of tendon energy with respect to deformation:

$$\sigma_i = \frac{dE^i_{tendon}}{d\gamma_i} \tag{6.13}$$

The change in deformation per movement of each via point, $\frac{d\gamma_i}{d\mathbf{Y}^{t_i}}$, is equivalent to the sum of the unit vectors pointing from each via point to its neighbors (with trivial exceptions at the endpoints, where the contribution of a neighbor is 0):

$$\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}_k} = \frac{\boldsymbol{Y}^{t_i}_{k-1} - \boldsymbol{Y}^{t_i}_{k}}{||\boldsymbol{Y}^{t_i}_{k-1} - \boldsymbol{Y}^{t_i}_{k}||_2} + \frac{\boldsymbol{Y}^{t_i}_{k+1} - \boldsymbol{Y}^{t_i}_{k}}{||\boldsymbol{Y}^{t_i}_{k+1} - \boldsymbol{Y}^{t_i}_{k}||_2} \tag{6.14}$$

The force $\mathbf{F}^i_{tendon} = -\frac{dE^i_{tendon}}{d\boldsymbol{Y}^{t_i}} = -\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}}\sigma_i$. Tendon stiffness is the force Jacobian:

$$\begin{aligned}
\frac{d\mathbf{F}^i_{tendon}}{d\boldsymbol{Y}^{t_i}} &= -\frac{d^2E^i_{tendon}}{(d\boldsymbol{Y}^{t_i})^2} \\
&= \frac{d}{d\boldsymbol{Y}^{t_i}}\left(-\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}}\sigma_i\right) \\
&= -\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}}\left(\frac{d\sigma_i}{d\boldsymbol{Y}^{t_i}}\right)^{\mathsf{T}} - \frac{d^2\gamma_i}{(d\boldsymbol{Y}^{t_i})^2}\sigma_i \\
&= -\frac{d^2E^i_{tendon}}{d\gamma_i^2}\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}}\left(\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}}\right)^{\mathsf{T}} - \frac{d^2\gamma}{(d\boldsymbol{Y}^{t_i})^2}\sigma_i
\end{aligned} \tag{6.15}$$

This requires the second derivative of energy with respect to deformation $\frac{d^2E^i_{tendon}}{d\gamma_i^2}$, which is straightforward from (6.12), as well as the gradient $\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}}$ in (6.14). It also requires the tendon deformation Hessian $\frac{d^2\gamma_i}{(d\boldsymbol{Y}^{t_i})^2}$, which is the Jacobian of $\frac{d\gamma_i}{d\boldsymbol{Y}^{t_i}}$ (6.14). Each element of (6.14) is the sum of two unit vectors, so we first determine the 3×3 Jacobian of a unit vector $\hat{\boldsymbol{q}} = [q_1, q_2, q_3]^{\mathsf{T}}/\sqrt{q_1^2+q_2^2+q_3^2}$ with respect to its components:

$$\frac{d\hat{\boldsymbol{q}}_l}{dq_m} = \begin{cases} \frac{q_{l-1}^2+q_{l-2}^2}{\sqrt{q_1^2+q_2^2+q_3^2}^3} & \text{if } l = m \\ \\ \frac{-q_l q_m}{\sqrt{q_1^2+q_2^2+q_3^2}^3} & \text{if } l \neq m \end{cases} \tag{6.16}$$

The deformation Hessian $\frac{d^2\gamma_i}{(d\boldsymbol{Y}^{t_i})^2}$ is sparse, with blocks on the main 3×3 diagonal and those immediately above and below it. The main diagonal block k is the sum of the Jacobians of the unit vectors towards the neighbors of $\boldsymbol{Y}^{t_i}_k$, the block above it is the negated Jacobian of the vector towards its predecessor, and the block below it is the negated Jacobian towards its successor. As above, there are trivial exceptions for endpoints. In other words, if we define $\boldsymbol{q}^b_a$ as the unit vector from node $\boldsymbol{Y}^{t_i}_a$ to $\boldsymbol{Y}^{t_i}_b$

and $\hat{\boldsymbol{q}}_a^b = \frac{\boldsymbol{q}_a^b}{||\boldsymbol{q}_a^b||}$ the structure of $\frac{d^2\gamma_i}{(d\boldsymbol{Y}^{t_i})^2}$ is as follows:

$$\frac{d^2\gamma_i}{(d\boldsymbol{Y}^{t_i})^2} = \begin{bmatrix} \frac{d\hat{\boldsymbol{q}}_1^2}{d\boldsymbol{q}_1^2} & -\frac{d\hat{\boldsymbol{q}}_2^1}{d\boldsymbol{q}_2^1} & 0 & \cdots & 0 \\ -\frac{d\hat{\boldsymbol{q}}_1^2}{d\boldsymbol{q}_1^2} & \frac{d\hat{\boldsymbol{q}}_2^1}{d\boldsymbol{q}_2^1} + \frac{d\hat{\boldsymbol{q}}_2^3}{d\boldsymbol{q}_2^3} & & & 0 \\ 0 & -\frac{d\hat{\boldsymbol{q}}_2^3}{d\boldsymbol{q}_2^3} & \ddots & -\frac{d\hat{\boldsymbol{q}}_{n-1}^{n-2}}{d\boldsymbol{q}_{n-1}^{n-2}} & 0 \\ \vdots & & & \frac{d\hat{\boldsymbol{q}}_{n-1}^{n-2}}{d\boldsymbol{q}_{n-1}^{n-2}} + \frac{d\hat{\boldsymbol{q}}_{n-1}^{n}}{d\boldsymbol{q}_{n-1}^{n}} & -\frac{d\hat{\boldsymbol{q}}_n^{n-1}}{d\boldsymbol{q}_n^{n-1}} \\ 0 & 0 & 0 & -\frac{d\hat{\boldsymbol{q}}_{n-1}^{n}}{d\boldsymbol{q}_{n-1}^{n}} & \frac{d\hat{\boldsymbol{q}}_n^{n-1}}{d\boldsymbol{q}_n^{n-1}} \end{bmatrix} \tag{6.17}$$

This is used to compute the tendon stiffness as in (6.15).

The tendon energy E_{tendon}, force $\mathbf{F}_{tendon}$, and stiffness $\frac{d\mathbf{F}_{tendon}}{d\boldsymbol{Y}}$ are the sums of the contributions of all tendons.

6.1.4 Pin Energy

Each pin i is modeled as a spring with constant κ_{pin}, connecting a mesh node $\boldsymbol{y}_i$ belonging to the soft gripper, to a point $\boldsymbol{x}_i^{pin}$ belonging to the quadrotor base (for a given drone state $\boldsymbol{X}$). The energy for each pin i is as follows:

$$E_{pin}^i(\boldsymbol{y}_i, \boldsymbol{X}) = \kappa_{pin} \|\boldsymbol{y}_i - \boldsymbol{x}_i^{pin}\|_2 \tag{6.18}$$

The pin force $\mathbf{F}_{pin}^i = -\frac{dE_{pin}^i}{d\boldsymbol{y}_i} = -\kappa_{pin}(\boldsymbol{y}_i - \boldsymbol{x}_i^{pin})$ and stiffness $\frac{d\mathbf{F}_{pin}^i}{d\boldsymbol{y}_i} = -\kappa_{pin}\mathbf{I}$. The pin energy, force, and stiffness are the sums of the contributions of all pins.

