

Dynamics & Control for Collaborative Aerial Manipulation

by

Venkata Naga Prasanth Kotaru

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Associate Professor Koushil Sreenath, Chair

Assistant Professor Mark Mueller

Professor Francesco Borrelli

Fall 2022

Dynamics & Control for Collaborative Aerial Manipulation

Copyright 2022
by
Venkata Naga Prasanth Kotaru

Abstract

Dynamics & Control for Collaborative Aerial Manipulation

by

Venkata Naga Prasanth Kotaru

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Associate Professor Koushil Sreenath, Chair

Aerial vehicles for physical interactions as a form of a *freely-floating manipulator* are of growing interest in recent times. Aerial vehicles like quadrotors enable us to address issues such as last-mile delivery and search & rescue. Using multiple such vehicles collaboratively increases the scope of manipulation beyond what a single vehicle can achieve. This dissertation presents theoretical and experimental contributions toward using multiple quadrotors for collaborative tasks, with an emphasis on cable-suspended payloads.

The dissertation studies the problem of *collaborative aerial manipulation* from two overarching views. The first half of the dissertation has an “individualistic view” and presents methods and algorithms for a single quadrotor/quadrotor with a cable-suspended payload. An L_1 adaptation scheme is implemented on a geometric attitude control for the quadrotor on $SO(3)$ in the presence of model uncertainties and disturbances. Next, the extended Kalman filter is modified to estimate states on S^2 . The concept of variation on manifolds is employed to linearize the system states and compute the variations in the state. Optimal and obstacle-free trajectories for a quadrotor with a suspended payload are generated using direct collocation. The planning method exploits the differential flatness for generating the trajectories in the flat space and converts non-differentiable obstacle avoidance constraints into smooth constraints using dual variables.

The second part of the dissertation has a “collaborative view” and presents results for two types of aerial manipulation, multiple quadrotors carrying a payload using suspended cables (parallel-aerial-manipulator) and a series of quadrotors connected using a flexible cable (serial-aerial-manipulator). The dissertation presents experimental results for grasping and controlling payload using a cable-suspended gripper using more than one quadrotor. Finally, the dissertation models and computes coordinate-free geometric dynamics for multiple quadrotors connected in series using a single flexible cable and shows that the system is differentially-flat.

To my family.

Contents

Contents	ii
List of Figures	iv
List of Tables	ix
1 Introduction	1
1.1 Collaborative Aerial Manipulation	2
1.2 Challenges & Contributions	4
1.3 Dissertation Outline	5
1.4 Summary	7
2 Mathematical Background	8
2.1 Notation	8
2.2 Differentiable Manifolds	9
2.3 Variation-based Linearization	11
2.4 Differential Flatness	13
2.5 Quadrotor with Suspended Payload	14
2.6 Summary	16
3 Geometric L_1 Adaptive Attitude Control on $SO(3)$	17
3.1 Introduction	17
3.2 Effects of Model Uncertainties and Disturbances on the Attitude Dynamics .	20
3.3 Geometric L_1 Adaption for Attitude Tracking Control	21
3.4 Numerical Validation	27
3.5 Experimental Results	34
3.6 Summary	37
4 Variation-based Extended Kalman Filter on S^2	38
4.1 Introduction	38
4.2 Variation Extended Kalman Filter	39
4.3 Spherical Pendulum	43
4.4 Summary	49

5	Direct Collocation for Quadrotor with a Suspended Payload	51
5.1	Introduction	51
5.2	Dynamics Modeling and System Variables	52
5.3	Collocation based Path Planning	54
5.4	Numerical Simulations	59
5.5	Experimental Results	62
5.6	Results and Discussion	65
5.7	Summary	67
6	Cable-suspended Grasping and Control for Collaborative Aerial Transportation	68
6.1	Introduction	68
6.2	Control for a Quadrotor with a Suspended Payload	70
6.3	Cable-suspended Aerial Grasping	75
6.4	Control for Two Quadrotors with a Suspended Payload	79
6.5	Control Barrier Functions for Collision Avoidance	81
6.6	Experimental Results	82
6.7	Summary	89
7	Geometric Modeling for Flexible Cable Suspended from Multiple Quadrotors	90
7.1	Introduction	90
7.2	Dynamics	92
7.3	Differential Flatness	95
7.4	Control	97
7.5	Numerical Simulations	100
7.6	Experimental Demonstrations	103
7.7	Summary	105
8	Conclusion & Future Work	106
8.1	Conclusion	106
8.2	Future Work	107
	Bibliography	109
A	Proofs	120
A.1	Proof of Theorem 3.1	120
A.2	Proof of Lemma 7.1	125
B	Derivations	127
B.1	Mapping Operators for Variation-based Extended Kalman Filter	127
B.2	Dynamics for Multiple Quadrotors Carrying a Flexible Cable	129
B.3	Variation-based Linearized Dynamics for Multiple Quadrotors Carrying a Flexible Cable	131

List of Figures

1.1	Experimental setup showing multiple quadrotors carrying (a) a triangular payload suspended using cables (b) an empty fire hose attached to two quadrotors. . . .	1
1.2	Conceptual visualization of various advantages of quadrotors with suspended payloads (a) decoupled dynamics between the payload and the quadrotor, (b) potential for collaboration using multiple quadrotors for heavier payloads, (c) makes it possible to carry payloads with larger surface areas.	2
1.3	An outline of the various chapters in the dissertation.	5
2.1	An illustration of the mathematical concepts, (a) A manifold showing the tangent vector and the tangent space. (b) Differential flatness mapping the nonlinear system to a flat space.	11
2.2	Schematic of a quadrotor with suspended payload, illustrating (a) the various system states, (b) and the two hybrid modes, slack and taut.	14
3.1	The Quadrotor model evolves on $SE(3)$, with the attitude represented by a rotation matrix $R \in SO(3)$, from body-frame to inertial-frame. A reference quadrotor evolving on $SE(3)$ with attitude states $\hat{R}, \hat{\Omega}$	18
3.2	L_1 Adaptive scheme on the Geometric Attitude Control. The <i>true model</i> captures the dynamics of the quadrotor and represents the physical plant.	27
3.3	<i>Circular Trajectory - Case I</i> : Comparison of errors between <i>true model</i> and <i>reference model</i> in the presence of constant external disturbance $\theta_t = [.95, .25, -.5]^T$ and model uncertainty ($m_a = 0.5kg$ at $r = [0.2, 0.2, 0.2]^T m$). (a) & (c) show that $\tilde{\Psi}$ and $\ \tilde{e}_\Omega\ $ decrease to zero, (b) and (d) show that the errors do not converge to zero in case of geometric control <i>without</i> L_1	28
3.4	<i>Circular Trajectory - Case I</i> (a) Uncertainty in the system calculated using (3.18), (b) Trajectory response with and without L_1 adaptation. Geometric control <i>without</i> L_1 and with model uncertainty results in an unstable response (shown in red).	29

3.5	<i>Circular Trajectory - Case II</i> : Comparison of errors between <i>true model</i> and <i>reference model</i> in the presence of time-varying external disturbances and model uncertainties (see (3.47), (3.44)). The <i>true model</i> errors and <i>reference model</i> errors did not converge to zero, however reached a bounded region about zero in case of geometric control <i>with</i> L_1 . Errors in case of geometric control <i>without</i> L_1 are not-bounded as shown in 3.8b and 3.5d.	30
3.7	Comparison between Euler L_1 and Geometric L_1 for various initialization errors: Attitude configuration error, Ψ , after 3 seconds of simulation for different added mass m_a , different initial roll angle errors between reference model and the desired, $\hat{\phi}(0) - \phi_d(0)$, and between true model and desired, $\phi(0) - \phi_d(0)$. Initial roll angles for <i>true model</i> $\phi(0)$ and <i>reference model</i> $\hat{\phi}(0)$ are swept from 0° to 179° with 5° increments, while the desired angles are $roll \equiv \phi_d \equiv pitch \equiv yaw \equiv 0^\circ$. Note, that geometric L_1 works for all considered configuration errors while the Euler L_1 failed for larger configuration errors (shown in grey). With increased added mass, Ψ increases and the range of errors for which the Euler L_1 control works decreases.	31
3.8	Step response comparison between different controls (i) Geometric L_1 , (ii) Geometric PD <i>without</i> L_1 , (iii) Geometric PID (<i>without</i> L_1). Following desired Euler angles $roll=30^\circ$, $pitch=30^\circ$, $yaw=30^\circ$ are considered with added mass $m_a = 0.5kg$ and a time-varying disturbance. Note that PID control improves the steady-state performance when there is no input saturation and L_1 control has better transient performance in both cases.	33
3.9	Experimental setup and the control architecture to track the quadrotor trajectory with L_1 adaptation to estimate the uncertainty θ . (a) Setup used to validate the Geometric L_1 control consisting of (i) Motion capture system to track the pose of the quadrotor; (ii) Ground control with the position control for the quadrotor; and (iii) Autel quadrotor with on-board Geometric Attitude control with L_1 adaptation. (b) Control architecture used in the experiments showing the ground and on-board controls.	34
3.10	<i>Hover</i> : Tracking performance with and without L_1 controller on quadrotor with attached weight. (a) the position error, and (b) the attitude error.	36
3.11	<i>Circular Trajectory</i> : Tracking performance with and without L_1 controller on quadrotor with attached weight. (a) the position error, (b) the attitude error.	36
4.1	Schematic for variation extended Kalman filter on S^2 : For each new measurement, the state is transformed into the Euclidean space using \ominus , where the measurement update is performed on the variation. The variation is mapped back onto the manifold using \oplus	39
4.2	Variation based extended Kalman filter illustrating the <i>time</i> update, <i>measurement</i> update and the <i>constraint</i> update.	40

4.3	The Spherical Pendulum model evolves on S^2 with the attitude of the pendulum represented by a unit-vector $q \in S^2$. Angular velocity of the pendulum at a point q in S^2 is in the tangent space $T_q S^2 = \{\omega \in \mathbb{R}^3 \mid \omega \cdot q = 0\}$	43
4.4	<i>Convergence of VEKF</i> : VEKF was run 100 times with different random initial states, while using the same measurements, noise covariances and initial covariance. Figure shows convergence of the pendulum attitude (top row) and the angular velocity (bottom row) estimates and the dashed black line shows the true state.	45
4.5	Comparison between VEKF, EKF (standard EKF applied on the dynamics in (4.15) through Jacobian linearization) and Angle-EKF (EKF applied on the dynamics represented using spherical coordinates); (Top) Plots for the norm of the load attitude estimate. (Bottom) Plot showing the norm of the estimated angular velocity.	46
4.6	<i>Constraint on various variation estimates</i> : The two rows illustrates the two constraints in (2.24). Constraint equation is valid for prior update (<i>first column</i>), however the measurement update (<i>second column</i>) does not satisfy the constraint equation. Finally, the constraint equation is satisfied after the constraint projection update (<i>third column</i>).	47
4.7	Eigenvalues of the state-covariance for a numerical simulation, with and without the constraint update. In both cases, two eigenvalues are zero (shown in black), corresponding to the two constraints in (2.24).	47
4.8	Comparison between the experiment results obtained through VEKF estimation (<i>solid line</i>), EKF estimation (<i>dashed line</i>) and the measurements(for \hat{q})/finite difference calculations(for $\hat{\omega}$) (<i>dotted line</i>). Top row shows load attitude estimate and angular velocities are shown in bottom row.	48
4.9	Plots of the norm of load attitude obtained through (i). measurments (dotted line), (ii). VEKF (solid line) and (iii). EKF (dashed line). Inaccurate measurement of the length of the spherical pendulum cable resulted in the offset in the norm for measurements and EKF. Also, note that the norm of q is preserved only in VEKF estimation.	48
4.10	Experimental setup for spherical pendulum model, a 3D-printed cube with reflective markers and a motion capture system to track center-of-mass position of the cube. (a) 3D-Printed cube with reflective markers suspended by a string, (b) Optitrack motion capture system	49
5.1	Obstacle avoidance constraint, $dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min}$, where the control object \mathbb{E}_{QL} is obtained by rotating R_{QL} & translating P_{QL} the initial cuboid configuration \mathbb{B}_{QL} , i.e., $\mathbb{E}_{QL} = R_{QL}\mathbb{B}_{QL} + P_{QL}$ and \mathbb{O} is the obstacle.	55
5.2	Obstacle avoidance task with the system starting at $[0, 0, 0]^T$, goal $[4, 0, 0]^T$, and a cubic obstacle centered at $[2, 0, 0]^T$ with side length equal to $1m$. The system avoids the obstacle laterally if the obstacle's height is relatively high and vertically if the obstacle's height is smaller.	60

5.3	(a) Top-view of the desired payload position trajectories for navigating three waypoints without position error tolerance for varying costs \mathcal{R}_0 , (b) $x-z$ plane view of the desired payload trajectory for the throwing task. The payload is released after being transported from the origin and reaches the desired target of $[5, 0, 0]^T$ after a projectile trajectory.	61
5.4	(a) Sideview of the quadrotor-payload system passing through a window with vertical gap greater than the cable length, (b) Trajectory of the system passing through a narrow window with a predefined waypoint in the open space of the window.	62
5.5	(a) Snapshots of passing through a window with hybrid modes. By tuning cost parameters, the optimization generates the hybrid modes to pass through the window. (b) the cable length l showing the two hybrid modes, $l = 1$ for taut and $l < 1$ for slack mode.	62
5.6	Experimental snapshots of the quadrotor-payload system executing successful obstacle avoidance and waypoint navigation maneuvers.	64
5.7	Computation time for various planning tasks with different constraint formulations. Computation time decreases with relaxed constraints and better initial guesses.	64
5.8	Complexity and robustness analysis of planning algorithm for navigation tasks in crowded environments. (a)-(d) show four successful trails with 8 random obstacles, where the difficulty of obstacle avoidance could vary depending on their locations. (e) planner fails for 12 randomly sampled obstacles due to reduced obstacle-free areas for any feasible maneuver.	66
6.1	Control structure for payload tracking control. The control architecture corresponding to the payload-cable subsystem is shown in the dashed box.	71
6.2	Types of grippers experimented for suspended-cable grasping (a) a screw-based gripper (b) electromagnet for grasping magnetic payloads	75
6.3	Various modes during grasping and transportating a target object using a quadrotor with cable suspended gripper.	77
6.4	Hybrid control approach for the quadrotor with suspended <i>payload</i> during aerial grasping. Additional (Σ_t, π_Q) mode is introduced to reduce the impact during slack-to-taut transition.	78
6.5	Two quadrotors with a common linear payload suspended from cables.	79
6.6	Position control barrier function illustration for multiple quadrotors	81
6.7	Hardware setup used for control of mulitple quadrotors with payload	83
6.8	Payload position tracking error using three controls for the payload-cable subsystem, F_{ff} - a quadrotor position control with feed-forward forces for tracking the <i>payload</i> position, F_{GeoPD} - Payload position control using the geometric PD control on S^2 defined in (6.7), F_{VMPC} - Variation-based MPC on the payload-cable subsystem defined in (6.15) . Each controller is tuned to achieve best performance individually.	84

6.9	Grasping and transportation of the <i>target payload</i> using a quadrotor with cable suspended electromagnet gripper.	85
6.10	Payload and quadrotor position during suspended aerial grasping. (i) quadrotor position control (white background) during ((a) takeoff, (c) grasping, (e) release and (g) landing), and (ii) <i>payload</i> position control (blue background) during (b), (f) <i>payload</i> tracking and (d) <i>target payload</i> transportation.	86
6.11	Distance between quadrotor and <i>payload</i> during the various stages of (a) takeoff, (b) payload tracking, (c) grasping, (d) transporting, (e) release and (g) landing.	86
6.12	Percentage of successful grasping with and without <i>payload</i> position tracking (tested on two different quadrotors with atleast 10 attempts)	87
6.13	Suspended cable grasping of a linear payload (wooden plank with metal plates on either end) using two quadrotors.	87
6.14	Tracking plots for linear (wooden plank) payload suspended using cables from two quadrotors.	88
6.15	Snapshots of the collision avoidance experiment, implementing a centralized CBF-QP on the quadrotor positions. At $t=12s$, shows the right quadrotor moving away when the left quadrotor is manually pushed towards it.	88
7.1	Multiple quadrotors carrying a flexible cable (with the cable modeled as a series of n discrete-links). Links are massless with lumped mass at the ends and indexed through $\mathcal{S} = \{0, 1, \dots, n\}$. The set $\mathcal{I} \subseteq \mathcal{S}$ gives the set of indices where the cable is attached to a quadrotor. Each link is modeled as a unit-vector $q_i \in \mathbb{S}^2$. The configuration space of this system is $Q := \mathbb{R}^3 \times (\mathbb{S}^2)^n \times (SO(3))^{n_Q}$ ($n_Q = \mathcal{I} $).	91
7.2	Configuration of the <i>multiple quadrotors carrying a flexible cable</i> illustrating the differential-flatness flat-outputs (shown in red)	97
7.3	(a). List of errors comparing the LQR control on the whole system (lqr) and feed-forward control on the quadrotor-position (ff) (b). Tracking errors for desired set-point for $n_Q = 3$ with the LQR control. Ψ_q, Ψ_R are as defined in (7.27)-(7.28)	102
7.4	Snapshots of 3 quadrotor-10 link system while tracking a setpoint. Setpoints for the quadrotor position is shown by the red-hexagrams.	102
7.5	(a). Snapshots of the <i>multiple quadrotors carrying a flexible cable</i> system while tracking the desired trajectory (b). Errors for the trajectory tracking control	103
7.6	Experiment demonstrating two quadrotors tethered using a single cable, supplying power from an external power source. Multiple quadrotors can be used collaboratively to achieve tethered flight over unknown/challenging terrain while increasing the horizontal reachability of the quadrotor.	104
7.7	Experimental demonstrations of tethered quadrotor for grasping over obstacles	104

List of Tables

2.1	List of commonly used variables for the quadrotor-payload model.	15
3.1	Notations used for the geometric L_1 adaptive control.	23
3.2	Mean and standard-deviation for position and attitude errors for Hover and Circular Trajectory with and without L_1 adaptation.	36
5.1	Parameters for planning experiments for a quadrotor with suspended payload. .	63
5.2	Complexity and robustness analysis for path planning algorithm in a crowded environment with obstacles.	66
6.1	Pros and cons of a cable suspended gripper <i>vs</i> an aerial manipulator arm	76
7.1	List of various symbols used for modeling <i>multiple quadrotors carrying a flexible cable</i> . Note: $k \in \mathcal{S}, i \in \mathcal{S} \setminus \{0\}, j \in \mathcal{I}$, WF - World frame, BF-Body-frame, $ \cdot $ represents cardinality of a set.	93

Acknowledgments

This opportunity to write the acknowledgments to my dissertation would not have been possible if not for my advisor Professor Koushil Sreenath, for trusting and providing me the opportunity to do the Ph.D. I'm immensely grateful for his support and guidance, both academic and personal, throughout my graduate studies. His feedback, dedication, patience, and mentorship helped me grow as a researcher. I would like to thank my dissertation and qualifying exam committee members, Professor Mark Mueller and Professor Francesco Borrelli, for their support and feedback. I would also like to thank Professor Murat Arcak and Professor Venkat Anantharam for serving on my qualifying exam committee.

This work was supported by funding from National Science Foundation grants CMMI-1840219, CMMI-1931853, Army Research Laboratory Grant W911NF-20-2-0105, Hellman Fellows Program, Berkeley Fire Research, Berkeley Deep Drive, Tsinghua-Berkeley Shenzhen Institute (TBSI), and the InnoHK of the Government of Hong Kong via the Hong Kong Centre for Logistics Robotics.

The research presented in this work has been shaped through numerous discussions with my labmates and collaborators. I would like to thank all my collaborators, Dr. Guofan Wu, Nikhil Baheti, Ryan Edmonson, Dr. Jun Zeng, Dr. Katherine Poggensee, Bike Zhang, and Karan Jain. Special mention to Massimiliano de Sa and Daniel Wong; I'm incredibly thankful for your support with the hardware and experiments. Hybrid Robotics has transformed from an empty room to incredible space housing numerous bipedal, quadrupedal, and flying robots during my journey as a Ph.D. student, and this has been possible only because of its amazing members. I would like to thank all my lab mates for their support and guidance, from paper parties to late-night experiments. Thank you, Ayush Agrawal, Shuxiao Chen, Zhongyu Li, Fernando Castaneda, Jason Choi, Akshay Thirugnanam, Matthew Wen, Dr. Avinash Siravuru, and Dr. Quan Nguyen, and all current and former members.

The experimental validations presented in this work would not be possible without HiPeRLab flight space. I would like to sincerely thank Prof. Mueller and the members of HiPeRLab for making me an honorary member of the lab and giving me the freedom to do the experiments in the lab.

I would like to thank Ayush for being a collaborator, labmate, housemate, travel companion, and, most of all, a fantastic friend, Bala, for your support and friendship, Avinash for the endless discussions on geometric control. Thanks to Saketh, Abhinav, Ravi Kiran, Mouni, Chaitanya, Kumar, Anirudh, Deeksha, Vidya, Radhika, and R12 group for keeping me sane and grounded. Thank you to all my friends and family for their support and encouragement. Finally, I would like to thank my brother, Kiran, for putting up with my impatience and my parents for all their support and sacrifices that made it possible to cross the finish line.

Chapter 1

Introduction

Aerial manipulation is the concept of using unmanned aerial vehicles (UAVs), typically quadrotors or hexacopters, for manipulation tasks such as grasping, transporting, etc. These vehicles are sometimes equipped with additional mechanisms such as a gripper, manipulator, or suspended cables. Quadrotors have been used in a wide range of manipulation tasks [1–5], grasping [6, 7], transporting [8–10], and sensing [11–13]. Various designs, mechanisms, and control schemes have been presented over the years to achieve the necessary manipulation objectives. The typical aerial manipulation systems are largely constrained in their flight time and payload capacity due to their limited hardware and power supply. However, simplicity in their design and control makes quadrotors ideal agents for collaboration where multiple quadrotor UAVs can operate together to achieve the common manipulation goal, thereby increasing their payload capacity.

This dissertation addresses the issue of multi-agent collaborative manipulation using quadrotors. A multi-quadrotor collaborative manipulation system consists of two or more quadrotors working together on a common manipulation task; it typically involves physical

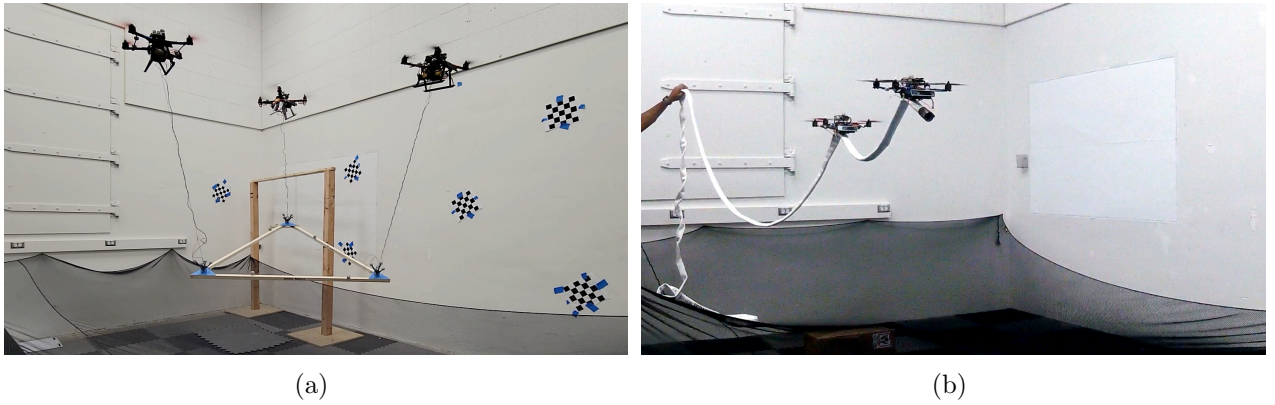


Figure 1.1: Experimental setup showing multiple quadrotors carrying (a) a triangular payload suspended using cables (b) an empty fire hose attached to two quadrotors.

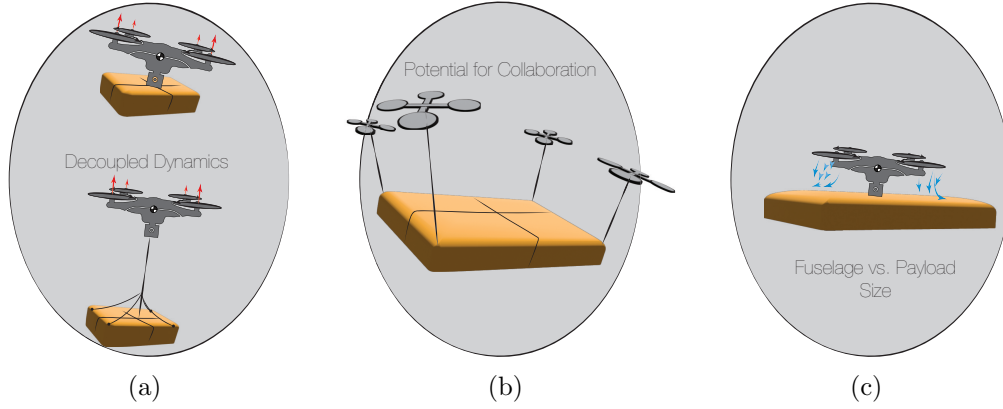


Figure 1.2: Conceptual visualization of various advantages of quadrotors with suspended payloads (a) decoupled dynamics between the payload and the quadrotor, (b) potential for collaboration using multiple quadrotors for heavier payloads, (c) makes it possible to carry payloads with larger surface areas.

interactions between the quadrotors via the shared object. Ensuring the stable and safe operation of the multi-agent system requires safe and reliable individual agents, as well as an understanding of how these agents behave when interacting physically. To this end, first, the dissertation looks into developing methods and tools for individual agents. An adaption scheme is employed for attitude control of the quadrotor in the presence of disturbances. Estimation and planning methods for a quadrotor with a suspended payload are presented. The second part of the dissertation presents methods of using multiple quadrotors for collaborative transportation, along with modeling and studying the dynamics of a flexible cable connected to multiple quadrotors.

The rest of the chapter discusses types of collaborative manipulations with an emphasis on suspended cable manipulation. The later part of the chapter discussed some of the challenges associated with aerial manipulation, along with the contributions of this dissertation. Finally, the dissertation outline is presented with a brief note on the content of each chapter.

1.1 Collaborative Aerial Manipulation

Collaborative aerial manipulation is when two or more aerial vehicles are used to perform a shared task, such as transporting a payload or manipulating an object together. For instance, Figure 1.1 shows an experimental setup for a couple of collaborative manipulation tasks, (a) carrying a triangular payload using three quadrotors and (b) two quadrotors supporting the end of (an empty) fire hose. These systems have complex behavior due to the physical interactions between individual agents and the shared object. These interactions and the corresponding forces/wrench depend on the type of connection and actuation between the object and the agent. For instance, (i) the agent can be rigidly attached (6 degrees-of-

freedom (DOFs) constraint), *(ii)* rigidly connected at a point (spherical/ball joint) (only the three translational DOF are constrained but is free to rotate), *(iii)* is suspended using a cable or *(iv)* is connected using a manipulation arm that adds additional DOF and degrees of actuation (DOA).

The choice of manipulation type, to some extent, is application specific; for instance, manipulating the pose of an object will require an actuated manipulator arm. Whereas, to pick and place an object, in addition to an actuated manipulator arm, can also be achieved by rigidly attaching the object to the quadrotor or using a gripper that is either rigidly attached or suspended from the quadrotor. The active mechanisms offer more degrees-of-freedom (DOFs) to the UAV and help in active aerial manipulation. However, these advantages come at the cost of loss of agility of the small UAVs due to the added inertia to the system. Moreover, actuated gripper arms introduce dynamic coupling between the manipulator and quadrotor, which must be accurately captured in the mathematical model and compensated by the controller.

A cable-suspended gripper/payload decouples the suspended object from the quadrotor to a single point of contact. This decoupling behavior comes at the disadvantage of increasing the degrees of under actuation in the system. If addressed correctly, some of the disadvantages can be ignored or overcome, making it feasible to exploit the advantages of cable-suspended manipulation. The advantages of using suspended cables (illustrated in Figure 1.2) with grippers for collaborative manipulation are discussed below.

- Decouples the dynamics between the suspended gripper/payload and the quadrotor; this makes the quadrotor more agile and responsive from external disturbances, i.e., the quadrotor has rotational freedom while the cable-suspended gripper performs the necessary manipulation task. The added distance between the gripper and quadrotor makes it possible to approach sensitive locations to grasp the object while reducing the impact of the quadrotor's downwash on the surrounding area.
- Cable suspended payload makes it possible for collaborative transportation. A swarm of quadrotors with suspended grippers can be used to approach, grasp and transport larger payloads. The decoupled nature also ensures that the interactions between the quadrotors are only via the cables, i.e., as forces on the quadrotors, keeping the quadrotors' rotation free to react as needed.
- Transporting the payload by attaching them to the quadrotor's frame limits the volumetric dimensions of the payload. The payload dimensions should always be less than distance between the propellers; if not, the displacement of the air column by the propellers is restricted, making lift difficult. Adding distance between the quadrotor and payload makes it possible for lift. Please note while the lift is possible, the payload surface still adds drag to the system.

This dissertation focuses on aerial manipulation using cable-suspended payloads and grippers. Exploring the collaborative behavior to grasp and transport using multiple quadrotors

first requires a working individual quadrotor with a cable-suspended gripper. Thus, a significant component of the dissertation discusses control, estimation, and planning for individual agents. Finally, collaborative methods such as collaborative grasping, safe-critical control are discussed. The following section presents the challenges associated with the multi-agent system, quadrotors, and quadrotors with suspended cables, and the methods to address them are discussed.

1.2 Challenges & Contributions

Challenges

A cable-suspended collaborative agent system is a nonlinear and hybrid system. Due to the suspended nature of cables, i.e., no actuation, these systems are usually highly underactuated. The system dynamics is hybrid in nature and switches based on the slackness of the cable. The whole system can act as a single dynamics model. However, the inputs to the system are from the individual quadrotors; thus, the control schemes have to be carefully chosen to ensure no unwanted behaviors are introduced into the system. For instance, quadrotors could pull on one another to control the payload, leading to undesired results. Due to the proximity of multiple quadrotors carrying a shared payload, the downwash from the propellers adds disturbance to each other. This is a safety-critical system, and precautions to ensure no internal collisions happen with the system have to be taken.

Contributions

The contribution of this dissertation are as follows:

- Adaptive control to estimate and reject disturbances. Implements an L_1 adaptation scheme on geometric attitude control of a quadrotor directly on the tangent bundle of the $SO(3)$ manifold. Also proves that the proposed control schemes are input-to-stable.
- Estimation on S^2 manifold: Formulates an extended Kalman filter for cable attitude estimation; to preserve the manifold structure of the estimated quantities.
- Fast collision-free offline trajectory generation for a quadrotor with suspended payload: Formulates a nonlinear optimization problem, using direct collocation on the flat variables for trajectory generation of a quadrotor with a suspended payload.
- Collaborative grasping and transportation using cable-suspended grippers: Control schemes and framework for grasping using suspended grippers and its extension to multiple agents for collaborative grasping are discussed. Centralized controllers are presented for transporting the payloads.

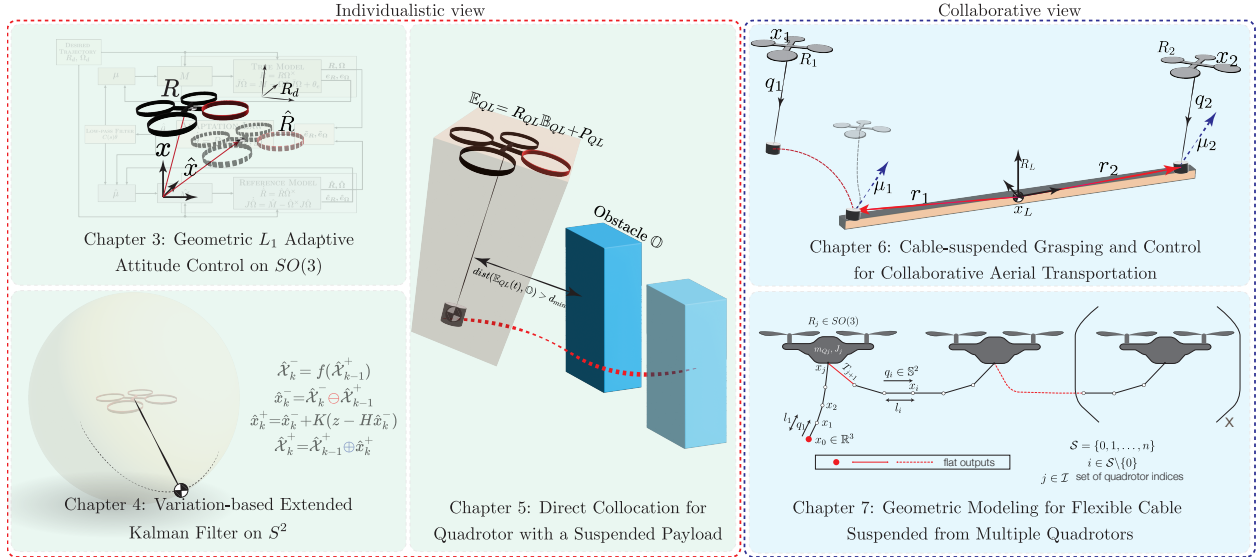


Figure 1.3: An outline of the various chapters in the dissertation.

- Modeling and dynamics of multiple quadrotors with a shared flexible payload: The dissertation presents coordinate-free full nonlinear dynamics for multiple quadrotors carrying a flexible cable and its variation-linearized dynamics. It also shows that the system is differentially flat.

1.3 Dissertation Outline

The rest of the dissertation's chapters are briefly discussed in this section. The overarching individualistic and collaborative views of the dissertation and their corresponding chapters are presented in Figure 1.3. The chapters 3, 4, 5, discussing control, estimation and planning methodologies related to single quadrotor (individualistic) and chapters 6 and 7 present the collaborative view.

Chapter 2: Mathematical Background

This chapter briefly presents a brief mathematical background of differentiable manifolds, namely S^2 and $SO(3)$, differential flatness, and variation-based linearization. It also introduces the dynamics of the quadrotor with cable-suspended payload and its hybrid model. These models are used in the later chapters for estimation, planning, and control.

Chapter 3: Geometric L_1 Adaptive Attitude Control on $SO(3)$

Control implementation for quadrotor systems, more specifically a multi-quadrotor payload system, typically consisting of cascaded control loops, with the innermost being the attitude

controller of the quadrotor. This chapter addresses the problem of disturbances and model uncertainties for quadrotors using geometric control techniques. A model reference adaptive control (MRAC) approach is used with the attitude tracking errors between the robot and the reference defined on the manifold. Controllers for the robot and the reference model, with L_1 adaptation to estimate the uncertainty in the robot model, are defined. Finally, the chapter presents various numerical simulations and experiments results validating the the proposed control scheme.

Chapter 4: Variation-based Extended Kalman Filter on S^2

Control of quadrotor with suspended cable requires the best estimation of the cable attitude to accurately track the payload position. This chapter presents an extended Kalman filter (EKF) for estimating the cable attitude on S^2 . This is achieved by mapping the state from the manifold to an Euclidean space using variation on manifold concept. Kalman filter measurement update is performed on this Euclidean space and is mapping back onto the manifold. The proposed Variation-based EKF is implemented on a spherical pendulum and validated using numerical simulations and experiments.

Chapter 5: Direct Collocation for Quadrotor with a Suspended Payload

Transportation using quadrotor with suspended payload in complex cluttered environments requires generating obstacle free trajectories. This chapters presents direct collocation based trajectory optimization for quadrotor suspended payload system. Differential flat variables are used for planning the trajectories by formulating a nonlinear optimization problem. This chapter presents various numerical simulations and experimental demonstrations of the generated obstacle free trajectories.

Chapter 6: Cable-suspended Grasping and Control for Collaborative Aerial Transportation

A framework for grasping and transporting target object using a cable suspended electromagnet is discussed in this chapter. The framework is extended to using multiple quadrotors with suspended grippers to collaboratively grasp and transport a common payload is presented. Two centralized control approaches are discussed *(i)* a payload control scheme to track desired trajectories for payload is presented, *(ii)* a control barrier function based collision free centralized control for slow stationary trajectories. The proposed control and grasping methods are verified in experiments.

Chapter 7: Geometric Modeling for Flexible Cable Suspended from Multiple Quadrotors

In this chapter, a full nonlinear geometric dynamics model for multiple quadrotors carrying a flexible cable is derived along with its linearized dynamics. This system, under certain

conditions, is shown to be differentially flat. The linearized equations are then used to compute the LQR gains, and a feedback controller is implemented to track trajectories generated using the flat variables. The final part of the chapter motivates the need for collaborative aerial systems with shared cable payload through experimental demonstrations, using two quadrotors externally tethered using a single power cable, for grasping.

Chapter 8: Conclusion

The final chapter presents the concluding remarks on cable-suspended collaborative aerial manipulation and the proposed methods used to achieve it. The later part of the chapter discusses some of the shortcomings and potential future research directions.

1.4 Summary

This chapter presents an overview of collaborative aerial manipulation and the advantages of cable suspended manipulation. The associated challenges using suspended cables and the contributions of dissertation are discussed. A brief description for the rest of the chapters is presented. In the next chapter, mathematical preliminaries and dynamics for quadrotor with suspended payload are presented.

Chapter 2

Mathematical Background

This chapter briefly presents the mathematical preliminaries used in this dissertation, such as the two common manifolds used in defining the dynamics of various quadrotor systems, differential flatness, and dynamics for quadrotor and quadrotor with suspended payload.

2.1 Notation

The common notations used in this work are given below, unless explicitly mentioned in each chapter, the follow conventions are applicable through out the dissertation.

- I_n represents a $n \times n$ Identity matrix
- O_n for $n \times n$ Zero matrix and 0_{mn} represents of a Zero matrix of size $m \times n$.
- e_1, e_2, e_3 represent the standard orthonormal basis vectors of \mathbb{R}^3 , i.e.,

$$e_1 = [1 \ 0 \ 0]^T, \ e_2 = [0 \ 1 \ 0]^T, \ e_3 = [0 \ 0 \ 1]^T$$

- The inertial frame has vertical axis pointing up, i.e., gravity vector is along $-e_3$.
- g is the acceleration due to gravity.
- The cross-map/hat-map is defined as $(\cdot)^\times : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ s.t $x^\times y = x \times y, \forall x, y \in \mathbb{R}^3$
- The vee-map, $\cdot^\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$ is defined as the inverse of the cross-map, i.e., $(y^\times)^\vee = y$
- Few useful properties of hat-map and vee-map. For any $x \in \mathbb{R}^3$, $A \in \mathbb{R}^{3 \times 3}$ and $R \in SO(3)$,

$$tr[x^\times A] = tr[Ax^\times] = \frac{1}{2}tr[x^\times (A - A^T)] = -x^T(A - A^T)^\vee, \quad (2.1)$$

$$Rx^\times R^T = (Rx)^\times, \quad (2.2)$$

$$x^\times A + A^T x^\times = (\{tr[A]I - A\}x)^\times. \quad (2.3)$$

- **Agent** can refer to just a quadrotor (like in Chapter 7) or a quadrotor with suspended gripper (Chapter 6) as a single entity depending on the context.
- **DOF**- Degrees of freedom, **DOA**- Degrees of Actuation and **DOuA**- Degrees of under actuation.

2.2 Differentiable Manifolds

Robotic systems often evolve on nonlinear manifolds, and it is not always ideal to model them using just Euclidean space, \mathbb{R}^n . Manifolds and, more frequently, *differentiable manifolds* are typically used to represent spaces in the field of robotics. This section briefly describes manifolds and the two main manifolds used in this work. Please refer to the many great references on differential geometry, and lie theory for more details on this; for instance, see [14–19].

Definition 2.1. Differentiable Manifold, M , [17], is an embedding in \mathbb{R}^n , such that,

$$M = \{x \in \mathbb{R}^n \mid f_i(x) = 0, i = 1, 2, \dots, l\} \quad (2.4)$$

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^1, i = 1, \dots, l$ are scalar differentiable functions with $\frac{\partial f_i(x)}{\partial x}, i = 1, \dots, l$ are linearly independent vectors in \mathbb{R}^n for each $x \in M$.

For a differentiable curve $\gamma : [-1, 1] \rightarrow M$ on M with $\gamma(0) = x \in M$, $\left. \frac{d\gamma(s)}{ds} \right|_{s=0}$ is a tangent vector to M at $x \in M$. For each $x \in M$, set of all tangent vectors to M is referred to as *Tangent Space*, $T_x M \subset \mathbb{R}^n$. Let $\xi \in T_x M$ be a tangent to M at x , then the tangent space,

$$T_x M = \{\xi \in \mathbb{R}^n \mid \left(\frac{\partial f_i(x)}{\partial x} \cdot \xi \right) = 0, i = 1, 2, \dots, m\}.$$

2D Unit Sphere: S^2

A two-dimensional unit sphere, represented as S^2 is a differential manifold embedding in \mathbb{R}^3 , defined as,

$$S^2 := \{q \in \mathbb{R}^3 \mid \|q\| = 1\}. \quad (2.5)$$

For any $q \in S^2$, the tangent space to S^2 at q is,

$$T_q S^2 := \{\xi \in \mathbb{R}^3 \mid q \cdot \xi = 0\}, \quad (2.6)$$

for unit-sphere, $T_q S^2$, is the plane tangent to the sphere. For a time-varying $q(t) \in S^2$ and its time-derivative $\dot{q} \in T_q S^2$. The angular velocity, $\omega \in \mathbb{R}^3$ is defined as,

$$\omega = q \times \dot{q}. \quad (2.7)$$

For two elements $q, q_d \in S^2$, the distance metric between the two elements can be represented using the configuration error,

$$\Psi_q = 1 - q \cdot q_d \quad (2.8)$$

The error between q, q_d and the corresponding angular velocities ω, ω_d can be defined as,

$$e_q = q_d \times q, \quad (2.9)$$

$$e_\omega = \omega + q \times (q \times \omega_d), \quad (2.10)$$

or alternately,

$$\tilde{e}_q = q \times q \times q_d \quad (2.11)$$

$$\tilde{e}_{\dot{q}} = \dot{q} - (q_d \times \dot{q}_d) \times q. \quad (2.12)$$

Note, $e_q = q \times \tilde{e}_q$ and $e_\omega \times q \times \tilde{e}_{\dot{q}}$. The S^2 manifold can be used to represent orientations of links or cables, for systems, where rotation about its own axes are insignificant.