6.1.5 Gravity Energy

We approximate the gripper mass as concentrated in the mesh nodes, and denote with m_i the mass of node i. The gravitational potential energy is determined by the mass and height of each node i:

$$E_{gravity}^i(\boldsymbol{Y}) = -m_i \, \boldsymbol{g}^{\mathsf{T}} \boldsymbol{y}_i \tag{6.19}$$

where $\boldsymbol{g} \doteq [0, 0, -9.81]^\intercal \mathrm{m/s}^2$ is the gravity vector. Gravitational force $\mathbf{F}^i_{gravity} = m_i \boldsymbol{g}$ and gravity has no stiffness. The gravitational energy and force are the sum of the contribution of all nodes.

6.1.6 Actuator Jacobian

Finally, our algorithm requires the actuator Jacobian $\frac{d\boldsymbol{Y}}{d\boldsymbol{l}}$. Our solution follows [4].

First, we note that the quasi-static assumption defines a subspace on which the overall force $\mathbf{F}$ is zero everywhere. Thus, all derivatives of $\mathbf{F}$ are likewise 0 on this subspace. Changing the tendon rest lengths $\boldsymbol{l}$ results in a change in tendon tensions $\boldsymbol{\sigma}$ and node positions $\boldsymbol{Y}$; the total derivative of $\mathbf{F}$ with respect to $\boldsymbol{l}$ yields partial derivatives in $\boldsymbol{\sigma}$ and $\boldsymbol{Y}$, which sum to zero:

$$\frac{d\mathbf{F}}{d\boldsymbol{l}} = \frac{\delta\mathbf{F}}{\delta\boldsymbol{\sigma}}\frac{d\boldsymbol{\sigma}}{d\boldsymbol{l}} + \frac{\delta\mathbf{F}}{\delta\boldsymbol{Y}}\frac{d\boldsymbol{Y}}{d\boldsymbol{l}} = 0 \tag{6.20}$$

Besides $\frac{d\boldsymbol{Y}}{d\boldsymbol{l}}$, which is the quantity for which we are solving, all the remaining terms are known. $\frac{\delta\mathbf{F}}{\delta\boldsymbol{\sigma}}$ is exactly the aggregation of the matrices described in equation (6.14) for each tendon. $\frac{d\boldsymbol{\sigma}}{d\boldsymbol{l}}$ is straightforward to compute from equations (6.11), (6.12), (6.13). And $-\frac{\delta\mathbf{F}}{\delta\boldsymbol{Y}}$ is the sparse system Hessian $\frac{d^2E}{d\boldsymbol{Y}^2}$:

$$\begin{aligned}\frac{d^2E}{d\boldsymbol{Y}^2} = &- \frac{d\mathbf{F}_{mesh}}{d\boldsymbol{Y}} - \frac{d\mathbf{F}_{tendon}}{d\boldsymbol{Y}} \\ &- \frac{d\mathbf{F}_{pin}}{d\boldsymbol{Y}} - \frac{d\mathbf{F}_{gravity}}{d\boldsymbol{Y}}\end{aligned} \tag{6.21}$$

Given these matrices, the resulting sparse linear system (6.20) can be solved with any linear equation solver.

6.2 Proof of Theorem 2

Here we prove that the quadrotor velocity and attitude, controlled as discussed in Section 2.3, converge to the desired values despite the presence of the soft gripper. The challenge lies in the fact that the soft load exerts a torque on the quadrotor center of mass which was not accounted for in the original geometric controller design.

6.2.1 Outline

We show that the attitude dynamics stabilize to a unique equilibrium, and the velocity error asymptotically approaches a limit proportional to the deviation of the attitude from equilibrium. Therefore, an unmodified geometric controller converges to a desired quadrotor attitude and velocity even in the presence of the disturbance induced by the soft load. The proof proceeds as follows:

- We analyze the tracking error and restate the theorem assumptions when restricting the quadrotor to the vertical plane.
- We show that the attitude stabilizes asymptotically to $\theta = \theta_{eq}$. In general, $\theta_{eq} = \theta_d$ iff. $\theta_d = 0$.
- We bound the attitude errors in terms of $d\theta$, the deviation of the attitude from θ_{eq}; the previous section showed that $d\theta$ asymptotically approaches zero.
- We bound the asymptotic magnitude of the total velocity error proportional to the horizontal velocity error and $|d\theta|$.
- We show that the horizontal velocity error asymptotically approaches zero. In light of the previous section, this further implies that the total velocity error vanishes asymptotically; however, we show explicitly that the vertical velocity error vanishes as well in the next section.
- Finally, we show that the vertical velocity error also asymptotically approaches zero.

6.2.2 Tracking Errors and Assumptions in the Plane

As is common in aerial manipulation (see, *e.g.,* Thomas *et al.* [51]), we consider a case in which the quadrotor is confined to the vertical plane, with the quadrotor velocity and attitude denoted as (v_x, v_z, θ) (Fig. 6-1). We denote the desired velocity and attitude as $(v_{xd}, v_{zd}, \theta_d)$ where θ_d is chosen such that velocity converges to the desired velocity (as described in (6.47) below). This allows us to express all control quantities defined on the rotation manifold SO(2) in terms of a single angle. In particular, the rotation errors in (2.18) simplify to elementary trigonometric functions:

$$\begin{aligned} \Psi &= 1 - \cos(\theta - \theta_d) \\ e_r &= \frac{d\Psi}{d\theta} = \sin(\theta - \theta_d) \\ e_\Omega &= \dot{\theta} - \dot{\theta}_d \end{aligned} \tag{6.22}$$

Similarly, the velocity errors become:

$$\boldsymbol{e}_v = [e_{v_x}, e_{v_z}]^\mathsf{T} = [v_x, v_z]^\mathsf{T} - [v_{xd}, v_{zd}]^\mathsf{T} \tag{6.23}$$

And the quadrotor body frame $\boldsymbol{R}$ is defined solely in terms of the angle θ:

$$\begin{aligned} R &= [\boldsymbol{b}_x, \boldsymbol{b}_z] \\ &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \end{aligned} \tag{6.24}$$

We define the load center of mass position relative to the quadrotor center of mass x_{load}, z_{load}, mass m_{load} and distance from the attachment point to the load center of mass when the quadrotor is vertical L_{CM} (Fig. 6-1).

Without aerodynamic drag, the load does not deform and there is no uncompensated load torque. Drag (assumed to act on the quadrotor center of mass) results in a deformation of the load and an uncompensated torque on the quadrotor (Fig. 6-2). This can act either to increase or decrease θ, depending on whether the x components of velocity and thrust are aligned. However, the state in which velocity and thrust point in opposite directions is inherently transient (because both drag and thrust act against velocity, thus quickly reducing velocity and drag); similarly for the case in which drag forces exceed thrust forces. Our assumptions on the load torque in (2.22) focus on the limiting case in which velocity and acceleration are aligned and the quadrotor has achieved static equilibrium (zero acceleration), but apply to any scenario in which the drag force is opposite and smaller than the thrust force (or does

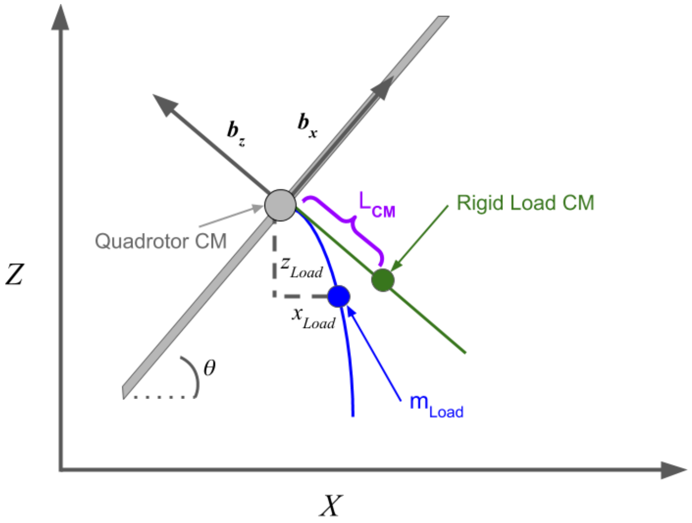


Figure 6-1: Quadrotor confined to the vertical plane. The figure shows the quadrotor body frame $\boldsymbol{R} = [\boldsymbol{b}_x, \boldsymbol{b}_z]$, angle θ, soft load center of mass relative to the quadrotor x_{Load}, z_{Load}, and the rigid load centers of mass (also compare to Fig. 2-3(b)).

not exist at all).

We denote the magnitude of the load's first moment of mass with $m_L = m_{load} L_{CM}$. We further define $g \doteq \|\boldsymbol{g}\|$, which is the norm of the gravity vector. With this notation, the assumptions in (2.22) reduce to:

$$-\pi/2 < \theta < \pi/2 \quad \text{and} \quad -\pi/2 < \theta_d < \pi/2 \tag{6.25}$$

$$|\tau_{load}| \leq m_L g \, |\sin(\theta)| \tag{6.26}$$

$$\operatorname{sgn}(\tau_{load}) = -\operatorname{sgn}(\theta) \tag{6.27}$$

Intuitively, (6.25) requires the the drone is not upside-down, while (6.26)-(6.27) require that (i) when the quadrotor is tilted, the load deforms under gravity such that its center of mass is lower and closer to the vertical than that of a corresponding rigid load (Fig. 2-3 and Fig. 6-1), and (ii) at rest and with the quadrotor level, the load center of mass is directly below the quadrotor center of mass. This implies that the torque τ_{load} (exerted by the soft load) is always of the opposite sign as θ and is upper-bounded by the torque exerted by a rigid load. We remark that these assumptions are satisfied by a symmetric soft load like that considered in this thesis.

We further define E_{load} as the gravitational potential energy of the load in the

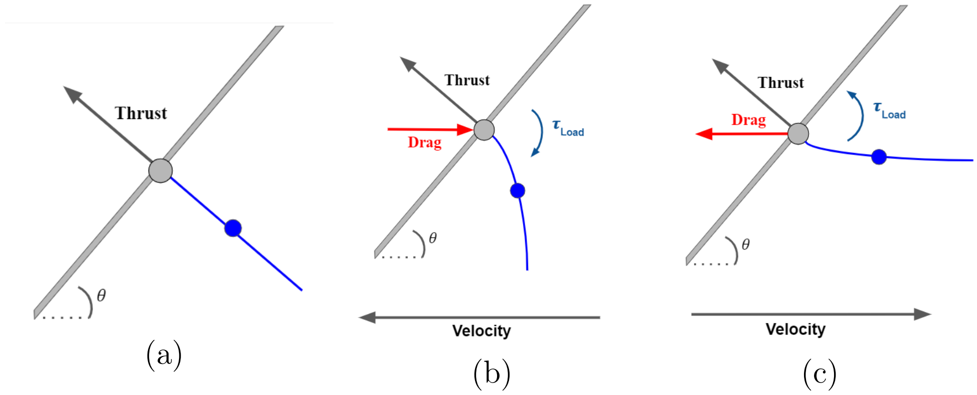


Figure 6-2: (a) With no aerodynamic force and no torque, the load does not deform and exerts no torque. (b) With aerodynamic forces opposing thrust forces, the load deforms and exerts a torque which decreases the quadrotor angle. (c) With aerodynamic forces acting with thrust forces, the load exerts a torque which increases the quadrotor angle.

non-rotating frame fixed to the quadrotor center of mass. Under our assumptions this is symmetric about $\theta = 0$ and upper-bounded by the energy of a rigid load:

$$\begin{aligned} E_{load} &= m_{load}\, z_{load}\, g \\ &\leq -m_L\ \cos(\theta)\ g \end{aligned} \tag{6.28}$$

By conservation of energy, the gradient of gravitational potential E_{load} with respect to θ corresponds to the gravitational torque τ_{load}:

$$\frac{dE_{load}}{d\theta} = -\tau_{load} \tag{6.29}$$

6.2.3 Attitude Stability

Given moment of inertia J, the attitude dynamics (restricted to the vertical plane) are:

$$J\ddot{\theta} = \tau + \tau_{load} \tag{6.30}$$

We choose the control τ as in (6.22):

$$\tau = -k_r e_r - k_\Omega e_\Omega + J\ddot{\theta}_d \tag{6.31}$$

where k_r, k_Ω are the control gains. Substituting the controller (6.31) back into the dynamics (6.30), the closed-loop attitude dynamics become:

$$J\ddot{\theta} = -k_r e_r - k_\Omega e_\Omega + J\ddot{\theta}_d + \tau_{load} \tag{6.32}$$

We show that, under assumptions (6.25)-(6.27), the closed-loop system (6.32) is stable. We define a Lyapunov function V_θ and show that this is negative semi-definite so the attitude asymptotically approaches the equilibrium angle $\theta = \theta_{eq}$. A Lyapunov function V_θ can be defined by analogy to the total energy of a double pendulum, where the upper pendulum stabilizes to θ_d rather than the vertical, the lower pendulum is non-rigid and in quasi-static equilibrium and there is angular velocity damping proportional to e_Ω:

$$V_\theta = \frac{J}{2} e_\Omega^2 + k_r \Psi + E_{load} \tag{6.33}$$

We compute the gradient of V_θ using (6.32), (6.22) and (6.28):

$$\begin{aligned}\dot{V}_\theta &= e_\Omega(J\ddot{\theta} - J\ddot{\theta}_d) + k_r e_r e_\Omega - \tau_{load} e_\Omega \\ &= e_\Omega(-k_r e_r - k_\Omega e_\Omega + \tau_{load}) + k_r e_r e_\Omega - \tau_{load} e_\Omega \\ &= -k_\Omega e_\Omega^2 \leq 0\end{aligned} \tag{6.34}$$