Remark 2.1. *The choice of the error (e_q, e_ω) and $(\tilde{e}_q, \tilde{e}_{\dot{q}})$ can be chosen based on the system dynamics, especially the input type. (e_q, e_ω) error vectors are suitable for torque inputs, and $(\tilde{e}_q, \tilde{e}_{\dot{q}})$ for force inputs.*

Special Orthogonal Group: $SO(3)$

A special orthogonal group $SO(3)$ in an embedding in $\mathbb{R}^{3 \times 3}$, defined as,

$$SO(3) := \{R \in \mathbb{R}^{3 \times 3} \mid R^T R = I, \det(R) = +1\}, \quad (2.13)$$

the tangent vector for $R \in SO(3)$ in the tangent space be $\Omega \in T_R SO(3)$. $SO(3)$ is used to represent the space of rotations, and in robotics is used to represent the orientation of rigid-bodies, and the tangent vector Ω is the angular-velocity of the body in the body-frame R . The kinematic equation for the rotation matrix is,

$$\dot{R} = R\Omega^\times. \quad (2.14)$$

Configuration error the $SO(3)$ between two elements $R, R_d \in SO(3)$ is given as,

$$\Psi_R = \frac{1}{2} \text{tr}(I - R_d R). \quad (2.15)$$

The error vectors for R, R_d and their corresponding angular velocities Ω, Ω_d is given as,

$$e_R = \frac{1}{2}(R_d^T R - R_d^T R)^\vee, \quad (2.16)$$

$$e_\Omega = \Omega - R^T R_d \Omega_d. \quad (2.17)$$

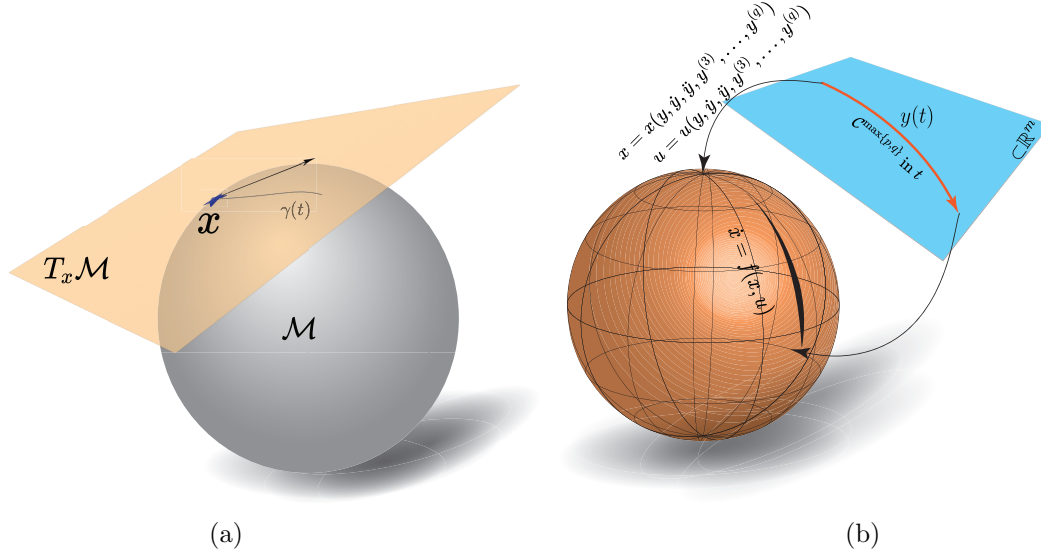


Figure 2.1: An illustration of the mathematical concepts, (a) A manifold showing the tangent vector and the tangent space. (b) Differential flatness mapping the nonlinear system to a flat space.

Tangent space at the identity element in the manifold group is called Lie-algebra. Any element M on the manifold can be mapped to the tangent space at identity. The Lie algebra for $SO(3)$ is the space of skew-symmetric matrices of $\mathbb{R}^{3 \times 3}$.

$$\forall x \in \mathbb{R}^3, \exp(x^\times) \in SO(3). \quad (2.18)$$

Finally, while S^2 is not a group, for an element $q \in S^2$ transformation using a rotation matrix $R \in SO(3)$ satisfy the closure properties, i.e., $Rq \in S^2$.

2.3 Variation-based Linearization

Linearization of nonlinear systems is a frequent occurrence in the field of robotics and/or control-theory. It is not always feasible to work directly on nonlinear systems and often required to linearize the system dynamics about a reference trajectory to get a first order approximation of the system behaviour within a neighborhood. This is often seen, for instance, in designing LQR control or developing an extended Kalman filter.

Systems evolving on nonlinear manifolds, too, require linearization about a reference trajectory. However, using standard Jacobian linearization, the resulting dynamics do not always adhere to the manifold constraints. Using variational vector fields [20], a trajectory on a manifold is perturbed such that the perturbed trajectory is also on the manifold. Thus, to linearize along a reference trajectory on a manifold, variations are taken with respect to the reference trajectory. Here the variation is referred to as an infinitesimal variation which could be roughly treated as a linear approximation of the distance between two points on a manifold [21, Chapters 5-7]. Variations for S^2 and $SO(3)$ manifolds are given below.

Variation on S^2

For a given trajectory $q(t)$ on S^2 , variation of the state q is such that the perturbed trajectory is also on S^2 . This is achieved by rotating the vector q , so that the unit-length of the vector q is preserved. Variation on S^2 is expressed in terms of a rotation matrix (represented as an exponential map) in [22] as follows,

$$q^\epsilon(t) = \exp[\epsilon \xi^\times] q(t), \quad (2.19)$$

for a curve $\xi(t) \in \mathbb{R}^3$ satisfying $\xi(t) \cdot q(t) = 0, \forall t$. The cross-map $(\cdot)^\times : \mathbb{R}^3 \rightarrow so(3)$ is defined such that, $u^\times v = u \times v$, for any $u, v \in \mathbb{R}^3$. The corresponding infinitesimal variation for S^2 is given as,

$$\delta q(t) = \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \exp[\epsilon \xi^\times] q(t) = \xi^\times q(t). \quad (2.20)$$

The infinitesimal variation could be roughly treated as a linear approximation of the distance between two points on S^2 . Infinitesimal variation of the angular velocity [22] is denoted by $\delta\omega(t)$ in \mathbb{R}^3 satisfying $q(t) \cdot \omega(t) = 0$. For future use, we define the *variation state* as,

$$x = \begin{bmatrix} \xi \\ \delta\omega \end{bmatrix} \in \mathbb{R}^6. \quad (2.21)$$

The infinitesimal variation ξ in (2.19) is always orthogonal to q . Taking the time-derivative of the constraint results in,

$$\xi \cdot q \equiv 0 \implies \dot{\xi} q + \xi \dot{q} = 0. \quad (2.22)$$

Moreover, the angular velocity ω in (2.8) is always orthogonal to q . Taking the variation of the constraint results in,

$$q \cdot \omega \equiv 0 \implies (\xi^\times q) \cdot \omega + q \cdot \delta\omega = 0. \quad (2.23)$$

Thus any variation of $q \in S^2$ and its angular velocity in $T_q S^2$ has to satisfy the constraints (2.22) and (2.23),

$$\begin{bmatrix} q^T & 0_{1 \times 3} \\ -\omega^T q^\times & q^T \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.24)$$

Variation on $SO(3)$

Infinitesimal variation of $R \in SO(3)$ with respect to a reference $R_d \in SO(3)$ (from [23]),

$$\delta R = \left. \frac{d}{ds} R_d \exp(s \eta^\times) \right|_{s=0} = R_d \eta^\times, \quad (2.25)$$

where $\eta \in \mathbb{R}^3$. Similarly, for the angular velocity the infinitesimal variation can be computed by comparing the time-derivative of the rotational variation in (2.25) and variation on the rotational kinematics in (2.14). The infinitesimal variation of the angular velocity is,

$$\delta\Omega = \Omega_d^\times \eta + \dot{\eta}. \quad (2.26)$$

The variation-based linearization of control affine systems is computed by taking a first-order approximation of the dynamics. For more details on variation-based linearization, see [24]. For a system evolving on $X := X_1 \times X_2 \times \dots \times X_n$, let the system equation be $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$, where the state \mathbf{x} encompasses all the various states belong to X_i and X_i is either an \mathbb{R}^n or S^2 or $SO(3)$. Thus the variation-based linearized dynamics are computed as,

$$\delta(\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}), \quad (2.27)$$

where $\mathbf{x} \in X, \mathbf{u} \in \mathbb{R}^m$ and $\delta(\cdot)$ refers to taking the variation. (2.27) is recursively solved using the following rules,

$$\delta(x + y) = \delta(x) + \delta(y), \quad (2.28)$$

$$\delta(x \cdot y) = \delta(x) \cdot y + x \cdot \delta(y), \quad (2.29)$$

where $x, y \in$ any combinations of $\mathbb{R}^n, S^2, SO(3)$. Finally, using (2.19), (2.25) and matrix manipulations, the variation-based linearization of (2.27) has the following form,

$$(\delta\dot{\mathbf{x}}) = \mathcal{A}(\mathbf{x}, \mathbf{u})\delta\mathbf{x} + \mathcal{B}(\mathbf{x}, \mathbf{u})\delta\mathbf{u}, \quad (2.30)$$

where $(\delta\dot{\mathbf{x}}), \delta\mathbf{x} \in \mathbb{R}^{3n}, \delta\mathbf{u} \in \mathbb{R}^m, \mathcal{A} \in \mathbb{R}^{3n \times 3n}$ and $\mathcal{B} \in \mathbb{R}^{3n \times m}$.

Remark 2.2. *The variation $\delta(\cdot)$ of the dynamics can be taken about the current system state or a reference trajectory, i.e., \mathbf{x}, \mathbf{u} in (2.30) is current system state & input or reference state & input depending on the context. When computing LQR gain, the system is variation-based linearized about reference trajectory, while in the extended Kalman filter, the dynamics are variation-based linearized about the current state for the covariance propagation step.*

2.4 Differential Flatness

A nonlinear system is differentially-flat if a set of outputs of the system (equal to the number of inputs) and their derivatives can be used to determine the states and inputs without integration, see Figure 2.1.

Definition 2.2. *Differentially-Flat System, [25]: A system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, $\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m$, is differentially flat if there exists flat outputs $\mathbf{y} \in \mathbb{R}^m$ of the form $\mathbf{y} = \mathbf{y}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^{(p)})$ such that the states and the inputs can be expressed as $\mathbf{x} = \mathbf{x}(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(q)})$, $\mathbf{u} = \mathbf{u}(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(q)})$, where p, q are positive finite integers.*

Differential flatness in quadrotor systems

A quadrotor is a differentially-flat system with the quadrotor center-of-mass, x_Q , and yaw ψ_Q as the flat outputs [26]. A quadrotor with cable suspended load (with the cable modeled as a

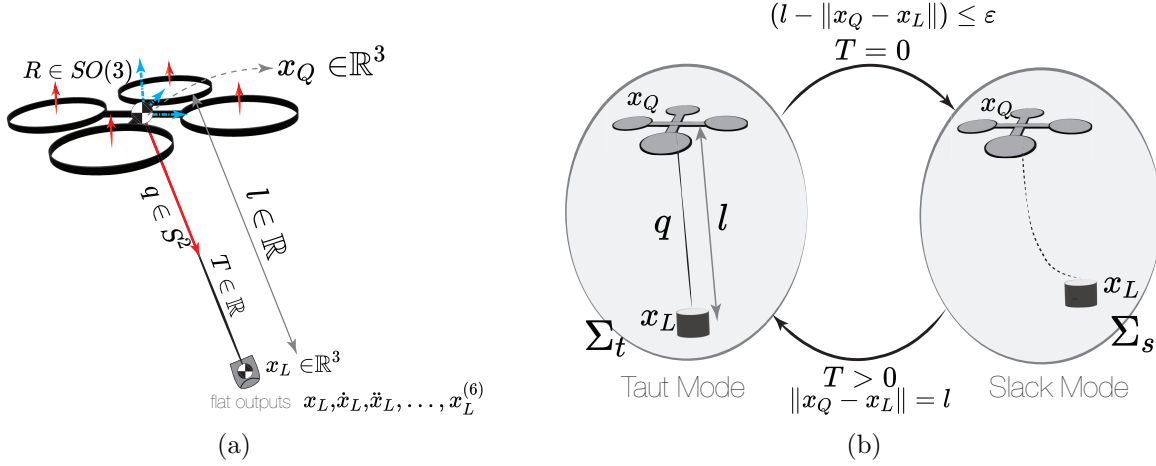


Figure 2.2: Schematic of a quadrotor with suspended payload, illustrating (a) the various system states, (b) and the two hybrid modes, slack and taut.

massless link) is also shown to be differentially flat with load position x_L , and quadrotor yaw ψ_Q as the flat outputs [27]. Similarly, a quadrotor with flexible cable suspended load, with the cable modeled as a series of smaller links, is shown to be differentially-flat [28]. Again, here load position and quadrotor yaw are the flat outputs. Multiple quadrotors carrying a rigid payload is differential flat system [29], as well.

2.5 Quadrotor with Suspended Payload

A collaborative aerial manipulation system consists of two/more quadrotors carrying a common payload. This requires each system to be able to carry or grasp the object successfully. In this section, dynamics for the single agent, a quadrotor with suspended payload is reproduced.

Remark 2.3. *For consistency, the object on the other end of the suspended cable from the quadrotor is always referred to as payload, irrespective of whether it is a payload or gripper/electromagnet used for grasping.*

As with the dynamics in [27, 30], a coordinate-free geometric representation is used as states of a quadrotor with suspended payload. Quadrotor is modeled as a rigid body and its attitude is represented using a rotation matrix $R \in SO(3)$, cable attitude with a Unit Sphere $q \in S^2$, the payload is modeled as a point mass with its position $x_L \in \mathbb{R}^3$ and the quadrotor position $x_Q \in \mathbb{R}^3$, where the relation between x_L & x_Q is,

$$x_L = x_Q + lq, \quad (2.31)$$

with l as the length of the cable. The cable is massless and is assumed to be attached to the center-of-mass of the quadrotors as shown in Figure 2.2. For most applications, this is a

Table 2.1: List of commonly used variables for the quadrotor-payload model.

$m_Q \in \mathbb{R}$	Mass of the quadrotor
$J \in \mathbb{R}^{3 \times 3}$	Inertia matrix of the quadrotor with respect to the body-fixed frame
$R \in SO(3)$	Rotation of the quadrotor from body-fixed frame to the inertial frame
$\Omega \in \mathbb{R}^3$	Angular velocity of the quadrotor wrt inertial frame in the body-frame
$x_Q \in \mathbb{R}^3$	Position vector of the quadrotor's center-of-mass in the inertial frame
$v_Q \in \mathbb{R}^3$	Velocity vector of the quadrotor's center-of-mass in the inertial frame
$f \in \mathbb{R}$	Magnitude of the thrust of the quadrotor and is in the direction of $R\mathbf{e}_3$
$M \in \mathbb{R}^3$	Moment vector of the quadrotor in the body-fixed frame
$q \in S^2$	Cable attitude showing the unit vector from x_Q to x_L in the inertial frame
$\omega \in T_q S^2$	Angular velocity of the cable attitude in the inertial-frame.
m_L	Mass of the payload in SI units
l	Cable length
x_L	Payload position in inertial-frame
v_L	Payload velocity in inertial-frame

reasonable approximation. When the cable is attached to a point, offset from the center-of-mass, the cable adds additional torque to the attitude dynamics, this is discussed in detail in [31].

The cable slackness in the model defines two different dynamics modes for the quadrotor-payload, slack mode Σ_s or taut mode Σ_t . Let the state of the system in taut mode be X_t ,

$$X_t = \{x_L, v_L, q, \omega, R, \Omega\}, \quad (2.32)$$

and slack mode be X_s ,

$$X_s = \{x_L, v_L, x_Q, v_Q, R, \Omega\}. \quad (2.33)$$

Dynamics of the system for taut mode,

$$\Sigma_t : \begin{cases} \dot{x}_L = v_L \\ (\dot{v}_L + g\mathbf{e}_3) = \frac{(q^T f R\mathbf{e}_3 - m_Q l \dot{q}^T \dot{q})}{(m_Q + m_L)} q \\ \dot{q} = \omega \times q \\ m_Q l \dot{\omega} = -q \times f R\mathbf{e}_3 \\ \dot{R} = R\hat{\Omega} \\ J\dot{\Omega} = M - \Omega \times J\Omega \end{cases} \quad (2.34)$$

where the various variables are as listed in Table 2.1. The quadrotor-payload system in slack mode has two different subsystems consisting of the quadrotor system and the payload

system. Dynamics for the slack mode as defined as,

$$\Sigma_s : \left\{ \begin{array}{l} \dot{x}_Q = v_Q \\ \dot{v}_Q = -m_Q g \mathbf{e}_3 + f R \mathbf{e}_3 \\ \dot{R} = R \hat{\Omega} \\ J_Q \dot{\Omega} = M - \Omega \times J_Q \Omega \end{array} \right\} \Sigma_s^Q \quad (2.35)$$

$$\left\{ \begin{array}{l} \dot{x}_L = v_L \\ \dot{v}_L = -m_L g \mathbf{e}_3 + F_{ext} \end{array} \right\} \Sigma_s^L$$

the quadrotor subsystem represented using Σ_s^Q and payload system Σ_s^L is a free-falling projectile. Variation based linearization for the quadrotor payload dynamics is shown in [24]. The quadrotor dynamics on $SE(3)$ are given as,

$$\Sigma^Q : \left\{ \begin{array}{l} \dot{x}_Q = v_Q, \\ \dot{v}_Q = -m_Q g \mathbf{e}_3 + f R \mathbf{e}_3, \\ \dot{R} = R \hat{\Omega}, \\ J_Q \dot{\Omega} = M - \Omega \times J_Q \Omega. \end{array} \right. \quad (2.36)$$

An almost global exponentially stable control for the quadrotor is defined in [32]. Typical position control architecture for quadrotors involves cascading approach, consisting of a position control run at a lower loop rate either offboard/onboard and an inner attitude loop run onboard the quadrotor. The inner attitude loop can again be comprised of an inner angular velocity tracking loop. The payload position control can be implemented as a combination of offboard/onboard or fully onboard. For payload position and attitude control, a slow dynamics model can be considered, used in [27], and the dynamics model for the same is given below as,

$$\Sigma_t^L : \left\{ \begin{array}{l} \dot{x}_L = v_L \\ (\dot{v}_L + g \mathbf{e}_3) = \frac{(q^T F - m_Q l \dot{q}^T \dot{q})}{(m_Q + m_L)} q \\ \dot{q} = \omega \times q \\ m_Q l \dot{\omega} = -q \times F \end{array} \right. \quad (2.37)$$

where, the F is the command input to the attitude control.

2.6 Summary

In this chapter, differential manifolds, in particular, S^2 and $SO(3)$ are introduced along with their variation linearization. The differential flatness concept is discussed, which will be used in the later chapters for planning. Finally, dynamics for quadrotors with cable-suspended payload and their hybrid modes are discussed.

Chapter 3

Geometric L_1 Adaptive Attitude Control on $SO(3)$

3.1 Introduction

Due to their small size and simple mechanical structure, quadrotors have a large range of potential applications including visual inspection and transportation, as well as a medium for testing control techniques for research purposes. Generally, an attitude controller for a quadrotor uses Euler angles or quaternions as the attitude states of the system. In geometric control, the entries of the rotation matrix between the body-fixed frame and inertial frame are used as the attitude states. Geometric control can be used for complex flight maneuvers as seen in [33] and it completely avoids singularities and complexities that arise when using local coordinates. Geometric control has also been used in robust tracking [33] and carrying suspended loads with cables [34–36].

However, these controllers are model dependent and assume accurate representation of the dynamics. In the presence of disturbances or uncertainties, the controllers would result in unstable control or large tracking errors. Lately, there has been much work done with adaptive control, to achieve higher performance and robustness in the presence of model uncertainties and disturbances.

Related Work

Model Reference Adaptive Control (MRAC) is one of the standard approach to adaptive control, where an ideal reference system is simulated, and the differences between the real and reference systems are used to predict and cancel the disturbances in the system [37, 38]. However, this approach has some important practical drawbacks. If the adaptation rate is pushed too high, the system is given high-frequency input commands, which may not be feasible for the system. Also, high adaptation gain might result in deteriorated closed-loop stability or unpredictable transient behaviours [39, 40]. Furthermore, the high-frequency commands may excite the system, which is not desired. Because of this, the adaptation rate

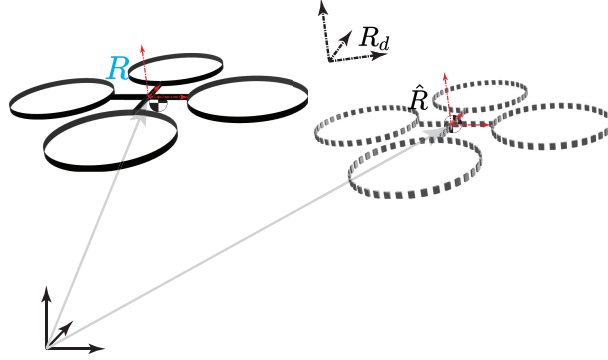


Figure 3.1: The Quadrotor model evolves on $SE(3)$, with the attitude represented by a rotation matrix $R \in SO(3)$, from body-frame to inertial-frame. A reference quadrotor evolving on $SE(3)$ with attitude states $\hat{R}, \hat{\Omega}$.

will be low, leading to a long prediction in the system uncertainties. L_1 adaptive control is an extension of MRAC, where the adaptation and control are decoupled through the use of a low pass filter. The designer can choose the cutoff frequency for the low pass filter, removing the high-frequency components of the adaptation from the control input. This promotes a very fast adaptation rate while keeping the control input sufficiently smooth, making L_1 a much more practical adaptive control technique [41, 42].

L_1 attitude controllers have been developed for attitude control of quadrotors, using Euler angles (or quaternions) as attitude states [39, 43–45] or for general linear systems [46]. Geometric adaptive schemes have been previously developed for quadrotor control in [47, 48]. Similar to [47], this work also develops the attitude dynamics and control laws for the quadrotor directly on the $SO(3)$ manifold without any local simplifications. However, in contrast to [47], we implement a model reference approach, consider time-varying disturbances and uncertainties, estimate uncertainties using a L_1 adaptation scheme, and show exponential input-to-state stability.

Challenges

Challenges when implementing the L_1 adaptation scheme to a geometric setting is to ensure correct formulation for the errors between the *reference model* (or state predictor) and *true model* (or plant). A simple difference between the states of the state predictor and the plant cannot be used[49], because these states belong to different tangent spaces and thus their difference often lead to incorrect measure of the error, especially for large errors. Similarly, care should be taken when defining the uncertainty prediction error. The adaptation law for the uncertainty requires a careful choice of prediction law that ensures the Lyapunov function for the prediction errors decreases with time.

Background

Geometric dynamics model

The dynamics of a quadrotor in a coordinate-free framework in Section 2.5. Figure 3.1 illustrates the quadrotor with reference frames. Equations of motion for the quadrotor discussed in [33] are given below,

$$\dot{x}_Q = v_Q, \quad (3.1)$$

$$m\dot{v}_Q = mg\mathbf{e}_3 - fR\mathbf{e}_3, \quad (3.2)$$

$$\dot{R} = R\Omega^\times, \quad (3.3)$$

$$J\dot{\Omega} = M - (\Omega \times J\Omega), \quad (3.4)$$

where Table 2.1 enumerates various symbols used in defining the dynamics.

Geometric Tracking Control for Quadrotor

The geometric tracking control presented in [33] tracks the desired quadrotor trajectory $x_d(t)$. The position controller calculates thrust f and the desired orientation R_d , while the attitude control calculates moment M to track the desired orientation R_d . The tracking errors for the attitude dynamics, with desired orientation R_d and body angular velocity Ω_d , are defined in (2.16), (2.17), respectively. The configuration error Ψ function between R and R_d is in (2.15). Note that, Ψ is almost globally positive definite and upper bounded by 2. The attitude error dynamics are then given as, see [33] for details,

$$\dot{e}_R = C(R_d^T R)e_\Omega, \quad (3.5)$$

$$J\dot{e}_\Omega = J\dot{\Omega} + J(\Omega^\times R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d), \quad (3.6)$$

$$J\dot{e}_\Omega = M - (\Omega \times J\Omega) + J(\Omega^\times R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d). \quad (3.7)$$

where $C(R_d^T R) := \frac{1}{2}(\text{Trace}[R^T R_d]I - R^T R_d)$. It can be easily noticed that choosing the control moment M as,

$$M = \mu + (\Omega \times J\Omega) - J(\Omega^\times R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d), \quad (3.8)$$

cancels the nonlinearities in (3.8). Then a choice of μ as,

$$\mu = -k_R e_R - k_\Omega e_\Omega, \quad (3.9)$$

for any positive constants, k_R, k_Ω , would result in $J\dot{e}_\Omega = -k_R e_R - k_\Omega e_\Omega$. For initial conditions that satisfy,

$$\Psi(R(0), R_d(0)) < 2, \quad (3.10)$$

$$\|e_\Omega(0)\|^2 < \frac{2}{\lambda_m(J)} k_R (2 - \Psi(R(0), R_d(0))), \quad (3.11)$$

(where λ_m is the minimum eigenvalue of the matrix and similarly, λ_M is the maximum eigenvalue), it is shown in [33] that the zero equilibrium of the attitude tracking errors (e_R, e_Ω) is exponentially stable. Moreover, a control Lyapunov candidate,

$$V = \frac{1}{2}e_\Omega \cdot J e_\Omega + k_R \Psi(R, R_d) + c_2 e_R \cdot e_\Omega, \quad (3.12)$$

is defined to show that the above error dynamics is exponentially stable for i.e., $\dot{V} \leq -\eta^T W \eta$, where $\eta = [e_R \ e_\Omega]^T$ and W is a positive definite matrix. A detailed proof can be found in [33]. Having discussed the geometric attitude control, the following section describes the attitude dynamics of quadrotor in the presence of disturbances and model uncertainties.

3.2 Effects of Model Uncertainties and Disturbances on the Attitude Dynamics

As discussed in the previous section, the control moment M in (3.8) ensures that the zero equilibrium of the error dynamics in (3.5)-(3.7) is exponentially stable. This controller assumes an accurate dynamical model of the quadrotor. However, presence of any uncertainties in the model properties, like in mass, m , and in inertia, J , can result in large tracking errors. External disturbances on the system can also result in similar adverse effects.

The unknown external disturbances can be captured in the attitude dynamics of the quadrotor (3.3)-(3.4). In particular, the attitude dynamics along with the external disturbances are defined as,

$$\dot{R} = R\Omega^\times, \quad (3.13)$$

$$\mathcal{J}\dot{\Omega} = M - \Omega \times \mathcal{J}\Omega + \theta_e, \quad (3.14)$$

where θ_e represents the unknown external disturbance and \mathcal{J} is the true (unknown) inertia of the quadrotor. The corresponding attitude error dynamics (3.5), (3.7), whose errors are defined as (2.16) and (2.17), can be modified and represented as below,

$$\dot{e}_R = C(R_d^T R) e_\Omega, \quad (3.15)$$

$$\begin{aligned} \mathcal{J}\dot{e}_\Omega &= M - (\Omega \times \mathcal{J}\Omega) \\ &+ \mathcal{J}(\Omega^\times R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d) + \theta_e. \end{aligned} \quad (3.16)$$

Note, in the above equation, the external disturbance θ_e is the matched uncertainty [50, Chapter 2].

Additionally, the control moment M in (3.8) assumes accurate knowledge of inertia of the quadrotor. However, if the true inertia, \mathcal{J} , of the quadrotor is not same as the nominal inertia of the quadrotor, J , it results in further uncertainties in the closed-loop system. In

particular, substituting for M from (3.8) in (3.16), we obtain,

$$\begin{aligned} \mathcal{J}\dot{e}_\Omega &= \mu + \theta_e \\ &+ \underbrace{[-\Omega \times (\mathcal{J} - J)\Omega + (\mathcal{J} - J)(\Omega^\times R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d)]}_{\triangleq \theta_m}, \end{aligned} \quad (3.17)$$

where, θ_m is zero when $J = \mathcal{J}$. We define δJ relating the true inertia \mathcal{J} and the nominal inertia J as, $\delta J := J\mathcal{J}^{-1} - I$. Thus, (3.17) can be written in the following manner,

$$J\dot{e}_\Omega = \mu + \underbrace{(\delta J)\mu + J\mathcal{J}^{-1}(\theta_m + \theta_e)}_{\triangleq \theta} \quad (3.18)$$

where θ is the combined expression for the uncertainties and disturbances. Therefore, the closed-loop attitude error dynamics of the quadrotor with control moment M defined in (3.8), along with the model uncertainties and disturbances can be given as,

$$\dot{e}_R = C(R_d^T R)e_\Omega, \quad (3.19a)$$

$$J\dot{e}_\Omega = \mu + \theta. \quad (3.19b)$$

For $\theta = 0$ in (3.19b), representing no model uncertainty, a choice of μ as a PD control similar to (3.9), would result in a stabilizing control. However, if $\theta \neq 0$, the choice of PD control alone will not be sufficient to guarantee stability. A choice of μ that can cancel the uncertainty θ would be helpful in achieving stability. A similar approach for biped robots is developed in [51], however, they do not address the case of dynamics evolving on manifolds. Also note that the uncertainty is a nonlinear function of states R, Ω and control input μ . In the next section, we propose a Geometric L_1 adaptation to predict the uncertainty θ .

3.3 Geometric L_1 Adaption for Attitude Tracking Control

In section 3.2, the attitude dynamics were described in the presence of model uncertainties. In particular, (3.19a) & (3.19b) present the attitude error dynamics along with the uncertainty θ and input μ . (Note that the actual control moment M is calculated using (3.8).) In this section, we proceed to present a geometric L_1 adaptation law to estimate the uncertainty θ , and compute the input μ , to track a desired time-varying trajectory (R_d, Ω_d) .

Remark 3.1. *The dynamical model given in (3.16) with the controller in (3.8) resulting in the closed-loop system in (3.19) is referred to as true model. States and inputs corresponding to the true model are given below,*

$$\boxed{R, \Omega, e_R, e_\Omega, \theta, M.}$$

Reference model

L_1 control architecture employs a *reference model* (also referred as reference/nominal system or state predictor) to predict the uncertainty θ in the system. In this chapter, we consider a *reference model* with a nominal inertia matrix, J , and without any disturbances θ_e . Let \hat{R} be the attitude of the *reference model* and $\hat{\Omega}$ be the body-angular velocity of the *reference model*. Dynamics of the *reference model* is written as,

$$\dot{\hat{R}} = \hat{R}\hat{\Omega}^\times, \quad (3.20)$$

$$J\dot{\hat{\Omega}} = \hat{M} - \hat{\Omega} \times J\hat{\Omega}, \quad (3.21)$$

with control moment \hat{M} defined similar to (3.8), i.e.,

$$\hat{M} = \hat{\mu} + (\hat{\Omega} \times J\hat{\Omega}) - J(\hat{\Omega}^\times \hat{R}^T R_d \Omega_d - \hat{R}^T R_d \dot{\Omega}_d). \quad (3.22)$$

Here, $R_d(t)$, $\Omega_d(t)$ are the same desired trajectory (attitude and body-angular velocity) considered in section 3.2. Similar to the attitude error vectors defined for the *true model*, we define configuration errors for the *reference model*. The attitude tracking error is defined as,

$$\hat{e}_R = \frac{1}{2}(R_d^T \hat{R} - \hat{R}^T R_d)^\vee, \quad (3.23)$$

and the angular velocity tracking error on $T_{\hat{R}}SO(3)$ is,

$$\hat{e}_\Omega = \hat{\Omega} - \hat{R}^T R_d \Omega_d. \quad (3.24)$$

The error dynamics for the *reference model* is similar to (3.5), (3.7). We present the attitude error dynamics with control moment defined in (3.22) below,

$$\dot{\hat{e}}_R = C(R_d^T \hat{R})\hat{e}_\Omega, \quad (3.25a)$$

$$J\dot{\hat{e}}_\Omega = \hat{\mu}. \quad (3.25b)$$

Comparing (3.25b) to (3.19b), we notice the presence of additional term θ (uncertainty) in the *true model*. L_1 adaptation is used to estimate this uncertainty, with $\hat{\theta}$ denoting the uncertainty. Discussion regarding the uncertainty estimation and its relevance to the control is presented below,

- (i) L_1 adaptation makes use of both the *true model* and the *reference model* to estimate the uncertainty $\hat{\theta}$.
- (ii) In-order for the *reference model* to account for the uncertainty in the *true model*, estimated uncertainty $\hat{\theta}$ (or a transported version of $\hat{\theta}$ in case of manifolds) is included in the control input $\hat{\mu}$.

Table 3.1: Notations used for the geometric L_1 adaptive control.

Symbol	Example	Model/Errors
no sub/super-script	R	<i>True model</i>
\cdot_d	R_d	Desired Trajectory
$\hat{\cdot}$	\hat{R}	<i>Reference model</i>
no sub/super-script	e_R	Error between <i>True model</i> & Desired Trajectory
$\hat{\cdot}$	\hat{e}_R	Error between <i>Reference model</i> & Desired Trajectory
$\tilde{\cdot}$	\tilde{e}_R	Error between <i>True model</i> & <i>Reference model</i>
	\mathcal{J}	True inertia
	J	Nominal inertia used by the control law

- (iii) Control inputs μ (*true model*) and $\hat{\mu}$ (*reference model*) are used to track a desired trajectory while canceling the uncertainty.
- (iv) Uncertainty is countered by including $(-\hat{\theta})$ in the control inputs μ and $\hat{\mu}$. However, $\hat{\theta}$ typically contains high frequency components due to fast estimation. L_1 adaptive control architecture is used to decouple estimation and adaption [46]. A low-pass filter is used to exclude the high frequency content in the input. Thus, $(-C(s)\hat{\theta})$ is included in the control inputs, where $C(s)$ is a low pass filter and with $\|C(0)\| = 1$. The low-pass filter is key to the trade-off between the performance and robustness.

Remark 3.2. The reference model is distinguished using the superscript $\hat{\cdot}$. Therefore, states and inputs corresponding to the reference model are as given below,

$$\boxed{\hat{R}, \hat{\Omega}, \hat{e}_R, \hat{e}_\Omega, \hat{\theta}, \hat{M}}.$$

Table 3.1 presents the different notations used in this chapter.

Errors between the *True model* and the *Reference model*

In the previous sections, we presented the *true model* and the corresponding *reference model* to mimic the actual system and its uncertainties. The goal of our work is to present an adaptive controller to reduce the differences between the *true model* and the *reference model*. We, also present a tracking controller for the *reference model*, that enables the *true model* to track the desired trajectory.

Attitude tracking errors between the *true model* and the desired time-varying trajectory are e_R and e_Ω . Errors between the *reference model* and the desired trajectory are \hat{e}_R and \hat{e}_Ω . Similarly, we define a new set of attitude errors to capture the difference between the

true model and the *reference model* as below,

$$\tilde{e}_R = \frac{1}{2}(R^T \hat{R} - \hat{R}^T R)^\vee, \quad (3.26)$$

$$\tilde{e}_\Omega = \hat{\Omega} - \hat{R}^T R \Omega, \quad (3.27)$$

and the corresponding configuration error function is given as,

$$\tilde{\Psi} = \frac{1}{2} \text{Trace}[I - R^T \hat{R}]. \quad (3.28)$$

From (3.19b) and (3.8), the uncertainty θ can be canceled out (if it was known) by the control moment. This shows that θ lies in the same dual space as the moment M , i.e., $\theta \in T_R^*SO(3)$, and similarly, the predicted uncertainty is in the dual space $T_{\hat{R}}^*SO(3)$. The difference between the actual uncertainty and the predicted uncertainty is thus calculated by transporting θ to the space of $\hat{\theta}$,

$$\tilde{\theta} = \hat{\theta} - \hat{R}^T R \theta. \quad (3.29)$$

Having presented the error definitions for orientation \tilde{e}_R , angular velocity \tilde{e}_Ω and uncertainty $\tilde{\theta}$ between the *true model* and the *reference model*, we now present the control design for μ and $\hat{\mu}_1$ for the attitude dynamics of *true model*, and *reference model* respectively, such that these errors, $(\tilde{e}_R, \tilde{e}_\Omega)$, exponentially reach an arbitrarily small neighborhood of the origin $(0,0)$.

Geometric Attitude Tracking Control for quadrotor with uncertainty

In this section, we define the control inputs μ and $\hat{\mu}$. Intuition behind the definition of the control inputs is presented below,

- *Reference model:*

- (i) Control moment (\hat{M}) and the control input ($\hat{\mu}$) for the *reference model* is given in (3.30)
- (ii) $\hat{\mu}$ is designed to track the desired trajectory (through $\hat{\mu}_1$), i.e., $(\hat{R}, \hat{\Omega}) \rightarrow (R_d, \Omega_d)$, alternately $(\hat{e}_R, \hat{e}_\Omega) \rightarrow (0, 0)$
- (iii) $\hat{\mu}$ also includes the terms to account for the true uncertainty and low-pass filtered estimate of the uncertainty to counter the uncertainty (through $\hat{\mu}_2$)

- *True model:*

- (i) Control moment (M) and the control input (μ) for the *true model* is given in (3.31)

- (ii) μ is defined such that the *true model* tracks the *reference model*, i.e., $(R, \Omega) \rightarrow (\hat{R}, \hat{\Omega})$, alternately $(\tilde{e}_R, \tilde{e}_\Omega) \rightarrow (0, 0)$
- (iii) Low-pass filtered uncertainty estimate, $-C(s)\hat{\theta}$, is included to cancel the uncertainty in the *true model*. However, $-C(s)\hat{\theta}$ has to be transported to the μ space.
- (iv) μ is chosen such that the error dynamics for the errors in (3.26), (3.27) are feedback linearized and results in $J\dot{\tilde{e}}_\Omega = -\tilde{k}_R\tilde{e}_R - \tilde{k}_\Omega\tilde{e}_\Omega + P\tilde{\theta}$.

Control moment, (\hat{M}) , for the *reference model*

$$\hat{M} = \hat{\mu} + (\hat{\Omega} \times J\hat{\Omega}) - J(\hat{\Omega}^\times \hat{R}^T R_d \Omega_d - \hat{R}^T R_d \dot{\Omega}_d), \quad (3.30a)$$

$$\hat{\mu} = \hat{\mu}_1 + \hat{\mu}_2, \quad (3.30b)$$

$$\hat{\mu}_1 = -\hat{k}_R \hat{e}_R - \tilde{k}_\Omega \hat{e}_\Omega, \quad (3.30c)$$

$$\hat{\mu}_2 = J\hat{R}^T R J^{-1} R^T \hat{R} \hat{\theta} - C(s)\hat{\theta}. \quad (3.30d)$$

Control moment, (M) , for the *true model*

$$M = \mu + (\Omega \times J\Omega) - J(\Omega^\times R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d), \quad (3.31a)$$

$$\mu = \mu_1 + \mu_2, \quad (3.31b)$$

$$\mu_1 = J R^T \hat{R} J^{-1} (\hat{\mu}_1 - C(s)\hat{\theta} + \tilde{k}\tilde{e}_R + \tilde{k}_\Omega \tilde{e}_\Omega), \quad (3.31c)$$

$$\mu_2 = J R^T \hat{R} \tilde{e}_\Omega^\times \hat{R}^T R e_\Omega. \quad (3.31d)$$

In (3.30), (3.31) \hat{k}_R , \tilde{k}_R and \tilde{k}_Ω are positive constants. The resulting attitude error dynamics are presented in Appendix A.1.

Remark 3.3. The relation between μ and $\hat{\mu}$ can be shown as follows,

$$\mu = J R^T \hat{R} J^{-1} (\hat{\mu} + \tilde{k}_R \tilde{e}_R + \tilde{k}_\Omega \tilde{e}_\Omega) - R^T \hat{R} \hat{\theta} + J R^T \hat{R} \tilde{e}_\Omega^\times \hat{R}^T R e_\Omega. \quad (3.32)$$

This relation is used to calculate $\dot{\tilde{e}}_\Omega$ in Appendix A.1. Note that the presence of \tilde{k}_Ω in both $\hat{\mu}_1$ (34c) and μ_1 (35c), ensures that the relation in (3.32) is obtained after simplification.

Theorem 3.1. Consider the control moments M and \hat{M} defined in (3.31) and (3.30) for the true model and the reference model respectively and

- (i) the adaptation law using Γ -Projection as,

$$\dot{\hat{\theta}} = Proj_\Gamma(\hat{\theta}, y), \quad (3.33)$$

with the projection operator as defined in Remark 3.4,

(ii) the definition of y given as,

$$y = -(P^T \tilde{e}_\Omega + cP^T J^{-T} \tilde{e}_R), \quad (3.34a)$$

$$P = J\hat{R}^T R J^{-1} R^T \hat{R}, \quad (3.34b)$$

and

(iii) with the initial condition that satisfies,

$$\tilde{\Psi}(\hat{R}(0), R(0)) < 2, \quad (3.35)$$

$$\|\tilde{e}_\Omega(0)\|^2 < \frac{2}{\lambda_{\min}(J)} \tilde{k}_R (2 - \tilde{\Psi}(\hat{R}(0)R(0))), \quad (3.36)$$

then the attitude tracking error $(\tilde{e}_R, \tilde{e}_\Omega)$, defined in (3.26), (3.27), is exponential input-to-state stable (e-ISS) [52] in the sense of Lyapunov.

Proof. Proof is given in Appendix A.1. □

Remark 3.4. Definition of the Γ -Projection operation is given in [53] as,

$$Proj_\Gamma(\hat{\theta}, y) = \begin{cases} \Gamma y - \Gamma \frac{\nabla f(\hat{\theta})(\nabla f(\hat{\theta}))^T}{(\nabla f(\hat{\theta}))^T (\nabla f(\hat{\theta}))} \Gamma y f(\hat{\theta}), \\ \quad \text{if } f(\hat{\theta}) > 0 \wedge y^T \Gamma \nabla f(\hat{\theta}) > 0, \\ \Gamma y, \quad \text{otherwise,} \end{cases} \quad (3.37)$$

where, $f(\hat{\theta}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ is any convex function.

Figure 3.2 presents the control architecture with L_1 adaptation. Also, note the presence of low-pass filter after the adaptation law. This is integral to the L_1 adaptive controller, where the high frequency noise is filtered from the system input [40]. The L_1 adaptive controller consists of the reference model (3.25), adaptation law (3.33), and the control law (3.31).