The Lyapunov function V_θ is negative semi-definite, so the system will converge to the largest invariant set $\{e_\Omega = 0, \dot{e}_\Omega = 0\}$; we substitute these values into (6.32) to find the equilibrium angle θ_{eq}:

$$\begin{aligned}J\ddot{\theta} - J\ddot{\theta}_d + k_\Omega e_\Omega &= -k_r e_r + \tau_{load} \\ J\dot{e}_\Omega + k_\Omega e_\Omega &= -k_r e_r + \tau_{load} \\ 0 &= -k_r e_r(\theta_{eq}) + \tau_{load}(\theta_{eq}) \\ \tau_{load}(\theta_{eq}) &= k_r \sin(\theta_{eq} - \theta_d)\end{aligned} \tag{6.35}$$

In general, the stable angle θ_{eq} is not equal to the desired angle θ_d. We establish the relationship between these two angles below. Because τ_{load} acts to decrease the magnitude of θ (6.27), it is clear that $|\theta_{eq}| \leq |\theta_d|$. This implies that $\operatorname{sgn}\sin(\theta_{eq} - \theta_d) = -\operatorname{sgn}\theta_d$. We denote $\operatorname{sgn}\theta_d$ as $\operatorname{sgn}_{\theta_d}$ and show that this is equal to $\operatorname{sgn}\theta_{eq}$ using (6.27), (6.35):

$$\begin{aligned} \operatorname{sgn}(\tau_{load}) &= \operatorname{sgn} k_r \sin(\theta_{eq} - \theta_d) \\ -\operatorname{sgn}\theta_{eq} &= \operatorname{sgn}\sin(\theta_{eq} - \theta_d) \\ \operatorname{sgn}\theta_{eq} &= \operatorname{sgn}_{\theta_d} \end{aligned} \tag{6.36}$$

We use (6.26), (6.36) to bound (6.35):

$$\begin{aligned} m_L\, g\ \ |\sin(\theta_{eq})| &\geq k_r |\sin(\theta_{eq} - \theta_d)| \\ m_L\, g\, |\sin(\theta_{eq})| &\geq -k_r \sin(\theta_{eq} - \theta_d) \operatorname{sgn}_{\theta_d} \end{aligned} \tag{6.37}$$

Which we expand using the trigonometric identity $\sin(a{-}b) = \sin(a)\cos(b){-}\cos(a)\sin(b)$:

$$\begin{aligned} m_L\, g\, |\sin(\theta_{eq})| &\geq -k_r (\sin(\theta_{eq})\cos(\theta_d) \\ &\qquad - \cos(\theta_{eq})\sin(\theta_d)) \operatorname{sgn}_{\theta_d} \\ m_L\, g\, |\sin(\theta_{eq})| &\geq -k_r (|\sin(\theta_{eq})| \cos(\theta_d) \\ &\qquad - \cos(\theta_{eq}) |\sin(\theta_d)|) \end{aligned} \tag{6.38}$$

We solve (6.38) for a bound on the magnitude of θ_{eq} as a function of θ_d:

$$\begin{aligned} m_L\, g &\geq -k_r (|\sin(\theta_{eq})| \cos(\theta_d) \\ &\qquad - \cos(\theta_{eq}) |\sin(\theta_d)|) / |\sin(\theta_{eq})| \\ m_L\, g\ &\geq -k_r \left(\cos(\theta_d) - \frac{|\sin(\theta_d)|}{|\tan(\theta_{eq})|} \right) \\ |\tan(\theta_{eq})| &\geq \frac{|\sin(\theta_d)|}{\frac{m_L\, g}{k_r} + \cos(\theta_d)} \\ |\theta_{eq}| &\geq \left| \tan^{-1} \left(\frac{\sin(\theta_d)}{\frac{m_L\, g}{k_r} + \cos(\theta_d)} \right) \right| \end{aligned} \tag{6.39}$$

Thus, we have $|\theta_d| \geq |\theta_{eq}| \geq \left|\tan^{-1}\left(\frac{\sin(\theta_d)}{(m_L\, g)/k_r + \cos(\theta_d)}\right)\right|$. When θ_d is zero, these bounds are equal and $\theta_{eq} = \theta_d = 0$; otherwise $\theta_{eq} \neq \theta_d$.

6.2.4 Bounding the Rotation Error

In the previous section, we have shown convergence to θ_{eq} rather than θ_d. This means that there exists some equilibrium rotation error $e_r(\theta_{eq})$. Further, it is convenient to express $e_r(\theta)$ in general in terms of $d\theta = \theta - \theta_{eq}$, rather than as $\theta - \theta_d$. We provide upper bounds for both these terms here. To simplify notation, in the following we denote $\sin(d\theta)$ by $s_{d\theta}$.

We can bound elementary trigonometric functions of θ_{eq} using (6.39) and the definition of the tangent:

$$|\sin(\theta_{eq})| \geq \left|\frac{\sin(\theta_d)}{\sqrt{\left(\frac{m_L\, g}{k_r}\right)^2 + 2\frac{m_L\, g}{k_r}\cos(\theta_d) + 1}}\right| \tag{6.40}$$

$$|\cos(\theta_{eq})| \leq \left|\frac{\frac{m_L\, g}{k_r} + \cos(\theta_d)}{\sqrt{\left(\frac{m_L\, g}{k_r}\right)^2 + 2\frac{m_L\, g}{k_r}\cos(\theta_d) + 1}}\right| \tag{6.41}$$

We define a constant $\eta = \frac{\frac{m_L\, g}{k_r}}{\sqrt{\left(\frac{m_L\, g}{k_r}\right)^2 + 1}}$. From (6.40), (6.41) and (6.25), we show that the rotation error associated with the equilibrium angle, $e_r(\theta_{eq})$, can be bounded as

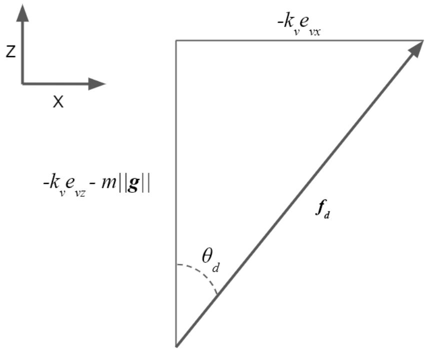


Figure 6-3: The desired force $\mathbf{f_d}$ and desired angle θ_d.