Position Control

Now that we have an attitude controller, we can use the position controller from [33]. For some smooth position tracking command $x_d(t) \in \mathbb{R}^3$, we can define the position and velocity tracking errors as,

$$e_x = x - x_d, \quad e_v = v - v_d. \quad (3.38)$$

The desired thrust vector for the quadrotor and the scalar thrust are computed as

$$F = -k_x e_x - k_v e_v + m g \mathbf{e}_3 + m \ddot{x}_d, \quad (3.39)$$

$$f = F \cdot R \mathbf{e}_3, \quad (3.40)$$

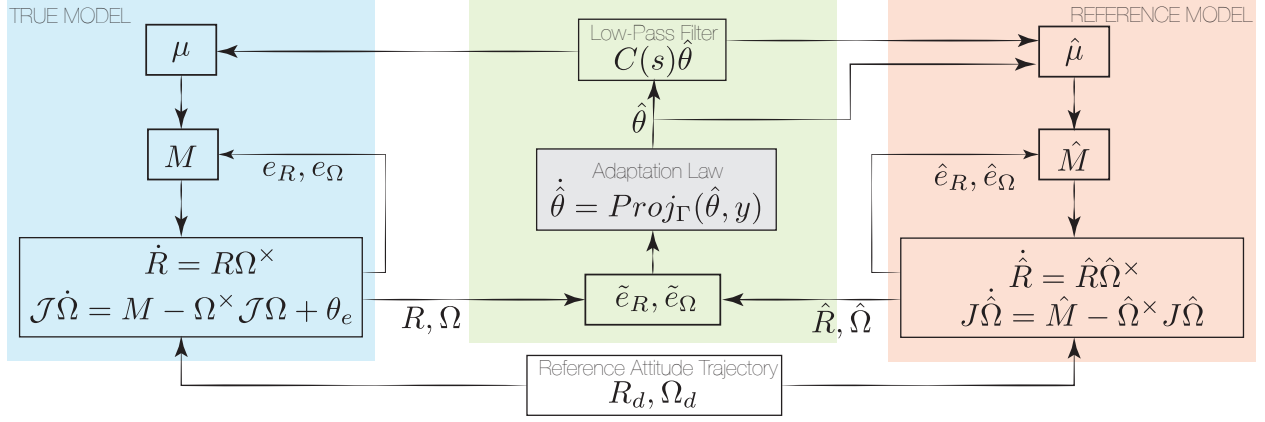


Figure 3.2: L_1 Adaptive scheme on the Geometric Attitude Control. The *true model* captures the dynamics of the quadrotor and represents the physical plant.

where k_x and k_v are positive constants and the thrust is in direction $b_3 := R e_3$. The desired orientation and angular velocity are given by,

$$R_d = [b_{1d} \quad b_{3d} \times b_{1d} \quad b_{3d}], \quad \Omega_d^\times = R_d^T \dot{R}_d, \quad (3.41)$$

$$b_{3d} = -\frac{-k_x e_x - k_v e_v + m g e_3 + m \ddot{x}_d}{\| -k_x e_x - k_v e_v + m g e_3 + m \ddot{x}_d \|}, \quad (3.42)$$

and is selected to be orthogonal to b_{1d} , such that $R_d \in SO(3)$ - see [33], [54, chapter 11] for more details.

Remark 3.5. A similar L_1 approach, used in the attitude control, can also be used to deal with disturbances and uncertainties in position control as studied in [39]. Since position and velocity evolve in the Cartesian space, we use a traditional (non-geometric) position L_1 adaptive control in the simulations and experiments discussed in the later sections.

3.4 Numerical Validation

In this section, we present numerical examples to validate the controller presented in the previous section. We discuss the performance of geometric L_1 control compared to geometric control *without* L_1 (geometric PD control) [33]. Two different scenarios are considered to evaluate the performance of geometric L_1 controller. We also compare geometric L_1 control performance to Euler angle L_1 control to present a better insight into the performance of geometric control.

The system and control parameters are ,

$$m = 1.129 \text{ kg}, J = \text{diag}([6.968, 6.211, 10.34]) e^{-3} \text{ kg} \cdot \text{m}^2,$$

$$k_x = 4, k_v = 3.2, k_R = \tilde{k}_R = 2, k_\Omega = \hat{k}_\Omega = \tilde{k}_\Omega = 0.25.$$

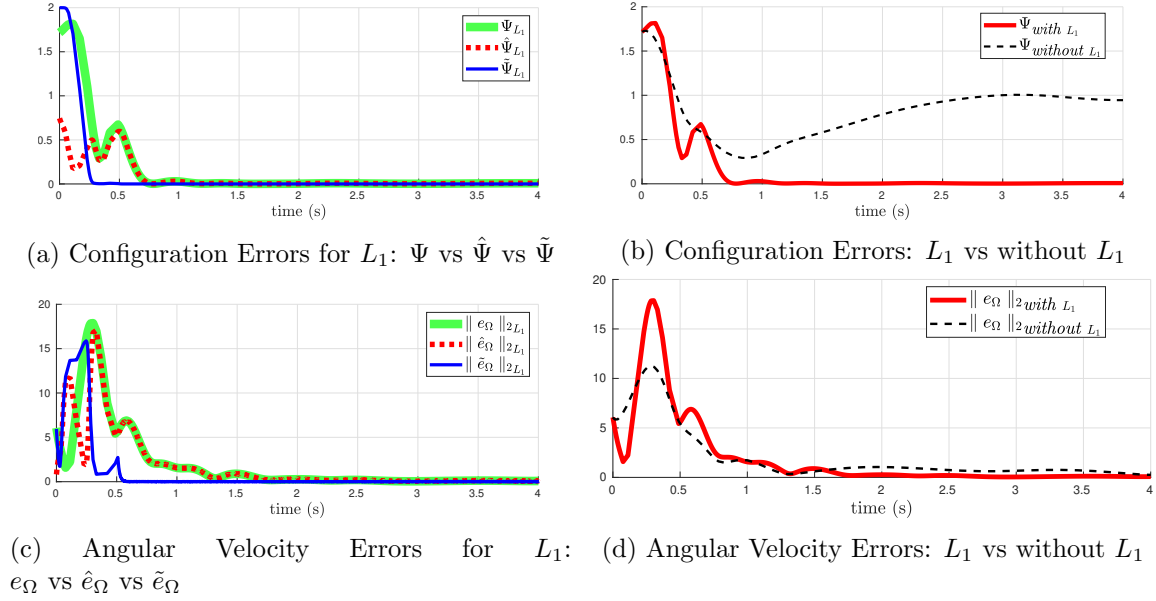


Figure 3.3: *Circular Trajectory - Case I*: Comparison of errors between *true model* and *reference model* in the presence of constant external disturbance $\theta_t = [.95, .25, -.5]^T$ and model uncertainty ($m_a = 0.5kg$ at $r = [0.2, 0.2, 0.2]^T m$). (a) & (c) show that $\tilde{\Psi}$ and $\|\tilde{e}_\Omega\|$ decrease to zero, (b) and (d) show that the errors do not converge to zero in case of geometric control *without* L_1 .

A first-order low-pass filter of form, $C(s) = \frac{a}{s+a}$, is used with $a = 2$ and an adaptation gain of $\Gamma = 10^6 I_{3 \times 3}$. We present simulations for a quadrotor tracking a desired circular trajectory given below,

$$x(t) = [\rho \cos(\omega t), \rho \sin(\omega t), 0]^T, \psi(t) = 0, \quad (3.43)$$

where $\rho = 1$, $\omega = 2$. The two different scenarios considered are, (i) a constant external disturbance and (ii) a time varying disturbance. In both scenarios, we consider a mass $m_a = 0.5kg$ attached to the quadrotor at $r = [0.2, 0.2, 0.2]^T m$ in the body frame, this added mass and its location is unknown to the controller. Presence of this added mass will also result in a moment about the center-of-mass due to gravity. Thus the external disturbance due to the mass is given as,

$$\theta_e = m_a g (r \times R^T e_3) \quad (3.44)$$

$$\mathcal{J} = J + J_{m_a}, \quad (3.45)$$

where, \mathcal{J} is the true inertia used in simulating the dynamics of the *true model* and J is the nominal inertia used to calculate the control moment and $J_{m_a} = -m_a (r^\times)^2$ is inertia tensor due to m_a in the body-frame.

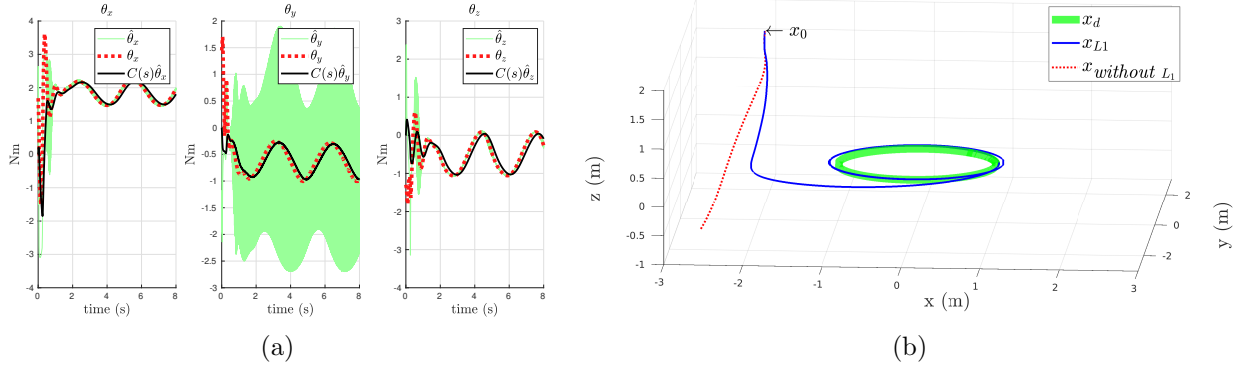


Figure 3.4: *Circular Trajectory - Case I* (a) Uncertainty in the system calculated using (3.18), (b) Trajectory response with and without L_1 adaptation. Geometric control *without* L_1 and with model uncertainty results in an unstable response (shown in red).

Geometric L_1 control vs Geometric control *without* L_1

Case I: Constant External Disturbance θ_e

An arbitrary external disturbance of $\theta_t = [0.95, 0.25, -0.5]^T Nm$ in the body-frame is considered. θ_t along with an off-center added mass results in,

$$\theta_e = [0.95, 0.25, -0.5]^T + m_a g (r \times R^T e_3) Nm. \quad (3.46)$$

Note that, this value is unknown to the controller. The initial condition for the controller is almost an inverted case with a roll of 178° . Figure 3.3 shows different errors in the system, including the errors between the *true model* and the *reference model* in the sub-figures 3.3a and 3.3c. Comparison between the controllers with L_1 and without L_1 are shown in the subfigures 3.3b and 3.3d. As shown in these figures, only L_1 control errors converge to zero.

Figure 3.4a shows the uncertainty θ in the system, along with the predicted disturbance $\hat{\theta}$ and the filtered disturbance $C(s)\hat{\theta}$. As shown in the figure, the unfiltered uncertainty estimate is noisy while the filtered uncertainty estimate closely follows the actual uncertainty.

The quadrotor position, x , using the two different Controllers L_1 , *without* L_1 is shown in Figure 3.4b. Note that the control *without* L_1 fails to track the desired trajectory due to the large unknown model uncertainty in the system.

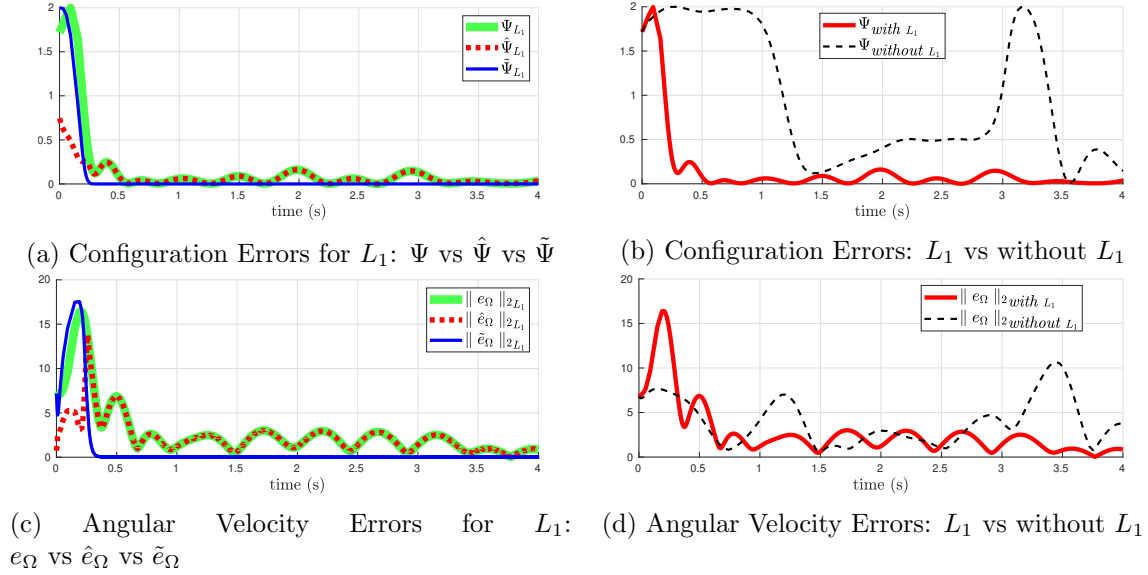


Figure 3.5: *Circular Trajectory - Case II*: Comparison of errors between *true model* and *reference model* in the presence of time-varying external disturbances and model uncertainties (see (3.47), (3.44)). The *true model* errors and *reference model* errors did not converge to zero, however reached a bounded region about zero in case of geometric control *with* L_1 . Errors in case of geometric control *without* L_1 are not-bounded as shown in 3.8b and 3.5d.

Case II: Time varying disturbance $\theta_e(t)$

Here, we consider a time varying disturbance along with model uncertainty. The following expressions denote the external disturbance used in this simulation example:

$$\begin{aligned} \theta_t = & \frac{1}{2}(\cos(t) + 0.5 \cos(3t + 0.23) \\ & + 0.5 \cos(5t - 0.4) + 0.5 \cos(7t + 2.09)), \end{aligned} \quad (3.47)$$

$$\theta_e = \theta_t + m_a g(r \times R^T \mathbf{e}_3). \quad (3.48)$$

The uncertainty due to both the time-varying disturbance and model uncertainty and the corresponding estimation through L_1 adaptation is shown in Figure 3.6a. Attitude tracking errors and resultant trajectory is shown in Figures 3.5 and 3.6b. From sub-figures 3.8a and 3.5c it can be noticed that the errors between *true model* and the *reference model*, $\tilde{\Psi}$ converge to zero very quickly, however the error between the *true model* and the desired trajectory Ψ doesn't completely converge to zero but reaches a bounded region about zero, this corresponds to the input-to-state stability discussed in Appendix A.1. As seen from Figure 3.6b, L_1 control performs better at tracking the trajectory even in the presence of time-varying disturbances.

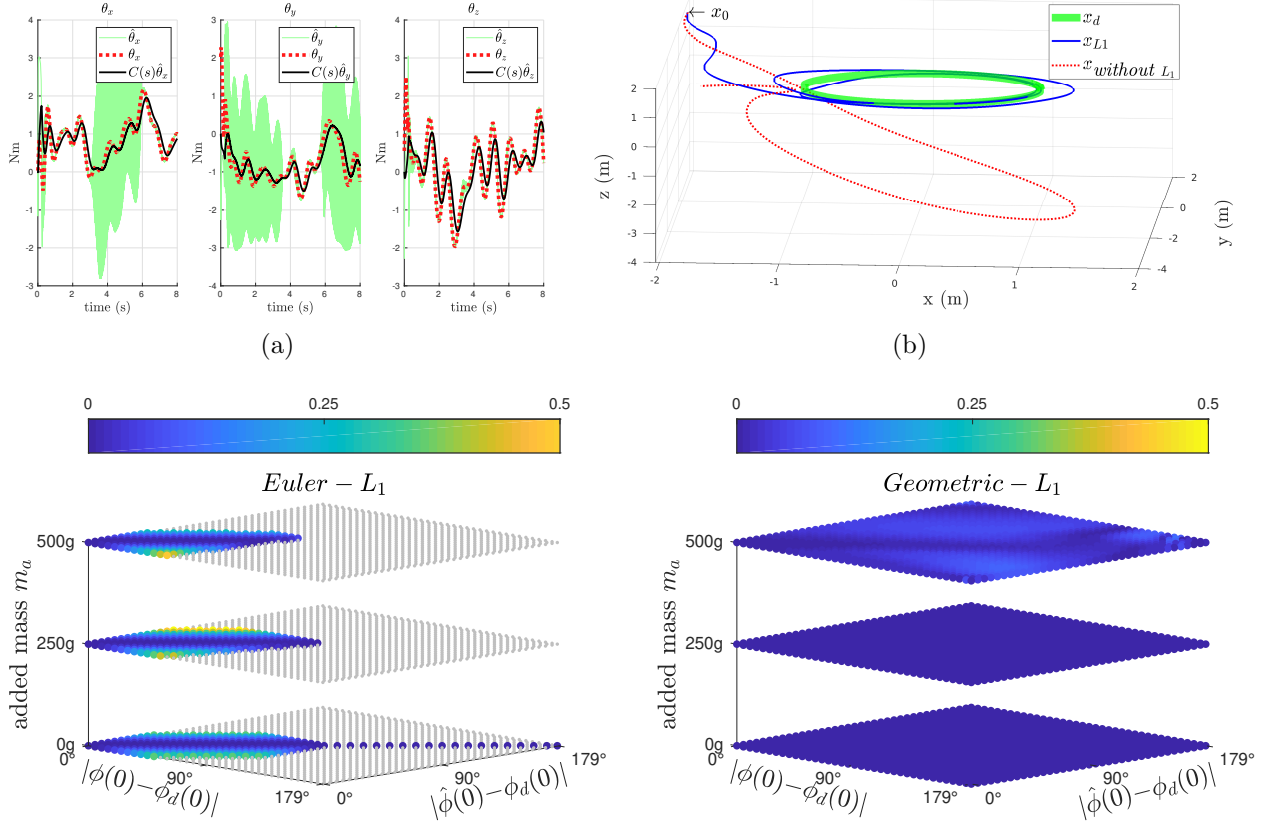


Figure 3.7: Comparison between Euler L_1 and Geometric L_1 for various initialization errors: Attitude configuration error, Ψ , after 3 seconds of simulation for different added mass m_a , different initial roll angle errors between reference model and the desired, $\hat{\phi}(0) - \phi_d(0)$, and between true model and desired, $\phi(0) - \phi_d(0)$. Initial roll angles for *true model* $\phi(0)$ and *reference model* $\hat{\phi}(0)$ are swept from 0° to 179° with 5° increments, while the desired angles are *roll* $\equiv \phi_d \equiv \text{pitch} \equiv \text{yaw} \equiv 0^\circ$. Note, that geometric L_1 works for all considered configuration errors while the Euler L_1 failed for larger configuration errors (shown in grey). With increased added mass, Ψ increases and the range of errors for which the Euler L_1 control works decreases.

Geometric L_1 vs Euler L_1

In order to establish the need for the geometric L_1 control, in this section we study and compare the initial condition response of Euler L_1 and geometric L_1 . Similar to geometric L_1 , an Euler L_1 control consists of a *reference model* and a *true model*, with the control moment computed using Euler angles and body-rates. The L_1 adaptation is implemented by considering a Lyapunov candidate for the Euler dynamics. In order to keep the comparison fair and avoid singularities, both controllers are simulated with the geometric rotational dynamics in (3.3)-(3.4).

We consider initial condition responses for several different initial roll angle errors between

the *reference model* and the *desired*, $\hat{\phi}(0) - \phi_d(0)$, and the *true model* and the *desired*, $\phi(0) - \phi_d(0)$, for various added masses $m_a \in \{0, 250g, 500g\}$. We keep all other errors zero and consider zero desired angles, $roll \equiv \phi_d \equiv 0^\circ$, $pitch \equiv 0^\circ$, $yaw \equiv 0^\circ$. We ran the simulations varying $\hat{\phi}(0) - \phi_d(0)$ and $\phi(0) - \phi_d(0)$ from 0° and 179° in 5° increments. The attitude configurations Ψ after 3 seconds of simulations is shown in Figure 3.7 indicated by the colormap. Each of the plots in Figure 3.7 illustrate $37 \times 37 \times 3 = 4107$ simulations.

We infer the following observations from the Figure,

- (i) Euler L_1 fails for larger attitude errors (shown in grey in the Figure), while geometric L_1 works for all considered configuration errors.
- (ii) Attitude configuration errors Ψ increases with increasing model uncertainty (i.e., the added mass m_a)
- (iii) The range of errors for which the Euler L_1 control works decreases with increased added mass m_a
- (iv) Also, note that in the case of no added mass ($m_a = 0$), Euler L_1 works for all initial angles when *reference model* and *true model* are initialized to the same angle, i.e., $\phi(0) = \hat{\phi}(0)$ corresponding to the diagonal line.

Step Input Response

In the section, we study the step response for geometric L_1 as well as other controllers. We consider a step input change in desired angles $roll=30^\circ$, $pitch=30^\circ$, $yaw=30^\circ$, along with an uncertainty of $0.5kg$ added mass and a time-varying disturbance. We compare the performance of geometric L_1 , geometric *PD without* L_1 , and geometric PID [55] (*without* L_1). We compare these controllers both without and with an input saturation of 5 Nm. The PD gains for all three controllers are chosen the same. The integral gain for the PID was increased to achieve the best performance for the case of no input saturation, and for the case of input saturation, the integral gain was chosen so that the input is just below the saturation in order to avoid integral windup. The resulting step responses are shown in Figure 3.8. As seen from the Figure, the geometric L_1 control outperforms the geometric PD and also has a better transient performance than the geometric PID, irrespective of input saturation. Furthermore, the geometric L_1 has a better steady-state performance compared to the geometric PID (with input saturation) and a similar steady-state performance compared to the geometric PID (without input saturation). It is important to note that comparing different controllers is difficult as it involves tuning several parameters. For instance, PID control depends on the integral gain, while L_1 control depends on the adaptation gain and the cut-off frequency. In order to have a fair comparison, we applied the same input saturation to all controllers and studied their step response performance.

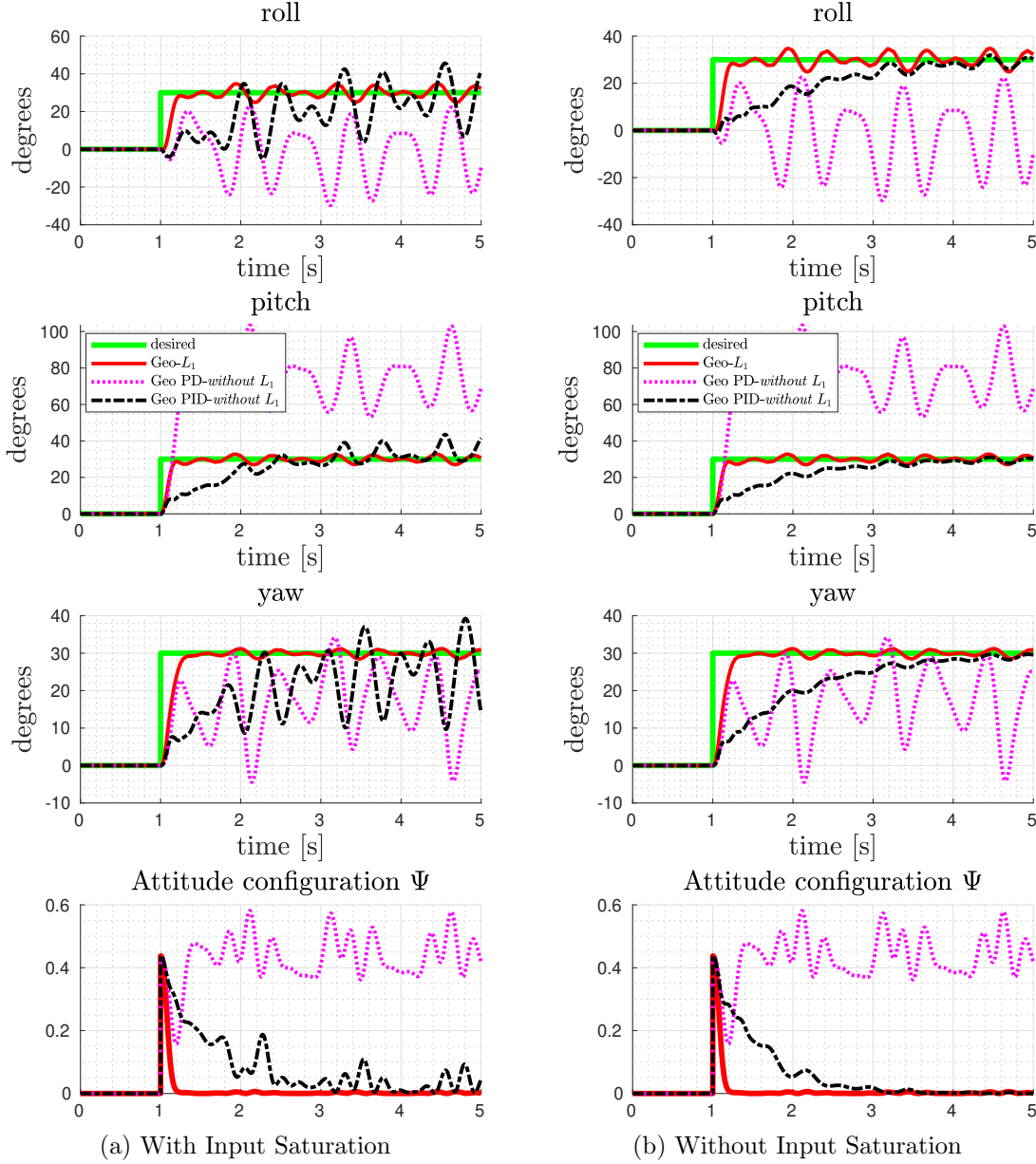


Figure 3.8: Step response comparison between different controls (i) Geometric L_1 , (ii) Geometric PD *without* L_1 , (iii) Geometric PID (*without* L_1). Following desired Euler angles $roll=30^\circ$, $pitch=30^\circ$, $yaw=30^\circ$ are considered with added mass $m_a = 0.5kg$ and a time-varying disturbance. Note that PID control improves the steady-state performance when there is no input saturation and L_1 control has better transient performance in both cases.

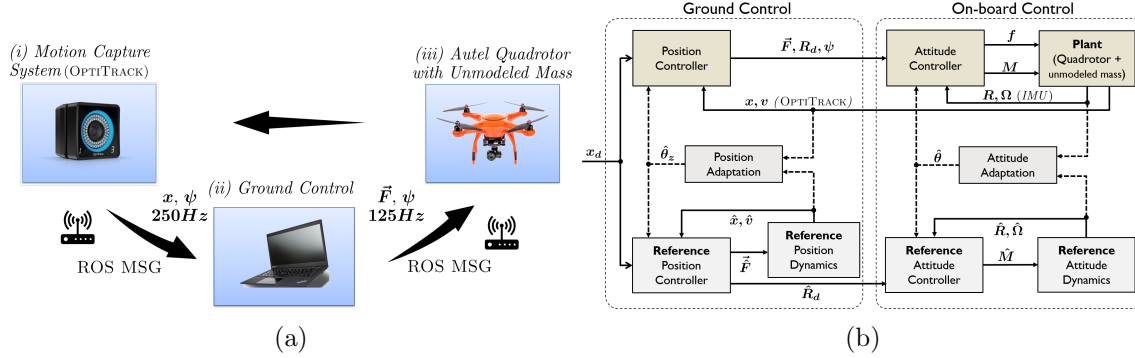


Figure 3.9: Experimental setup and the control architecture to track the quadrotor trajectory with L_1 adaptation to estimate the uncertainty θ . (a) Setup used to validate the Geometric L_1 control consisting of (i) Motion capture system to track the pose of the quadrotor; (ii) Ground control with the position control for the quadrotor; and (iii) Autel quadrotor with on-board Geometric Attitude control with L_1 adaptation. (b) Control architecture used in the experiments showing the ground and on-board controls.

3.5 Experimental Results

In this Section, we present the experimental results for the Geometric L_1 Adaptive Control developed in the previous sections.

Setup

The experiments are conducted using the Autel-X star quadrotor equipped with a Raspberry Pi 3 based Navio-2. A ROS node on Raspberry Pi 3 runs the on-board attitude control at rate of $1kHz$. Figure 3.9a illustrates the experimental setup used in this chapter. A motion capture system Optitrack is used to estimate the pose, velocity and yaw of the quadrotor at $250Hz$. The Inertial Measurement Unit (IMU) on the Navio2 is used to estimate the body-attitude and body-rates. A Lenovo-Thinkpad with Ubuntu 14.04 and ROS constitutes the ground control. A ROS node runs the position control on the ground control and communicates with the onboard control through WiFi at $125Hz$.

To generate disturbances and model uncertainty to the quadrotor system, a weight of $m_a = 0.2kg$ is rigidly attached to the fuselage of the quadrotor at $[0.1, 0.1, 0.1]^T m$ in the body-frame. This additional mass is unknown to the controller(s) and constitutes both model uncertainty, with true inertial \mathcal{J} (see (3.45)) and disturbance in the system; added mass generates a moment due to gravity about the quadrotor center-of-mass, given as,

$$\theta_m = m_a g (r \times R^T e_3).$$

Since, this moment varies with R , especially for a circular trajectory it would result in a time-varying disturbance.

Control Architecture

The control technique used to track the desired trajectory using L_1 Geometric Attitude controller consists of (i) a position controller, along with a reference model for position dynamics and (ii) an attitude controller with a reference attitude dynamics. Figure 3.9b illustrates the control architecture used in the experiments. From the desired trajectory, positions and its higher derivatives are calculated and used to calculate the desired states and the feed forward inputs using differential flatness [26]. From the desired position, velocity and acceleration, the desired thrust vector \vec{F} is calculated. A reference position dynamics is simulated using Euler integration and the uncertainty in the position dynamics is estimated and is compensated in the thrust calculation. Based on this thrust vector information, the desired attitude is calculated on-board and is used to compute the desired moments for the quadrotor M and reference model \hat{M} . \hat{M} is used in the reference attitude dynamics simulation, achieved through on-board Euler integration. The scalar thrust, f , is calculated using the thrust vector, \vec{F} , and the attitude R using (3.40). The moment, M , along with the thrust, f , is used to generate the desired angular speed for the motors as presented in [56].

Results

To validate the developed controller, we show the tracking performance of the quadrotor (with the weight m_a attached) with geometric L_1 adaptive control and geometric PD control in (3.8) without L_1 adaptation (referred to as *without L_1*).

Remark 3.6. Note that the position controller in the ground control (see Figure 3.9b) is same for both L_1 and without L_1 controllers (i.e, we are comparing only the attitude control).

The different experimental system parameters for the quadrotor are,

$$m = 1.129kg, m_a = 0.2kg, r = [0.1, 0.1, 0.1]^T m$$

$$J = \begin{bmatrix} 6.968 & -0.02909 & -0.2456 \\ -0.02909 & 6.211 & 0.3871 \\ -0.2456 & 0.3871 & 10.34 \end{bmatrix} 10^{-3} kgm^2.$$

We show the performance for Hover and a Circular trajectory tracking in the following subsections¹.

Hover

The desired states for the Hover are, $x_d = [0, -1, 1]^T$, $v_d = [0, 0, 0]^T$, $R = I_{3 \times 3}$ and $\Omega_d = [0, 0, 0]^T$. Figure 3.10 presents the tracking performance in terms of (a) position error, and (b) the attitude configuration error for Hover. The mean and standard-deviation for these errors are presented in Table 3.2.

¹A video showcasing the experiments can be found at this link, <https://youtu.be/nBDDxpkz6Pg>.

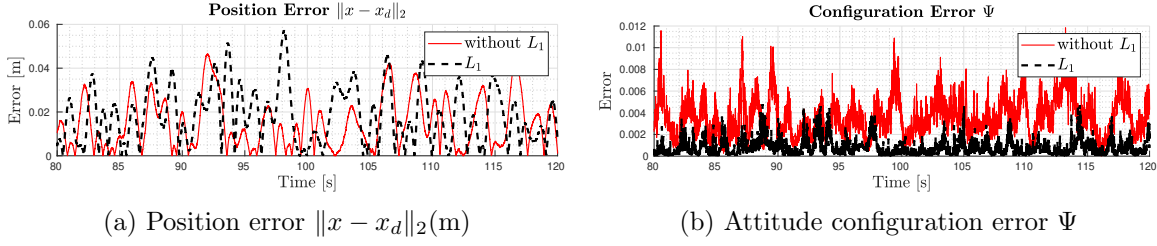


Figure 3.10: *Hover*: Tracking performance with and without L_1 controller on quadrotor with attached weight. (a) the position error, and (b) the attitude error.

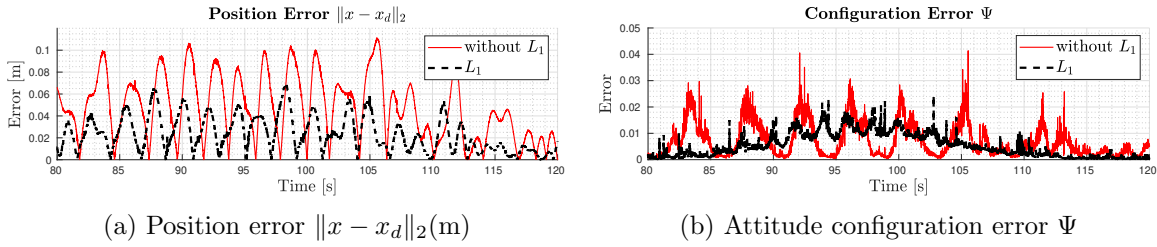


Figure 3.11: *Circular Trajectory*: Tracking performance with and without L_1 controller on quadrotor with attached weight. (a) the position error, (b) the attitude error.

Experiment	Error	Without L_1 mean [std-dev]	With L_1 mean [std-dev]
Hover	$\ x - x_d\ _2$	0.0152 [0.0118]	0.0182 [0.0132]
	Ψ	0.0055 [0.0462]	0.0021 [0.0452]
Circular Trajectory	$\ x - x_d\ _2$	0.0250 [0.0220]	0.0126 [0.0120]
	Ψ	0.0056 [0.0428]	0.0027 [0.0423]

Table 3.2: Mean and standard-deviation for position and attitude errors for Hover and Circular Trajectory with and without L_1 adaptation.

Circular Trajectory

We chose the following circular trajectory in flat-outputs:

$$x = [x_0 + \rho \cos(\omega(t)), y_0 + \rho \sin(\omega(t)), z_0]^T, \psi(t) = 0 \quad (3.49)$$

where, $\omega(t) = \frac{2\pi a}{b + \exp^{-c(t-t_0)}}$ and $[x_0, y_0, z_0]$ is the center of the circle and ρ is the radius of the circle. The parameters used in the experiment are $\rho = 1$, $x_0 = y_0 = 0$, $z_0 = 1.5$, $a = 13$, $b = 1$, $c = 0.1$ and $t_0 = 80$. These values result in a circular trajectory with increasing speed to a maximum of 2m/s and then decreasing to zero. Tracking performance for this trajectory is presented in Figure 3.11 and the mean and standard-deviation of the errors are presented in Table 3.2.

Remark 3.7. As it can be noticed from Figures 3.10 & 3.11 and Table 3.2, controllers with and without L_1 have similar performance in case of Hover, while the L_1 controller shows

approximately a factor of two better performance in the circular trajectory case. This could be attributed to the fact that the states and especially the body-angular velocity has an effect on the uncertainty as can be seen from (3.17) & (3.18). In particular, in case of hover the angular velocity Ω is close to zero resulting in a small uncertainties $\theta_m \approx 0$ while in case of the circular trajectory, the feed-forward angular velocity results in significant uncertainty θ .

3.6 Summary

We address the problem of disturbances and model uncertainties in cases of quadrotor control using geometric control techniques, where coordinate-free dynamics are used to avoid singularities. We develop the error dynamics with uncertainties in the system and a reference model without any uncertainty. Attitude tracking errors are defined between the robot model and the reference model. We develop control for the robot model and the reference model with L_1 adaptation to estimate the uncertainty in the robot model. Control Lyapunov candidate is defined to show that the proposed control strategy results in exponential stability between the robot model and reference model beyond a small bound about the origin. The bounded region is inversely proportional to the chosen adaptation gain Γ . Numerical simulations are presented to validate the control in the presence of disturbances and model uncertainties. Experimental results show the improved performance due to the L_1 adaptive control. In the next chapter, a variation-based extended Kalman filter is presented for estimating states evolving on S^2 manifolds.

Chapter 4

Variation-based Extended Kalman Filter on S^2

Being able to manipulate a cable-suspended system/gripper accurately is essential to achieving suspended grasping and collaborative grasping using multiple quadrotors. As described in the earlier chapters, the cable attitude is represented using the 2D unit-sphere S^2 and controllers [30, 57] have been designed that exhibit almost global stability. The controllers assume complete and accurate knowledge of the state and thus require the best state estimate, like link attitude and angular velocity, to achieve the desired results. Typically, systems that evolve on S^2 are parametrized by azimuthal angles (spherical coordinates) wrt a local reference system. Prior estimation algorithms on S^2 use the local parameterizations and estimate the azimuthal angles. However, such estimators are not valid globally and have singularities.

In this chapter, we propose modifications to the extended Kalman filter (EKF) to estimate the states on TS^2 and their products, called *Variation EKF*, or VEKF for short. The rest of the chapter presents a brief introduction to the Kalman filter, followed by operators mapping to and fro from the space of TS^2 to \mathbb{R}^n , where the measurement update of the EKF is implemented. The proposed EKF is implemented on a spherical pendulum. Finally, the VEKF is experimentally validated for both systems.

4.1 Introduction

Kalman filters provide the optimal estimate for linear systems with the gaussian process and measurement noises. Most real-world systems are typically nonlinear, and extensions of the Kalman filter, like the extended Kalman filter (EKF) and unscented Kalman filter (UKF), are the usual goto. The standard measurement update for systems evolving on nonlinear manifolds does not guarantee that the estimated state remains on the manifold. Thus the need for estimators that estimate directly on the manifold. One such estimator, a set-bounded estimator for the spherical pendulum that uses a global, coordinate-free parametrization, is

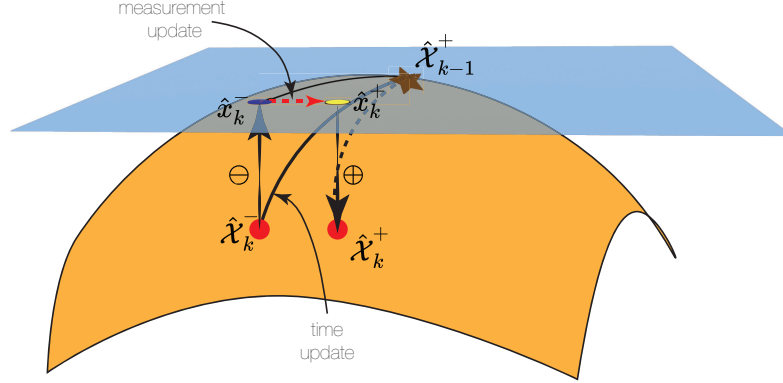


Figure 4.1: Schematic for variation extended Kalman filter on S^2 : For each new measurement, the state is transformed into the Euclidean space using \ominus , where the measurement update is performed on the variation. The variation is mapped back onto the manifold using \oplus .

developed in [58]. In the set-bounded estimator, the pendulum attitude is represented by $SO(3)$ with a symmetry axis, effectively reducing the attitude to S^2 and assuming bounded measurement noise. Unlike the set-bounded estimators, this chapter provides a global formulation of an EKF for S^2 estimation using variation-based methods, similar to the estimation of states on $SE(3)$ in [59].

For hardware implementation of a quadrotor with cable-suspended payload systems, external vision systems are usually used to measure the payload poses. Most often than not, the control loop is closed around the quadrotor position, not the payload position, due to the difficulty in accurately estimating the states. In [60], a downward-payload facing camera is used to estimate the payload position in the quadrotor frame. An EKF is used for the payload dynamics to estimate the cable attitude. A similar vision-based setup is used in [61] to ensure the payload lies in the camera's FOV, and in [62], the setup is used to estimate the pose of a rigid payload suspended from multiple quadrotors. These works are primarily focused towards achieving onboard vision based estimates for the cable attitude and, use EKF. The same vision-based setup can extend the VEKF presented in this chapter. In the following sections, the Kalman filter is briefly discussed primarily to establish the terminology and the notations, and is followed by the discussion on variation extended Kalman filter.

4.2 Variation Extended Kalman Filter

Kalman filter generates an optimal state estimate for a linear system with a linear measurement model with known Gaussian noise. Kalman filters consist of two steps: (i) A *time* update, where the prior knowledge of the state along with the system dynamics are used to estimate the state at a later time, and (ii) A *measurement* update, during which the estimate of the state and its covariance is updated based on the measurements received. An

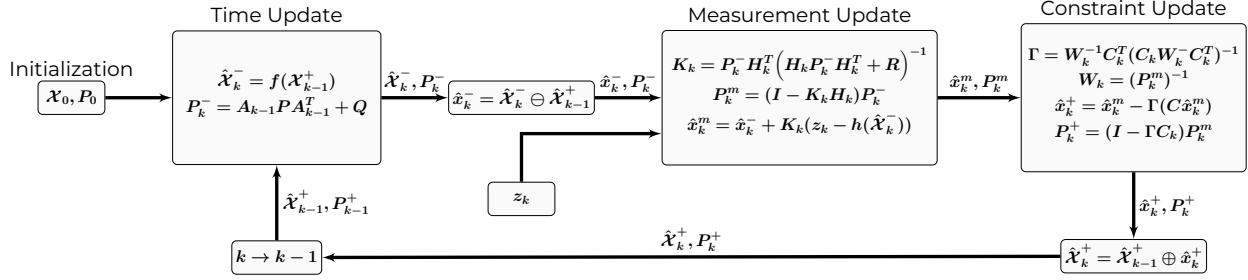


Figure 4.2: Variation based extended Kalman filter illustrating the *time* update, *measurement* update and the *constraint* update.

EKF is used for state estimation of nonlinear systems and measurement models by linearizing the dynamics and measurements. For EKF, in the *time* update, nonlinear dynamics of the system are integrated to propagate the state estimate through time. In contrast, linearized dynamics are used to update the covariance of the estimated state.

Variation extended Kalman filter estimates the cable attitude directly on S^2 . In VEKF, while the states evolve on the manifold, covariance and the measurement update on the tangent space. VEKF calculates the variation in \mathbb{R}^n between the system states at time t_{k-1} and t_k , obtained by integrating the dynamics in time update, and the measurement update is performed on the variation between the states. The variation state is updated with the measurement at t_k . Finally, the variation estimate is used to obtain the full system state estimate at t_k by projecting it back to the manifold. An illustration of the proposed method is presented in Figure 4.1.

Variations between two states on S^2

Infinitesimal variation on S^2 is defined in Section 2.3; for two states between $\mathcal{X}_1 \in TS^2$ and $\mathcal{X}_2 \in TS^2$, let $x \in \mathbb{R}^6$ be the variation between these states. The following two operators are defined to compute the transformation between the states. Consider $\ominus : TS^2 \times TS^2 \rightarrow \mathbb{R}^6$ and $\oplus : TS^2 \times \mathbb{R}^6 \rightarrow TS^2$ defined as

$$\mathcal{X}_2 \ominus \mathcal{X}_1 = x \quad \text{and} \quad \mathcal{X}_1 \oplus x = \mathcal{X}_2. \quad (4.1)$$

The mathematical formulae for the operators \ominus, \oplus are presented in Appendix B.1. Notation followed by the VEKF in this work is presented below.

Notation

- Let the estimates be represented by $\hat{\cdot}$, i.e., $\hat{\mathcal{X}}_k$ represents the full state estimate at time t_k and \hat{x}_k represents the variation state estimate between the system states $\hat{\mathcal{X}}_{k-1}$ & $\hat{\mathcal{X}}_k$.
- Let $\hat{\mathcal{X}}_k^-$ be *time* update of the system state, \hat{x}_k^- be *time* update of the variation state and P_k^- be *time* update of the covariance of the variation state at time-step k .

- Let \hat{x}_k^m and P_k^m be the *measurement* updated variation state and the corresponding covaraince.
- Let \hat{x}_k^+ and P_k^+ be the variation state and its covaraince after the *constraint* update. Finally, let $\hat{\mathcal{X}}_k^+$ be the final system state update.

Algorithm 1 *Variation Based Extended Kalman Filter*

System model and measurement model

$$\begin{aligned}\mathcal{X}_k &= f(\mathcal{X}_{k-1}, \mathcal{V}), \\ x_k &= A_{k-1}x_{k-1} + v \\ z_k &= h(\mathcal{X}_k) + w_k \\ \mathcal{X}(0) &\sim (\bar{\mathcal{X}}_0, \mathcal{P}_0), x \sim (0, P_0), v \sim (0, \mathcal{Q}), w \sim (0, \mathcal{R})\end{aligned}$$

where $f(X)$ and A are given in (4.2), (4.18). $v(t)$ & $w(t)$ are white noise processes uncorrelated, with $\mathcal{X}(0)$ and with each other.

Initialization

$$P_0^+ = P_0, \hat{\mathcal{X}}_0^+ = \bar{\mathcal{X}}_0.$$

Prior (Time) update

- 1: *A priori* state update by integrating the dynamics using (4.4), $\hat{\mathcal{X}}_k^- = f(\hat{\mathcal{X}}_{k-1}^+)$
- 2: Variation between $\hat{\mathcal{X}}_{k-1}^+$, $\hat{\mathcal{X}}_k^-$ is calculated, $\hat{x}_k^- = \hat{\mathcal{X}}_k^- \ominus \hat{\mathcal{X}}_{k-1}^+$
- 3: Covaraince update of the variation is obtained using (4.6), $P_k^- = A_{k-1}P_{k-1}^+A_{k-1}^T + \mathcal{Q}$.

Measurement update

- 1: Kalman gain is calculated using the *a priori* covaraince and the measurement matrix H_k , $K_k = P_k^- H_k^T [H_k P_k^- H_k^T + \mathcal{R}_k]^{-1}$
- 2: Measurement update of the variation using the Kalman gain and measurement z , $\hat{x}_k^m = \hat{x}_k^- + K_k(z - h(\hat{x}_k^-))$
- 3: Corresponding covaraince update of the variation, $P_k^m = (I - K_k H_k) P_k^-$

Constraint update

- 1: Constraint update by projecting the variation estimate into the constraint space, $\hat{x}_k^+ = \hat{x}_k^m - \Gamma(C\hat{x}_k^m)$
- 2: Similarly, projecting the Covaraince to lie in the constraint space.

$$\begin{aligned}P_k^+ &= (I - \Gamma C_k) P_k^m \\ \Gamma &= W_k^{-1} C_k^T (C_k W_k^{-1} C_k^T)^{-1} \\ W_k &= (P_k)^{-1}\end{aligned}$$

- 3: *A posteriori* state estimate is calculated using the updated variation, $\hat{\mathcal{X}}_k^+ = \hat{\mathcal{X}}_{k-1}^+ \oplus \hat{x}_k^+$.
-

Time Update

The system equations with gaussian process noise be represented as,

$$\mathcal{X}_k = f(\mathcal{X}_{k-1}, \mathcal{V}), \quad (4.2)$$

where \mathcal{V} is the process noise in the system. Since, the dynamics evolve on the product space of S^2 , the process noise $\mathcal{V} \notin \mathbb{R}^n$ but belongs to the tangent space of S^2 . The variation-based linearization, as described in Section 2.3, of the above dynamics be

$$x_k = A_{k-1}x_{k-1} + v, \quad (4.3)$$

and the process noise of the variation state is $v \in \mathbb{R}^n$ with covariance $\mathcal{Q} = \mathbb{E}[vv^T] \in \mathbb{R}^{n \times n}$. For practical applications, \mathcal{Q} can be considered a design parameter. Let the mean of the initial state be $\hat{\mathcal{X}}_0$ and let the initial covaraince of the variation state be $P_0 \in \mathbb{R}^{n \times n}$. The current *time* estimate $\hat{\mathcal{X}}_k^-$ is obtained by integrating the dynamics with the previous state estimate $\hat{\mathcal{X}}_{k-1}^+$ as initial condition. This is formulated as

$$\hat{\mathcal{X}}_k^- = f(\hat{\mathcal{X}}_{k-1}^+). \quad (4.4)$$

Next, the variation between the states $\hat{\mathcal{X}}_{k-1}^+$ and $\hat{\mathcal{X}}_k^-$ is computed as,

$$\hat{x}_k^- = \hat{\mathcal{X}}_k^- \ominus \hat{\mathcal{X}}_{k-1}^+, \quad (4.5)$$

where \ominus is defined (4.1) and in Appendix B.1. Rest of the estimation is performed on the variation \hat{x}_k^- , effectively transforming the estimation from TS^2 to \mathbb{R}^n . Covariance of the variation state is updated using, (see [63]),

$$P_k^- = A_{k-1}P_{k-1}^+A_{k-1}^T + \mathcal{Q}. \quad (4.6)$$

Measurement Update

The measurement $z \in \mathbb{R}^p$ is a nonlinear function of the state, as given below,

$$z = h(\mathcal{X}) + w, \quad (4.7)$$

where $w \in \mathbb{R}^p$ is Gaussian measurement noise with covariance $\mathcal{R} = \mathbb{E}[ww^T] \in \mathbb{R}^{p \times p}$. Similar to an EKF [63], the above measurement model is linearized resulting in

$$z = Hx, \quad (4.8)$$

where $H \in \mathbb{R}^{p \times 6}$ is the measurement matrix. The *time* updated variation state \hat{x}_k^- and covariance P_k^- is fused with the measurement z_k at time t_k to obtain \hat{x}_k^m, P_k^m using the following measurement update:

$$\hat{x}_k^m = \hat{x}_k^- + K_k(z - h(\hat{x}_k^-)), \quad (4.9)$$

$$P_k^m = (I - K_k H_k)P_k^-, \quad (4.10)$$

where the Kalman gain $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$.

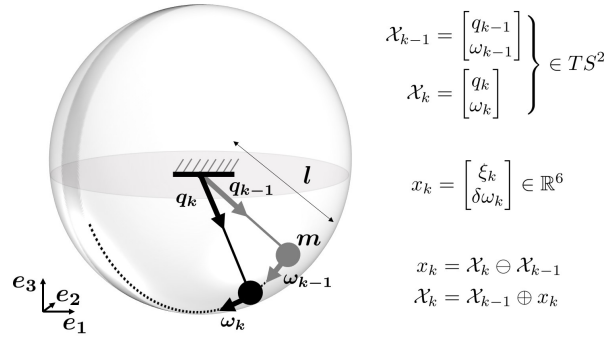


Figure 4.3: The Spherical Pendulum model evolves on S^2 with the attitude of the pendulum represented by a unit-vector $q \in S^2$. Angular velocity of the pendulum at a point q in S^2 is in the tangent space $T_q S^2 = \{\omega \in \mathbb{R}^3 \mid \omega \cdot q = 0\}$.

Constraint Update

The variation state and its covariance \hat{x}_k^m, P_k^m obtained from the measurement update are not guaranteed to satisfy the constraint (2.24). Estimated variation is then projected into the constraint surface. This is achieved by projecting the estimates into the Null-space of the constraint matrix C , as explained in [64, 65]. The constraint update is thus given by,

$$\hat{x}_k^+ = \hat{x}_k^m - \Gamma(C\hat{x}_k^m), \quad (4.11)$$

$$P_k^+ = (I - \Gamma C)P_k^m, \quad (4.12)$$

where, $\Gamma = W_k^{-1}C_k^T(C_k W_k^{-1}C_k^T)^{-1}$ with W_k being a positive definite symmetric weight matrix. Choosing $W_k = (P_k)^{-1}$ to obtain the smallest projected covariance [64]. Finally, the variation state estimate \hat{x}_k^+ is used to calculate the pendulum state estimate \mathcal{X}_k^+ as explained in Appendix B.1 i.e.,

$$\hat{\mathcal{X}}_k^+ = \hat{\mathcal{X}}_{k-1}^+ \oplus \hat{x}_k^+. \quad (4.13)$$

An overview of the variation based extended Kalman filter (VEKF) is illustrated in the Algorithm 1 and Figure 4.2. In the next sections, VEKF is implemented first on a spherical pendulum and later on quadrotor with suspended payload system.

Remark 4.1. *VEKF can also be used to estimate states on other manifolds such as $SO(3)$ and $SE(3)$, and requires computing the variations on the respective manifolds.*

4.3 Spherical Pendulum

Consider the spherical pendulum model presented in Figure 4.3. Mass of the suspended cable is negligible compared to the attached mass and the cable is considered to be mass-less. The spherical pendulum lies in the space of two-sphere S^2 .

Dynamics

Spherical pendulum states are given by a unit vector $q \in S^2$ and its corresponding angular velocity $\omega \in T_q S^2$. Let \mathcal{X} represent the *pendulum state* i.e.,

$$\mathcal{X} = \begin{bmatrix} q \\ \omega \end{bmatrix} \in TS^2. \quad (4.14)$$

The continuous time equations of motion for spherical pendulum is given as,

$$\begin{bmatrix} \dot{q} \\ \dot{\omega} \end{bmatrix} =: \dot{\mathcal{X}} = \begin{bmatrix} \omega \times q \\ -\frac{g}{l}(q \times e_3) \end{bmatrix}, \quad (4.15)$$

where m is the mass of the pendulum mass, l is the length of the pendulum. Dynamics in (4.15) are linearized about the current states using the variaton-linearization described in Section 2.3. Taking variation of the spherical pendulum dynamics in (4.15) about the current state \mathcal{X} yields the following variation dynamics (see [66]) ,

$$\delta \dot{q} = \delta \omega \times q + \omega \times \delta q, \quad (4.16)$$

$$\delta \dot{\omega} = \frac{-g}{l}(\delta q \times e_3). \quad (4.17)$$

Substituting for $\delta q = \xi^\times q$ from (2.20), we have the following coordinate-free linearized dynamics,

$$\begin{bmatrix} \dot{\xi} \\ \delta \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} qq^T \omega^\times & I_3 - qq^T \\ -\frac{g}{l} e_3^\times q^\times & O_3 \end{bmatrix}}_{\triangleq A} \begin{bmatrix} \xi \\ \delta \omega \end{bmatrix} \quad (4.18)$$

and let $x = [\xi, \delta \omega]^T$ is as defined in (2.21) along with the constraint equation in (2.24).

Remark 4.2. *Note that (4.18) and (2.24) constitute a constrained linear time-varying system. Furthermore, as noted in [66, Lemma 1], the constraint space is time-invariant, i.e., $Cx(0) = 0 \implies Cx(t) = 0, \forall t \geq 0$. Note that, unlike in [66], we obtained the variation-based linearization (4.16)-(4.17) by taking variations about the current state \mathcal{X} and not a desired state.*

The dynamics defined above are in continuous time domain, a first order Euler integration can be used to convert them to discrete domain. For the sperical pendulum, a variation dynamics integrator is used to update the state dynamics, see [67].

Measurement Equation

To estimate spherical pendulum attitude, position measurements are considered during the measurement update of the EKF. Measurement model given the state of the spherical pendulum is,

$$h(\mathcal{X}) = lq. \quad (4.19)$$

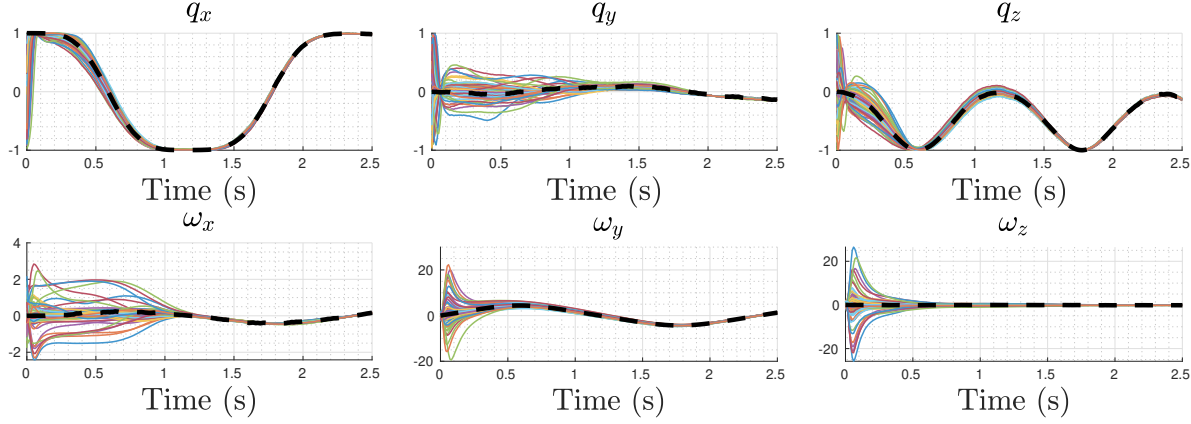


Figure 4.4: *Convergence of VEKF*: VEKF was run 100 times with different random initial states, while using the same measurements, noise covariances and initial covariance. Figure shows convergence of the pendulum attitude (top row) and the angular velocity (bottom row) estimates and the dashed black line shows the true state.

Similarly, the measurment equation after prior update is, $h(\hat{\mathcal{X}}_k^-) = l\hat{q}_k^-$. This is expressed as function of previous estimate and the variation as,

$$h(\hat{\mathcal{X}}_k^-) \equiv h(\hat{x}_k^-) = l \exp[(\xi_k^-)^\times] \hat{q}_{k-1}^+. \quad (4.20)$$

h is linearized to obtain the measurement matrix H_k as shown below,

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{x=\hat{x}_k^-} = \left. \frac{\partial \left(l \exp[(\xi)^\times] \hat{q}_{k-1}^+ \right)}{\partial \begin{bmatrix} \xi \\ \delta\omega \end{bmatrix}} \right|_{x=\hat{x}_k^-} \implies H_k = l \left[\frac{\partial \left(\exp[(\xi)^\times] \hat{q}_{k-1}^+ \right)}{\partial \xi} \quad 0_{3 \times 3} \right] \Big|_{x=\hat{x}_k^-} \quad (4.21)$$

Derivative of an exponential-map of a vector w.r.t. the vector is given in [68] or can be symbolically calculated in toolboxes such as (MATLAB, mathematica).

Numerical Simulations

For numerical simulations of the spherical pendulum system, we use the following system parameters, $m = 1kg$, $l = 1m$. We choose the noise statistics as follows: The initial covaraince of the variation state is chosen as $P_0 = I_6$, the measurement noise covariance is chosen as $\mathcal{R} = 1e^{-3}I_3$, and the process noise is $\mathcal{Q} = 1e^{-5}I_6$. The mean of the initial variation state is $\hat{x}_0 = 0_{6 \times 1}$, while the initial state of the pendulum is,

$$\hat{\mathcal{X}}_0 = [1, 0, 0, 0, 0, 0]^T.$$

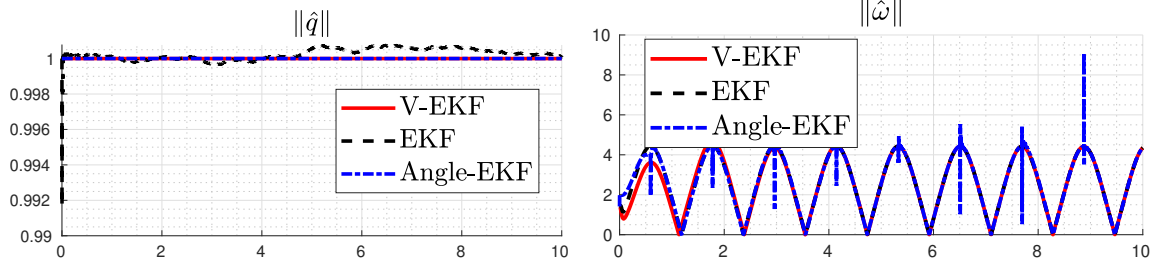


Figure 4.5: Comparison between VEKF, EKF (standard EKF applied on the dynamics in (4.15) through Jacobian linearization) and Angle-EKF (EKF applied on the dynamics represented using spherical coordinates); (Top) Plots for the norm of the load attitude estimate. (Bottom) Plot showing the norm of the estimated angular velocity.

Figure 4.4, illustrates the result of simulating the system with the VEKF for 100 random initial conditions with the aforementioned measurement model and covariances. As seen from the Figure, VEKF estimates of the state are close to the true state even with random initial states. More importantly, even with almost maximum initial attitude error, the estimates from the VEKF still converge to the true values. In these simulations, measurements are obtained by running the dynamics simulation of the spherical pendulum with initial state $\hat{\mathcal{X}}_0$ and adding random process and measurement noises with covariances \mathcal{Q} and \mathcal{R} , respectively. Random initial conditions are selected by generating random variations with covariance P_0 and transforming the initial condition $\hat{\mathcal{X}}_0$ through these variations.

Furthermore, the pendulum state estimated by the VEKF lies on S^2 .i.e, the unit norm of the pendulum attitude is preserved and is shown in the Figure 4.5. In the Figure, norm of the pendulum attitude obtained through the VEKF estimate is compared to that of a standard EKF (where the dynamics of the pendulum in (4.15) are linearized using Jacobians without taking variations on S^2 into consideration) and an angle-EKF (the dynamics of the pendulum are presented in the spherical coordinates). Angle-EKF satisfies the unit-norm constraint, since the state q is as a function sin and cos of the angles. However, this representation results in singularities in the system and can be observed from the $\|\hat{\omega}\|$ plots in the Figure. This results in very high values for the angular velocity norm near the regions of singularities.

In order for the estimated variations to be valid, the variation estimates should also satisfy the constraint in (2.24). Figure 4.6 shows the constraint values during different stages of the estimation, namely (i) *time* update, (ii) *measurement* update and (iii) *constraint* update. As shown in the Figure, constraint projection update ensures that the variation estimate satisfies the constraint in (2.24). As seen from the Figures 4.4, 4.5, 4.6, the variation based extended Kalman filter is validated using numerical examples. In the next section, VEKF is validated through experiments. Figure 4.7 presents the eigenvalues of state-covariance during a numerical simulations with and without constraint update. In both cases, the two eigenvalues are zero. This stands to reason as the variation on S^2 is constrained as given in (2.24). Note, in practice, it is observed that the estimates converge even without the constraint update.

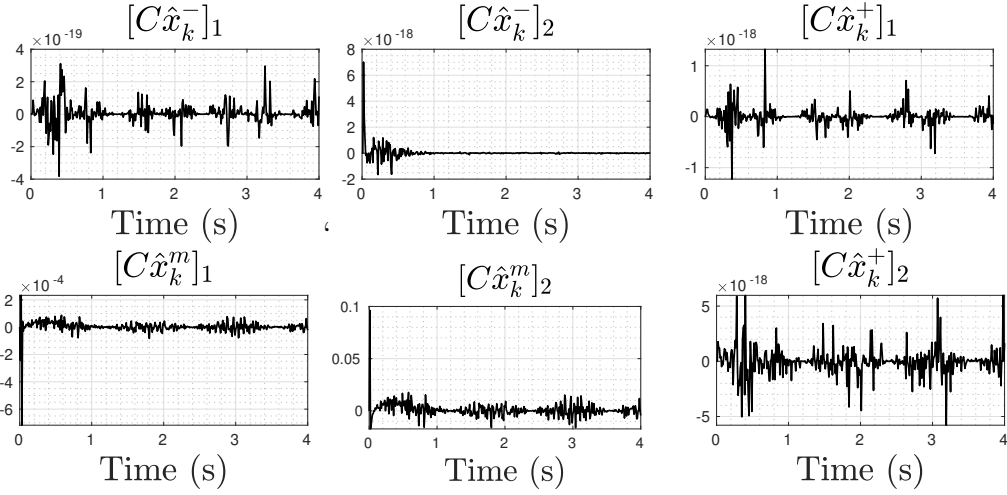


Figure 4.6: *Constraint on various variation estimates*: The two rows illustrates the two constraints in (2.24). Constraint equation is valid for prior update (*first column*), however the measurement update (*second column*) does not satisfy the constraint equation. Finally, the constraint equation is satisfied after the constraint projection update (*third column*).

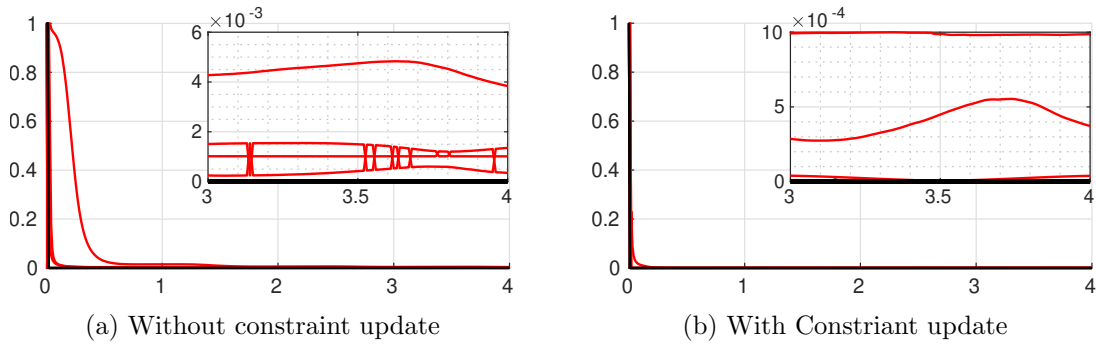


Figure 4.7: Eigenvalues of the state-covariance for a numerical simulation, with and without the constraint update. In both cases, two eigenvalues are zero (shown in black), corresponding to the two constraints in (2.24).

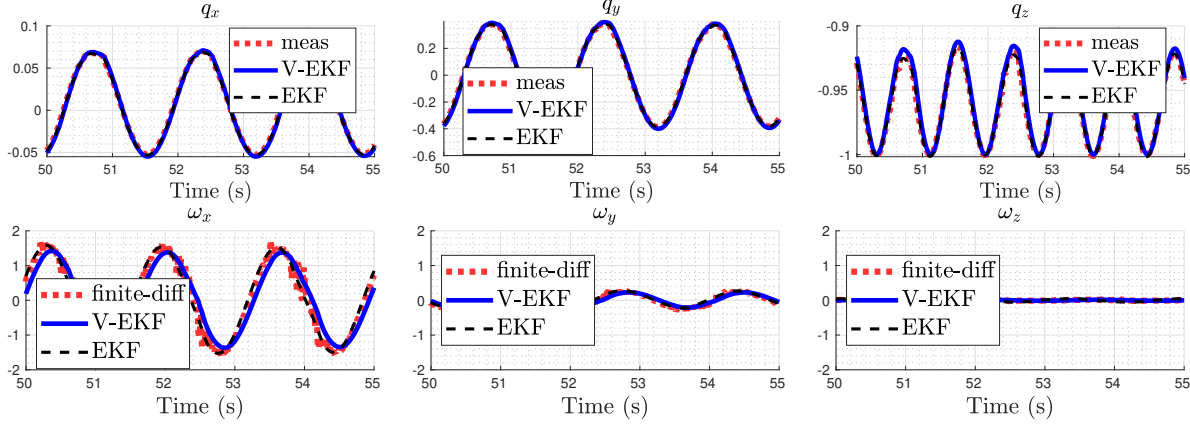


Figure 4.8: Comparison between the experiment results obtained through VEKF estimation (*solid line*), EKF estimation (*dashed line*) and the measurements (for \hat{q})/finite difference calculations (for $\hat{\omega}$) (*dotted line*). Top row shows load attitude estimate and angular velocities are shown in bottom row.

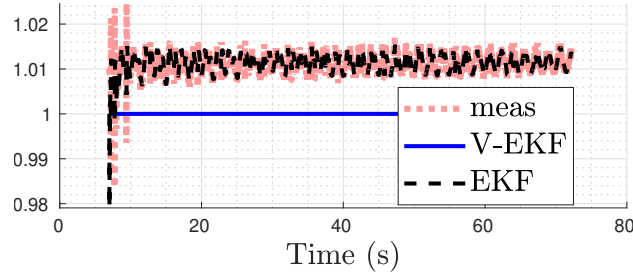


Figure 4.9: Plots of the norm of load attitude obtained through (i). measurements (dotted line), (ii). VEKF (solid line) and (iii). EKF (dashed line). Inaccurate measurement of the length of the spherical pendulum cable resulted in the offset in the norm for measurements and EKF. Also, note that the norm of q is preserved only in VEKF estimation.

Experiments

A 3D printed cube with motion capture markers suspended from a string is used for a spherical pendulum setup. The string is rigidly fixed on the other end. Figure 4.10 shows the spherical pendulum used in the experiments. An Optitrack motion capture system and reflective markers on the pendulum and the cable suspension point are used to measure the relative position of the pendulum bob. System properties of the experimental setup are, $m = 0.0580 \text{ kg}$, $l = 0.6665 \text{ m}$. Measurements are taken for freely swinging the pendulum by releasing the pendulum from a fixed position and zero velocity. The VEKF is used to estimate the state of the spherical pendulum using only the position measurements of the pendulum bob.

To validate the proposed estimation through VEKF, we show the estimated states of

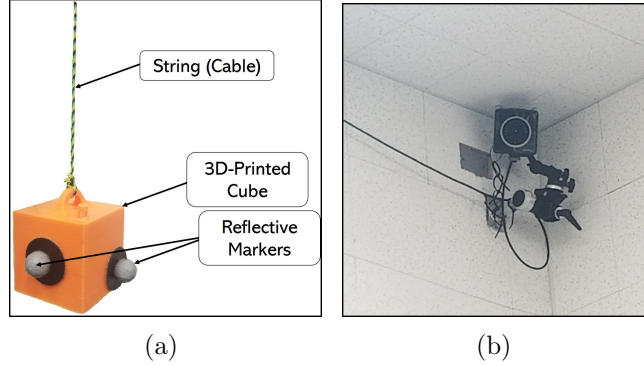


Figure 4.10: Experimental setup for spherical pendulum model, a 3D-printed cube with reflective markers and a motion capture system to track center-of-mass position of the cube. (a) 3D-Printed cube with reflective markers suspended by a string, (b) Optitrack motion capture system

the pendulum and compare the results with the sensor measurements and standard EKF. Following noise statistics for the process model and measurement model are considered,

$$\mathcal{Q} = \text{diag}([1, 1, 1, 100, 100, 100])e^{-5}, \mathcal{R} = \text{diag}([1, 1, 1])e^{-3}.$$

Initial position of the pendulum-mass is measured for over a period of 10 seconds and the measured data is used to calculate the mean of the initial state and initial covariance of the variation state. Calculated $\bar{\mathcal{X}}_0$ and P_0 for the experiments are,

$$\begin{aligned} \bar{\mathcal{X}}_0 &= [0.0070, 0.4017, -0.9157, 0, 0, 0]^T, \\ P_0 &= \text{diag}([0.25, 5e^{-4}, 1e^{-4}, 3.49, 1e^{-2}, 1e^{-3}]). \end{aligned}$$

Figure 4.8 shows the experiment results for state estimation of the spherical pendulum. Comparison between the VEKF estimation, EKF estimation and the measurements is illustrated in the Figure. As shown in the Figure, VEKF and EKF have similar estimates. However, advantage of VEKF can be seen in Figure 4.9, which shows the plots for norm of q for various estimates. As seen from the Figure, norm of q is preserved only in the case of VEKF. The offset in the case of the EKF estimate and the measurement can be attributed to the inaccurate measurement of the cable length. Such inaccuracies do not effect the structure of the state estimate for the VEKF. Also shown in the plots for ω are the values of the angular velocity computed from finite differences of the measurement. The finite difference estimates are very noisy while the VEKF estimates are smoother. Thus, as we have seen, variation based extended Kalman filter can be used to estimate the states on S^2 .

4.4 Summary

In this chapter, a variation-based Extended Kalman Filter (VEKF) is developed to estimate the states of a spherical pendulum evolving on the two-sphere S^2 . Geometric variations on

S^2 are used to obtain a variation-based linearization of the nonlinear geometric dynamics of the pendulum. The resulting linearized dynamics are time-varying with state constraints. The proposed VEKF consists of a time update, a measurement update, and a constraint update, resulting in the estimation of the states directly on TS^2 . The proposed method is validated through numerical simulations and hardware experiments on a spherical pendulum and a quadrotor with a suspended cable. In the later chapters, these estimates are required to control the quadrotor with a suspended gripper. The next chapter presents methods for fast trajectory generation for quadrotor-suspended payload.

Chapter 5

Direct Collocation for Quadrotor with a Suspended Payload

This chapter, presents a method for fast trajectory generation for a quadrotor with suspended payload. A nonlinear optimization framework is presented for offline trajectory generation along with techniques and simplifications to improve the optimization time.

5.1 Introduction

Aerial manipulation with Unmanned Aerial Vehicles (UAVs) has wide applicability in aerial package delivery. Early aerial manipulation research employed a payload that was either rigidly attached to the quadrotor or attached to an actuated manipulator arm, see [69, 70]. An alternative method is to use a cable suspension system to replace the actuated manipulator. While this preserves the agility of the UAV, the resulting system is underactuated and hybrid, due to the multiple dynamical models depending on whether the cable is slack or taut. The nonlinear underactuated and hybrid model makes fast trajectory optimization for this system challenging. This chapter addresses this by implementing direct collocation on the differentially-flat hybrid system, using complementarity constraints to avoid specifying hybrid mode sequences, using a coordinate-free geometric formulation to avoid singularities, and reformulating the obstacle avoidance constraint as a dual problem, resulting in faster optimization than prior work.

The state of the art for controllers in cable-suspended aerial manipulation can be generally classified into two approaches. The first approach is to use a feedback controller to minimize the swinging behavior of the transported load [71–73]. This method is not energetically efficient and the quadrotor has to counteract the swinging motion of the payload. Therefore, it does not exploit the system dynamics and prevents high-speed agile maneuvers. The other approach is to make use of the swing agility of the system and design the geometric controller for a quadrotor with a suspended load system [31, 74–78].

Path planning for agile navigation leveraging the swinging dynamics of the system has

been studied in [79–81]. Obstacles are represented using mixed integers to capture hard constraints of the obstacle faces in [80], while a conservative signed distance was proposed in [81]. Next, tasks such as waypoint navigation or obstacle avoidance often require a large number of parameters and must be tuned carefully in [79, 80]. Moreover, the exact order for switching hybrid modes are required to be predefined in [79, 80]. Furthermore, in [81], when constructing the optimization problem, the full dynamics was used as constraints, increasing the computational complexity. Some recent work [82, 83] on obstacle avoidance for aerial manipulation use different manipulator structures, which is quite innovative but less energy efficient and make the system more difficult to control. In this chapter,

- We model the quadrotor-load system using states as high order derivatives of load position, load attitude, cable tension and the distance between quadrotor and payload, (see Figure 2.2a). These states fully describe the system since the quadrotor-load system is differentially-flat with load position as flat outputs.
- We use direct collocation and differential-flatness to speed up the optimization and reduce the number of nonlinear constraints. Complementarity constraints are also used to represent the hybrid modes, so that mode sequences don't have to be predefined.
- The non-differentiable collision-free constraints between quadrotor-load system and obstacles are reformulated into smooth (differentiable) nonlinear constraints using strong duality of optimization. This formulation leads us to compute *energy-efficient, dynamically feasible, but safe* trajectories for a variety of tasks.

In the later sections, formulation for the trajectory generation including modeling the dynamics and the direct-collocation optimization problem is discussed.

5.2 Dynamics Modeling and System Variables

In this section, we discuss ways to model the dynamics of a quadrotor with a suspended payload for trajectory generation. The dynamics model for the quadrotor with a cable-suspended payload and its hybrid modes are discussed in detail in Section 2.5. The two hybrid modes (slack and taut) are illustrated in Figure 2.2, and the various variables are explained in Table 2.1. Hybrid modeling of the cable-suspended payloads makes trajectory optimization a complex problem since it requires computing the hybrid sequences subject to non-convex constraints on the system states.

The dynamics for the quadrotor payload system are given in (2.34), as seen in these equations, the quadrotor attitude is represented using a rotation matrix $R \in SO(3)$ and the cable attitude using $q \in S^2$. Since R and q evolve on smooth manifolds, they require manifold constraints to be satisfied for all time. Most optimization solvers and their gradient descent methods are developed in Euclidean space. The optimal solution from the trajectory generation with dynamics using (2.34) does not necessarily satisfy the manifold constraints.

The Lie group property of the rotation matrix is used to redefine the decision variables into Euclidean space. For any $\eta \in \mathbb{R}^3$, the exponential map of the lie algebra is a lie group, such that,

$$\exp_m(\eta^\times) \in SO(3), \quad (5.1)$$

where \exp_m is matrix exponential. The rotation matrix R can be expressed by rotating a rotation matrix R_0 about η , i.e.,

$$R = R_0 \exp_m(\eta^\times), \text{ for any } \eta \in \mathbb{R}^3, \quad (5.2)$$

and R_0 is initial rotation matrix and without loss of generality, $R_0 = I_3$. Rotation of a unit-vector is a unit-vector, thus, any $q \in S^2$ can be written as,

$$q = \exp_m(\xi^\times)q_0, \text{ for any } \xi \in \mathbb{R}^3, \quad (5.3)$$

where q_0 is the initial orientation of q , without loss of generality, $q_0 = [0 \ 0 \ -1]^T$. Using the expressions in (5.2) and (5.3), the quadrotor payload taut dynamics can be written as,

$$q = \exp_m(\xi^\times)q_0, \quad (5.4a)$$

$$R = R_0 \exp_m(\eta^\times), \quad (5.4b)$$

$$\dot{x}_L = v_L, \quad (5.4c)$$

$$\dot{v}_L = (1/(m_Q + m_L))(q^T f R e_3 - m_Q l \dot{q}^T \dot{q})q, \quad (5.4d)$$

$$\dot{\xi} \approx \omega, \quad (5.4e)$$

$$m_Q l \dot{\omega} = -q \times f R e_3, \quad (5.4f)$$

$$\dot{\eta} \approx \Omega, \quad (5.4g)$$

$$J_Q \dot{\Omega} = M - \Omega \times J_Q \Omega. \quad (5.4h)$$

Note, (5.4e) and (5.4g) are first-order approximations and are obtained from the kinematic equations,

$$\dot{R} = R \Omega^\times \implies \frac{d}{dt}(\exp_m(\eta^\times)) = \exp_m(\eta^\times) \Omega^\times, \quad (5.5)$$

$$\dot{q} = \omega \times q \implies \frac{d}{dt}(\exp_m(\xi^\times))q_0 = \omega^\times \exp_m(\xi^\times)q_0. \quad (5.6)$$

Equations in (5.4) are in the standard form, $\dot{\mathcal{X}} = f(\mathcal{X})$, where the decision variables $\mathcal{X} \in \mathbb{R}^{18}$ are in the Euclidean space. Planning and trajectory optimization methods from the literature, such as direct collocation, and multiple/single shooting, can be applied directly to these equations. It is important to note that the angular velocity equations are only first-order approximations; feedback controllers are required to track these trajectories on hardware successfully.

The dynamics defined in (5.4) is for quadrotor payload system in taut mode. A separate set of dynamics for slack mode are given in (2.35). Transition maps to switch the states to and fro the modes are required for unified trajectory generation for both modes together. Adding transition maps makes the optimization problem more complex and nonlinear. To address the hybrid modes of the system for optimization, we introduce a differential-flatness based parameterization of the quadrotor-load system and model the cable as a prismatic joint.

Assume the cable to be a prismatic joint of length l , and is less than or equal to full cable nominal length l_0 , i.e., $l < l_0$ when the cable is slack and $l = l_0$ when taut. With abuse of notation, the unit vector between the quadrotor center-of-mass and the payload position is denoted by $q \in S^2$, even when the cable is slack. Therefore the kinematic relation between x_Q and x_L is,

$$x_Q = x_L - l \cdot q. \quad (5.7)$$

The payload position x_L is a flat output for the quadrotor payload system. Consider the flat outputs and their derivatives as the generalized coordinates for the quadrotor-payload system. $x_L^{(i)}$ is i^{th} derivative of the payload position, with $i \in \{0, 1, \dots, 6\}$,

$$\frac{dx_L^{(i)}}{dt} = x_L^{(i+1)} \in \mathbb{R}^3 \quad \text{for } i \in \{0, 1, \dots, 5\}. \quad (5.8)$$

The trajectory generation problem is formulated on the dynamics in (5.8) and solves for the optimal flat outputs $x_L^{(i)}$. The rest of the states and inputs can be computed using differential-flatness equations from the flat outputs and system parameters. To fully describe hybrid dynamics of the system, the collocation relation described in (5.8) is not sufficient. An additional scalar magnitude for cable tension $T \in \mathbb{R}^+$ is introduced, such that,

$$-Tq = m_L \ddot{x}_L + m_L g e_3, \quad (5.9)$$

$$T(l - l_0) = 0, \quad (5.10)$$

where (5.9) and (5.10) (similar to [81]) represent the payload dynamics and the complementarity constraint, respectively. The complementarity constraint requires either $T = 0$ with $l \leq l_0$ or $l = l_0$ with $T > 0 \forall t$, captures the slack and taut modes together. With (5.8), (5.9), (5.10), we can fully describe the system hybrid dynamics.

5.3 Collocation based Path Planning

Direct Collocation is a nonlinear-optimization technique for formulating the optimal control problem for dynamical systems. In direct collocation, the system trajectory is divided into N nodes with piecewise continuous polynomials (typically cubic, Bezier, and Legendre polynomials). Numerical integration techniques such as Hermite-Simpson, first-order Euler integration, etc., are used to integrate the system dynamics within the node segments.

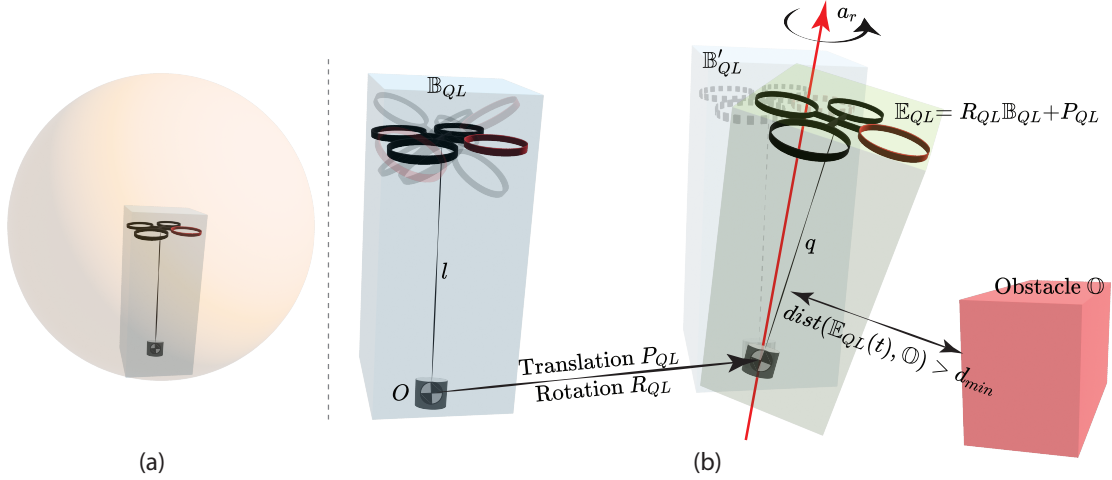


Figure 5.1: Obstacle avoidance constraint, $\text{dist}(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{\min}$, where the control object \mathbb{E}_{QL} is obtained by rotating R_{QL} & translating P_{QL} the initial cuboid configuration \mathbb{B}_{QL} , i.e., $\mathbb{E}_{QL} = R_{QL}\mathbb{B}_{QL} + P_{QL}$ and \mathbb{O} is the obstacle.

An optimization problem to reduce the error between the integrated system dynamics and the polynomials is solved, that results in optimal trajectory for the dynamical systems as piecewise continuous polynomials. For more details on the optimal trajectory generation methods, we urge the readers to refer to [84–86] and to [87–89] for their applications in the robotics field. In this section, the trajectory generation for the quadrotor-payload system is formulated as a direct-collocation problem.

The optimization problem for the dynamics defined using differential-flatness in (5.8) with N nodes is formulated as,

$$\min_{\{x_L^{(i)}, T, l, q\}} J(x_L^{(i)}, T, l, q), \quad \text{for } i \in \{0, 1, \dots, 5\} \quad (5.11a)$$

$$\text{s.t. } x_L^{(i)}(t_{k+1}) - x_L^{(i)}(t_k) = \frac{\Delta t_k}{2}(x_L^{(i+1)}(t_{k+1}) + x_L^{(i+1)}(t_k)), \text{ for } i \in \{0, 1, \dots, 5\} \quad (5.11b)$$

$$m_L x_L^{(2)}(t_k) = -T(t_k)q(t_k) - m_L g e_3, \quad (5.11c)$$

$$T(t_k) \geq 0, \quad (5.11d)$$

$$T(l(t_k) - l_0) = 0, \quad (5.11e)$$

$$0 \leq l(t_k) \leq l_0. \quad (5.11f)$$

In the rest of the section, the formulation in (5.11) is explained in detail. (5.11b) represents the direct trapezoidal collocation [84] between two neighboring nodes with $k \in \{1, 2, 3, \dots, N-1\}$. The complementarity constraint is implemented in (5.11d), (5.11e) and we limit the prismatic distance between the quadrotor and the payload in (5.11f). The trapezoidal collocation presented in (5.11b) results in a second order approximation, enabling the computational efficiency.

Cost Function

Full definition for the cost function J in (5.11a) is,

$$J = \mathcal{S}_f t_f + \sum_{k=1}^N [x_L^{(6)}(t_k)^T \mathcal{Q} x_L^{(6)}(t_k) + \mathcal{R}_0 T(t_k) + \mathcal{R}_1 (l_0 - l(t_k)) \Delta t_k + \mathcal{R}_2 (l(t_k) - l(t_{k-1}))^2 \Delta t_k] \quad (5.12)$$

where $t_f = \sum_{k=1}^N \Delta t_k$ and $\mathcal{S}_f \in \mathbb{R}^+$ represent the final time and the terminal cost parameter, respectively. $\mathcal{Q} \in \mathbb{R}^{3 \times 3}$ is a diagonal positive-definite matrix and $\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2 \in \mathbb{R}^+$. The optimization will find the optimal time while considering energy efficiency by adding a terminal time cost. The rest of the terms in the cost function are,

- $\sum_{k=1}^N x_L^{(6)}(t_k)^T \mathcal{Q} x_L^{(6)}(t_k) \Delta t_k$: To minimize the energy consumption, total input to the quadrotor-payload system moment M and thrust f is captured by the 6th derivative of the payload position (via differential-flatness).
- $\sum_{k=1}^N \mathcal{R}_0 T(t_k) \Delta t_k$: Cost on the taut mode.
- $\sum_{k=1}^N \mathcal{R}_1 (l_0 - l(t_k)) \Delta t_k$: Cost for the slack mode.
- $\sum_{k=1}^N \mathcal{R}_2 (l(t_k) - l(t_{k-1}))^2 \Delta t_k$: Cost to ensure smooth transition during slack and taut modes.