a function of $\sin(\theta_d)$:

$$\begin{aligned}
|e_r(\theta_{eq})| &= |\sin(\theta_{eq} - \theta_d)| \\
&= |\sin(\theta_{eq})\cos(\theta_d) - \cos(\theta_{eq})\sin(\theta_d)| \\
&\leq \left| \frac{\sin(\theta_d)\cos(\theta_d) - [\frac{m_L\,g}{k_r} + \cos(\theta_d)]\sin(\theta_d)}{\sqrt{\left(\frac{m_L\,g}{k_r}\right)^2 + 2\frac{m_L\,g}{k_r}\cos(\theta_d) + 1}} \right|^{1} \\
&\leq \frac{\frac{m_L\,g}{k_r}}{\sqrt{\left(\frac{m_L\,g}{k_r}\right)^2 + 2\frac{m_L\,g}{k_r}\cos(\theta_d) + 1}} |\sin(\theta_d)| \\
&\leq \eta|\sin(\theta_d)|
\end{aligned} \tag{6.42}$$

And from (6.42), $e_r(\theta)$ in general can be bounded as:

$$\begin{aligned}
|e_r(\theta)| &= |\sin(\theta_{eq} - \theta_d + d\theta)| \\
&\leq |\sin(\theta_{eq} - \theta_d)| + |s_{d\theta}| \\
&\leq \eta|\sin(\theta_d)| + |s_{d\theta}|
\end{aligned} \tag{6.43}$$

These allow us to discuss the evolution of rotation error e_r as $d\theta$ asymptotically approaches 0.

[1] [Footnote to Eq. 6.42]: This is non-trivial. Assume without loss of generality that θ_d, θ_{eq} are positive; otherwise we can multiply by sgn_{θ_d} as we do elsewhere. This implies that $\sin(\theta_d), \sin(\theta_{eq})$ are positive; $\cos(\theta_d), \cos(\theta_{eq})$ are positive from (6.25). Further, from (6.36) $\sin(\theta_{eq} - \theta_d)$ is negative if θ_d, θ_{eq} are positive. We have $\sin(\theta_{eq} - \theta_d) = \sin(\theta_{eq})\cos(\theta_d) - \cos(\theta_{eq})\sin(\theta_d)$, so $\cos(\theta_{eq})\sin(\theta_d)$ must be larger than $\sin(\theta_{eq})\cos(\theta_d)$. Therefore, to maximize the magnitude of $\sin(\theta_{eq} - \theta_d)$ we upper-bound $\cos(\theta_{eq})$ and lower-bound $\sin(\theta_{eq})$. These are the bounds provided by (6.40), (6.41).

6.2.5 Bounding the Total Velocity Error

The velocity error dynamics in the vertical plane can be written as:

$$m\dot{\boldsymbol{e}}_v = m\boldsymbol{g} + f\boldsymbol{b}_z \tag{6.44}$$

We choose the desired thrust force $\mathbf{f_d}$ and actual thrust force f as in the geometric controller equations (2.19), which when restricted to the vertical plane and under the assumptions of the theorem becomes:

$$\begin{aligned} \mathbf{f_d} &= -k_v\boldsymbol{e}_v - m\boldsymbol{g} && (6.45) \\ f &= \mathbf{f_d} \cdot \boldsymbol{b}_z && (6.46) \end{aligned}$$

We choose the desired angle θ_d based on (6.45) to align $\boldsymbol{b}_z$ with the desired force $\mathbf{f_d}$ (Fig. 6-3).

$$\theta_d = \sin^{-1}\left(\frac{-k_v e_{v_z}}{||\mathbf{f_d}||}\right) \tag{6.47}$$

We define rotation errors and choose torques as in the previous section such that θ asymptotically approaches θ_{eq}.

Similarly to Lee *et al.* [28], we rewrite the closed-loop dynamics in terms of attitude error e_r. First we add and subtract $\frac{f}{\cos(\theta-\theta_d)}\frac{\mathbf{f_d}}{||\mathbf{f_d}||}$:

$$\begin{aligned} m\dot{\boldsymbol{e}}_v = m\boldsymbol{g} &+ \frac{f}{\cos(\theta-\theta_d)}\frac{\mathbf{f_d}}{||\mathbf{f_d}||} \\ &+ \frac{f}{\cos(\theta-\theta_d)}\left(\cos(\theta-\theta_d)\boldsymbol{b}_z - \frac{\mathbf{f_d}}{||\mathbf{f_d}||}\right) \\ = m\boldsymbol{g} &+ \frac{f}{\cos(\theta-\theta_d)}\frac{\mathbf{f_d}}{||\mathbf{f_d}||} + \frac{f}{\cos(\theta-\theta_d)}\boldsymbol{w} \end{aligned} \tag{6.48}$$

Where $\boldsymbol{w}$ is defined as:

$$\boldsymbol{w} = \cos(\theta-\theta_d)\boldsymbol{b}_z - \frac{\mathbf{f_d}}{||\mathbf{f_d}||} \tag{6.49}$$

$\frac{\mathbf{f_d}}{||\mathbf{f_d}||}$ is the desired $\boldsymbol{b}_z$ orientation, which from (6.47) is $[-\sin(\theta_d), \cos(\theta_d)]^\mathsf{T}$. From (6.24) $\boldsymbol{b}_z = [-\sin(\theta), \cos(\theta)]^\mathsf{T}$. We substitute these values into (6.49).

$$\boldsymbol{w} = \cos(\theta - \theta_d) \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix} - \begin{bmatrix} -\sin(\theta_d) \\ \cos(\theta_d) \end{bmatrix} \tag{6.50}$$

Using common trigonometric identities, we show that $\boldsymbol{w}$ (6.50) is proportional to the rotation error e_r and aligned with the quadrotor axis $\boldsymbol{b}_x = [\cos(\theta), \sin(\theta)]^\mathsf{T}$ (6.24).

$$\begin{aligned} \boldsymbol{w} &= (\sin(\theta)\sin(\theta_d) + \cos(\theta)\cos(\theta_d)) \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix} \\ &\quad - \begin{bmatrix} -\sin(\theta_d) \\ \cos(\theta_d) \end{bmatrix} \\ &= \begin{bmatrix} \sin(\theta_d)(1 - \sin^2(\theta)) - \sin(\theta)\cos(\theta_d)\cos(\theta) \\ -\cos(\theta_d)(1 - \cos^2(\theta)) + \cos(\theta)\sin(\theta_d)\cos(\theta) \end{bmatrix} \\ &= \begin{bmatrix} \sin(\theta_d)\cos^2(\theta) - \sin(\theta)\cos(\theta_d)\cos(\theta) \\ -\cos(\theta_d)\sin^2(\theta) + \cos(\theta)\sin(\theta_d)\cos(\theta) \end{bmatrix} \\ &= (\sin(\theta_d)\cos(\theta) - \cos(\theta_d)\sin(\theta)) \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \\ &= \sin(\theta_d - \theta) \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \\ &= -e_r \boldsymbol{b}_x \end{aligned} \tag{6.51}$$