Note, that increasing \mathcal{R}_1 penalizes slack mode and resulting trajectory would be taut and similarly increasing \mathcal{R}_0 will generate trajectory tending towards slack mode. Thus, with these two parts of cost function and the complementarity constraints, the hybrid modes need not be predefined, with the optimization finding the appropriate modes.

Input Constraints

Inputs to physical systems are bounded and the inputs $u = [f, M]$ are computed from the flat outputs' derivatives $x_L^{(i)}$,

$$u = g(x_L^{(i)}). \quad (5.13)$$

In (5.13), g is a nonlinear function, and adding bounds on the inputs results in nonlinear constraints. Instead, the derivatives $x_L^{(i)}$ are bounded, and their bounds are precomputed based on the input bounds. The cable tension is bounded by the maximum tension T_{max} ,

$$0 \leq T(t_k) \leq T_{max}. \quad (5.14)$$

and T_{max} is pre-estimated from the physical input constraints.

Geometric Constraints

The cable attitude $q(t_k)$ can be bounded as,

$$-q \cdot e_3 \geq \cos(\alpha_{max}), \quad (5.15)$$

where the α_{max} represents the maximum swing angle for load attitude. Adding the constraint in (5.15), prevents the cable from aggressive swings and prevents the cable from colliding with the propellers or getting stuck in the quadrotor frame/legs.

Obstacle Avoidance

Traditional and the most common way of formulating the obstacle avoidance constraint is by adding l_2 distance norm to the obstacle. Adding l_2 norm constraint with minimum radius $l_0 + \varepsilon$ is very conservative. The conservative nature makes it challenging to find a feasible trajectory in highly constrained spaces. In this section, the quadrotor payload is represented as a cuboid, shown in Figure 5.1. We use the duality formulation from [90], transferring the obstacle avoidance constraints into smooth nonlinear constraints using the strong duality of optimization, where the quadrotor-load system is considered as a full-dimensional controlled object.

We model the quadrotor-load system as a controlled object, $\mathbb{E}_{QL}(t)$, with a cuboid shape, whose central axis represents the prismatic joint $l(t)$. The cuboid dimensions are ϵ_x , ϵ_y (the cross-section of the cuboid), ϵ_z (by the maximum prismatic length). The ϵ_x and ϵ_y are determined such that all quadrotor orientations are bounded within the cuboid. The controlled object at each timestamp is obtained through the rotation $R_{QL}(t)$ and the translation $P_{QL}(t)$ from an initial convex set $\mathbb{B}_{QL}(t) = \{y : Gy \leq g\}$, where $G \in \mathbb{R}^{6 \times 3}$ and $g \in \mathbb{R}^{6 \times 1}$. In this chapter, the initial convex set $\mathbb{B}_{QL}(t)$ represents the vertical connection between the quadrotor and the payload through the prismatic joint. The transformation relation from convex set \mathbb{B}_{QL} to \mathbb{E}_{QL} could be written as,

$$\mathbb{E}_{QL}(t) = R_{QL}(t)\mathbb{B}_{QL} + P_{QL}(t), \quad (5.16)$$

where R_{QL} and P_{QL} are defined later in (5.23) and (5.21) respectively. The initial convex set $\mathbb{B}_{QL}(t)$ is defined as,

$$\mathbb{B}_{QL}(t) = \{y \in \mathbb{R}^3 \mid \begin{bmatrix} -\epsilon_x \\ -\epsilon_y \\ 0 \end{bmatrix} \leq y \leq \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z + l(t) \end{bmatrix}\}. \quad (5.17)$$

The obstacles are modeled as convex compact sets with non-empty relative interior, such that each of them can be represented as

$$\mathbb{O} = \{y : Ay \leq b\}. \quad (5.18)$$

Then the collision-free trajectory generation with $d_{min} > 0$ as a desired safety margin can be formulated as

$$\begin{aligned} dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min} &\Leftrightarrow \exists \lambda, \mu > 0 : \\ -g^T \mu + (AP_{QL}(x) - b)^T \lambda &> d_{min}, \\ G^T \mu + R_{QL}(x)^T A^T \lambda &= 0, \quad \|A^T \lambda\| < 1, \end{aligned} \quad (5.19)$$

where λ and μ represent the dual variables associated with the obstacle \mathbb{O} . The detailed proof of (5.19) can be found [90, Theorem 1]. The collision constraint can be relaxed using a slack variable s to represent the penetration distance (when the safety margin is not needed to be strictly guaranteed), then the obstacle avoidance with minimum penetration is formulated as,

$$\begin{aligned} dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min} &\Leftrightarrow \exists \lambda, \mu > 0, s \geq 0 : \\ -g^T \mu + (AP_{QL}(x) - b)^T \lambda &> d_{min} - s, \\ G^T \mu + R_{QL}(x)^T A^T \lambda &= 0, \quad \|A^T \lambda\| < 1. \end{aligned} \quad (5.20)$$

Introducing the slack variable s as the penetration distance in (5.20) would make this non-linear optimization smoother. To keep the penetration distance to minimum, additional cost term Ks , (K is a larger positive scalar) to penalize s can be added to the cost function (5.12).

Remark 5.1. *The constraint $dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min}$ is non-differentiable, while the dual variables in the formulation (5.19) or (5.20) could make the constraint become differentiable and smooth.*

Without loss of generality, consider the initial cuboid \mathbb{B}_{QL} to be located at the origin, then the translation P_{QL} is given as,

$$P_{QL}(t) = x_L(t). \quad (5.21)$$

After the translation \mathbb{B}_{QL} is transformed to $\mathbb{B}'_{QL}(t)$, presented in Figure 5.1. Finally, rotation of the cuboid is given by R_{QL} , and the rotation can be expressed using angle-axis representation, i.e., $\exists a_r \in \mathbb{R}^3$, such that the R_{QL} is computed using (5.2). R_{QL} can also be computed using Rodrigues' rotation formula,

$$\theta_r = \|a_r\|, \hat{a}_r = a_r / \|a_r\|, \quad (5.22)$$

$$R_{QL} = I_3 + \sin(\theta_r)(\hat{a}_r)^\times + (1 - \cos(\theta_r))(\hat{a}_r)^\times{}^2, \quad (5.23)$$

further, for $\theta_r = \pi$, $a_r = \frac{q_0 + q}{\|q_0 + q\|}$ and $q_0 = [0, 0, -1]^T$ represents the vertical load attitude of cuboid $\mathbb{B}'_{QL}(t)$.

Special Tasks

In this section, the general formulation in (5.11) is modified for specific tasks, such as waypoint navigation, payload throwing towards a desired target and passing through narrow windows.

Waypoint navigation The waypoint navigation can be specified either on load position or quadrotor position. For numerical stability of the solvers, the equality constraints are reformulated always reformulated as inequality constraints with a margin ϵ ,

$$\begin{aligned} p_{wL,i} - \epsilon &\leq x_L(t) \leq p_{wL,i} + \epsilon \text{ (payload waypoints),} \\ p_{wQ,i} - \epsilon &\leq x_Q(t) \leq p_{wQ,i} + \epsilon \text{ (quadrotor waypoints),} \end{aligned} \quad (5.24)$$

where $x_Q(t) = x_L(t) - l(t)q(t)$. Moreover, it would also be possible to set the position of the load or the quadrotor for specific node, since it is fully defined in the system state.

Payload throwing A payload throwing task can be achieved through a final release of payload after quadrotor-load navigation so that it free falls towards a given target. The ballistic trajectory is well defined as function of the velocity at the moment of release. Assume x_r be the position, v_r the velocity of the payload in the world frame at the moment of release and x_t the target position. The released payload arriving at the given target x_t can be written as a constraint,

$$x_r + v_r t_r + \begin{bmatrix} 0 & 0 & -0.5gt_r^2 \end{bmatrix}^T = x_t, \quad (5.25)$$

where t_r represents the travel time from the release moment to the target position. Both the travel time t_r and the position of release x_r are solved by the optimization.

Window task In this task, the quadrotor and the load have to move through a window of height being smaller than the cable length, so that the quadrotor is forced to swing up the load to make it pass through the window, or to have hybrid mode transitions to achieve aggressive window avoidance maneuvers. Since a window-shaped obstacle can be considered as the combination of four obstacles together, the window task could be considered as a task to avoid multiple obstacles.

5.4 Numerical Simulations

This section presents the numerical simulations and optimal trajectories for various tasks such as navigation of quadrotor-load system through waypoints, obstacle avoidance etc., generated using the proposed formulation.

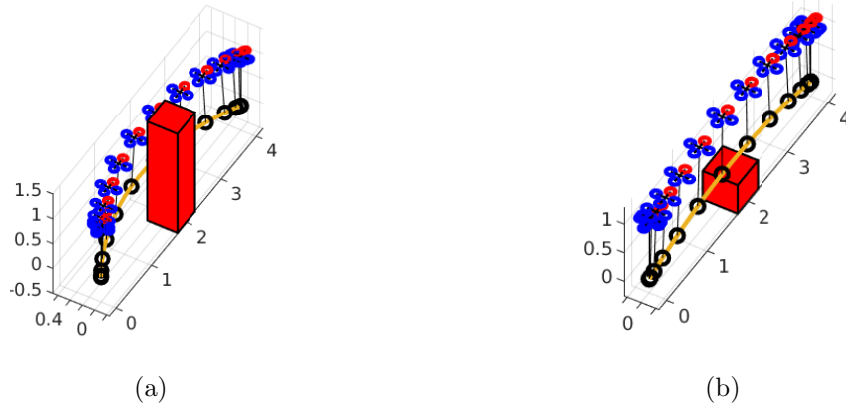


Figure 5.2: Obstacle avoidance task with the system starting at $[0, 0, 0]^T$, goal $[4, 0, 0]^T$, and a cubic obstacle centered at $[2, 0, 0]^T$ with side length equal to $1m$. The system avoids the obstacle laterally if the obstacle's height is relatively high and vertically if the obstacle's height is smaller.

Obstacle Avoidance

The proposed planning algorithm can find dynamically-feasible trajectories for navigation tasks with obstacle avoidance. Figure 5.2a shows the path planning results for an example scenario with 50 nodes with obstacle avoidance. Snapshots show that the system initially hovers at the origin and navigates to another target payload position by avoiding the obstacle between them. Moreover, since our path planning problem formulation considers energy efficiency, the quadrotor-load system will avoid obstacles with different poses based on it. For example, when the height of this cuboid is decreased to $0.5m$, the quadrotor-load system will navigate this open space by avoiding the obstacle vertically instead of laterally, shown in Figure 5.2b.

Waypoint Navigation

The algorithm can find solutions for the waypoint navigation task with computational efficiency. The quadrotor-load system is required to maneuver from a hover at the origin and back to the origin while navigating through three waypoints without any tolerance ($\epsilon = 0$). We have 100 nodes along the trajectories and 25 nodes assigned for each trajectory segment. We show in Figure 5.3a that the waypoint navigation task is well executed. As shown in the figure, The three waypoints are $[0, 2, 0]^T$, $[2, 2, 0]^T$ and $[2, 0, 0]^T$, marked with diamonds. The three payload trajectories colored blue, green, and red are optimized with $\mathcal{R}_0 = 10^4, 10^5, 10^6$, respectively. Note that the scalar \mathcal{R}_0 is the weight parameter in the cost function in (5.12).

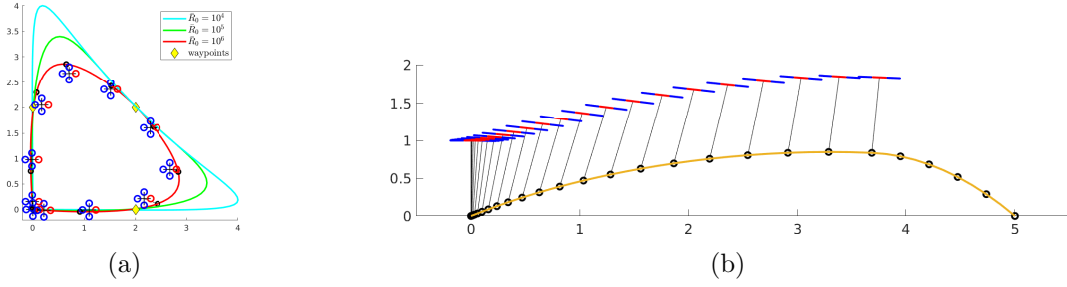


Figure 5.3: (a) Top-view of the desired payload position trajectories for navigating three waypoints without position error tolerance for varying costs \mathcal{R}_0 , (b) $x-z$ plane view of the desired payload trajectory for the throwing task. The payload is released after being transported from the origin and reaches the desired target of $[5, 0, 0]^T$ after a projectile trajectory.

Payload Throwing

For the payload-throwing task, the system needs to swing the payload up and move toward the target position. The quadrotor might need to lift the payload and impart a positive horizontal velocity at release to ensure the payload has sufficient time during the free fall. We don't need additional constraints to specify the movement while letting the optimization find the appropriate time for payload release. In the simulation presented in Figure 5.3b, the quadrotor lifts the payload from the initial position and swings it to pump energy before releasing the payload.

Window Tasks

Non-Guided Window Task For a window whose height is bigger than the length of cable, the optimization could generate a dynamically feasible trajectory to pass through this kind of window, presented in Figure 5.4a. For window heights greater than the cable length, the optimization generates dynamically feasible trajectories to pass through the windows, presented in Figure 5.4a. In this numerical simulation, the window's height and width are $1.2m$ and $1.0m$, respectively. Notice that the cable is always taut in the resulting trajectory.

State-Guided Window Task For the narrow window, to decrease the computation time, we could add constraints at the specific node during the optimization. In Figure 5.4b, we show that our system could navigate through a narrow window by defining an additional waypoint in the open space of the window obstacle as guidance for optimization. Using 30 nodes for the path planning optimization and using additional constraints at node number 15: (i) load position is inside the open space $\mathbb{S} = \{y = (y_1, y_2, y_3) : y_1 \in [-0.5, 0.5], y_2 \in [-1, 1], y_3 \in [1.5, 2.5]\}$; (ii) load attitude angle α is bigger than α_{min} , where $\alpha_{min} = \pi/4$. Here

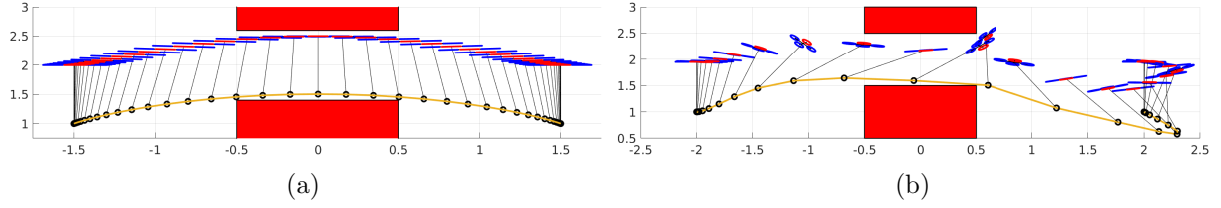


Figure 5.4: (a) Sideview of the quadrotor-payload system passing through a window with vertical gap greater than the cable length, (b) Trajectory of the system passing through a narrow window with a predefined waypoint in the open space of the window.

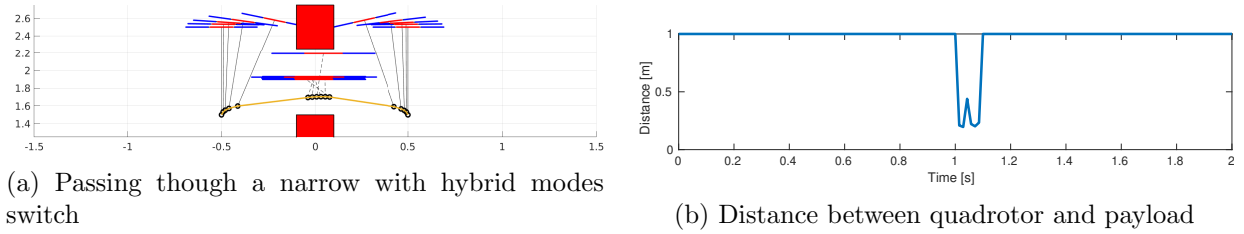


Figure 5.5: (a) Snapshots of passing through a window with hybrid modes. By tuning cost parameters, the optimization generates the hybrid modes to pass through the window. (b) the cable length l showing the two hybrid modes, $l = 1$ for taut and $l < 1$ for slack mode.

load attitude angle α represents the angle between cable attitude and vertical direction,

$$\cos(\alpha) = [0, 0, -1]^T \cdot q.$$

Notice from the figure, the final trajectory is always taut.

Mode-Guided Window Task An alternative method to generate trajectories to pass through narrow windows is through mode sequencing. However, instead of explicitly specifying the modes at each collocation node, the cost \mathcal{R}_0 is adjusted to force the optimization to generate slack/taut modes. Having a lower \mathcal{R}_0 at the initial nodes, then a larger value followed by a smaller value leads the optimization to generate motions where the cable will be taut, slack, and taut when passing through the narrow window. Figure 5.5 presents the snapshots of maneuvering through a narrow window. Notice that, instead of an aggressive swing in Figure 5.4b, the quadrotor moves down to decrease the distance between itself and the suspended payload to avoid obstacles, where the cable becomes slack.

5.5 Experimental Results

A custom build quadrotor with a Raspberry Pi 3 with Navio-2 autopilot is used to test the trajectories generated using our planner. A ROS node on Raspberry Pi 3 runs the onboard

Table 5.1: Parameters for planning experiments for a quadrotor with suspended payload.

Description	Notation	Value
mass of the quadrotor	m_Q	0.825Kg
mass of the payload	m_L	0.065Kg
length of cable	l_0	1.097m
load attitude angle limitation	α_{max}	$\pi/2$
quadrotor dimension	$2\epsilon_x/2\epsilon_y$	0.255/0.255m
safety distance	d_{min}	5cm

attitude control at $500Hz$. The ground station consists of a MacBook Pro running Ubuntu 18.04 and ROS melodic. A ROS node runs the position control at $200Hz$ on the ground control and communicates with the onboard control through WiFi. A 3D-printed load is suspended using a string from the quadrotor. Parameters for the hardware setup are listed in Table 5.1. Optitrack motion capture system is used to estimate the pose, velocity of the quadrotor, and load. An Inertial Measurement Unit (IMU) on the Navio2 is used to estimate the body attitude and body-rates.

The optimal trajectories generated using the techniques described in Sections 5.3 & 5.4 are implemented in the experiments using the following control architecture. The optimal trajectory outputs the load trajectory and its derivatives $x_L^{(i)}$, and the corresponding quadrotor trajectory is computed through differential flatness. The control loop is closed around this trajectory, i.e., we implement trajectory tracking control on the quadrotor. We use open source solver IPOPT [91] in Matlab with modeling language Yalmip [92] for path planning optimization. In each experiment, we first generated an offline trajectory on a laptop, and this is sent to the ground control using a Python interface.

Finally, experimental snapshots for some scenarios are shown in Figure 5.6¹. The experiments illustrate obstacle avoidance with and without waypoint guidance. In the video, we also show more complex cases, such as generating an s-shaped trajectory to avoid wall-shaped obstacles in a closed space.

Computation Time: To verify the computation time advantage of our approach, we time the path planning optimization for each scenario. A MacBook Pro with Intel Core i7 (CPU 2.6GHz base speed) running Ubuntu 18.04 was used for offline path planning computation. For each scenario mentioned in Section 5.4, the computation time is measured 10 times. We use the safe margin distance constraints in (5.19) for obstacle avoidance formulation. The results are shown in Figure 5.7. The algorithm outperforms state-of-the-art approaches and is over 10 times faster in hover-to-hover with no obstacles, payload throwing, and waypoint navigation, where it generally takes less than 1s with our algorithm. The time metrics when using constraints with safety margin distance in (5.19) or with minimum-penetration

¹A video of the experiments can be viewed at https://youtu.be/e09RZ0x_nZk

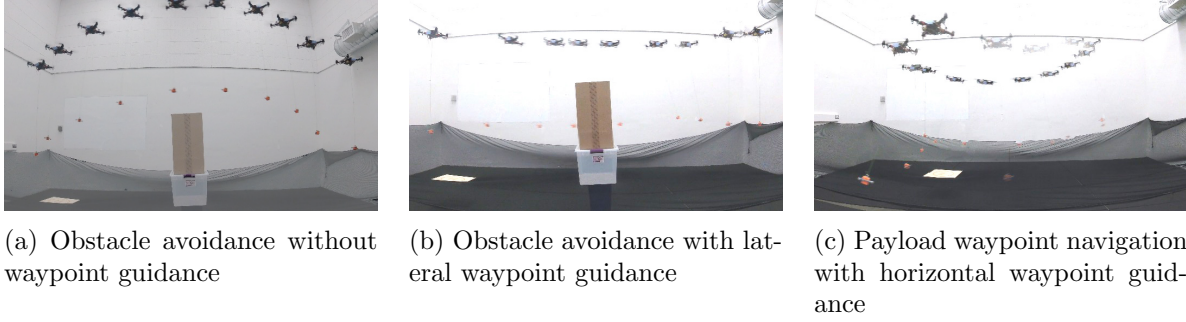


Figure 5.6: Experimental snapshots of the quadrotor-payload system executing successful obstacle avoidance and waypoint navigation maneuvers.

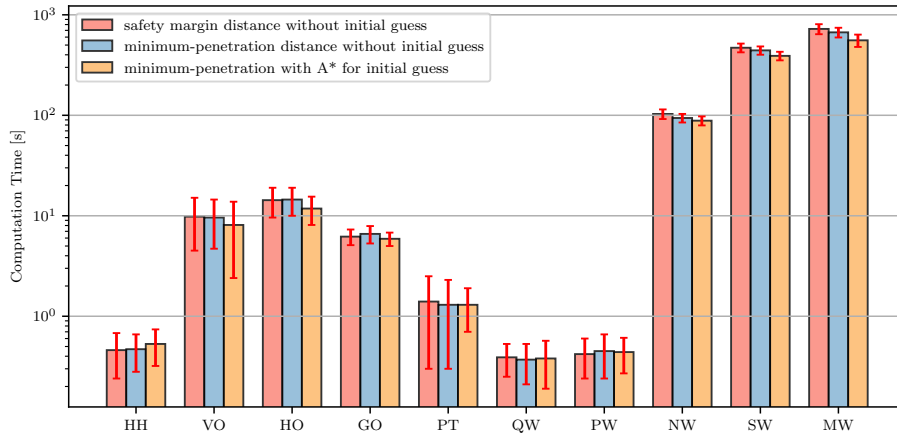


Figure 5.7: Computation time for various planning tasks with different constraint formulations. Computation time decreases with relaxed constraints and better initial guesses.

distance in (5.20) for different tasks mentioned in Section 5.4 are presented in the figure. The dual infeasibility tolerance for solver IPOPT needs to be higher for state-guided and window-guided for window tasks. The standard deviation of computation time comes from fluctuations in CPU performance. Also, while the computation time difference might not seem significant, note the y-axis of the plot is in log-scale. In Section 5.3, we noted that the minimum-penetration formulation in (5.20) could be used as obstacle avoidance constraints, which makes the problem smoother. In this case, the computation time is further reduced about 10-20% when the obstacles are crowded.

The various planning tasks along with (the number of nodes) are, HH- hover to hover (50), VO- Vertical obstacle avoidance (50), HO- Horizontal obstacle avoidance (50), GO- Guided obstacle avoidance (50), PT- Payload throwing (50), QW- Triangle flight for quadrotor waypoints (100), PW- Triangle flight for payload waypoints (100), NW- guided for window task (50), SW- State-guided for window task (30), MW- Mode-guided for window task (30).

Initial Guess

In the results presented, no initial guess is provided to the optimization solver. The optimization formulation in (5.11) is a non-convex optimization problem and hence computationally challenging to solve in general. For nonlinear optimization, the optimality of the solution depends on the initial guesses provided to the solvers, and that different initial guesses can lead to different (local) optima. Computing a good initial guess is often difficult and highly problem dependent; ideally, the initial guess should be obstacle-free and approximately satisfy the system dynamics. A* algorithm can be used to get initial guesses for x_L and its high order derivatives with collocation constraints in (5.11b). The initial guess of cable tension T and load attitude q could be calculated from the load acceleration and the distance $l(t)$ between the quadrotor and the payload. The initial guess of penetration distance s is assumed to be the obstacle's biggest dimension. The results with the minimum-penetration method and A* as the initial guess are also listed in Figure 5.7.

5.6 Results and Discussion

Advantages The optimization approach generates energy-efficient, dynamically feasible trajectories, which satisfy the constraints of obstacle avoidance, waypoint navigation, and hybrid mode transitions. The algorithm could handle almost all the cases for the system containing a quadrotor with a suspended payload without excessive cost tuning. However, specifying the initial waypoints will result in lower computation times. The problem formulation exploits the differential-flatness property with complementarity constraints for representing either slack or taut for the cable. As the full dynamics are not used in the collocation, and instead differential flatness is formulated as the dynamics, this reduces the number of nonlinear constraints in the optimization problem and the computation time of motion planning is far more optimized compared to prior work.

Parameter tuning The proposed planner generates feasible and sub-optimal solutions without extensive parameter tuning. For example, the following costs $Q = \text{diag}([1, 1, 1])$, $\mathcal{R}_0 = 1$, $\mathcal{R}_1 = 1$, $\mathcal{R}_2 = 1$ were used for both obstacle avoidance and waypoint navigation tasks. For example, for the waypoint navigation case, if we increase the value of \mathcal{R}_0 , we could observe different triangular trajectories, presented in Figure 5.3a, where the payload trajectories become smoother.

Complexity and Robustness Assume we have m nodes for our optimization and n obstacles (assumed as cuboids, then matrix A in (5.18) becomes 6×3) need to be considered in the collision-free trajectory generation, then we have $(3 \times 7 + 1 + 3 + 1)n + (6 + 3)nm = 26n + 9mn$ variables to optimize, where $21n$ variables are for payload positions and their derivatives, $5n$ ones for load attitude, cable tension, quadrotor-load distance and $9mn$ ones for dual variables λ and μ in (5.19). The problem becomes challenging if the environment

# obstacles	1	2	3	4	5	6	7	8	9	10	11	12
mean [s]	5.7	17.0	23.5	72.1	115.2	120.5	153.1	165.2	218.7	252.3	292.1	342.9
std [s]	1.5	8.3	9.2	45.0	60.1	53.7	62.1	86.5	79.7	83.8	102.5	136.7
min [s]	3.5	7.2	16.8	25.6	33.2	61.9	88.5	82.4	141.4	136.6	158.0	191.2
max [s]	8.1	32.0	40.5	172.3	223.5	226.4	272.8	289.9	332.0	428.9	569.1	652.7
failing cases	0	0	0	0	0	0	1	1	0	2	5	10

Table 5.2: Complexity and robustness analysis for path planning algorithm in a crowded environment with obstacles.

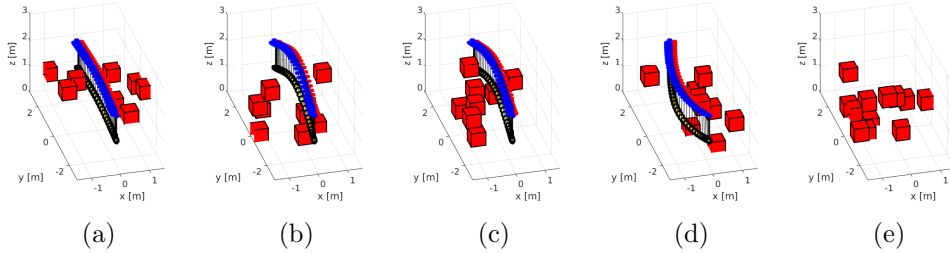


Figure 5.8: Complexity and robustness analysis of planning algorithm for navigation tasks in crowded environments. (a)-(d) show four successful trails with 8 random obstacles, where the difficulty of obstacle avoidance could vary depending on their locations. (e) planner fails for 12 randomly sampled obstacles due to reduced obstacle-free areas for any feasible maneuver.

becomes crowded with obstacles. We test the scalability of our planning algorithm with different numbers of obstacles. Table 5.2 and Figure 5.8 present the environment setup and scalability results.

The planning problem is set up with $[0, -2.5, 1.0]$, $[0, 2.5, 1.0]$ as payload initial and final positions, and no waypoints are predefined in the planner. In the closed space $\mathbb{S} = \{y = (y_1, y_2, y_3) : y_1 \in [-1.5, 1.5], y_2 \in [-1.5, 1.5], y_3 \in [0, 2]\}$, random cubes with dimensions 0.5m by 0.5 by 0.5m are generated as obstacles. Moreover, we limit the payload system movement in the closed space $\mathbb{W} = \{y = (y_1, y_2, y_3) : y_1 \in [-1.5, 1.5], y_2 \in [-3, 3], y_3 \in [0, 2]\}$, which forces the system to navigate between obstacles and ensure the initial and the final configurations of the system are collision-free. For each trial, we sample the designed number of obstacles and test the path planning algorithm, and we have 20 trials for each number of obstacles. Initial guesses with A^* are used, and minimum penetration for obstacle avoidance is applied for optimization formulation.

Limitations and Future Work One limitation of the proposed approach is that it is quite difficult to directly set physical limitations of quadrotor movement. For instance, to set the maximum thrust f or moment M of the quadrotor, a nonlinear constraint must be formulated with the differential-flatness property. Without physical limitations on the quadrotor, the generated trajectories might be too aggressive that are infeasible to implement on the hardware. A possible improvement would be to assign some additional upper or lower

bounds on the payload jerk, snap, or other high-order derivatives (linear constraints). Since the focus is on an optimization-based planning approach with collocation for the quadrotor-payload system, the complexity of the algorithm increases with the number of obstacles, and the algorithm may not be efficient when there are too many obstacles. Combining the sampling-based and optimization-based methods in future work could reduce the computation time in environments crowded with obstacles and keep the trajectories dynamically feasible.

5.7 Summary

In this chapter, the quadrotor with a suspended payload is parameterized using the differential-flatness property with complementarity constraints. With additional constraints from the robot, task, environment, and customized guidance, the path planning was formulated as a nonlinear optimization problem on collocation points. The proposed solutions for the trajectory generation problem outperform the state-of-the-art in the computation time. Several numerical simulations and experiments for tasks such as waypoint navigation, obstacle avoidance, and passing through a window are presented to validate the differential flatness and direct-collocation-based planner for a quadrotor with a suspended payload.

The last three chapters discuss methods for individual agents (quadrotor or quadrotor with a suspended payload). The next chapter presents control methods for collaborative grasping using quadrotors with suspended gripper.

Chapter 6

Cable-suspended Grasping and Control for Collaborative Aerial Transportation

This chapter explores the aspects of aerial manipulation, specific to grasping and transportation. Grasping and transportation are a significant share of aerial manipulation tasks. Using a suspended cable with a gripper reduces the weight and inertia associated with a manipulation arm and the additional mechanism but adds additional under-actuation to the system and requires control of the suspended payload. This chapter looks into grasping and transportation using cable-suspended grippers. Specifically, this chapter discusses the following three. *(i)* Grasping using a quadrotor with cable-suspended payload (gripper). Magnetic payloads are grasped using a cable-suspended electromagnet from quadrotor(s). Payload grasping makes use of payload control and the hybrid modes of the quadrotor with suspended payload, *(ii)* Centralized payload control for trajectory tracking of the payload suspended from multiple quadrotors and *(iii)* Finally, presents safety-critical offboard control to avoid collisions.

Remark 6.1. *The object suspended on the other end of the suspended cable from the quadrotor is referred to as payload, irrespective of whether it is a gripper (electromagnet or otherwise) or electromagnet grasping an object. The object grasped using the suspended gripper is referred to as the target payload .*

6.1 Introduction

Transportation using aerial vehicles addresses the last-mile delivery problem. Aerial transportation using unmanned aerial vehicles has two main steps, grasping or loading the aerial with the payload/package and transporting the payload to the destination. Most research in aerial transportation, especially cooperative aerial transportation using more than one vehicle, addresses the control and planning problem. Various centralized and decentralized

control approaches with and without communications have been proposed in the literature. However, the initial grasping operation is usually skipped with the assumption that the aerial vehicles are already equipped with the payload (usually through user assistance).

Most aerial grasping consists of an aerial vehicle equipped with a grasping mechanism directly attached to the vehicle (referred to as *direct-grasping* in the rest of the chapter) or using a manipulator arm to grasp the target objects (*mechanism-grasping*). Having the gripper directly attached to the vehicle frame solves the control and planning, as it only requires the control of the aerial vehicle. Direct grasping requires the vehicle to come close to the target and might even require the vehicle to land and take off. Coming close to the payload surface can also lead to ground-effect disturbances. The downwash from the propellers could be adversarial to the payload and its surrounding area.

In mechanism-grasping, aerial vehicles are typically equipped with an actuated manipulator arm, and is usually fully actuated and has controllability in all the required degrees of freedom. Having the additional mechanism enables inflight grasping operation and is ideal for manipulating the pose of the target object, including picking and placing over a reasonably large distance than typical manipulators. However, an actuated arm mechanism adds additional mass and inertia to the system and constrains the already limited (power/payload capacity) resources. In addition, grasping for transport, for instance, debris removal during search & rescue or payload transportation, the objective is typically to pick up and move the objects. Such object removal or payload transportation can be done without a fully actuated arm to pick up and transport.

Alternately, suspending just the grasping mechanism (*suspended-grasping*) using a cable either at a fixed length or via a pulley reduces the system's weight and enables inflight grasping with lesser added weight. This chapter presents control and experiment results for suspended grasping magnetic payloads using quadrotors with cable-suspended electromagnets.

Related Work

Various design and grasping mechanisms have been proposed over the years for aerial grasping and transportation. Light-weight and low-complexity grippers for grasping and perching of quadrotors were proposed in [93]. Such grippers were used for constructing 3D structures in [94, 95]. Other variations, such as vacuum suction [96] and soft grippers [97], were used as alternatives to push the boundaries of dynamic aerial grasping. All these grippers are mounted below the quadrotors and require the quadrotors to approach from above and come close to the payload to grasp. Using a manipulator-arm addresses these issues and enables inflight grasping. An actuated arm [98–100] can be used for dynamic aerial grasping such as the avian-inspired grasping [7, 101] while in motion. In [6], a model predictive control for pick and place using a quadrotor actuated arm was presented, or in the case of [102], the quadrotor is equipped with three actuated arms.

Agile load transportation can be achieved by suspending the payload while transporting [103]. Due to the suspended nature of the cable, such systems are under-actuated. Re-

searchers proposed various control methods such as treating the payload as disturbance and employed adaptive or robust disturbance rejection methods [104, 105], methods for swing attenuation [106–108] while transporting. Finally, active control of the suspended payload position [27, 30, 60, 109, 110]. While disturbance rejection and swing attenuation can minimize the swinging in the cable, the resulting control will not be ideal for grasping using a suspended gripper due to the lack of active control on the suspended gripper. Payload control methods have been successful in tracking the suspended payload, and some of the literature also experimentally validates the result. However, there exist minimal experimental results for suspended aerial grasping. A wire-suspended gripper with swing suppression is used for grasping in flight in [111]. Swing suppression adds undesired weight to the system, which can be removed through active control of the cable. This work uses active payload control for aerial grasp in flight.

6.2 Control for a Quadrotor with a Suspended Payload

A quadrotor with suspended *payload* is under-actuated, and its dynamics and hybrid modes are discussed in Section 2.5. Grasping a *target payload* requires generating a reference trajectory to *target payload* and tracking this reference. Reference trajectory for the *payload* is generated by exploiting the differential flatness (see Section 2.4) of the quadrotor-suspended-payload system, either by generating splines for the payload directly or by using trajectory generation methods on the flat-variables or full-system states as discussed in Chapter 5.

A *payload* tracking controller is required to track the trajectory to the *target payload*. Two payload controllers are discussed in this section, (i) geometric control defined in [27] is used for payload tracking with an emphasis on hardware implementation and (ii) an alternate tracking control, a variation-based MPC on the variation-based linearized dynamics of the cable-attitude dynamics. Both these approaches present control strategies for the payload-cable subsystem, shown in the dashed-box in Figure 6.1. The controllers through singular perturbation argument [112] assume a slow dynamics for payload-cable subsystem and a fast dynamics for the quadrotor attitude dynamics. Dynamics for the payload-cable subsystem Σ_t^L are described in (2.37) and reproduced below,

$$\dot{x}_L = v_L, \quad (6.1a)$$

$$(\dot{v}_L + g e_3) = \frac{(q^T F - m_Q l \dot{q}^T \dot{q})}{(m_Q + m_L)} q, \quad (6.1b)$$

$$\dot{q} = \omega \times q, \quad (6.1c)$$

$$m_Q l \dot{\omega} = -q \times F. \quad (6.1d)$$

The input to the subsystem Σ_t^L is thrust-vector F , and is tracked by the quadrotor onboard control. The quadrotor onboard control tracking a desired thrust vector F command is

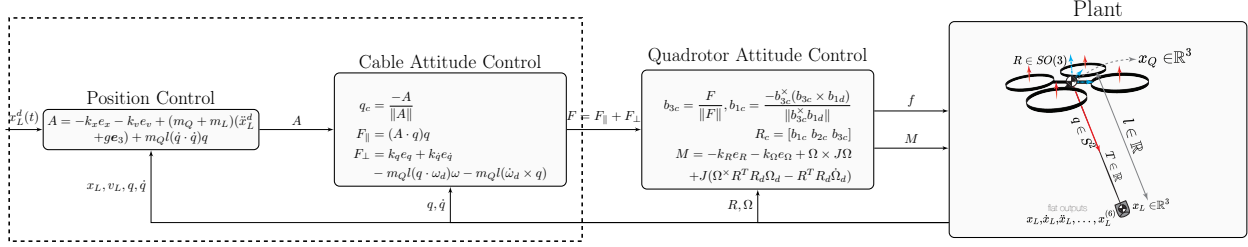


Figure 6.1: Control structure for payload tracking control. The control architecture corresponding to the payload-cable subsystem is shown in the dashed box.

given,

$$f = F \cdot R e_3, \quad (6.2)$$

$$M = -k_R e_R - k_\Omega e_\Omega + \Omega \times J \Omega + J(\Omega^\times R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d), \quad (6.3)$$

where e_R, e_Ω are defined in (2.16) and (2.17), tracking the command rotation matrix defined as,

$$R_c = [b_{1c} \quad b_{2c} \quad b_{3c}] \quad (6.4a)$$

$$b_{3c} = \frac{F}{\|F\|}, \quad b_{1c} = \frac{-b_{3c}^\times (b_{3c} \times b_{1d})}{\|b_{3c}^\times b_{1d}\|}, \quad b_{2c} = b_{3c} \times b_{1c}. \quad (6.4b)$$

A faster convergence rate for the inner attitude loop ensures that any command thrust vector F is tracked reasonably quickly enough to track the payload. The faster convergence of the inner loop is implemented in hardware through high loop rate. Thus, the control design for the payload-cable subsystem is to drive the tracking errors to zero $(e_x, e_v, e_q, e_{\dot{q}}) \rightarrow 0$.

Geometric Control

Geometric control for quadrotor with suspended payload described in [27] uses a cascaded approach for the *payload* control as shown in Figure 6.1. The outer-loop does the *payload* position control with the control input, A , given as,

$$A = -k_x e_x - k_v e_v + (m_Q + m_L)(\ddot{x}_L^d + g e_3) + m_Q l(\dot{q} \cdot \dot{q})q, \quad (6.5)$$

where $e_x = x_L - x_L^d$, $e_v = v_L - v_L^d$ are the position and velocity errors for the *payload*. The control input consists of the PD terms $-k_x e_x - k_v e_v$ as well the feedforward terms for the desired acceleration along with terms to cancel out the centrifugal forces. However, any force on the payload can act only along the cable, i.e., along $-q$ direction. The input force required now consists of two components, a parallel component where the component of A along $-q$ is used to track the payload position, and a perpendicular component orienting the cable in the desired direction A . The perpendicular component implements the cable

attitude control S^2 from [27] with the errors defined in (2.11), (2.12). The cable attitude control is given as,

$$q_c = \frac{-A}{\|A\|}, \quad (6.6a)$$

$$F_{\parallel} = (A \cdot q)q, \quad (6.6b)$$

$$F_{\perp} = k_q \tilde{e}_q + k_{\dot{q}} \tilde{e}_{\dot{q}} - m_Q l (q \cdot \omega_d) \omega - m_Q l (\dot{\omega}_d \times q), \quad (6.6c)$$

and the errors $\tilde{e}_q, \tilde{e}_{\dot{q}}$ are computed against the commanded attitude q_c . Finally, the total input force the payload-cable subsystem is given as,

$$F_{\text{GeoPD}} = F_{\parallel} + F_{\perp}, \quad (6.7)$$

F_{GeoPD} is the commanded input the onboard attitude control.

Discussion on gain tuning

The payload position control as shown in the previous section and in Figure 6.1 has three cascaded loops. Thus implementation of the control requires designing three sets of PD gains, k_x, k_v for the payload position, $k_q, k_{\dot{q}}$ for the cable attitude and finally, k_R, k_{Ω} for the quadrotor attitude control. From the propositions 1-3 in [27], with the conditions satisfied the controller is shown to exhibit almost global stability properties. However, hardware implementation of the controller is more restricted due to bounded inputs, restricted convergence rates due to hardware limitations etc. Thus, designing the control gains is more challenging, stiff gains in the outerloop leads to undesired oscillations in the system and relaxed gains will not track the desired trajectory.

For most quadrotor systems, the onboard gains are pre-tuned to satisfy the necessary system requirements of the quadrotor without any additional mechanism, (the suspended cable system in this work). Thus, a payload control system, requires tuning the gains for the payload-cable sub-system, $k_x, k_v, k_q, k_{\dot{q}}$. It is observed during hardware implementation, the choice $k_q, k_{\dot{q}}$ is more important for a stable control. The input defined in (6.6c) drives the tracking error $(\tilde{e}_q, \tilde{e}_{\dot{q}})$ to zero equilibrium for cable attitude system defined in (6.1c) and (6.1d). The choice of the feedback and feedforward terms follows from [19, Lemma 11.23]. It can also shown using the Lyapunov candidate defined in [113], given below,

$$\mathcal{V}_q = \frac{1}{2} m_Q l e_{\omega} \cdot e_{\omega} + k_q \Psi_q + c m_Q l e_q \cdot e_{\omega}. \quad (6.8)$$

The PD gains in (6.6c) are re-written as $k_q = \frac{1}{m_Q l} k'_q$ and $k_{\dot{q}} = \frac{1}{m_Q l} k'_{\dot{q}}$ and the scaled version gains $k'_q, k'_{\dot{q}}$ are tuned to achieved desired convergence rate for the cable-attitude control independent of the system parameters such as cable length and the payload mass.

Variation-based Model Predictive Control

In this section, an model-predictive control (MPC) approach for the payload-cable subsystem is presented, to exploit the advantages of MPC [114]. MPC follows a receding horizon approach considering the future behaviour of the system and is typically implemented as a constrained optimization problem. Behavior of the system is captured through the system dynamics, nonlinear or linear. Most systems, especially robotics systems are nonlinear in nature due to interaction between mulitple components or agents, for instance, cable attitude and quadrotor attitude for the quadrotor payload system. For nonlinear MPC, while using the full nonlinear system dynamics will better predict the future system behavior, but this is sometimes challenging to implement on hardware in real-time with limited computational resources. Linearized dynamics are first-order approximation of the nonlinear-systems and are sufficient for most hardware implementation.

Variation-based linearized dynamics for the systems evolving smooth manifolds is emperically observed to exhibit larger region-of-attraction. In this section for implementing the MPC, the payload-cable subsystem is variation-based linearized as described in Section 2.3. The variation-based linearized dynamics about a time-varying reference trajectory for the system in (6.1) results in the following linearized error dynamics,

$$\delta \dot{\mathbf{x}} = A_{\Sigma^{CL}} \delta \mathbf{x} + B_{\Sigma^{CL}} \delta \mathbf{u}, \quad (6.9)$$

$$C_{\Sigma^{CL}} \delta \mathbf{x} = 0, \quad (6.10)$$

with

$$\delta \mathbf{x} = [e_x^T \quad e_v^T \quad e_q^T \quad e_\omega^T]^T, \quad (6.11)$$

and the linear system given as,

$$A_{\Sigma^{CL}} = \begin{bmatrix} O & I & O & O \\ O & O & a_{23} & a_{24} \\ O & O & a_{33} & a_{34} \\ O & O & a_{43} & O \end{bmatrix}, B_{\Sigma^{CL}} = \begin{bmatrix} O \\ b_{21} \\ O \\ b_{41} \end{bmatrix}, \quad (6.12)$$

$$\begin{aligned} a_{23} &= \frac{-1}{m_Q + m_L} \left[(q_d^T F_d - m_Q l \dot{q}_d^T \dot{q}_d) I_3 + q_d F_d^T \right] q_d^\times, \\ a_{24} &= \frac{2m_Q l}{m_Q + m_L} q_d \dot{q}_d^T q_d^\times, a_{33} = q_d q_d^T \omega_d^\times, a_{34} = I - q_d q_d^T, a_{43} = \frac{-1}{m_Q l} F_d^\times q_d^\times \\ b_{21} &= \frac{1}{m_Q + m_L} q_d q_d^T, b_{41} = -\frac{1}{m_Q l} q_d^\times. \end{aligned}$$

Note, subscript d refers to the states of the desired reference trajectory. The reference states can be generated from the payload position trajectory x_L^d through differential flatness.

Finally, finite-time constrained optimal control formulation for the MPC with time-horizon N at time t is given below,

$$J_t^*(\delta \mathbf{x}_t) = \min_{u_{t:t+N-1|t}} \left((\delta \mathbf{x}_{t+N|t})^T P (\delta \mathbf{x}_{t+N|t}) + \sum_{i=0}^{N-1} [(\delta \mathbf{x}_{t+i|t})^T Q^x (\delta \mathbf{x}_{t+i|t}) + (\delta \mathbf{u}_{t+i|t})^T Q^u (\delta \mathbf{u}_{t+i|t})] \right), \quad (6.13a)$$

$$\text{s.t. } \delta \mathbf{x}_{t+i+1|t} = \mathbf{A}(t) \delta \mathbf{x}_{t+i|t} + \mathbf{B}(t) \delta \mathbf{u}_{t+i|t}, i = 0, 1, \dots, N-1, \quad (6.13b)$$

$$\delta \mathbf{x}_{t+i|t} \in \mathcal{X}, \delta \mathbf{u}_{t+i|t} \in \mathcal{U}, i = 0, 1, \dots, N-1, \quad (6.13c)$$

$$\delta \mathbf{x}_{t+N|t} \in \mathcal{X}_f. \quad (6.13d)$$

$$\delta \mathbf{x}_{t|t} = \delta \mathbf{x}_t, \quad (6.13e)$$

The description of the various variables used in the above formulation are as follows,

- $\delta \mathbf{x}_{t+i|t}$ is the $t+i$ predicted error of the payload-cable subsystem with the initial error computed using (6.11) starting at time t , $\delta \mathbf{x}_t$.
- $\delta \mathbf{u}_{t:t+N|t}$ is the input sequence to the error dynamics given in (6.13b) at time t
- $\mathbf{A}(t), \mathbf{B}(t)$ are the discrete-time error dynamics at time t for the payload-cable system. $\mathbf{A}(t), \mathbf{B}(t)$ are computed by taking the zero-order hold (ZOH) on the continuous-time variation-based linearized dynamics $A_{\Sigma CL}, B_{\Sigma CL}$ in (6.10) for time-step h . (such that, time at $t+1$ is time at t plus h).
- Q^x, Q^u are the state and input stage cost weight for the MPC and P is the terminal state cost weight. Choice of P should be such that the terminal cost $(\delta \mathbf{x}_{t+N|t})^T P (\delta \mathbf{x}_{t+N|t})$ is a Lyapunov function. In practice, this is obtained by computing the steady-state LQR gain for the linearized dynamics.
- The error state and input bounds in (6.13c) are computed by transforming the appropriate bounds about the current reference state. For instance, $\mathcal{U} = \{\delta \mathbf{u} \in \mathbb{R}^3 \mid u_{min} \leq \delta \mathbf{u} + u_d \leq u_{max}\}$.
- The terminal state constraint \mathcal{X}_f is ideally set to $0_{12,1}$ to drive the error to zero. However, reaching the terminal state in N time-steps doesn't always results in a feasible solution to the optimization problem on hardware. Thus, the terminal constraint in (6.13d) can be ignored, if the terminal cost gain P is appropriately selected.
- Finally, the initial error state $\delta \mathbf{x}_t$ in (6.13e) is computed using (6.11) about desired states obtained from the reference trajectory of the payload $x_L^d(t)$ through differential flatness.

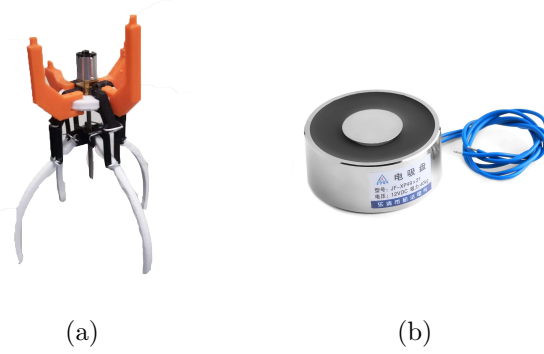


Figure 6.2: Types of grippers experimented for suspended-cable grasping (a) a screw-based gripper (b) electromagnet for grasping magnetic payloads

The optimal solution to the cost function in (6.13a) at time t is the sequence of inputs given below,

$$\delta \mathbf{u}_{t:t+N-1|t}^* = \{\delta \mathbf{u}_{t|t}^*, \delta \mathbf{u}_{t+1|t}^*, \dots, \delta \mathbf{u}_{t+N-1|t}^*\}. \quad (6.14)$$

The first element in the input sequence is considered as the control law for the linearized dynamics and is used for the payload-tracking. The input to the payload-cable subsystem for trajectory tracking at time t is thus computed as,

$$F_{\text{VMPC}} = u_t^d + \delta \mathbf{u}_{t|t}^*. \quad (6.15)$$

Similar to the geometric control in the previous section, the F_{VMPC} is the command input to the quadrotor onboard-control.

Remark 6.2. *The variation model predictive control in this section implements the MPC on the variation-linearized dynamics i.e., on the error dynamics of the system, unlike, traditional way of directly implementing on the system states.*

6.3 Cable-suspended Aerial Grasping

This section discusses aerial grasping using cable suspended gripper. The right design choice of gripper is paramount for any grasping operation. For cable suspended gripper, the number of DOF for the gripper is constrained. Since the gripper is suspended from the cable, unlike an aerial manipulator arm, the necessary counter balancing wrenches for grasping cannot be supported by the quadrotor. As the cable can apply only a pulling force on the gripper, quadrotor with suspended gripper can only capture objects in the plane perpendicular to the suspended cable (say *grasping plane*). For most practical applications, grasping in flight using quadrotor with suspended cable happens at close to hovering situations. The gripper design should be such that grasping is symmetric in the *grasping plane*. Various differences between a cable suspended gripper and manipulator arm based grasping are presented in Table 6.1

Table 6.1: Pros and cons of a cable suspended gripper *vs* an aerial manipulator arm

Cable suspended grasping	Aerial manipulator grasping
<ol style="list-style-type: none"> 1. Reduced inertia and mass 2. Decoupled dynamics between the gripper and the quadrotor 3. Restricted degrees-of-freedom due for grasping to the suspended nature of the cable 4. Grasping is only possible in the <i>grasping plane</i> 5. Requires a specialized control design due to under-actuation in the cable 	<ol style="list-style-type: none"> 1. Higher inertia and mass due to the added manipulator 2. Coupled dynamics between the quadrotor and manipulator arm 3. Degrees of freedom depends on the manipulator arm 4. Grasping along any pose is possible based on the DOF of the manipulator arm 5. Requires a specialized control design to account for the wrenches in the manipulator arm

Gripper

The design of a gripper in itself is a huge research question that requires extensive study and analysis. For the purpose of this dissertation, where the focus is on cable-suspended grasping, the gripper design is kept simple with minimum requirements. Design choices for the gripper are a lightweight, simple design with one or zero actuation mechanisms, no additional sensors (to reduce weight), and quick grasping action.

A couple of gripper designs (see Figure 6.2) were experimented as part of the aerial grasping using suspended gripper. A custom-design gripper using a screw mechanism as shown in Figure 6.2a is considered. A single screw is used to actuate the equidistant 3D printed arms. However, the screw nature of the mechanisms makes the grasping too slow and the dimension of the grasped object is restricted to the gripper size. Also, due to the nature of the gripper and the arm, the grasping capacity is limited.

Alternately, an electromagnet can be used for grasping the payload but requires the payloads to be magnetic. Electromagnets are simpler in design, easy to maintain, and provide larger payload capacity over the 3D printed gripper. An electromagnet requires complete surface contact with the target object for successful grasping. Any cable-suspended gripper requires a tracking control to reach the desired *target payload* location. Tracking controls for a quadrotor with suspended cable discussed in the previous section can be used to reach the target location. Due to their simple nature and larger capacity, electromagnets are used for aerial grasping in this chapter.

Modes for suspended aerial grasping

As mentioned, an electromagnet as a gripper requires a full surface with the target magnetic object for a successful grasp. A cable-suspended gripper in flight always has a pulling tension

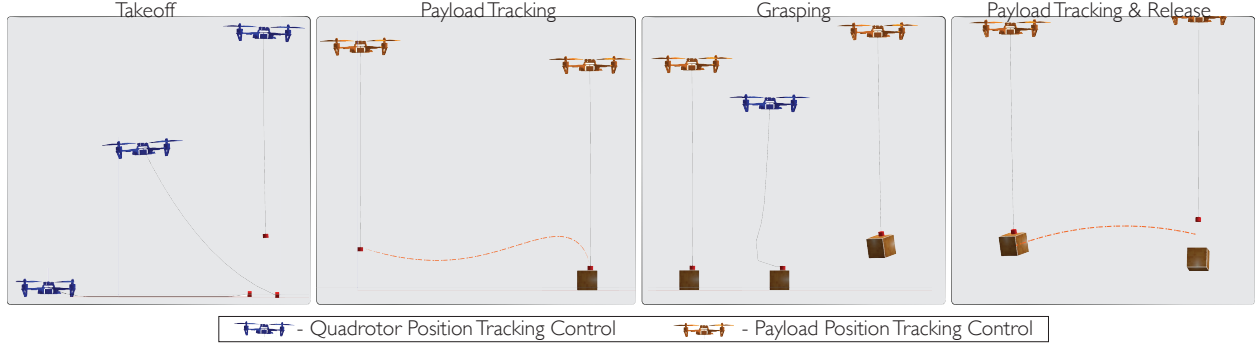


Figure 6.3: Various modes during grasping and transporting a target object using a quadrotor with cable suspended gripper.

$T \geq 0$, in the cable; thus, even with accurate position control cannot guarantee a successful surface contact. A proper contact requires a downward force on the gripper. In addition, in hardware, due to noise in the system, the position control is not entirely precise and has a non-zero mean error in the position control.

The hybrid nature of the quadrotor with suspended *payload* is exploited for successful grasping using an electromagnet. Section 2.5 discusses the hybrid modes of the quadrotor with suspended payload based on cable slackness. When the cable is slack, the quadrotor and the *payload* are fully decoupled into independent entities, with the *payload* as a free-falling object. Thus, in slack mode, the gravity on the *payload* (electromagnetic-gripper) provides the necessary downward force required for successful grasping.

Different modes for grasping and transporting a payload using cable suspended gripper are shown in Figure 6.3. During takeoff mode, system is in quadrotor position tracking control mode, with the takeoff height for the quadrotor set to be cable length plus desired height from the ground level. While takeoff is treated as a single mode, takeoff for quadrotor with suspended payload can also be further divided into various stages, for instance, [115] models the takeoff into multi-stages.

After takeoff, the system switches to a payload position tracking mode and the control strategies presented in Section 6.2 are used for the payload position control. After reaching the *target payload* position using the tracking mode, the setpoint of the *payload* is changed to a point below the *target payload*. This ensures, that the cable goes slack after the *payload* (electromagnet) makes the contact with the *target payload* surface. When the cable is slack the payload position control is no longer valid, as it was defined for the taut dynamics Σ_t (2.34). In additional, the estimation of the cable attitude and its angular-velocities will be off and can lead to bad outcomes. Following hybrid control approach is proposed to address cable slackness during grasping or even in flight.

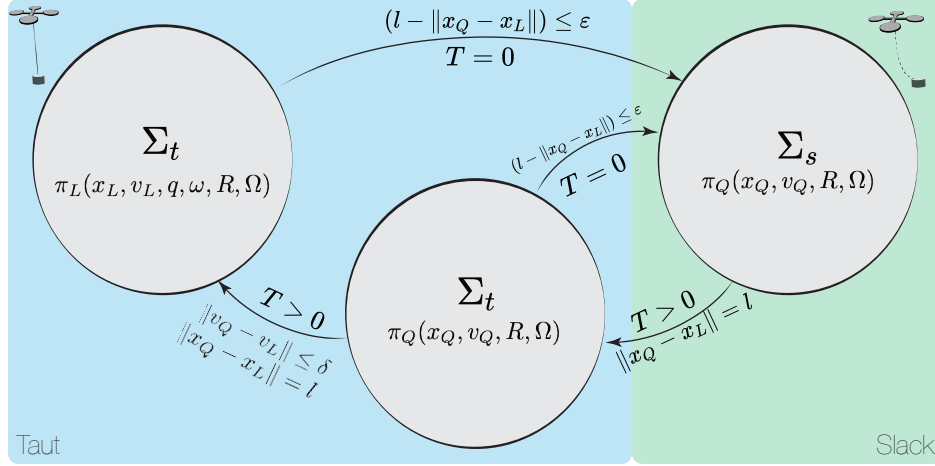


Figure 6.4: Hybrid control approach for the quadrotor with suspended *payload* during aerial grasping. Additional (Σ_t, π_Q) mode is introduced to reduce the impact during slack-to-taut transition.

Hybrid control for quadrotor with suspended payload

The dynamics for the two hybrid modes of the quadrotor with suspended *payload* are encoded as Σ_t (taut) and Σ_s (slack). Let the quadrotor position control, with the initial conditions satisfied, from [32] be denoted as $\pi_Q(x_Q, v_Q, R, \Omega)$. The controller depends on the full state of the quadrotor and is almost globally exponentially stable. Similarly, let $\pi_L(x_L, v_L, q, \omega, R, \Omega)$ be the payload position control from [27], with the respective initial conditions satisfied. Therefore, the closed loop systems (Σ_t, π_L) , (Σ_s^Q, π_Q) has global stability properties, where Σ_s^Q is the quadrotor subsystem in slack mode.

The hybrid control approach proposed in Figure 6.4 is used for aerial grasping. When the cable is fully taut, *payload* position control is used to track the payload position. When the cable goes slack quadrotor position control with the desired quadrotor trajectory is computed from differential flatness. From the slack mode, when the cable becomes taut, while the dynamics of the system change, control policy still uses the quadrotor position control π_Q and only switches to π_L when the error between the relative velocities of the quadrotor and *payload* are sufficiently small.

While the full mathematical proof for the hybrid control is not presented here, the intuition behind the approach is as follows. Even though the closed-loop systems (Σ_t, π_L) , (Σ_s^Q, π_Q) are independently stable, it doesn't guarantee the full hybrid model is stable. The issue with the stability arises during the transitions. The transition from taut to slack $\Sigma_t \rightarrow \Sigma_s$ is defined by the identity map in [27]. i.e., the initial states in the slack mode are same as final states of the taut mode. Thus, quadrotor control in the slack mode is still stable, while payload is free-falling independent entity and has no bearing on the quadrotor.

Transition from the slack to taut is modeled as an inelastic collision ensuring zero relative velocity between the quadrotor and the *payload*. However, in the real-world, the collisions are not truly fully inelastic, especially when cable/strings were used for suspending the

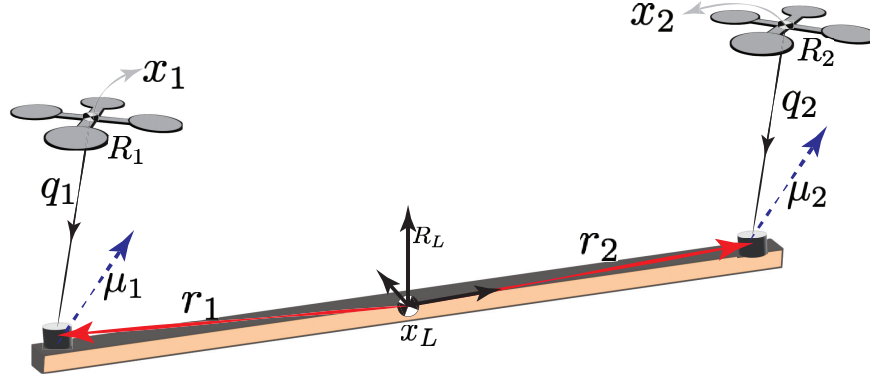


Figure 6.5: Two quadrotors with a common linear payload suspended from cables.

payload. Thus, the imperfect inelastic collision can result in non-zero relative velocities and can cause the system to switch to slack mode. Using payload control (as well as cable estimation) during this switching is not ideal and leads to instability. To avoid this, the switching to payload control π_L is restricted until the relative velocity difference is sufficiently small. During this mode, quadrotor position control is used, with bounded inputs (hardware constraints) and reasonable trajectories for the payload (no flips etc), the suspended payload can only apply a bounded force disturbance. While the (Σ_t, π_Q) is not an asymptotically stable system, for bounded disturbance it can still remain stable.

After grasping, the system is again in payload control mode to transport the grasped *target payload* to the destination and releasing it. Same grasping approach can now be used in multiple agents to grasp larger *target payloads*. In the next section, tracking control for payloads suspended from two quadrotors (that can be extended to more quadrotors) is discussed.

6.4 Control for Two Quadrotors with a Suspended Payload

In this section, a linear payload suspended using cables from two quadrotors is considered as shown in Figure 6.5. Each quadrotor has a cable-suspended gripper rigidly attached to the payload, for the linear rod considered in this section, they roughly on the each end of the payload. The force required to track the common payload is generated by the attached gripper and is treated as feed-forward to the position control in (6.5). The methodology presented in this section extends the controller presented in the previous section for single quadrotor with suspended payload. Alternate but similar approaches are presented in [116, 117]. Even though framework provided in the rest of the section assumes two quadrotors connected to the linear payload, the same can be extended to more than two quadrotors, numerical simulations for systems with three and four quadrotors are presented in later

sections.

System description and the states for the quadrotor and cable-suspended gripper follows from the previous section and from Section 2.5 with the subscript showing the system index (i.e., $(\cdot)_i, i = \{1, 2\}$). The quadrotor states are given as, x_i position, v_i velocity, R_i orientation and Ω_i for angular-velocities. Note, x_{Li}, v_{Li} refers to the gripper position velocities for each quadrotor. The suspended payload is a rigid body with the states index L , i.e., position of the payload in the inertial frame is $x_L \in \mathbb{R}^3$ and the payload attitude is $R_L \in SO(3)$ with their corresponding velocities given as v_L and Ω_L . The point of contact of the cable on the payload with the payload-frame R_L is r_i . Again, the forces on the rigid payload acts via the cable along $-q_i$ at the point of contacts.

Goal is to transport the payload along a desired reference trajectory in $SE(3)$, say $(x_L^d(t), R_L^d(t))$. Control input for the payload can be computed using the geometric control defined in [32]. The force and torque for the payload are given as,

$$F = -k_x e_x - k_v e_v + m_L(\ddot{x}_L^d + g e_3), \quad (6.16a)$$

$$M = -k_R e_{R_L} - k_\Omega e_{\Omega_L} + \Omega_L \times J_L \Omega_L + J_L(\Omega_L^\times R_L^T R_{Ld} \Omega_{Ld} - R_L^T R_{Ld} \dot{\Omega}_{Ld}), \quad (6.16b)$$

where J_L and m_L are the inertial and mass of the rigid payload with F is the force in the inertial frame and M is the torque in the payload frame. The required wrench is generated through the contact forces of the gripper. Since the gripper is suspended using cable it can only generate forces and not torques. Let the contact forces be denoted by μ_i in the payload frame and the grasp map for the payload is given as,

$$G = \begin{bmatrix} I & I & \dots & I \\ r_1^\times & r_2^\times & \dots & r_n^\times \end{bmatrix}, \quad (6.17)$$

i.e., the net force generated is $\sum_{i=1}^n \mu_i$ for n quadrotors and net torque is $\sum_{i=1}^n r_i^\times \mu_i$. Contact forces are computed using the following QP formulation,

$$\bar{\mu}^* = \min_{\mu_i} (G\bar{\mu} - W)^T (G\bar{\mu} - W) + \lambda \bar{\mu}^T \bar{\mu}, \quad (6.18a)$$

$$\text{s.t.} \quad \begin{bmatrix} -\mu_{max}^h \\ -\mu_{max}^h \\ 0 \end{bmatrix} \leq \mu_i \leq \begin{bmatrix} \mu_{max}^h \\ \mu_{max}^h \\ \mu_{max}^v \end{bmatrix}, i = 1, \dots, n \quad (6.18b)$$

where $\lambda \bar{\mu}^T \bar{\mu}$ is the regularization term and W is the payload wrench in the body-frame

$$W = \begin{bmatrix} R_L^T F \\ M \end{bmatrix}, \quad (6.19)$$

and μ_h^{max} and μ_{max}^v are the horizontal and vertical force bounds for the gripper and $\bar{\mu} = [\mu_1^T \ \mu_2^T \ \dots \ \mu_n^T]^T$. When the bounds on the contact forces are sufficiently large enough, such that that inequality constraint in (6.18b) is not active, the contact forces can be computed as,

$$\bar{\mu}^* = G^\dagger W, \quad (6.20)$$

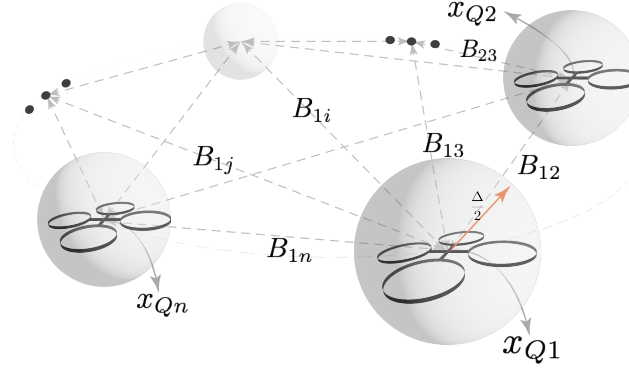


Figure 6.6: Position control barrier function illustration for multiple quadrotors

where G^\dagger is the pseudo-inverse of G and can be pre-computed and implemented on hardware without need for solving the optimization. Numerical simulations and experimental results implementing the control is shown in the later section.

6.5 Control Barrier Functions for Collision Avoidance

When working with dynamic agents like quadrotors in close proximity, such as when transporting or manipulating together, safety is paramount to the users and the system. This section briefly discuss how control barrier functions (CBF) [118] can be used to add a safety layer after the payload control. Note, that the added safety comes at the expense of stability and thus needs to be carefully calibrated and tuned, especially in hardware, for successful operations.

Control barrier function QP formulation is formulated on a point-mass double integrator dynamics for i^{th} quadrotor in the system,

$$\dot{x}_{Qi} = v_{Qi}, \quad (6.21)$$

$$m_{Qi}\dot{v}_{Qi} = F_i, \quad (6.22)$$

where F_i is the commanded input, computed using one of the control methods in the earlier sections, or just a PD control $\mu_i = -k_p e_{xi} - k_d e_{vi} + a_i^d + g e_3 + F_{ff}$ with F_{ff} feed-forward force for payload tracking. Distance between the quadrotors with a minimum radius is considered as the barrier function, and its implemented as a pair-wise constraint on all the quadrotors, see Figure 6.6.

Thus, for n quadrotors, consider the following control barrier function between quadrotor i and quadrotor j , $\forall i = \{1, \dots, n\}, j = \{i + 1, \dots, n\}$ and $i \neq j$ as,

$$B_{ij} := (x_i - x_j)^T W (x_i - x_j) - \Delta^2, \quad (6.23)$$

where Δ is the minimum safety distance between the quadrotors. Derivates of the barrier function are,

$$\dot{B}_{ij} = 2(x_i - x_j)^T W(v_i - v_j) \quad (6.24)$$

$$\ddot{B}_{ij} = 2(v_i - v_j)^T W(v_i - v_j) + 2(x_i - x_j)^T W(\alpha_i - \alpha_j), \quad (6.25)$$

and $\alpha_i = \frac{F_i}{m_{Q_i}}$ is the command acceleration.

Exponential barrier constraint [119] is thus given as,

$$\ddot{B}_{ij} \geq -k_1 B_{ij} - k_2 \dot{B}_{ij} \quad (6.26)$$

$$L_g L_f B_{ij} \mu_i - L_g L_f B_{ij} \mu_j \geq -k_1 B_{ij} - k_2 \dot{B}_{ij} - L_f^2 B_{ij}, \quad (6.27)$$

where $L_g L_f B_{ij} = 2(x_i - x_j)^T W$ and $L_f^2 B_{ij} = 2(v_i - v_j)^T W(v_i - v_j)$. Rewriting the above inequality for all $\forall i, j = \{1, \dots, n\}$ and $i \neq j$ and using the relation $B_{ij} \equiv B_{ji}$, gives,

$$\underbrace{\begin{bmatrix} L_g L_f B_{12} & -L_g L_f B_{12} & 0_{13} & \dots & 0_{13} \\ L_g L_f B_{13} & 0_{13} & -L_g L_f B_{13} & \dots & 0_{13} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{13} & L_g L_f B_{23} & -L_g L_f B_{23} & \dots & 0_{13} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{13} & 0_{13} & \dots & \dots & -L_g L_f B_{(n-1)n} \end{bmatrix}}_{:=\mathcal{A}} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \geq \underbrace{\begin{bmatrix} -k_1 B_{12} - k_2 \dot{B}_{12} - L_f^2 B_{12} \\ -k_1 B_{13} - k_2 \dot{B}_{13} - L_f^2 B_{13} \\ \vdots \\ -k_1 B_{23} - k_2 \dot{B}_{23} - L_f^2 B_{23} \\ \vdots \\ \left(-k_1 B_{(n-1)n} - k_2 \dot{B}_{(n-1)n} - L_f^2 B_{(n-1)n} \right) \end{bmatrix}}_{:=b} \quad (6.28)$$

Using the above inequality formulation, the centralized control barrier function quadratic program (CBF-QP) formulation is,

$$\min_{\bar{\alpha}} \quad (\bar{\alpha} - \alpha_{ref})^T H(\bar{\alpha} - \alpha_{ref}), \quad (6.29a)$$

$$\text{s.t.} \quad \mathcal{A}\bar{\alpha} \geq b, \quad (6.29b)$$

$$\bar{\alpha}_{min} \leq \bar{\alpha} \leq \bar{\alpha}_{max}, \quad (6.29c)$$

where $\bar{\alpha} = [\alpha_1^T \ \alpha_2^T \ \dots \ \alpha_n^T]^T$ and u_{ref} is the payload control input acceleration computed using the previously defined controllers.

6.6 Experimental Results

This section demonstrates the proposed control designs for payload grasping and tracking on an experimental platform. The experimental platform used for multiple quadrotors transporting the payloads is described below.

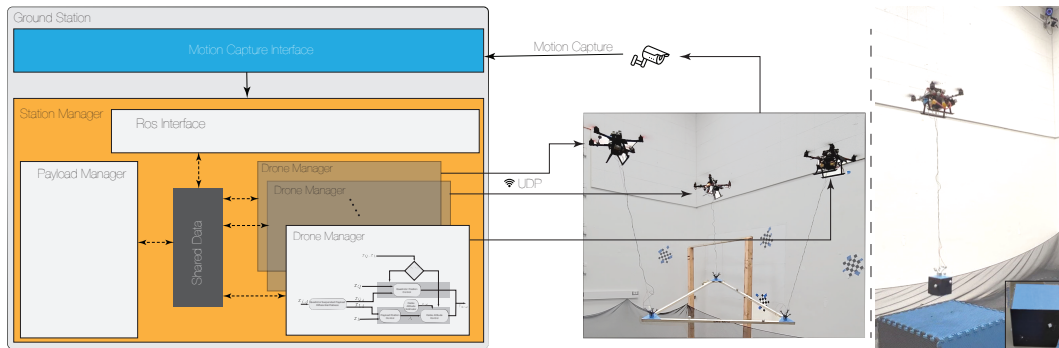


Figure 6.7: Hardware setup used for control of multiple quadrotors with payload

Hardware Setup

All the experiments presented here¹ were conducted in a $4m \times 5m \times 4m$ flight space, equipped with an OptiTrack motion capture system for tracking poses of the quadrotors and payloads. The hardware setup consists of a ground station system, custom-design quadrotors, payload and grippers, and a local wifi network for communications (see Figure 6.7).

- **Quadrotor:** All quadrotors have a similar setup, are custom built, and weigh about 750g. Each quadrotor is equipped with a Raspberry Pi² and Navio2³ autopilot shield for the raspberry pi. Raspberry pi runs a custom autopilot firmware and communicates with the ground station using mavros via UDP in the local wifi network. Firmware runs the onboard attitude estimation and control at 800Hz. Each quadrotor is assigned a unique id and port and has motion capture markers to track the pose.
- **Ground Station:** The ground station consists of a single C++ node, with multiple threads running individual drone-manager classes. Each drone manager class communicates and manages a single quadrotor via UDP and has a ROS layer to receive the motion capture data. The drone manager implements the offboard quadrotor/payload control at 200Hz and sends the command thrust to the corresponding quadrotor.
- **Electromagnet Gripper:** Each quadrotor has an (off-the-shelf available) electromagnet gripper suspended from the quadrotor via a string. The electromagnet is electronically connected to the Raspberry Pi/Navio2 flight control unit via a relay. Electromagnets have about 200N capacity and weighs about 110g.
- **Magnetic Payload:** Each *target payload* is equipped with $10cm \times 10cm$ steel metal plates. For single quadrotor experiments as $10cm \times 10cm \times 10cm$ 3D printed cube

¹A video showing experimental results and simulation code can be found at https://hybrid-robotics.berkeley.edu/collab_aero

²<https://www.raspberrypi.org/>

³<https://navio2.emlid.com/>

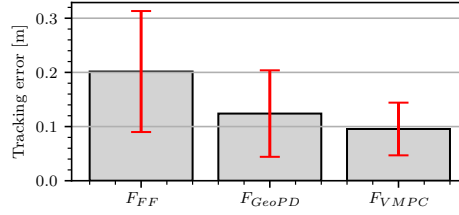


Figure 6.8: Payload position tracking error using three controls for the payload-cable subsystem, F_{ff} - a quadrotor position control with feed-forward forces for tracking the *payload* position, F_{GeoPD} - Payload position control using the geometric PD control on S^2 defined in (6.7), F_{VMPC} - Variation-based MPC on the payload-cable subsystem defined in (6.15) . Each controller is tuned to achieve best performance individually.

(weighing 150g)is used as the base for the metal plate. For multi-quadrotor experiments, metal plates are attached to the wooden sticks.

Quadrotor with suspended payload trajectory tracking

Payload position tracking control is implemented on a quadrotor with suspended payload (electro-magnet with tracking markers). Three different offboard controllers are tested (note, the onboard attitude control is the same for all), (i) F_{FF} : Quadrotor position control with desired quadrotor position and the feed-forward tension in the cable (for tracking the payload position) are computed using differential-flatness, (ii) F_{GeoPD} : Payload position control using the geometric PD control on S^2 defined in (6.7) and (iii) F_{VMPC} : Variation-based MPC on the payload-cable subsystem defined in (6.15).

Each controller is tuned to achieve the best result individually. Controllers are tested against different trajectories (setpoint, square, and circular trajectory). The resulting position tracking error plots are shown in Figure 6.8. The figure shows that the geometric PD and VMPC controllers have lower tracking errors, as expected, as the feedback loop is closed around the payload position. Having a small tracking error is ideal, as it can be used for grasping, as shown in the next section.

Grasping using a single quadrotor with suspended electromagnet

In this experiment, a quadrotor with a suspended electromagnet is used for grasping a stationary target magnetic-payload (described in the setup, 3D printed cube with a metal plate). Given the position for the *target payload*, payload trajectory is computed as,

$$x_L^d(t) = 0.5(p_0 + p_f) + 0.5(p_f - p_0) \sin(\pi(t/T_f) - (\pi/2)), \quad (6.30)$$

where p_0 is the starting position of the *payload* and p_f is the *target payload* position with additional buffer along the z-axis. Geometric PD control in (6.7) is used for tracking the

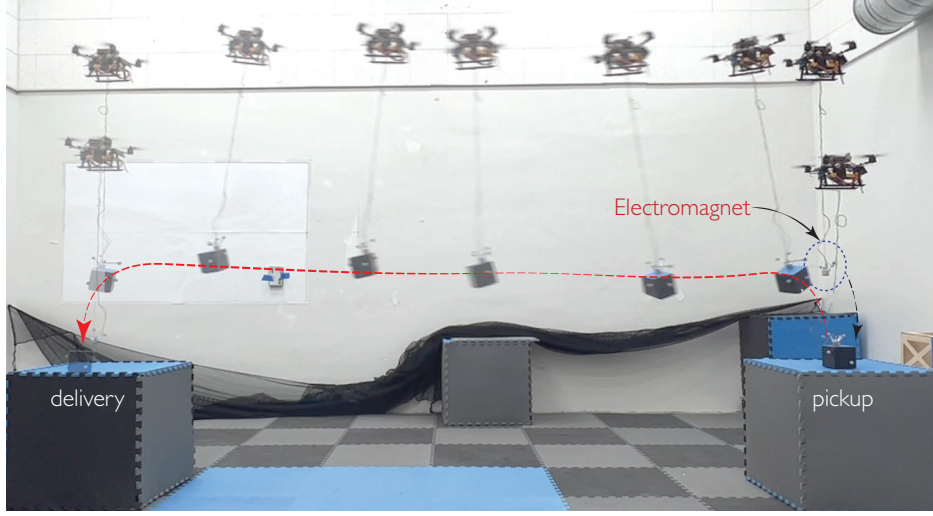


Figure 6.9: Grasping and transportation of the *target payload* using a quadrotor with cable suspended electromagnet gripper.

payload position and modes described in Section 6.3 is used for grasping the *target payload*. Figure 6.9 shows the experiment snapshots for pickup and delivery of the *target payload*.

Position tracking plots for *payload* and quadrotor during grasping and transportation for the single quadrotor with a suspended electromagnet are given in Figure 6.10. As shown in the figure, the two different control modes are shown in the blue and white background for *payload* and quadrotor position control, respectively. The *target payload* weight is not accounted for payload control during its transportation, and the same can be observed in the z-error in the figure after grasping (d), (e) stages. Figure 6.11 shows the distance between the quadrotor and *payload* position during the various stages. The cable is slack during grasping and release, as evident from the figure (as the distance is smaller than the nominal cable length). Switching the control to the quadrotor position in slack mode is crucial, as the payload control and estimation would be off when the cable is slack.

The repeatability of the suspended aerial grasping is tested by conducting multiple grasping experiments with F_{GeoPD} and without F_{FF} payload position control on two different quadrotors. The success rate for the grasping is reported in Figure 6.12. Using payload position control successfully grasped about 80% of the time, while just using quadrotor control has only 20% success. Figure 6.9 reported a mean tracking error of $0.1m$ for F_{GeoPD} , and payload has a surface of $0.1m \times 0.1m$, thus, this tracks with the observation of 80% success rate.

Grasping using two quadrotors

The grasping method defined in this chapter and experimentally validated in the previous section is used for grasping a $1.82m(6ft)$ wooden plank (with metal plates on either end). Grasping is achieved sequentially, where the first quadrotor (on the left) is user-requested to

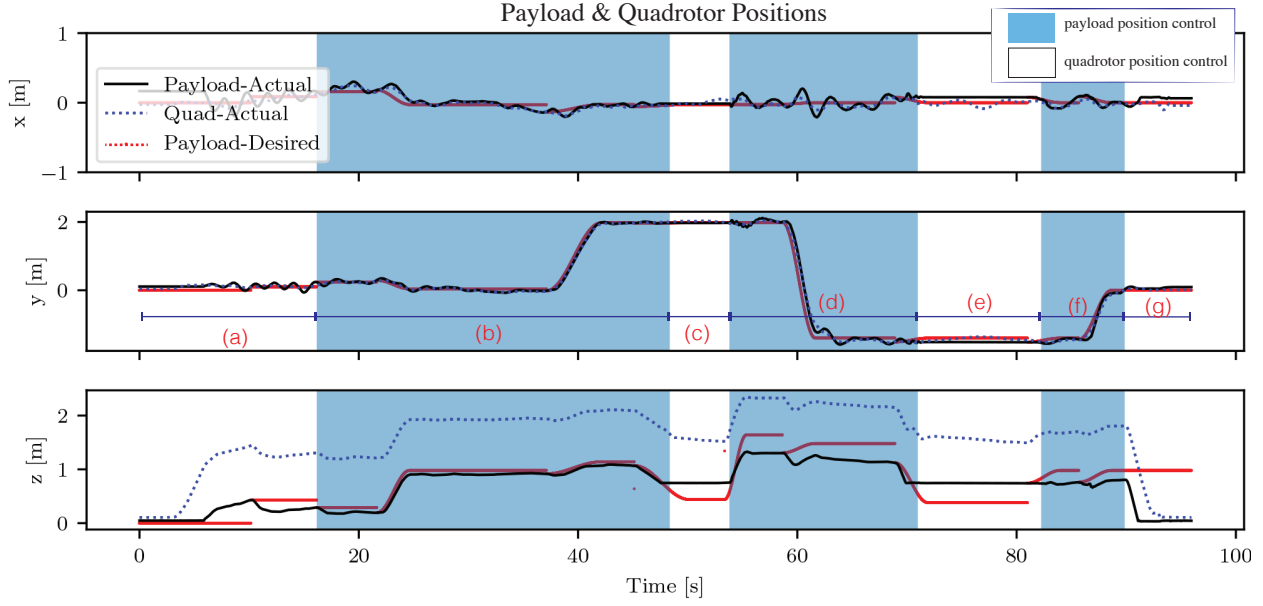


Figure 6.10: Payload and quadrotor position during suspended aerial grasping. (i) quadrotor position control (white background) during ((a) takeoff, (c) grasping, (e) release and (g) landing), and (ii) *payload* position control (blue background) during (b), (f) *payload* tracking and (d) *target payload* transportation.

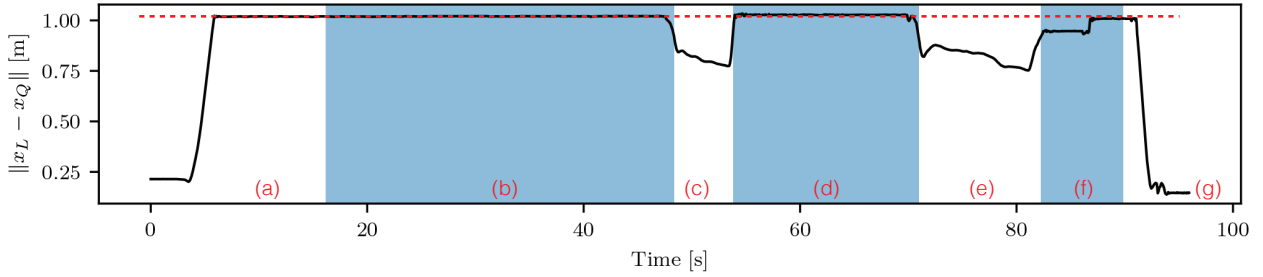


Figure 6.11: Distance between quadrotor and *payload* during the various stages of (a) takeoff, (b) payload tracking, (c) grasping, (d) transporting, (e) release and (g) landing.