From (6.46) $||\mathbf{f_d}|| = \frac{f}{\cos(\theta - \theta_d)}$. This, with (6.51) and (6.45), allows us to reduce the closed loop dynamics (6.48) to the desired correction term proportional to the velocity error, as well as a disturbance term along the quadrotor $\boldsymbol{b}_x$ axis proportional to the magnitude of the desired force $\mathbf{f_d}$ and the rotation error e_r:

$$\begin{aligned} m\dot{\boldsymbol{e}}_v &= m\boldsymbol{g} + ||\mathbf{f_d}||\frac{\mathbf{f_d}}{||\mathbf{f_d}||} + ||\mathbf{f_d}||\boldsymbol{w} \\ &= m\boldsymbol{g} + \mathbf{f_d} - ||\mathbf{f_d}||e_r\boldsymbol{b}_x \\ &= -k_v\boldsymbol{e}_v - ||\mathbf{f_d}||e_r\boldsymbol{b}_x \end{aligned} \tag{6.52}$$

We show that it is possible to define a bound defined by the horizontal velocity error e_{v_x} and the deviation from equilibrium attitude $d\theta$, above which the velocity error decreases. In order to do so we define a Lyapunov candidate V_v and show that this is negative definite when $||\boldsymbol{e}_v||$ exceeds some threshold. Consider the Lyapunov candidate V_v, which if negative definite would demonstrate velocity convergence:

$$V_v \quad = \tfrac{1}{2}m||\boldsymbol{e}_v||^2 \tag{6.53}$$

$$\dot{V}_v \quad = \boldsymbol{e}_v \cdot m\dot{\boldsymbol{e}}_v \tag{6.54}$$

$$\tag{6.55}$$

We can upper-bound $\dot{V}_v$ (6.54) using (6.52):

$$\begin{aligned} \dot{V}_v &= \boldsymbol{e}_v \cdot (-k_v\boldsymbol{e}_v - ||\mathbf{f_d}||e_r\boldsymbol{b}_x) \\ &\leq ||\boldsymbol{e}_v||(-k_v||\boldsymbol{e}_v||+||\mathbf{f_d}||\,|e_r|) \end{aligned} \tag{6.56}$$

We further upper-bound $\dot{V}_v$ by considering the term $||\mathbf{f_d}||\,|e_r|$. As in Appendix 6.2.4, we define $d\theta = \theta - \theta_{eq}$ and denote $\sin(d\theta)$ by $s_{d\theta}$. First, from (6.47) the horizontal component of the desired force is equal to $-\sin(\theta_d)||\mathbf{f_d}||$:

$$|\sin(\theta_d)|\,||\mathbf{f_d}||= k_v|e_{v_x}| \tag{6.57}$$

Equations (6.57) and (6.43) allow us to bound $||\mathbf{f_d}||\,|e_r|$ as jointly affine in e_{v_x}, $s_{d\theta}$:

$$\begin{aligned} ||\mathbf{f_d}||\,|e_r| &\leq ||\mathbf{f_d}||(\eta|\sin(\theta_d|) + |s_{d\theta}|) \\ &\leq k_v\eta|e_{v_x}|+||\mathbf{f_d}||\,|s_{d\theta}| \end{aligned} \tag{6.58}$$

Further, using (6.45) we can bound $||\mathbf{f_d}||$ in terms of its components:

$$||\mathbf{f_d}|| \leq k_v ||\boldsymbol{e}_v|| + m\,g \tag{6.59}$$

Finally, we can bound $\dot{V}_v$ (6.56) in terms of $e_{v_x}, s_{d\theta}$ by using (6.58), (6.59):

$$\begin{aligned} \dot{V}_v &\leq ||\boldsymbol{e}_v||(-k_v||\boldsymbol{e}_v|| + ||\mathbf{f_d}||\,|e_r|) \\ &\leq ||\boldsymbol{e}_v||(-k_v||\boldsymbol{e}_v|| + k_v\eta|e_{v_x}| + ||\mathbf{f_d}||\ |s_{d\theta}|) \\ &\leq ||\boldsymbol{e}_v||(-k_v||\boldsymbol{e}_v|| + k_v\eta|e_{v_x}| + (k_v||\boldsymbol{e}_v|| + m\,g\,)\ |s_{d\theta}|) \\ &\leq ||\boldsymbol{e}_v||(-k_v||\boldsymbol{e}_v||(1 - |s_{d\theta}|) + k_v|e_{v_x}|\eta + m\,g\ \ |s_{d\theta}|) \end{aligned} \tag{6.60}$$

When $\boldsymbol{e}_v$ exceeds some threshold e_v^{stable}, $\dot{V}_v$ is negative definite and $||\boldsymbol{e}_v||$ decreases monotonically. This occurs when the right-hand-side of (6.60) is negative:

$$\begin{aligned} 0 &\geq ||\boldsymbol{e}_v||(-k_v||\boldsymbol{e}_v||(1 - |s_{d\theta}|) + k_v|e_{v_x}|\eta + m\,g\ \ |s_{d\theta}|) \\ 0 &\geq -k_v||\boldsymbol{e}_v||(1 - |s_{d\theta}|) + k_v|e_{v_x}|\eta + m\,g\ \ |s_{d\theta}| \\ ||\mathbf{e_v}|| &\geq \frac{|e_{v_x}|\eta + \frac{m\,g}{k_v}|s_{d\theta}|}{1 - |s_{d\theta}|} \\ &\doteq e_v^{stable} \end{aligned} \tag{6.61}$$

Equation (6.61) bounds the norm of total velocity error $||\boldsymbol{e}_v||$, but e_{v_x} appears in the bound so velocity stability has not yet been shown. In the next section, we use the results above to show asymptotic convergence to the desired horizontal velocity as θ approaches θ_{eq}.

6.2.6 Bounding Horizontal Velocity

In the previous section we defined a bound on $||\boldsymbol{e}_v||$ affine in $|e_{v_x}|, |s_{d\theta}|$ above which velocity error decreases. However, clearly $|e_{v_x}| \leq ||\boldsymbol{e}_v||$, so $|e_{v_x}| \geq e_v^{stable}$ implies $||\boldsymbol{e}_v|| \geq e_v^{stable}$. Therefore, we can substitute $|e_{v_x}|$ for $||\boldsymbol{e}_v||$ in (6.61) to define a threshold $e_{v_x}^{stable}$ proportional only to $d\theta$, above which the magnitude of horizontal velocity error $|e_{v_x}|$

decreases monotonically:

$$
\begin{aligned}
|e_{v_x}| &> \frac{|e_{v_x}|\eta + \frac{m\,g}{k_v}|s_{d\theta}|}{1-|s_{d\theta}|} \\
|e_{v_x}|(1-|s_{d\theta}|-\eta) &> \frac{m\,g}{k_v}|s_{d\theta}| \\
|e_{v_x}| &> \frac{\frac{m\,g}{k_v}|s_{d\theta}|}{(1-|s_{d\theta}|-\eta)} \\
&\doteq e_{v_x}^{stable}
\end{aligned}
\tag{6.62}
$$

$|e_{v_x}|$ decreases monotonically to $e_{v_x}^{stable}$; this threshold is linear in $|s_{d\theta}|$, so as $d\theta$ asymptotically approaches zero e_{v_x} does likewise. Note that (6.47) implies that θ_d approaches zero with e_{v_x}, and that from (6.39) $\theta_d = 0$ implies $\theta_{eq} = \theta_d$. Therefore, horizontal velocity convergence also implies that rotation error e_r vanishes asymptotically.