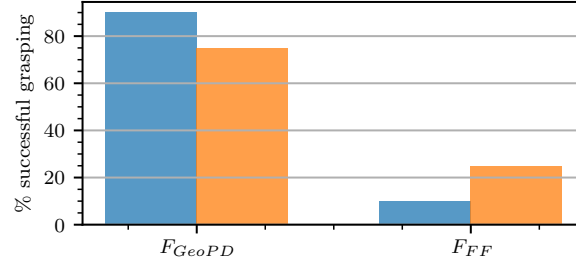


Figure 6.12: Percentage of successful grasping with and without *payload* position tracking (tested on two different quadrotors with atleast 10 attempts)

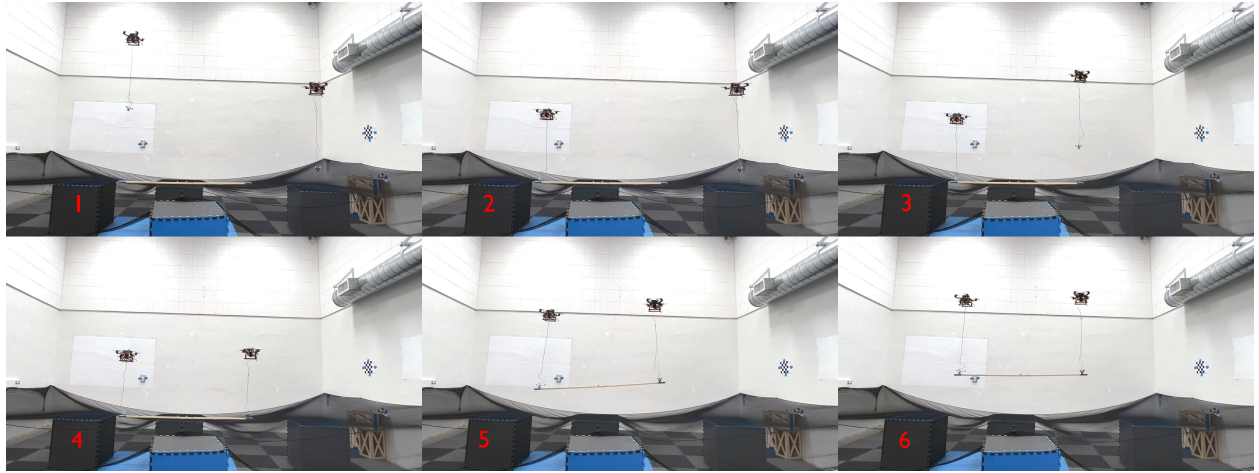
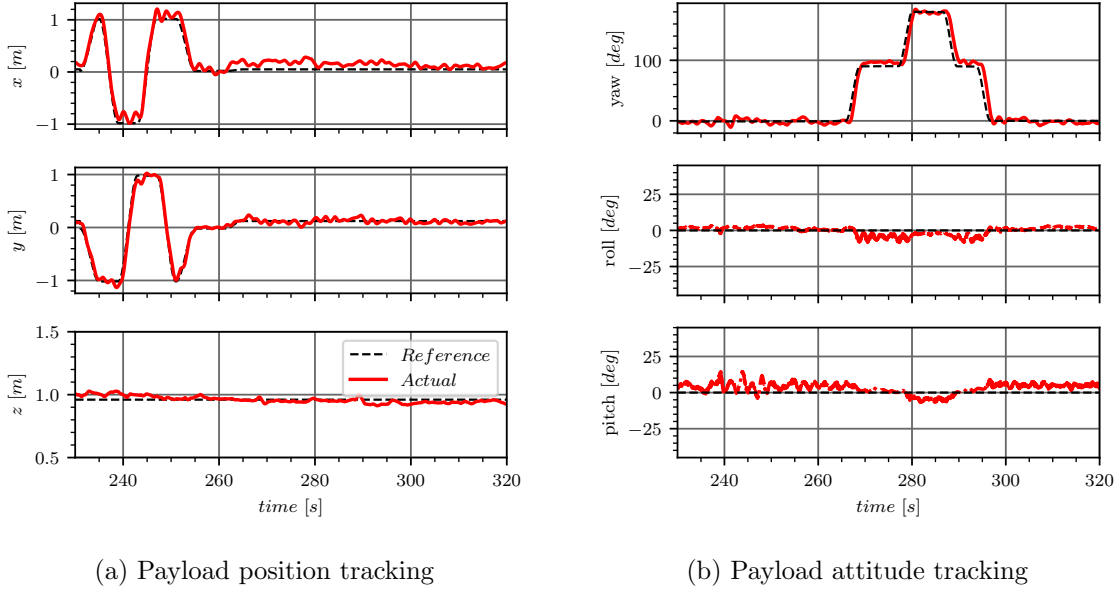


Figure 6.13: Suspended cable grasping of a linear payload (wooden plank with metal plates on either end) using two quadrotors.

grasp the metal plate of the stationary wooden plank. After the first quadrotor successfully grasps, the second quadrotor on the right is requested to grasp the payload. After both quadrotors successfully grasp the payload, both quadrotors are requested to lift the payload. Figure 6.13 shows the various shapshots of two quadrotors grasping the wooden plank. The next section presents experiment results for payload tracking for two quadrotors with a suspended wooden plank.

Payload tracking control suspended from two quadrotors

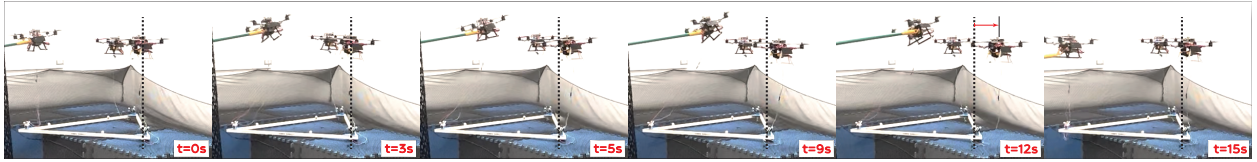
Here, the centralized control defined in Section 6.4 for payload tracking is experimentally validated on a linear (wooden plank) payload suspended from two quadrotors. A square trajectory of side $1m$ is used as the reference trajectory, with each side defined by the spline equation in (6.30). Note, the piece-wise square trajectory is only up to \mathcal{C}^1 continuous



(a) Payload position tracking

(b) Payload attitude tracking

Figure 6.14: Tracking plots for linear (wooden plank) payload suspended using cables from two quadrotors.


 Figure 6.15: Snapshots of the collision avoidance experiment, implementing a centralized CBF-QP on the quadrotor positions. At $t=12s$, shows the right quadrotor moving away when the left quadrotor is manually pushed towards it.

and discontinuity in the higher terms, i.e., the states generated using differential flatness have discrete jumps at each corner. This discontinuity causes the payload to jerk during trajectory tracking. Position tracking plots for the payload are shown in Figure 6.14a and attitude tracking in Figure 6.14b. As observed from the plot, the proposed control is able to successfully track the payload position and attitude. The payload tracking control can be used for transporting heavier loads using multiple quadrotors.

Position control barrier function for multiple quadrotors

A centralized quadratic program with position control-barrier-function constraints is implemented to avoid quadrotor collisions with each other. Three quadrotors with a cable-suspended triangular payload are considered for this experiment. Payload tracking using quadrotor position control with feed-forward input for the payload tracking is considered

along with the exponential barrier constraints defined in Section 6.5. Experiment snapshots showing the collision avoidance, when one of the quadrotors is manually pushed towards another, are shown in Figure 6.15. The piece-wise square trajectory tracking plot for three quadrotors with suspended triangular payload with safety control is shown in Figure 6.15. Note that the tracking is less accurate than the earlier two-quadrotor case because of the quadrotor position control and does not have any feedback on the payload position.

6.7 Summary

This chapter presents the framework for grasping using a quadrotor with cable-suspended electromagnetic grippers. Grasping using a suspended gripper requires closing the control loop around the gripper position to account for the under actuation in the system. This chapter presents control methods for tracking the *payload* position suspended from a quadrotor. It also discusses in detail the various stages involved in grasping and the hybrid control approach used to address the issue of the hybrid modes in the system. Especially since grasping using an electromagnet requires the cable to go slack. The chapter also presents how multiple quadrotor gripper systems can be used to grasp and transport larger payloads. Finally, the chapter discusses how the control barrier functions can be implemented as a safety layer to avoid collisions between the quadrotors. The next chapter discusses a different aerial manipulation system; specifically, quadrotors tethered in series using a single flexible cable.

Chapter 7

Geometric Modeling for Flexible Cable Suspended from Multiple Quadrotors

In this chapter, a different form of collaborative aerial manipulation system, a series of quadrotors connected using a single flexible cable with the end either tethered or free. Such a configuration is similar to a serial manipulator and is referred to as a *serial-aerial-manipulator*. A serial aerial manipulator can be used in applications where the quadrotors are used for manipulation over larger horizontal distances with the cable supplying the power/material for manipulation and sensing etc., or even to carry a fire-hose for active fire-fighting. Thus, it is crucial to understand the behavior of the complex dynamical system and observe how the cable swings affect the stability of the quadrotors or to plan and generate trajectories to avoid entanglements. In this chapter, the dynamics of *multiple quadrotors carrying a flexible cable* are modeled using coordinate-free representation using S^2 and $SO(3)$ manifolds.

7.1 Introduction

One extension of the payload-carrying research is developing multi-rotor vehicles for active fire-fighting [120, 121] using a tethered hose that carries water and power. This enables carrying a fire hose to heights higher than a typical fire-truck ladder and fly longer due to the tethered power supply. Multirotors are also used to help string power cables between poles [122], which typically is achieved using manned helicopters. To achieve stable and safe control of these complex systems, it is important to understand the underlying governing principles and dynamics. This chapter aims to model and control the dynamics of a multiple quadrotor system carrying a flexible cable/hose.

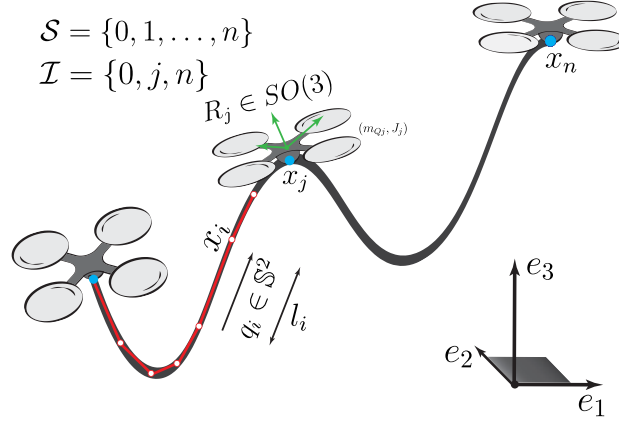


Figure 7.1: Multiple quadrotors carrying a flexible cable (with the cable modeled as a series of n discrete-links). Links are massless with lumped mass at the ends and indexed through $\mathcal{S} = \{0, 1, \dots, n\}$. The set $\mathcal{I} \subseteq \mathcal{S}$ gives the set of indices where the cable is attached to a quadrotor. Each link is modeled as a unit-vector $q_i \in \mathbb{S}^2$. The configuration space of this system is $Q := \mathbb{R}^3 \times (\mathbb{S}^2)^n \times (SO(3))^{n_Q}$ ($n_Q = |\mathcal{I}|$).

Related Work

Existing literature on co-operative aerial manipulation focuses on grasping and transporting payloads using multiple quadrotors [123], [9], [124], [125], [126]. Trajectory tracking control for point-mass/rigid-body payloads suspended from multiple quadrotors is studied in [57], [127], [29], [116]. Similarly, for loads suspended using flexible cables, stabilizing controllers are presented in [128], [129], and these systems are shown to be differentially-flat in [28]. Tethered aerial vehicles have also been extensively studied in the literature; for instance, stabilization of tethered quadrotors and nonlinear-observers for the same are discussed in [130], [131], [132]. Geometric control of a tethered quadrotor with a flexible tether is presented in [133].

Most of the work discussed in the previous section models the tethers/cables either as rigid links or as a series of links. Partial differential equations have also been used to model a continuous mass system, such as the aerial refueling cable shown in [134]. However, modeling the aerial cable as a finite-segment lumped mass [135], [136] is quite common in the literature due to the finite dimensionality of the state-space. Most of these works assume Euler angles in the local frame to represent the attitude of the links. This results in complex equations of motion for the system that are also prone to singularities in case of aggressive motions. Therefore, in this work, we make use of coordinate-free representation that results in singularity-free and compact equations of motion.

Challenges

Multiple quadrotors carrying a flexible cable has multiple challenges in modeling the dynamics as well as designing a controller. Even though modeling the cable as a finite-segment lumped mass results in a finite-dimensional state space, it would still result in a large number of states depending on the choice of the number of discrete links. In addition, developing a controller is challenging due to the high under-actuation in the system. The swing of the cable, when not accounted for in control, can have an adversarial effect.

Collaborative aerial manipulation involves dynamic interactions between various agents and their shared manipulation objects. Modeling these systems is important to understand their behavior and be able to design and control multi-agent manipulation systems. Using multiple quadrotors can be used for carrying a flexible payload, such as tethers; for instance, an externally powered system can be used for extended flights or grasping, etc. In this chapter, dynamics modeling for a quadrotor carrying a flexible hose is presented, and the system is shown to be a differentially-flat system. Variation-based linearized dynamics are presented, and the linearized dynamics are used to compute the LQR gain. Towards the end, an example system where two quadrotors powered using a single external tether is discussed.

Remark 7.1. *In this chapter, the terms cable and hose are synonymously used to refer to flexible payload suspended from multiple quadrotors. A rigidly fixed/connected cable on one end is referred to as tether.*

7.2 Dynamics

Consider a flexible cable connected to multiple quadrotor UAVs as shown in Figure 7.1. In this section, we present the coordinate-free dynamics for this system. We consider the following assumptions to derive the dynamics:

- (i) Cable is modeled as a series of n smaller links connected by spherical joints;
- (ii) Each link is massless with lumped point-masses at the end and the cable mechanical properties like stiffness and torsional forces ignored.
- (iii) The quadrotors attach to the cable at their respective center-of-masses.

In the following section, we present the notation used to describe the system.

Notation

Let the cable be discretized into n links with the cable joints indexed as $\mathcal{S} = \{0, 1, \dots, n\}$ as shown in Figure 7.1. The position of one (without loss of generality, say starting) end of the cable is given as $x_0 \in \mathbb{R}^3$ in the world-frame. The position of the link joints/point-masses is represented by $x_i \in \mathbb{R}^3$, where the link attitude between x_{i-1} and x_i is given by $q_i \in \mathbb{S}^2$ and

Table 7.1: List of various symbols used for modeling *multiple quadrotors carrying a flexible cable*.
 Note: $k \in \mathcal{S}, i \in \mathcal{S} \setminus \{0\}, j \in \mathcal{I}$, WF - World frame, BF-Body-frame, $|\cdot|$ represents cardinality of a set.

Variables	Definition
$n \in \mathbb{R}^+$	Number of links in the cable.
$\mathcal{S} = \{0, 1, \dots, n\}$	Set containing indices of the cable-segments.
$x_k \in \mathbb{R}^3$	Position of the k^{th} point-mass of the cable in WF.
$v_k \in \mathbb{R}^3$	Velocity of the k^{th} point-mass of the cable in WF.
$l_i \in \mathbb{R}^+$	Length of the i^{th} segment.
$m_k \in \mathbb{R}^+$	Mass of the k^{th} point-mass in the cable-segments.
$q_i \in \mathbb{S}^2$	Orientation of the i^{th} cable segment in WF.
$\omega_i \in T_{q_i} \mathbb{S}^2$	Angular velocity of the i^{th} cable segment in WF.
$\mathcal{I} \subset \mathcal{S}$	Set of indices where the cable is attached to the quadrotor.
$ \mathcal{I} = n_Q$	Number of quadrotors.
$x_{Qj} \equiv x_j$	Center-of-mass position of the j^{th} quadrotor in WF.
$R_j \in SO(3)$	Attitude of the j^{th} quadrotor w.r.t. WF.
$\Omega_j \in T_{R_j} SO(3)$	Angular velocity of the j^{th} quadrotor in BF.
m_{Qj}, J_j	Mass & inertia of the j^{th} quadrotor.
$f_j \in \mathbb{R}, M_j \in \mathbb{R}^3$	Thrust and moment of the j^{th} quadrotor in BF.
$\mathbf{1}_i := \mathbf{1}_{\mathcal{I}}(i) = \begin{cases} 1 & \text{if } i \in \mathcal{I} \\ 0 & \text{else} \end{cases}$	Indicator function for the set \mathcal{I} .
$\bar{m}_k = m_k + m_{Qk} \mathbf{1}_k$	Net mass at the k^{th} link joint.
$u_k = (-\bar{m}_k g \mathbf{e}_3 + f_k R_k \mathbf{e}_3 \mathbf{1}_k)$	Net force due to thrusters & gravity.

length of this link-segment is l_i i.e., $x_i = x_{i-1} + l_i q_i$. Also, m_i is the mass of the lumped point-mass for link i . Let the set $\mathcal{I} \subseteq \mathcal{S}$ be the set of indices where the cable is attached to the quadrotor and $n_Q = |\mathcal{I}|$ is the number of quadrotors. For the j^{th} quadrotor, $R_j \in SO(3)$ is the attitude, m_{Qj}, J_j is its mass and inertia matrix (in its body-frame) and $f_j \in \mathbb{R}, M_j \in \mathbb{R}^3$ are the corresponding thrust and moment for all $j \in \mathcal{I}$. Finally, the configuration space of this system is given as $Q := \mathbb{R}^3 \times (\mathbb{S}^2)^n \times (SO(3))^{n_Q}$. Table 7.1 lists the various symbols used in this chapter.

Derivation

The kinematic relation between the different link positions is given using link attitudes as,

$$x_i = x_0 + \sum_{k=1}^i l_k q_k, \quad \forall i \in \mathcal{S} \setminus \{0\}, \quad (7.1)$$

$$v_i = v_0 + \sum_{k=1}^i l_k \dot{q}_k, \quad (7.2)$$

$$a_i = a_0 + \sum_{k=1}^i l_k \ddot{q}_k. \quad (7.3)$$

Potential energy $\mathcal{U} : TQ \rightarrow \mathbb{R}$ of the system, due to cable and quadrotors' mass is,

$$\mathcal{U} = \sum_{i \in \mathcal{S}} \bar{m}_i x_i \cdot g \mathbf{e}_3, \quad (7.4)$$

where $\bar{m}_i = m_i + m_{Q_i} \mathbf{1}_i$ is the net-mass at index i and $\mathbf{1}_i := \mathbf{1}_{\mathcal{I}}(i) = \begin{cases} 1, & \text{if } i \in \mathcal{I} \\ 0, & \text{else} \end{cases}$ is an indicator function for the set \mathcal{I} . Kinetic energy $\mathcal{T} : TQ \rightarrow \mathbb{R}$ is similarly given as,

$$\mathcal{T} = \sum_{i \in \mathcal{S}} \frac{1}{2} \bar{m}_i \langle v_i, v_i \rangle + \sum_{j \in \mathcal{I}} \frac{1}{2} \langle \Omega_j, J_j \Omega_j \rangle, \quad (7.5)$$

where Ω_j is the angular velocity of the quadrotor j in its body-frame. Dynamics of the system are derived using the Lagrangian method, where Lagrangian $\mathcal{L} : TQ \rightarrow \mathbb{R}$, is given as, $\mathcal{L} = \mathcal{T} - \mathcal{U}$. We derive the equations of motion using the Langrange-d'Alembert principle of least action, given below,

$$\delta \int_{t_0}^{t_f} \mathcal{L} dt + \int_{t_0}^{t_f} \delta W_e dt = 0, \quad (7.6)$$

where δW_e is the infinitesimal work done by the external forces. δW_e can be computed as,

$$\delta W_e = \sum_{j \in \mathcal{I}} \left(\langle W_{1,j}, \hat{M}_j \rangle + \langle W_{2,j}, f_j R_j \mathbf{e}_3 \rangle \right), \quad (7.7)$$

$$W_{1,j} = R_j^T \delta R_j, \quad (7.8)$$

$$W_{2,j} = \delta x_j = \delta x_0 + \sum_{k=1}^j l_k \delta q_k, \quad (7.9)$$

are variational vector fields [128] corresponding to quadrotor attitudes and positions. The infinitesimal variations on q and R are expressed as,

$$\begin{aligned} \delta q &= \xi^\times q = -q^\times \xi, \quad \xi \in \mathbb{R}^3 \text{ s.t. } \xi \cdot q = 0, \\ \delta \dot{q} &= -q^\times \dot{\xi} - \dot{q}^\times \xi, \\ \delta R &= R \eta^\times, \quad \delta \Omega^\times = (\Omega^\times \eta)^\times + \dot{\eta}^\times, \quad \eta \in \mathbb{R}^3, \end{aligned}$$

with the constraints $q \cdot \dot{q} = 0$ and $q \cdot \omega = 0$, ω is the angular velocity of q , s.t. $\dot{q} = \omega \times q$. Similarly, variations on the link positions are given as,

$$\delta x_i = \delta x_0 + \sum_{k=1}^i l_k \delta q_k = \delta x_0 - \sum_{k=1}^i l_k q_k^\times \xi_k, \quad (7.10)$$

$$\delta v_i = \delta v_0 + \sum_{k=1}^i l_k \delta \dot{q}_k = \delta v_0 - \sum_{k=1}^i l_k (\dot{q}_k^\times \xi_k + \dot{q}_k^\times \xi_k). \quad (7.11)$$

Finally, the equations of motion for the system are computed by solving (7.6). See Section B.2 for the detailed derivation. Equations of motion for the *multiple quadrotors carrying a flexible cable* are given in (7.13)-(7.15). Note the mass-matrix $\mathbb{M}_{\{q_i\}}$ is a function of link attitudes $\{q_i\} = \{q_1, q_2, \dots, q_n\}$ and we use the following notation similar to [137],

$$\begin{aligned} M_{00} &= \sum_{k=0}^n \bar{m}_k, \quad M_{0i} = l_i \sum_{k=i}^n \bar{m}_k, \\ M_{i0} &= M_{0i}, \quad M_{ij} = \sum_{k=\max\{ij\}}^n \bar{m}_k l_i l_j. \end{aligned} \quad (7.12)$$

Remark 7.2. In (7.14), note the use of f_i, R_i for $i \notin \mathcal{I}$, (since $i \notin \mathcal{I}$ implies no quadrotor is attached at index i and thus cannot have f_i and R_i). Since $i \notin \mathcal{I} \implies \mathbf{1}_i = 0$ and thus $f_i R_i \mathbf{e}_3 \mathbf{1}_i = 0$, there by ensuring the right inputs to the system.

Remark 7.3. Degrees of freedom for the multiple quadrotors carrying a flexible cable is $DOF = 3(n_Q + 1) + 2n$ where $2n$ corresponds to the link attitudes DOF , $3n_Q$ the rotational DOF of the quadrotors and 3 for the initial position x_0 . Similarly, the degrees of actuation is $DOA = 4n_Q$ corresponding to the 4 inputs for each quadrotor. Thus, the degrees of under-actuation are $DOuA = 2n + 3 - n_Q$. For a typical setup we have $n \gg n_Q$, i.e., system is highly under-actuated.

Remark 7.4. For a tethered system, we can assume $x_0 \equiv 0 \forall t$, i.e. without loss of generality the system is tethered to origin of the inertial frame, and derive the dynamics as earlier. Equations of motion for this system would be same as (7.13)-(7.15), without the equation corresponding to \dot{v}_0 .

7.3 Differential Flatness

In the previous section, we derived the dynamics for *multiple quadrotors carrying a flexible cable*. The system is under-actuated and thus the control of the system is challenging. In this section, we show that this system is differentially-flat. For the system defined in this work, the quadrotor-flexible cable segments are connected in series. Unlike previous work where each quadrotor has only one segment connected to it, each quadrotor in this system can have 0, 1, or 2 segments connected to it.

In the following, we formalize the differential-flatness for certain configurations of the *multiple quadrotors carrying a flexible cable* system.

Lemma 7.1. $\mathcal{Y} = (x_0, \psi_j, T_{k+1}) \forall j \in \mathcal{I} \ \& \ k \in \mathcal{I} \setminus \{n\}$ are the set of flat-outputs for multiple quadrotors carrying a flexible cable with $n \in \mathcal{I}$ (i.e., end of the cable is always attached to a quadrotor as shown in Figure 7.2), where $x_0 \in \mathbb{R}^3$ is the position of the start of the cable, $\psi_j \in \mathbb{R}$ is the yaw angle of the j^{th} quadrotor and $T_{k+1} \in \mathbb{R}^3$ is the tension vector in the $(k+1)^{th}$ link (as shown in Figure 7.2).

Equations of motion for multiple quadrotors carrying a flexible cable

$$\dot{x}_0 = v_0, \quad \dot{q}_i = \omega_i \times q_i, \quad (7.13)$$

$$\begin{bmatrix} M_{00}I_3 & -q_1^\times M_{01} & -q_2^\times M_{02} & \dots & -q_n^\times M_{0n} \\ -M_{10}q_1^\times & -M_{11}I_3 & M_{12}q_1^\times q_2^\times & \dots & M_{1n}q_1^\times q_n^\times \\ -M_{20}q_2^\times & M_{21}q_2^\times q_1^\times & -M_{22}I_3 & \dots & M_{2n}q_2^\times q_n^\times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -M_{n0}q_n^\times & M_{n1}q_n^\times q_1^\times & M_{n2}q_n^\times q_2^\times & \dots & -M_{nn}I_3 \end{bmatrix} \begin{bmatrix} \dot{v}_0 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \\ \vdots \\ \dot{\omega}_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n M_{0i} \|\omega_i\|^2 q_i + \sum_{k=0}^n u_k \\ -\sum_{k=1}^n (M_{1k} \|\omega_k\|^2 q_1^\times q_k) - l_1 q_1^\times \sum_{k=1}^n u_k \\ -\sum_{k=1}^n (M_{2k} \|\omega_k\|^2 q_2^\times q_k) - l_2 q_2^\times \sum_{k=2}^n u_k \\ \vdots \\ -\sum_{k=1}^n (M_{nk} \|\omega_k\|^2 q_n^\times q_k) - l_n q_n^\times u_n \end{bmatrix}, \quad (7.14)$$

$$\dot{R}_j = R_j \Omega_j^\times, \quad J_j \dot{\Omega}_j = M_j - \Omega_j \times J_j \Omega_j, \quad (7.15)$$

$$\forall i \in \mathcal{S} \setminus \{0\}, \quad j \in \mathcal{I}, \quad u_i = (-\bar{m}_i g \mathbf{e}_3 + f_i R_i \mathbf{e}_3 \mathbf{1}_i).$$

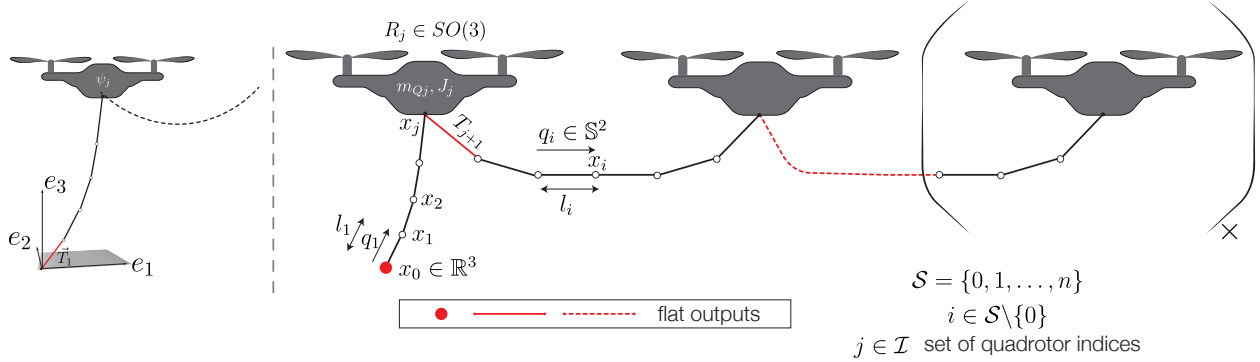


Figure 7.2: Configuration of the *multiple quadrotors carrying a flexible cable* illustrating the differential-flatness flat-outputs (shown in red)

Proof. See Appendix A.2. □

Remark 7.5. To determine the states and inputs of the system with n links, we require $(2n+4)$ derivatives of the flat-output x_0 , 2^{nd} derivative of the yaw angle ψ_j and $2(n-k)+2$ derivatives of the tension vector T_{k+1} .

Corollary 7.1. $\mathcal{Y} = (T_1, \psi_j, T_{k+1}) \forall j \in \mathcal{I} \text{ \& } k \in \mathcal{I} \setminus \{n\}$ are the flat-outputs for a tethered multiple quadrotors carrying a flexible cable shown in Figure 7.2, where $T_1 \in \mathbb{R}^3$ is the tension in the 1st link, $\psi_j \in \mathbb{R}$ is the yaw angle of the quadrotor at index j and $T_{k+1} \in \mathbb{R}^3$ is the tension vector in the $(k+1)^{th}$ link.

Proof. See Appendix A.2. □

Differential-flatness is used in planning the system trajectories, where the flat outputs are used to plan in the lower-dimension space and the corresponding desired states and inputs are computed using differential flatness. In the next section, linearized dynamics about a desired time-varying trajectory are derived and used towards an LQR to track desired trajectories.

7.4 Control

As presented in the Remark 7.3, the given system is highly underactuated and thus controlling the system is challenging. This section presents a way to control the system by linearizing the dynamics in (7.13)-(7.15) about a desired time-varying trajectory. States and inputs of the desired trajectories are represented with a subscript- d

$$(x_{0d}(t), v_{0d}(t), q_{id}(t), \omega_{id}(t), R_{jd}(t), \Omega_{jd}(t)) \forall i \in \mathcal{S}, j \in \mathcal{I} \quad (7.16)$$

Variation-based Linearization

In this sub-section, we present the coordinate-free linear dynamics, obtained through variation-based linearization of the nonlinear dynamics in (7.13)-(7.15). We use the variation linearization techniques described in Section 2.3 and [24] to obtain the linear dynamics. The error state of the linear-dynamics is given as,

$$\delta \mathbf{x} = [\delta x, \xi_1, \dots, \xi_n, \delta v, \delta \omega_1, \dots, \delta \omega_n, \eta_{j_1}, \dots, \eta_{j_{n_Q}}, \delta \Omega_{j_1}, \dots, \delta \Omega_{j_{n_Q}}]^\top, \quad (7.17)$$

and the corresponding inputs as,

$$\delta \mathbf{u} = [\delta f_{j_1}, \delta f_{j_2}, \dots, \delta f_{j_{n_Q}}, \delta M_{j_1}^\top, \delta M_{j_2}^\top, \dots, \delta M_{j_{n_Q}}^\top]^\top, \quad (7.18)$$

where j_1, j_2, \dots, j_{n_Q} are elements of \mathcal{I} arranged in increasing order. The individual elements of the error state are computed as,

$$\begin{aligned} \delta x &= x - x_d, \quad \delta v = v - v_d, \\ \xi_i &= q_{id}^\times q_i, \quad \delta \omega_i = \omega_i - (-(q_i^\times)^2) \omega_{id}, \\ \eta_j &= \frac{1}{2} (R_{jd}^T R_j - R_j^T R_{jd})^\vee, \quad \delta \Omega_j = \Omega_j - R_j^T R_{jd} \Omega_{jd}. \end{aligned}$$

Finally, the linearized dynamics (see Appendix B.3 for detailed derivation of the linearized dynamics) about a time-varying desired trajectory are given below,

$$\delta \dot{\mathbf{x}} = \mathcal{A} \delta \mathbf{x} + \mathcal{B} \delta \mathbf{u}, \quad (7.19)$$

$$\mathcal{C} \delta \mathbf{x} = 0. \quad (7.20)$$

The linear dynamics matrices \mathcal{A}, \mathcal{B} are,

$$\mathcal{A} = \begin{bmatrix} 0_{3,3} & 0_{3,3n} & I_{3,3} & 0_{3,3n} & 0_{3,3n_Q} & 0_{3,3n_Q} \\ 0_{3n,3} & \alpha & 0_{3,3} & \beta & 0_{3,3n_Q} & 0_{3,3n_Q} \\ & & & \mathbb{M}_{\{q_{id}\}}^{-1} F & & \\ 0_{3n_Q,3} & 0_{3n_Q,3n} & 0_{3n_Q,3} & 0_{3n_Q,3n} & \gamma & I_{3n_Q,3n_Q} \\ 0_{3n_Q,3} & 0_{3n_Q,3n} & 0_{3n_Q,3} & 0_{3n_Q,3n} & 0_{3n_Q,3n_Q} & \nu \end{bmatrix}, \quad (7.21)$$

with,

$$\begin{aligned} F &= \begin{bmatrix} O_{3,3} & [a]_i & O_{3,3} & [b]_i & [e]_j & O_{3n_Q,3n_Q} \\ O_{3n,3} & [c]_{i,j} & O_{3n,3} & [d]_{i,j} & [f]_{i,j} & O_{3n_Q,3n_Q} \end{bmatrix}, \\ \alpha &= b \text{diag}[q_{1d} q_{1d}^\top \omega_{1d}^\times, q_{2d} q_{2d}^\top \omega_{2d}^\times, \dots, q_{nd} q_{nd}^\top \omega_{nd}^\times], \\ \beta &= b \text{diag}[(I_3 - q_{1d} q_{1d}^\top), (I_3 - q_{2d} q_{2d}^\top), \dots, (I_3 - q_{nd} q_{nd}^\top)], \\ \gamma &= b \text{diag}[-\Omega_{j_1 d}^\times, -\Omega_{j_2 d}^\times, \dots, -\Omega_{j_{n_Q} d}^\times], \\ \nu &= b \text{diag}[J_1^{-1}((J_1 \Omega_{1d})^\times - \Omega_{1d}^\times J_1), \dots, J_n^{-1}((J_n \Omega_{nd})^\times - \Omega_{nd}^\times J_n)], \end{aligned}$$

and,

$$\mathcal{B} = \begin{bmatrix} O_{3(n+1),4n_Q} \\ \mathbb{M}_{\{q_{id}\}}^{-1} G \\ O_{3n_Q,4n_Q} \\ [O_{3n_Q,n_Q} \quad \mu] \end{bmatrix}, \text{ with } \mu = bdiag[J_{j1}^{-1}, \dots, J_{jn_Q}^{-1}], \quad (7.22)$$

$$G = \begin{bmatrix} [g]_i \\ [h]_{i,j} \end{bmatrix} O_{(3(n+1),3n_Q)}.$$

Next, the constraint matrix \mathcal{C} is defined as,

$$\mathcal{C} = \begin{bmatrix} O_{n,3} & \mathcal{C}_1 & O_{n,3} & O_{n,3n} & O_{n,6n_Q} \\ O_{n,3} & \mathcal{C}_2 & O_{n,3} & \mathcal{C}_1 & O_{n,6n_Q} \end{bmatrix}, \quad (7.23)$$

with

$$\mathcal{C}_1 = bdiag(q_{1d}^T, q_{2d}^T, \dots, q_{nd}^T), \mathcal{C}_2 = bdiag(-\omega_{1d}^T q_{1d}^\times, \dots, -\omega_{nd}^T q_{nd}^\times).$$

The rest of the elements are described below,

$$\begin{aligned} a_i &= M_{0i} [(\dot{\omega}_{id}^\times - \|\omega_{id}\|^2 I_3) q_{id}^\times, \\ b_i &= M_{0i} (2q_{id} \omega_{id}^\top), \quad i = \{1, \dots, n\}, \end{aligned}$$

$$c_{ij} = \begin{cases} [M_{io} \dot{q}_{0d}^\times - \sum_{j=1, j \neq i}^n M_{ij} ((q_{jd}^\times) \dot{\omega}_{jd})^\times + \\ \quad \|\omega_{jd}\|^2 q_{jd}^\times) - l_i (\sum_{k=i}^n u_k^\times)] (-q_i^\times), & i = j \\ [M_{ij} q_{id}^\times (\dot{\omega}_{jd}^\times - \|\omega_{jd}\|^2 I) q_{jd}^\times], & i \neq j \end{cases}$$

$$d_{ij} = \begin{cases} O_{3,3}, & i = j \\ M_{ij} [2q_{id}^\times q_{jd} \omega_{jd}^\top], & i \neq j \end{cases}$$

$$e_j = -f_{jd} R_{jd} e_3^\times, \quad j \in \mathcal{I}$$

$$f_{ij} = \begin{cases} \phi, & \text{if } j \notin \mathcal{I} \\ -(l_i q_i^\times) f_{jd} R_{jd} e_3^\times, & \text{if } j \in \mathcal{I}, j \geq i \\ O_{3,3}, & \text{if } j \in \mathcal{I}, j < i \end{cases}$$

$$g_j = R_{jd} e_3,$$

$$h_{ij} = \begin{cases} \phi, & \text{if } j \notin \mathcal{I} \\ (l_i q_i^\times) R_{jd} e_3, & \text{if } j \in \mathcal{I}, j \geq i \\ O_{3,1}, & \text{if } j \in \mathcal{I}, j < i \end{cases}$$

and $bdiag$ is block diagonal matrix. Note that $\mathbb{M}_{\{q_{id}\}}$ in (7.21), (7.22) is the same mass matrix in (7.14), except is the function of desired link attitudes $\{q_{id}\}$.

As seen, (7.19)-(7.20) is a time-varying constrained linear system. The constraints arise due to the variation constraint on S^2 as discussed in [24]. Controllability of the constrained linear equation can be argued along the similar lines to [24], however, due the complexity of the matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}$ computing the controllability matrix would be intractable.

Finite-Horizon LQR

Assuming, we have the complete reference trajectory we can implement any linear control technique for (7.19)-(7.20). Similar to [24, Lemma 1], we can show that the constraint (7.20) is time-invariant, i.e., if the initial condition satisfies the constraint, solution to the linear system would satisfy the constraint for all time. However, due to this constraint, the controllability matrix computed using \mathcal{A}, \mathcal{B} might not be full-rank and requires state transformation into the unconstrained space to result in full-rank controllability matrix.

Instead, we opt for a finite-horizon LQR controller for the variation-linearized dynamics about a time-varying desired trajectory. We chose a finite-time horizon T , the terminal cost matrix P_T and pick cost matrices for states $Q_1 = Q_1^T$ and inputs $Q_2 = Q_2^T$. Finally, we solve the continuous-time Ricatti equation backwards in time to obtain the gain matrix $P(t)$, that satisfies,

$$-\dot{P} = Q_1 - PBQ_2^{-1}\mathcal{B}^T P + \mathcal{A}^T P + P\mathcal{A}. \quad (7.24)$$

The above equation is solved offline and stored in a table for online computation. Note that the explicit time dependence of $P, \mathcal{A}, \mathcal{B}$ is dropped for convenience. Finally, the feedback gain for the control input is computed as,

$$K = R^{-1}\mathcal{B}P, \quad \delta \mathbf{u} = -K\delta \mathbf{x}. \quad (7.25)$$

Since the gains are computed backwards in time, the computed input would result in a stable control for the constrained linear-system. The net control-input to the nonlinear system can be computed as,

$$u(t) = u_d(t) + \delta \mathbf{u}. \quad (7.26)$$

In the next section, we present few numerical simulations with the finite-horizon LQR performing tracking control on the full nonlinear-dynamics.

7.5 Numerical Simulations

In this section, we present numerical results to validate the dynamics and control discussed in the earlier sections. We present numerical simulations for tracking control for a desired setpoint and circular trajectory.¹

¹MATLAB code for the simulations can be found at <https://github.com/HybridRobotics/multiple-quadrotor-flexible-hose>. Video for simulations is at <https://youtu.be/i3egJ4fcAKM>.

Setpoint Tracking

(i). Two Quadrotor system:

Following parameters are considered for the simulations,

$$n = 10, n_Q = 2, \mathcal{I} = \{0, 10\}, m_i = 0.0909kg, l_i = 0.1m, \\ m_{Qj} = 0.85kg, J_j = diag([.0557, .0557, 0.1050])kgm^2$$

and the setpoint is given as,

$$x_{0d} = [0, 0, 0]^T, x_{nd} = [0.6, 0.0, 0.0]^T,$$

with the cable hanging between these two points. Degrees of freedom and under-actuation for this setup are $\#DOF = 29$, $\#DOuA = 21$ respectively. The linear dynamics $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are computed about this setpoint x_d . Here, we compare two different controllers, (i) the finite-horizon LQR discussed in the previous-section and (ii) position-controllers on the two quadrotors with feed-forward forces due to the cable at steady state. We start with some initial error in the cable orientation and the resulting error plots are shown in Figure 7.3a. As seen in the Figure, errors for cable position x_0 , cable attitudes and quadrotors' attitude converge to origin. Attitude errors for the hose links is defined as the configuration error on S^2 ,

$$\Psi_q = 1 - q_{id}^T q_i, \quad (7.27)$$

and similar quadrotor attitude error is defined as,

$$\Psi_R = 0.5Tr(I - R_{jd}^T R_j). \quad (7.28)$$

For the position control with feed-forward forces, even though the quadrotor attitudes are zero, the initial error in cable orientation results in oscillations in the cable. These oscillations are not accounted for in the control and can be seen in Figure 7.3a.

(ii). Three Quadrotor system:

Setpoint tracking for cable suspended from three-quadrotors is presented here. Parameters for the system are as follows,

$$n = 10, n_Q = 3, \mathcal{I} = \{0, 5, 10\}, m_i = 0.0909kg, l_i = 0.2m,$$

and $\#DOF = 32, \#DOuA = 20$. Various tracking errors for the system are presented in Figure 7.3b and snapshots for the system are shown in Figure 7.4.

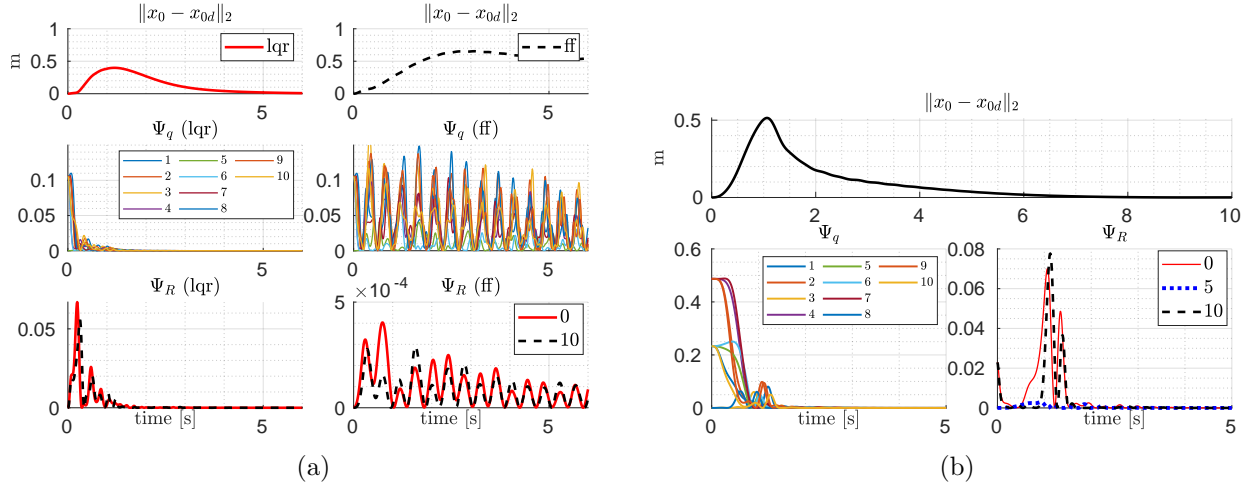


Figure 7.3: (a). List of errors comparing the LQR control on the whole system (lqr) and feed-forward control on the quadrotor-position (ff) (b). Tracking errors for desired set-point for $n_Q = 3$ with the LQR control. Ψ_q, Ψ_R are as defined in (7.27)-(7.28)

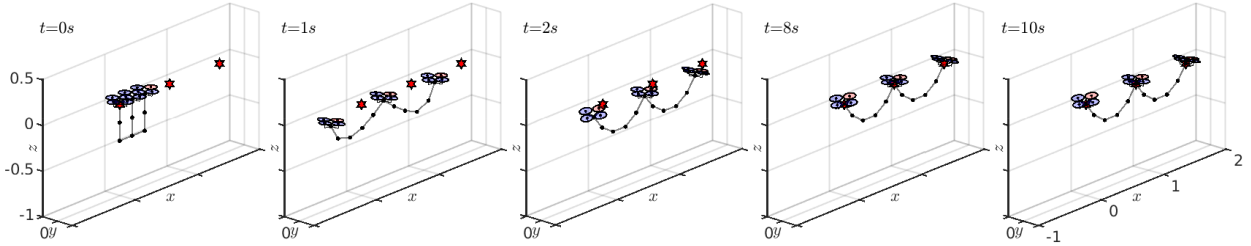


Figure 7.4: Snapshots of 3 quadrotor-10 link system while tracking a setpoint. Setpoints for the quadrotor position is shown by the red-hexagrams.