Equation (6.61) provides a bound affine in $|e_{v_x}|, |s_{d\theta}|$ above which velocity error decreases monotonically. We have now shown that both of these terms asymptotically approach zero; this is sufficient to establish that velocity error likewise vanishes over time. However, we also show explicitly below that vertical velocity error e_{v_z} vanishes.

6.2.7 Bounding Vertical Velocity

Given the previous results, we show that there exists a bound on e_{v_z} affine in $|e_{v_x}|, |s_{d\theta}|$ above which it decreases monotonically. Based on (6.48) and the definition $\boldsymbol{b}_x = [\cos(\theta), \sin(\theta)]^\mathsf{T}$, the vertical velocity dynamics are:

$$
m\dot{e_{v_z}} = -k_v e_{v_z} - ||\mathbf{f_d}||e_r \sin(\theta) \tag{6.63}
$$

Consider the Lyapunov candidate V_{v_z}:

$$
\begin{aligned}
V_{v_z} &= \frac{1}{2} m e_{v_z}^2 \\
\dot{V_{v_z}} &= e_{v_z}(m\dot{e_{v_z}})
\end{aligned}
\tag{6.64}
$$

As in (6.60), we can use (6.58), (6.59), (6.63) to bound $\dot{V_{v_z}}$ (6.64):

$$\begin{aligned} \dot{V_{v_z}} &\leq |e_{v_z}|(-k_v|e_{v_z}| \\ &\quad + k_v|e_{v_x}|\eta + (k_v||\boldsymbol{e}_v||+m\,g\,)|s_{d\theta}|) \end{aligned} \tag{6.65}$$

From (6.23) we have $||\boldsymbol{e}_v||\geq |e_{v_x}|+|e_{v_z}|$, so similarly to (6.61) we set the right-hand-side of (6.65) to zero in order to find a bound above which e_{v_z} decreases monotonically:

$$\begin{aligned} 0 &\geq -k_v|e_{v_z}| \\ &\quad + k_v|e_{v_x}|\eta + (k_v||\boldsymbol{e}_v||+m\,g\,)|s_{d\theta}| \\ k_v|e_{v_z}| &\geq k_v|e_{v_x}|\eta + (k_v||\boldsymbol{e}_v||+m\,g\,)|s_{d\theta}| \\ k_v|e_{v_z}| &\geq k_v|e_{v_x}|\eta + (k_v|e_{v_x}|+k_v|e_{v_z}|+m\,g\,)|s_{d\theta}| \\ k_v|e_{v_z}|(1-|s_{d\theta}|) &\geq (\eta+|s_{d\theta}|)k_v|e_{v_x}|+m\,g\;\;|s_{d\theta}| \\ |e_{v_z}| &\geq \frac{(\eta+|s_{d\theta}|)|e_{v_x}|+\frac{m\,g}{k_v}|s_{d\theta}|}{1-|s_{d\theta}|} \\ &\doteq e_{v_z}^{stable} \end{aligned} \tag{6.66}$$

$|e_{v_z}|$ decreases monotonically to $e_{v_z}^{stable}$. Because $d\theta, e_{v_x}$, asymptotically approach zero, $e_{v_z}^{stable}$ – and therefore e_{v_z} – do so as well. We have now shown explicitly that all components of the velocity error $\boldsymbol{e}_v$ asymptotically approach zero, so the velocity controller described here asymptotically tracks the desired velocity.

Bibliography

[1] Buryanov Alexander and Viktor Kotiuk. Proportions of hand segments. *International Journal of Morphology*, 28:755–758, 09 2010.

[2] S. B. Backus, L. U. Odhner, and A. M. Dollar. Design of hands for aerial manipulation: Actuator number and routing for grasping and perching. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[3] James M Bern, Pol Banzet, Roi Poranne, and Stelian Coros. Trajectory optimization for cable-driven soft robot locomotion. In *Robotics: Science and Systems (RSS)*, 2019.

[4] James M. Bern, Grace Kumagai, and Stelian Coros. Fabrication, modeling, and control of plush robots. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[5] Karen Bodie, Maximilian Brunner, Michael Pantic, Stefan Walser, Patrick Pfändler, Ueli Angst, Roland Siegwart, and Juan Nieto. An omnidirectional aerial manipulation platform for contact-based inspection. *Arxiv*, 2019.

[6] A. Bry, C. Richter, A. Bachrach, and N. Roy. Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. *Intl. J. of Robotics Research*, 37(7):969–1002, 2015.

[7] N. Bucki and M. W. Mueller. Design and control of a passively morphing quadcopter. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 9116–9122, 2019.

[8] Alvaro Caballero, Alejandro Suarez, Fran Real, Victor Vega, Manuel Bejar, Angel Rodríguez Castano, and Anibal Ollero. First experimental results on motion planning for transportation in aerial long-reach manipulators with two arms. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[9] Yitong Deng, Yaorui Zhang, Xingzhe He, Shuqi Yang, Yunjin Tong, Michael Zhang, Daniel DiPietro, and Bo Zhu. Soft multicopter control using neural dynamics identification. *arXiv preprint arXiv:2008.07689*, 08 2020.

[10] J. Dentler, S. Kannan, M. A. O. Mendez, and H. Voos. A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors. In *2016 IEEE Conference on Control Applications (CCA)*, pages 519–525, 2016.

[11] Aaron Dollar and Robert Howe. The highly adaptive SDM hand: Design and performance evaluation. *Intl. J. of Robotics Research*, 2010.

[12] Christian Duriez. Control of elastic soft robots based on real-time finite element method. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013.

[13] Davide Falanga, Kevin Kleber, Stefano Mintchev, Dario Floreano, and Davide Scaramuzza. The foldable drone: A morphing quadrotor that can squeeze and fly. *IEEE Robotics and Automation Letters*, 2018.

[14] F. Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, B. Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlík, and Stephane Cotin. SOFA: a multi-model framework for interactive physical simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, 2012.

[15] J. Fishman and L. Carlone. Control and trajectory optimization for soft aerial manipulation. In *IEEE Aerospace Conference.* , arXiv preprint arXiv: 2004.04238, pdf=https://arxiv.org/pdf/2004.04238.pdf, year=2021.

[16] J. Fishman, S. Ubellacker, N. Hughes, and L. Carlone. Dynamic grasping with a "soft" drone: From theory to practice. *arXiv preprint arXiv: 2103.06465*, 2021.

[17] Philipp Foehn, Davide Falanga, Naveen Kuppuswamy, Russ Tedrake, and Davide Scaramuzza. Fast Trajectory Optimization for Agile Quadrotor Maneuvers with a Cable-Suspended Payload. In *Robotics: Science and Systems (RSS)*, 2017.

[18] Thomas George Thuruthel, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. Control strategies for soft robotic manipulators: A survey. *Soft Robotics*, 5, 2018.

[19] F.A. Goodarzi, D. Lee, and T. Lee. Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on SE(3) for agile maneuvers. *J. Dyn. Sys., Meas., Control.*, (9), 2015.

[20] Farhad Goodarzi, Daewon Lee, and Taeyoung Lee. Geometric stabilization of quadrotor uav with a payload connected by flexible cable. *Proceedings of the American Control Conference*, 2014.

[21] T. Hassan, M. Manti, G. Passetti, N. d'Elia, M. Cianchetti, and C. Laschi. Design and development of a bio-inspired, under-actuated soft gripper. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015.

[22] C. Hintz, C. Torno, and L. R. García Carrillo. Design and dynamic modeling of a rotary wing aircraft with morphing capabilities. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 492–498, 2014.

[23] Florian Käslin, Thomas Baur, Philip Meier, Patrick Koller, Nina Buchmann, Petra D'Odorico, and Werner Eugster. Novel twig sampling method by unmanned aerial vehicle (UAV). In *Front. For. Glob. Change*, 2018.

[24] Hossein Khamseh, Farrokh Janabi-Sharifi, and Abdelkader Abdessameud. Aerial manipulation—a literature survey. *Robotics and Autonomous Systems*, 2018.

[25] Jonathan King, Dominik Bauer, Cornelia Schlagenhauf, Kai-Hung Chang, Daniele Moro, Nancy Pollard, and Stelian Coros. Design. fabrication, and evaluation of tendon-driven multi-fingered foam hands. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018.

[26] L. Meier et al. PX4 Autopilot. `https://px4.io/`, 2019.

[27] T. Lee, M. Leok, and N. H. McClamroch. Geometric tracking control of a quadrotor UAV on SE(3). In *IEEE Conf. on Decision and Control (CDC)*, pages 5420–5425, 2010.

[28] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. Control of complex maneuvers for a quadrotor UAV using geometric methods on SE(3). *ArXiv*, 2010.

[29] G. Loianno and V. Kumar. Cooperative transportation using small quadrotors using monocular vision and inertial sensing. *IEEE Robotics and Automation Letters*, 2018.

[30] G. Loianno, M. Watterson, and V. Kumar. Visual inertial odometry for quadrotors on SE(3). In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1544–1551, 2016.

[31] M. Di Luca, S. Mintchev, G. Heitz, F. Noca, and D. Floreano. Bioinspired morphing wings for extended flight envelope and roll control of small drones. *Interface Focus*, 2017.

[32] Mariangela Manti, Taimoor Hassan, Giovanni Passetti, Nicolò D'Elia, Cecilia Laschi, and Matteo Cianchetti. A bioinspired soft robotic gripper for adaptable and effective grasping. *Soft Robotics*, 2015.

[33] A. D. Marchese, K. Komorowski, C. D. Onal, and D. Rus. Design and control of a soft and continuously deformable 2d robotic manipulation system. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[34] A. D. Marchese, R. Tedrake, and D. Rus. Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[35] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2520–2525, 2011.

[36] S. Mintchev, S. de Rivaz, and D. Floreano. Insect-inspired mechanical resilience for multicopters. *IEEE Robotics and Automation Letters*, 2017.

[37] Joseph Moore and Russ Tedrake. Powerline perching with a fixed-wing uav. In *AIAA Infotech at Aerospace Conference*, 2009.

[38] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999.

[39] John-Paul Ore, Sebastian G. Elbaum, Amy J. Burgin, Baoliang Zhao, and Carrick Detweiler. Autonomous aerial water sampling. In *Field and Service Robotics*, 2013.

[40] Paul E.I. Pounds, Daniel R. Bersak, and Aaron M. Dollar. The Yale Aerial Manipulator: Grasping in flight. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[41] Pablo Ramon, Alejandro Ernesto Gómez Tamm, Francisco Javier Garcia-Rubiales, Begoña Arrue, and A. Ollero. Autonomous landing on pipes using soft gripper for inspection and maintenance in outdoor environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[42] V. Riviere, A. Manecy, and S. Viollet. Agile robotic fliers: A morphing-based approach. In *Soft robotics*, 2018.

[43] Roberto Rossi, Angel Santamaria-Navarro, Juan Andrade-Cetto, and Paolo Rocco. Trajectory Generation for Unmanned Aerial Manipulators Through Quadratic Programming. *IEEE Robotics and Automation Letters*, 2017.

[44] Daniela Rus and Michael T. Tolley. Design, fabrication and control of soft robots. *Nature*, 2015.

[45] M. Ryll, H. H. Bülthoff, and P. R. Giordano. First flight tests for a quadrotor UAV with tilting propellers. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 295–302, 2013.

[46] M. Ryll, H. H. Bülthoff, and P. R. Giordano. A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. *IEEE Trans. Robotics*, 23(2):540–556, 2015.

[47] Cornelia Schlagenhauf, Dominik Bauer, Kai-Hung Chang, Jonathan P. King, Daniele Moro, Stelian Coros, and Nancy Pollard. Control of Tendon-Driven Soft Foam Robot Hands. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, page 1—7, 2018.

[48] Eftychios Sifakis and Jernej Barbic. FEM Simulation of 3D Deformable Solids: A practitioner's guide to theory, discretization and model reduction. In *SIGGRAPH*, 2012.

[49] Koushil Sreenath, Nathan Michael, and Vijay Kumar. Trajectory generation and control of a quadrotor with a cable-suspended load - A differentially-flat hybrid system. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4888–4895, 2013.

[50] Cornelius Thiels, Johnathon Aho, Scott Zietlow, and Donald Jenkins. Use of unmanned aerial vehicles for medical product transport. *Air Medical Journal*, 2015.

[51] Justin Thomas, Giuseppe Loianno, Joseph Polin, Koushil Sreenath, and Vijay Kumar. Toward autonomous avian-inspired grasping for micro aerial vehicles. *Bioinspiration & Biomimetics*, 2014.

[52] Burak Yüksel, Gabriele Buondonno, and Antonio Franchi. Differential flatness and control of protocentric aerial manipulators with any number of arms and mixed rigid-/elastic-joints. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[53] N. Zhao, Y. Luo, H. Deng, and Y. Shen. The deformable quad-rotor: Design, kinematics and dynamics characterization, and flight performance validation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2391–2396, 2017.