Trajectory Tracking

In this section, we show that the presented controller tracks a desired time-varying trajectory with initial errors. We use the following system parameters,

$$n = 5, n_Q = 2, \mathcal{I} = \{0, 5\}, m_i = 0.1667kg, l_i = 0.2m$$

and the rest same as those given in Section 7.5. We consider the following flat output trajectory,

$$x_0 = \begin{bmatrix} a_x(1 - \cos(2f_1\pi t)) \\ a_y \sin(2f_2\pi t) \\ a_z \cos(2f_3\pi t) \end{bmatrix}, \bar{T}_1 = \begin{bmatrix} 2.74 \\ 0.0 \\ -3.27 \end{bmatrix}, \psi_0 \equiv \psi_5 \equiv 0,$$

$$f_1 = \frac{1}{4}, f_2 = \frac{1}{5}, f_3 = \frac{1}{7}, a_x = 2, a_y = 2.5, a_z = 1.5.$$

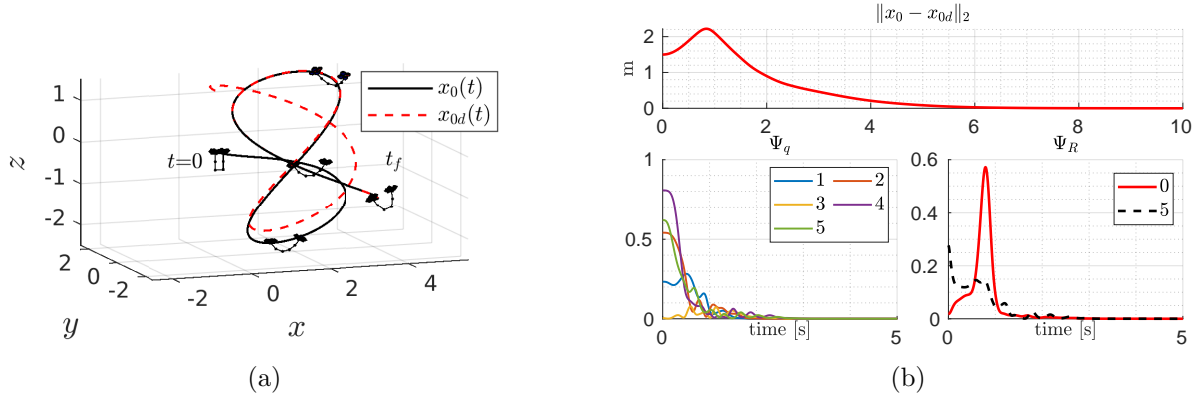


Figure 7.5: (a). Snapshots of the *multiple quadrotors carrying a flexible cable* system while tracking the desired trajectory (b). Errors for the trajectory tracking control

Rest of the states and inputs can be computed using differential-flatness. We use the linearized-dynamics and the finite-horizon LQR presented in the previous sections to achieve the tracking control. Following weights are used for the LQR,

$$\begin{aligned} Q_x &= 0.5I_6, \quad Q_q = 0.75I_{6n}, \quad Q_R = I_{3n_Q}, \quad Q_\Omega = 0.75I_{3n_Q}, \\ Q &= bdiag(Q_x, Q_q, Q_R, Q_\Omega), \\ R &= 0.2I_{4n_Q}, \quad P_T = 0.01I_{n_x}, \end{aligned}$$

where $n_x = 6+6n+6n_Q$. Figure 7.5a shows snapshots of the system at different instants along the trajectory. The proposed controller tracks the desired trajectory (shown in red) when started with an initial error.

7.6 Experimental Demonstrations

This section presents experiments demonstrating potential applications for two quadrotors powered using a single tether. A two-quadrotor system has the advantages of a single tethered quadrotor, such as extended flight time and a steady power supply, while also increasing maneuverability in the horizontal direction, especially over difficult/unknown terrains. In the rest of the section, we present two experiments, comparing the following setups, (i) a single tethered quadrotor (ii) a two quadrotor system powered by a single tether.

Passing Through Windows

In this experiment, we consider a window passing example with tethered quadrotors as shown in Figure 7.6. In setup (ii), the first quadrotor (left) supports the second quadrotor (right) exploring on the other side of the window. Similar exploration using single tethered

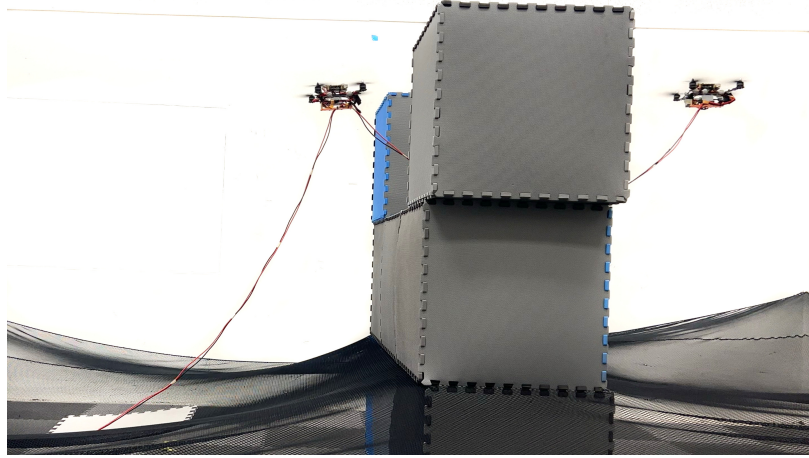
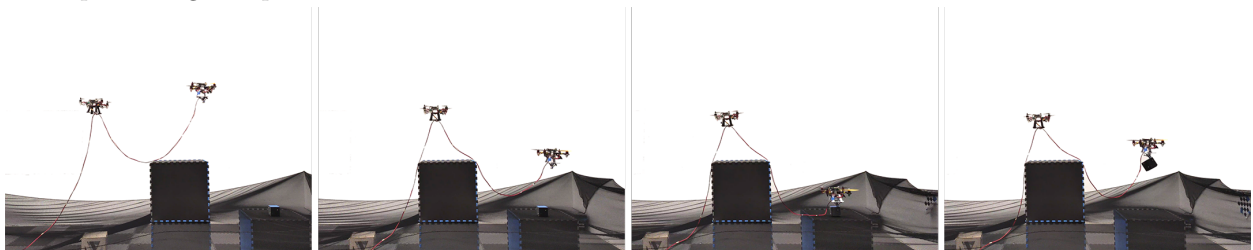


Figure 7.6: Experiment demonstrating two quadrotors tethered using a single cable, supplying power from an external power source. Multiple quadrotors can be used collaboratively to achieve tethered flight over unknown/challenging terrain while increasing the horizontal reachability of the quadrotor.



(a) Snapshots of a grasping attempt using a single tethered quadrotor (setup (i)) over an obstacle – this ends up crashing the quadrotor



(b) Snapshots of two quadrotors connected by single tether (ii) grasping over an obstacle

Figure 7.7: Experimental demonstrations of tethered quadrotor for grasping over obstacles

quadrotor are not always feasible in cluttered environments, as the obstacles could interfere with the tether.

Grasping Over Obstacles

A two quadrotor tethered system can act as an *aerial series-manipulator* in cluttered environments, as shown in Figure 7.7b. We do a grasping experiment using the two setups, to grasp an object located on the other side of an obstacle, in the vertical plane as shown in Figure 7.7. Waypoints are provided to the quadrotors to reach over the obstacles and grasp the object. An electromagnet is used as the gripper for grasping the metallic object. In setup (i), see Figure 7.7a, a single tethered quadrotor attempts to grasp the object, however, is unable to reach the grasp location due to the limitation of the quadrotor to drag the tether over obstacles. In setup (ii), the first quadrotor (left) acts an intermediate joint for the second drone (right) to help grasp the object over the obstacle as shown in Figure 7.7b.

7.7 Summary

In this chapter, we have studied the multiple quadrotors carrying a flexible cable system. We modeled the flexible cable as a series of smaller discrete links with lumped mass and derived the coordinate-free dynamics using Lagrange-d'Alembert's principle. We also showed that the given system is differentially-flat as long as the end of the hose is connected to a quadrotor. Variation-based linearized dynamics were derived about time-varying desired trajectory. We showed tracking control for the system using finite-horizon LQR for the linear dynamics and validated this through numerical simulations with up to 10 discretizations of the hose. Finally, we discussed some limitations due to the assumptions and directions for future work. In the next chapter, concluding remarks and potential future research directions are discussed.

Chapter 8

Conclusion & Future Work

8.1 Conclusion

This dissertation first presents a method to estimate and reject disturbances in the quadrotor attitude control. In any control architecture implementation of collaborative aerial manipulation, it is essential to have stable attitude control, the innermost loop. In aerial manipulation, vehicles are usually equipped with additional mechanisms such as manipulator arms or grippers and physically interact with the world. Added mechanisms or physical interactions can lead to model uncertainties and disturbances to the system. Geometric control exhibits global stability properties, while the L_1 adaptation scheme has fast adaptation while decoupling high-frequency estimates from the control. Geometric L_1 adaptive attitude control combines these control schemes to combine global properties with fast adaptation. Theoretical proof and experimental validations are presented for the geometric L_1 adaptive control to deal with the uncertainties in the attitude control.

A quadrotor with a cable-suspended gripper, as opposed to one rigidly fixed to the quadrotor frame, gives a couple of advantages at the cost of added under actuation in the system. Using cables allows for in-flight grasping without the need to land the quadrotor. Adds some distance to the target object and doesn't disturb the surrounding area while picking up the object. However, dealing with the under-actuation requires estimation and control of the cable attitude evolving on S^2 is crucial.

States evolving manifolds, such as S^2 , are variables embedded in Euclidean space with constraints, such as unit-norm for the S^2 state. Standard extended Kalman filters do not consider the manifold constraints when estimating the state. Variation extended Kalman filter uses the variation on manifold concept to linearize the dynamics that respect the manifold constraints. Operators are defined to compute the to and from transformation between the states on TS^2 to Euclidean space. The Kalman filter operations, such as the covariance and measurement updates, are performed in this Euclidean space and mapped back to the manifold. Numerical simulations show the convergence of various initial conditions with bounded initial angular velocity error. The proposed method is experimentally validated.

This dissertation proposes fast offline trajectory generation for a quadrotor with a suspended payload. Trajectory generation is formulated as a nonlinear optimization problem using the direction-collocation concept. A quadrotor with a cable-suspended payload model is parameterized using the differential-flatness property with complementarity constraints. In addition, non-convex optimization constraints are reformulated using dual variables into smooth constraints by considering the quadrotor-load system as a full-dimensional controlled object. The proposed algorithm outperforms the state-of-the-art in the computation time. Finally, the planning method is numerically and experimentally verified.

Multiple quadrotors with cable-suspended grippers can be deployed to grasp larger payloads and transport them. The dissertation presents the architecture to grasp and transport payloads using suspended cables using electromagnets. Two such quadrotor-gripper vehicles are used to grasp and transport a larger payload. Experiments show how aerial vehicles can be exploited for collaboration tasks even while the underlying system (quadrotor-payload) is underactuated.

Finally, the dissertation presents the dynamics of a “serial-aerial-manipulator” when multiple quadrotors are connected in series using a single flexible cable. The cable is modeled using the lumped-mass model as a series of mass-less links with mass at the end of the link. Each link is represented using a unit-vector (S^2), and a global coordinate free generalized dynamics are derived. It is shown that the system is differentially flat, and the trajectories generated using the flat variables are used to compute the full-time-varying reference state. Linearized dynamics for the system are computed using variation-based linearization. An instance of the tethered multi-quadrotor system, say an externally powered using tether, can be used to operate together in flight.

In conclusion, the dissertation is focused on ways to achieve collaborative aerial manipulation. Control, estimation, and planning methods are presented to facilitate individual quadrotor and quadrotor payload systems to perform manipulation, more specifically, collaborative aerial manipulation tasks. Experimental validations are presented on how the cable-suspended quadrotor systems can be used for grasping and transportation tasks. The methods presented in this dissertation help make collaborative aerial manipulation more practical for real-world applications.

8.2 Future Work

The dissertation covers various aspects of collaborative aerial manipulation. It discusses control, estimation, and planning methodologies for the individual agents and how they can be employed together to perform collaborative tasks. However, these methods and approaches consider certain assumptions, approximations, and simplifications necessary for taking a step toward achieving the overarching goal of real-world implementation of collaborative aerial manipulation. This section discusses some of the shortcomings and how they can be addressed in the future.

Outdoor & vision based experiments: All the flight experiments were performed in an indoor flight space with a motion-capture system. The motion capture system isolates the problem of control and grasping from estimation and perception, a significant component integral to real-world applications, while also creating a wind-disturbance-free flight space for experiments. New hardware makes it possible to run computationally expensive perception algorithms onboard the quadrotors. Thus, works like [60–62] can be employed to estimate the payload pose and perceive & locate target locations. While extensive research on wind disturbance rejection for individual quadrotors exists in the literature, a multi-quadrotor payload system makes it more complex to deal with external wind disturbance and internal disturbances due to downwash. We expect future work to focus on developing more robust methods for collaborative manipulation.

Decentralized approach: Chapter 6 primarily focuses on the centralized control approach for controlling and transporting the payload. As discussed in the chapter, implementing such approaches in hardware requires a centralized base station that computes the control commands for all the agents together and communicates them to the individual agent; this approach necessitates a centralized ground station. This approach requires high-speed communication with the individual agents for active control of the payload, which is ideal for developing collaborative systems and can potentially be used in real-world applications with short ranges, like in a warehouse. Fast communication with the ground station is not always ideal, especially outdoors; alternately, one of the agents should act as the ground station, thus requiring the agent to have higher computational power.

Drag due to downwash: Throughout this work, the payload/gripper on the other end of the quadrotor is modeled as a point mass. Even in the case of multiple quadrotors, the payloads are designed to have a small surface area. While suspending the payload using the cable increases the distance between the payload and quadrotor and enables the airflow from the propellers for necessary lift, it doesn't eliminate Drag. Payloads with larger surface areas experience significant Drag due to the downwash from the quadrotor. Thus, the drag force acts as an external disturbance on the quadrotor along the cable. Works like [138] can be employed to model the amount of Drag for a given surface and cable length. Modeling the Drag can help design system parameters, such as the adequate cable to minimize Drag for a given surface area, and develop algorithms to estimate and cancel the drag forces.

Learning for control: Extensive modeling and carefully designed control algorithms can be successfully employed on hardware for collaborative aerial manipulation. These models, however, do not always capture the complete system behavior, for example, the aerodynamic effects of the quadrotors on each other in close proximity, the communication delays, estimation noise, etc. Model-free and model-based learning algorithms can be employed to train for optimal control policies for transporting and manipulating using multiple vehicles.

Finally, addressing these research problems will enable a swarm of quadrotors to extend their abilities beyond surveillance, sensing, and inspection. In conclusion, these research directions help achieve a collaborative aerial manipulation that can be employed in real-world applications, for instance, grasping and transportation in warehouses and search & rescue, or may even be used for constructing habitats on the moon/mars.

Bibliography

- [1] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. “Aerial manipulation: A literature review”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1957–1964.
- [2] Hossein Bonyan Khamseh, Farrokh Janabi-Sharifi, and Abdelkader Abdessameud. “Aerial manipulation—A literature survey”. In: *Robotics and Autonomous Systems* 107 (2018), pp. 221–235.
- [3] Matko Orsag et al. *Aerial manipulation*. Springer, 2018.
- [4] DING Xilun et al. “A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems”. In: *Chinese Journal of Aeronautics* 32.1 (2019), pp. 200–214.
- [5] Alejandro Suarez et al. “Benchmarks for aerial manipulation”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2650–2657.
- [6] Gowtham Garimella and Marin Kobilarov. “Towards model-predictive control for aerial pick-and-place”. In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 4692–4697.
- [7] Justin Thomas et al. “Toward image based visual servoing for aerial grasping and perching”. In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 2113–2118.
- [8] Sarah Tang, Koushil Sreenath, and Vijay Kumar. “Multi-robot trajectory generation for an aerial payload transport system”. In: *Robotics Research*. Springer, 2020, pp. 1055–1071.
- [9] Daniel Mellinger et al. “Cooperative grasping and transport using multiple quadrotors”. In: *Distributed autonomous robotic systems*. Springer, 2013, pp. 545–558.
- [10] Federico Augugliaro et al. “The flight assembled architecture installation: Cooperative construction with flying machines”. In: *IEEE Control Systems Magazine* 34.4 (2014), pp. 46–64.
- [11] Yazeed Alnumay et al. “Carbon Footprint Reduction Through the Use of Drones for Inspection Activities”. In: *ADIPEC*. OnePetro. 2022.
- [12] Azarakhsh Keipour et al. “Integration of fully-actuated multirotors into real-world applications”. In: *arXiv preprint arXiv:2011.06666* (2020).

- [13] Jake Welde and Vijay Kumar. “Coordinate-free dynamics and differential flatness of a class of 6DOF aerial manipulators”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 4307–4313.
- [14] Brian F Doolin and Clyde F Martin. *Introduction to differential geometry for engineers*. Courier Corporation, 2013.
- [15] Robert Gilmore. *Lie groups, physics, and geometry: an introduction for physicists, engineers and chemists*. Cambridge University Press, 2008.
- [16] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. “A micro Lie theory for state estimation in robotics”. In: *arXiv preprint arXiv:1812.01537* (2018).
- [17] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. “Global formulations of Lagrangian and Hamiltonian dynamics on manifolds”. In: *Springer* 13 (2017), p. 31.
- [18] Frank C Park, James E Bobrow, and Scott R Ploen. “A Lie group formulation of robot dynamics”. In: *The International journal of robotics research* 14.6 (1995), pp. 609–618.
- [19] Francesco Bullo and Andrew D Lewis. *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Vol. 49. Springer, 2019.
- [20] John Milnor. “Morse Theory.(AM-51), Volume 51”. In: *Morse Theory.(AM-51), Volume 51*. Princeton university press, 2016.
- [21] Jerrold E Marsden and Tudor S Ratiu. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Vol. 17. Springer Science & Business Media, 2013.
- [22] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. “Lagrangian mechanics and variational integrators on two-spheres”. In: *International Journal for Numerical Methods in Engineering* 79.9 (2009), pp. 1147–1174.
- [23] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. “Stable manifolds of saddle equilibria for pendulum dynamics on S^2 and $SO(3)$ ”. In: *Decision and Control and European Control Conference*. 2011, pp. 3915–3921.
- [24] Guofan Wu and Koushil Sreenath. “Variation-Based Linearization of Nonlinear Systems Evolving on $SO(3)$ and S^2 ”. In: *IEEE Access* 3 (2015), pp. 1592–1604.
- [25] Richard M Murray, Muruhan Rathinam, and Willem Sluis. “Differential flatness of mechanical control systems: A catalog of prototype systems”. In: *ASME international mechanical engineering congress and exposition*. Citeseer. 1995.
- [26] Daniel Mellinger and Vijay Kumar. “Minimum snap trajectory generation and control for quadrotors”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 2520–2525.

- [27] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar. “Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load”. In: *Conference on Decision and Control*. 2013, pp. 2269–2274.
- [28] Prasanth Kotaru, Guofan Wu, and Koushil Sreenath. “Differential-flatness and control of quadrotor (s) with a payload suspended through flexible cable (s)”. In: *Indian Control Conference (ICC)*. 2018, pp. 352–357.
- [29] Koushil Sreenath and Vijay Kumar. “Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots”. In: *Robotics: Science and Systems (RSS)*. 2013.
- [30] Beomyeol Yu et al. “Geometric Control and Experimental Validation for a Quadrotor UAV Transporting a Payload”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 201–207.
- [31] Jun Zeng and Koushil Sreenath. “Geometric Control of a Quadrotor with a Load Suspended from an Offset”. In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 3044–3050.
- [32] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. “Geometric tracking control of a quadrotor UAV on SE (3)”. In: *49th IEEE conference on decision and control (CDC)*. IEEE. 2010, pp. 5420–5425.
- [33] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. “Control of complex maneuvers for a quadrotor UAV using geometric methods on SE(3)”. In: *arXiv preprint arXiv:1003.2005* (2010).
- [34] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar. “Geometric Control and Differential Flatness of a Quadrotor UAV with a Cable-Suspended Load”. In: *IEEE Conference on Decision and Control (CDC)*. Florence, Italy, Dec. 2013, pp. 2269–2274.
- [35] Farhad A Goodarzi, Daewon Lee, and Taeyoung Lee. “Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on SE(3) for agile maneuvers”. In: *Journal of Dynamic Systems, Measurement, and Control* 137.9 (2015), p. 091007.
- [36] Guofan Wu and Koushil Sreenath. “Geometric Control of Quadrotors Transporting a Rigid-body Load”. In: *IEEE Conference on Decision and Control*. Dec. 2014, pp. 6141–6148.
- [37] Brian Whitehead and Stefan Bieniawski. “Model reference adaptive control of a quadrotor UAV”. In: *AIAA Guidance, Navigation, and Control Conference, 2010*, p. 8148.
- [38] Alia F Abdul Ghaffar and Thomas S Richardson. “Position Tracking of an Under-actuated Quadrotor using Model Reference Adaptive Control”. In: *AIAA Guidance, Navigation, and Control Conference, 2016*, p. 1388.

- [39] Zongyu Zuo and Pengkai Ru. “Augmented L1 adaptive tracking control of quad-rotor unmanned aircrafts”. In: *IEEE Transactions on Aerospace and Electronic Systems*, 2014 50.4 (), pp. 3090–3101.
- [40] Chengyu Cao and Naira Hovakimyan. “Design and analysis of a novel L1 adaptive controller, Part I: Control signal and asymptotic stability”. In: *American Control Conference, 2006*, pp. 3397–3402.
- [41] Jens Dodenhöft et al. “Design and Evaluation of an L1 Adaptive Controller for NASA’s Transport Class Model”. In: *AIAA Guidance, Navigation, and Control Conference*. 2017, p. 1250.
- [42] Kasey Ackerman et al. “L1 Stability Augmentation System for Calspan’s Variable-Stability Learjet”. In: *AIAA Guidance, Navigation, and Control Conference*. 2016, p. 0631.
- [43] Bernard Michini. “Modeling and adaptive control of indoor unmanned aerial vehicles”. PhD thesis. Massachusetts Institute of Technology, 2009.
- [44] Paul De Monte and Boris Lohmann. “Position trajectory tracking of a quadrotor based on L1 adaptive control”. In: *at-Automatisierungstechnik*, 2014 62.3 (), pp. 188–202.
- [45] Srinath Mallikarjunan et al. “L1 adaptive controller for attitude control of multirotors”. In: *AIAA Guidance, Navigation, and Control Conference*. 2012, p. 4831.
- [46] Chengyu Cao and Naira Hovakimyan. “Design and analysis of a novel l1 adaptive controller, part ii: Guaranteed transient performance”. In: *American Control Conference, 2006*, pp. 3403–3408.
- [47] Taeyoung Lee. “Geometric adaptive control for aerial transportation of a rigid body”. In: *arXiv preprint arXiv:1503.01148* (2015).
- [48] Shankar Kulumani, Christopher Poole, and Taeyoung Lee. “Geometric adaptive control of attitude dynamics on SO(3) with state inequality constraints”. In: *American Control Conference, 2016*, pp. 4936–4941.
- [49] Naira Hovakimyan et al. “ L_1 adaptive control for safety-critical systems”. In: *IEEE Control Systems Magazine* 31.5 (2011), pp. 54–104.
- [50] Naira Hovakimyan and Chengyu Cao. *L1 adaptive control theory: guaranteed robustness with fast adaptation*. Vol. 21. SIAM-Society for Industrial and Applied Mathematics, 2010.
- [51] Quan Nguyen and Koushil Sreenath. “L1 adaptive control for bipedal robots with control Lyapunov function based quadratic programs”. In: *American Control Conference, 2015*, pp. 862–867.
- [52] Shishir Kolathaya, Jacob Reher, and Aaron D Ames. “Input to State Stability of Bipedal Walking Robots: Application to DURUS”. In: *arXiv preprint arXiv:1801.00618* (2018).

- [53] Eugene Lavretsky and Travis E Gibson. “Projection operator in adaptive systems”. In: *arXiv preprint arXiv:1112.4232* (2011).
- [54] Quan Quan. *Introduction to multicopter design and control*. Springer, 2017.
- [55] Farhad A Goodarzi, Daewon Lee, and Taeyoung Lee. “Geometric nonlinear PID control of a quadrotor UAV on $SE(3)$ ”. In: *European Control Conference, 2013*, pp. 3845–3850.
- [56] Robert Mahony, Vijay Kumar, and Peter Corke. “Multirotor aerial vehicles”. In: *IEEE Robotics and Automation magazine* 20.32 (2012).
- [57] Taeyoung Lee, Koushil Sreenath, and Vijay Kumar. “Geometric control of cooperating multiple quadrotor UAVs with a suspended payload”. In: *Conference on decision and control*. 2013, pp. 5510–5515.
- [58] Elizabeth Gregory and Amit K Sanyal. “A Comparison Study of State Estimators for Dynamics on the Sphere”. In: *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*. American Society of Mechanical Engineers. 2012, pp. 615–624.
- [59] Farhad A Goodarzi and Taeyoung Lee. “Global Formulation of an Extended Kalman Filter on $SE(3)$ for Geometric Control of a Quadrotor UAV”. In: *Journal of Intelligent & Robotic Systems* 88.2-4 (2017), pp. 395–413.
- [60] Sarah Tang, Valentin Wüest, and Vijay Kumar. “Aggressive flight with suspended payloads using vision-based control”. In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 1152–1159.
- [61] Guanrui Li, Alex Tunchez, and Giuseppe Loianno. “Pcmpc: Perception-constrained model predictive control for quadrotors with suspended loads using a single camera and imu”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 2012–2018.
- [62] Guanrui Li, Rundong Ge, and Giuseppe Loianno. “Cooperative transportation of cable suspended payloads with mavs using monocular vision and inertial sensing”. In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5316–5323.
- [63] Lihua Xie, Dan Popa, and Frank L Lewis. *Optimal and robust estimation: with an introduction to stochastic control theory*. CRC press, 2007.
- [64] Nachi Gupta and Raphael Hauser. “Kalman filtering with equality and inequality state constraints”. In: *arXiv preprint arXiv:0709.2791* (2007).
- [65] Dan Simon and Tien Li Chia. “Kalman filtering with state equality constraints”. In: *IEEE transactions on Aerospace and Electronic Systems* 38.1 (2002), pp. 128–136.
- [66] Guofan Wu and Koushil Sreenath. “Variation-based Linearization of Nonlinear Systems Evolving on $SO(3)$ and S^2 ”. In: *IEEE Access* 3 (Sept. 2015), pp. 1592–1604.
- [67] Prasanth Kotaru and Koushil Sreenath. “Variation based extended Kalman filter on S^2 ”. In: *European Control Conference (ECC)*. 2019, pp. 875–882.

- [68] Guillermo Gallego and Anthony Yezzi. “A compact formula for the derivative of a 3-D rotation in exponential coordinates”. In: *Journal of Mathematical Imaging and Vision* 51.3 (2015), pp. 378–384.
- [69] S. Kim, S. Choi, and H. Kim. “Aerial manipulation using a quadrotor with a two dof robotic arm”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 4990–4995.
- [70] Justin Thomas et al. “Toward Autonomous Avian-Inspired Dynamic Grasping and Perching”. In: *Bioinspiration & Biomimetics* 9.2 (June 2014), pp. 025010–025024.
- [71] I. Palunko, R. Fierro, and P. Cruz. “Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach”. In: *IEEE International Conference on Robotics and Automation*. 2012, pp. 2691–2697.
- [72] A. Faust et al. “Learning swing-free trajectories for UAVs with a suspended load”. In: *IEEE International Conference on Robotics and Automation*. pp. 4902–4909. 2013.
- [73] M. E. Guerrero et al. “Passivity based control for a quadrotor UAV transporting a cable-suspended payload with minimum swing”. In: *IEEE Conference on Decision and Control*. 2015, pp. 6718–6723.
- [74] K. Sreenath, T. Lee, and V. Kumar. “Geometric Control and Differential Flatness of a Quadrotor UAV with a Cable-Suspended Load”. In: *IEEE International Conference on Decision and Control*. pp. 2269–2274. 2013.
- [75] T. Lee, K. Sreenath, and V. Kumar. “Geometric control of cooperating multiple quadrotor UAVs with a suspended payload”. In: *IEEE Conference on Decision and Control*. 2013, pp. 5510–5515.
- [76] K. Sreenath, N. Michael, and V. Kumar. “Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system”. In: *IEEE International Conference on Robotics and Automation*. pp. 4873–4880. 2013.
- [77] K. Sreenath and V. Kumar. “Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots”. In: *Robotics: Science and Systems*. 2013.
- [78] G. Wu and K. Sreenath. “Geometric Control of Multiple Quadrotors Transporting a Rigid-body Load”. In: *IEEE International Conference on Decision and Control*. pp. 6141–6148. 2014.
- [79] C. de Crousaz, F. Farshidian, and J. Buchli. “Aggressive optimal control for agile flight with a slung load”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Machine Learning in Planning and Control of Robot Motion*. 2014.
- [80] S. Tang and V. Kumar. “Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload”. In: *IEEE International Conference on Robotics and Automation*. 2216–2222. 2015.

- [81] P. Foehn et al. “Fast Trajectory Optimization for Agile Quadrotor Maneuvers with a Cable-Suspended Payload”. In: *Robotics: Science and Systems*. 2017.
- [82] Alvaro Caballero et al. “Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms”. In: *International Journal of Advanced Robotic Systems* 15.3 (2018), p. 1729881418770525.
- [83] Jun Zeng, Prasanth Kotaru, and Koushil Sreenath. “Geometric Control and Differential Flatness of a Quadrotor UAV with Load Suspended from a Pulley”. In: *American Control Conference*. IEEE. 2019, pp. 2420–2427.
- [84] Matthew Kelly. “An introduction to trajectory optimization: How to do your own direct collocation”. In: *SIAM Review* 59.4 (2017), pp. 849–904.
- [85] Anil V Rao. “A survey of numerical methods for optimal control”. In: *Advances in the Astronautical Sciences* 135.1 (2009), pp. 497–528.
- [86] Francesco Topputo and Chen Zhang. “Survey of direct transcription for low-thrust space trajectory optimization with applications”. In: *Abstract and Applied Analysis*. Vol. 2014. Hindawi. 2014.
- [87] Moritz Diehl et al. “Fast direct multiple shooting algorithms for optimal robot control”. In: *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 65–93.
- [88] Brian Geiger et al. “Optimal path planning of UAVs using direct collocation with nonlinear programming”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2006, p. 6199.
- [89] Ayonga Hereid et al. “3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 1447–1454.
- [90] Xiaojing Zhang et al. “Autonomous parking using optimization-based collision avoidance”. In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 4327–4332.
- [91] R. H. Byrd, M. E. Hribar, and Jorge Nocedal. “An interior point algorithm for large-scale nonlinear programming”. In: *SIAM Journal on Optimization* 9.4 (1999), pp. 877–900.
- [92] J. Löfberg. “YALMIP : A Toolbox for Modeling and Optimization in MATLAB”. In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.
- [93] Daniel Mellinger et al. “Design, modeling, estimation and control for aerial grasping and manipulation”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2011, pp. 2668–2673.
- [94] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. “Construction with quadrotor teams”. In: *Autonomous Robots* 33.3 (2012), pp. 323–336.

- [95] Fran Real et al. “Experimental evaluation of a team of multiple unmanned aerial vehicles for cooperative construction”. In: *IEEE Access* 9 (2021), pp. 6817–6835.
- [96] Chad C Kessens et al. “Versatile aerial grasping using self-sealing suction”. In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 3249–3254.
- [97] Paul Kremer et al. “A Lightweight Universal Gripper with Low Activation Force for Aerial Grasping”. In: *arXiv preprint arXiv:2208.10768* (2022).
- [98] Suseong Kim, Seungwon Choi, and H Jin Kim. “Aerial manipulation using a quadrotor with a two dof robotic arm”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 4990–4995.
- [99] Guillermo Heredia et al. “Control of a multirotor outdoor aerial manipulator”. In: *2014 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2014, pp. 3417–3422.
- [100] AE Jimenez-Cano et al. “Aerial manipulator for structure inspection by contact from the underside”. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2015, pp. 1879–1884.
- [101] Justin Thomas et al. “Avian-inspired grasping for quadrotor micro UAVs”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 55935. American Society of Mechanical Engineers. 2013, V06AT07A014.
- [102] Hannibal Paul et al. “TAMS: development of a multipurpose three-arm aerial manipulator system”. In: *Advanced Robotics* 35.1 (2021), pp. 31–47.
- [103] Ivana Palunko, Patricio Cruz, and Rafael Fierro. “Agile load transportation: Safe and efficient load manipulation with aerial robots”. In: *IEEE robotics & automation magazine* 19.3 (2012), pp. 69–79.
- [104] Dengyan Duan et al. “Stabilization control for unmanned helicopter-slung load system based on active disturbance rejection control and improved sliding mode control”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 235.13 (2021), pp. 1803–1816.
- [105] Hongda Wang, Yongbin Huang, and Chao Xu. “ADRC methodology for a quadrotor UAV transporting hanged payload”. In: *2016 IEEE International Conference on Information and Automation (ICIA)*. IEEE. 2016, pp. 1641–1646.
- [106] M Eusebia Guerrero-Sánchez et al. “Swing-attenuation for a quadrotor transporting a cable-suspended payload”. In: *ISA transactions* 68 (2017), pp. 433–449.
- [107] Xiao Liang et al. “Antiswing Control for Aerial Transportation of the Suspended Cargo by Dual Quadrotor UAVs”. In: *IEEE/ASME Transactions on Mechatronics* (2022).

- [108] Maria Eusebia Guerrero et al. “Passivity based control for a quadrotor UAV transporting a cable-suspended payload with minimum swing”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 6718–6723.
- [109] Sarah Tang and Vijay Kumar. “Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload”. In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 2216–2222.
- [110] Gan Yu et al. “Aggressive maneuvers for a quadrotor-slung-load system through fast trajectory generation and tracking”. In: *Autonomous Robots* 46.4 (2022), pp. 499–513.
- [111] Ryo Miyazaki et al. “Long-reach aerial manipulation employing wire-suspended hand with swing-suppression device”. In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 3045–3052.
- [112] Hassan K Khalil. *Nonlinear control*. Vol. 406. Pearson New York, 2015.
- [113] Tse-Huai Wu et al. “Spacecraft relative attitude formation tracking on SO (3) based on line-of-sight measurements”. In: *2013 American Control Conference*. IEEE. 2013, pp. 4820–4825.
- [114] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [115] Patricio J Cruz and Rafael Fierro. “Cable-suspended load lifting by a quadrotor UAV: hybrid model, trajectory generation, and control”. In: *Autonomous Robots* 41.8 (2017), pp. 1629–1643.
- [116] Guofan Wu and Koushil Sreenath. “Geometric control of multiple quadrotors transporting a rigid-body load”. In: *Conference on Decision and Control*. 2014, pp. 6141–6148.
- [117] Taeyoung Lee. “Geometric control of quadrotor UAVs transporting a cable-suspended rigid body”. In: *IEEE Transactions on Control Systems Technology* 26.1 (2017), pp. 255–264.
- [118] Aaron D Ames et al. “Control barrier functions: Theory and applications”. In: *2019 18th European control conference (ECC)*. IEEE. 2019, pp. 3420–3431.
- [119] Quan Nguyen and Koushil Sreenath. “Exponential control barrier functions for enforcing high relative-degree safety-critical constraints”. In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 322–328.
- [120] Aerones. *Firefighting drone*. 2018. URL: https://www.aerones.com/eng/firefighting_drone/.
- [121] Blake Hament and Paul Oh. “A Pressure Washing Hosing-Drone–Mitigating Reaction Forces and Torques”. In: *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2022, pp. 468–477.

- [122] SkyScopes. *Sharper Shape and SkySkopes pull power lines*. 2017. URL: <https://www.skyskopes.com/post/sharper-shape-and-skyskopes-pull-power-lines> (visited on 08/27/2017).
- [123] Ivan Maza et al. “Multi-UAV cooperation and control for load transportation and deployment”. In: *International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. Springer. 2009, pp. 417–449.
- [124] Qimi Jiang and Vijay Kumar. “The inverse kinematics of cooperative transport with multiple aerial robots”. In: *IEEE Transactions on Robotics* 29.1 (2012), pp. 136–145.
- [125] Hyeonbeom Lee and H Jin Kim. “Constraint-based cooperative control of multiple aerial manipulators for handling an unknown payload”. In: *IEEE Transactions on Industrial Informatics* 13.6 (2017), pp. 2780–2790.
- [126] Nathan Michael, Jonathan Fink, and Vijay Kumar. “Cooperative manipulation and transportation with aerial robots”. In: *Autonomous Robots* 30.1 (2011), pp. 73–86.
- [127] Farhad A Goodarzi and Taeyoung Lee. “Stabilization of a rigid body payload with multiple cooperative quadrotors”. In: *Journal of Dynamic Systems, Measurement, and Control* 138.12 (2016), p. 121001.
- [128] Farhad A Goodarzi, Daewon Lee, and Taeyoung Lee. “Geometric control of a quadro- tor UAV transporting a payload connected via flexible cable”. In: *International Journal of Control, Automation and Systems* 13.6 (2015), pp. 1486–1498.
- [129] Farhad A Goodarzi and Taeyoung Lee. “Dynamics and control of quadrotor UAVs transporting a rigid body connected via flexible cables”. In: *American Control Conference*. 2015, pp. 4677–4682.
- [130] Sergei Lupashin and Raffaello D’Andrea. “Stabilization of a flying vehicle on a taut tether using inertial sensing”. In: *2013 IEEE/RSJ International Conference on Intel- ligent Robots and Systems*. 2013, pp. 2432–2438.
- [131] Marco M Nicotra, Roberto Naldi, and Emanuele Garone. “Taut cable control of a tethered UAV”. In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 3190–3195.
- [132] Marco Tognon and Antonio Franchi. “Nonlinear observer-based tracking control of link stress and elevation for a tethered aerial robot using inertial-only measurements”. In: *International Conference on Robotics and Automation (ICRA)*. 2015, pp. 3994–3999.
- [133] Taeyoung Lee. “Geometric controls for a tethered quadrotor UAV”. In: *Conference on Decision and Control (CDC)*. 2015, pp. 2749–2754.
- [134] Zhijie Liu, Jinkun Liu, and Wei He. “Modeling and vibration control of a flexible aerial refueling hose with variable lengths and input constraint”. In: *Automatica* 77 (2017), pp. 302–310.

- [135] Paul Williams and Pavel Trivailo. “Dynamics of circularly towed aerial cable systems, part I: optimal configurations and their stability”. In: *Journal of guidance, control, and dynamics* 30.3 (2007), pp. 753–765.
- [136] Kapseong Ro and James W Kamman. “Modeling and simulation of hose-paradrogue aerial refueling systems”. In: *Journal of guidance, control, and dynamics* 33.1 (2010), pp. 53–63.
- [137] Farhad A Goodarzi, Daewon Lee, and Taeyoung Lee. “Geometric stabilization of a quadrotor UAV with a payload connected by flexible cable”. In: *American Control Conference*. 2014, pp. 4925–4930.
- [138] Karan P Jain et al. “Modeling of aerodynamic disturbances for proximity flight of multirotors”. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2019, pp. 1261–1269.

Appendix A

Proofs

A.1 Proof of Theorem 3.1

Attitude Error Dynamics

As described in earlier sections, attitude is represented using a rotation matrix $R \in SO(3)$ and body-angular velocity $\Omega \in T_R SO(3)$. Errors between different attitude and angular velocities namely, true system, reference system and desired trajectory are different and defined in (2.16) & (2.17), (3.23) & (3.24) and (3.26) & (3.27). We compute the derivatives of the errors between *true model* and *reference model*, (\tilde{e}_R and \tilde{e}_Ω) as shown below.

$$\dot{\tilde{e}}_R = \frac{1}{2} \left(\frac{d}{dt}(R^T \hat{R}) - \frac{d}{dt}(\hat{R}^T R) \right)^\vee, \quad (\text{A.1})$$

Note,

$$\begin{aligned} \frac{d}{dt}(R^T \hat{R}) &= \dot{R}^T \hat{R} + R^T \dot{\hat{R}} \\ &= (R\Omega^\times)^T \hat{R} + R^T (\hat{R}\hat{\Omega}^\times) \\ &= R^T \hat{R} (-\hat{R}^T R\Omega^\times R^T \hat{R} + \hat{\Omega}^\times) \end{aligned} \quad (\text{A.2})$$

Using (2.2), we get

$$\frac{d}{dt}(R^T \hat{R}) = R^T \hat{R} (\hat{\Omega} - \hat{R}^T R\Omega)^\times = R^T \hat{R} \tilde{e}_\Omega^\times, \quad (\text{A.3})$$

and consequently,

$$\frac{d}{dt}(\hat{R}^T R) = -\tilde{e}_\Omega^\times \hat{R}^T R. \quad (\text{A.4})$$

Therefore,

$$\dot{\tilde{e}}_R = \frac{1}{2} (R^T \hat{R} \tilde{e}_\Omega^\times + \tilde{e}_\Omega^\times \hat{R}^T R)^\vee, \quad (\text{A.5})$$

Using the property in (2.3) we get,

$$\dot{e}_R = \frac{1}{2}(Tr[\hat{R}^T R]I - \hat{R}^T R)\tilde{e}_\Omega =: C(R^T \hat{R})\tilde{e}_\Omega, \quad (\text{A.6})$$

where it is shown in [33] that the function $C(R^T \hat{R})$ satisfies the property $\|C(R^T \hat{R})\tilde{e}_\Omega\|_2 \leq 1$ for any rotation matrix in $SO(3)$. Next, from (3.27) we have,

$$\tilde{e}_\Omega = \hat{\Omega} - \hat{R}^T R \Omega \quad (\text{A.7})$$

$$\begin{aligned} &= \hat{\Omega} - \hat{R}^T R_d \Omega_d + \hat{R}^T R_d \Omega_d - \hat{R}^T R \Omega \\ &= (\hat{\Omega} - \hat{R}^T R_d \Omega_d) - \hat{R}^T R (\Omega - R^T R_d \Omega_d) \end{aligned}$$

$$\implies \tilde{e}_\Omega = \hat{e}_\Omega - \hat{R}^T R e_\Omega. \quad (\text{A.8})$$

The derivative of \tilde{e}_Ω then is,

$$\dot{\tilde{e}}_\Omega = \dot{\hat{e}}_\Omega - (-\tilde{e}_\Omega^\times \hat{R}^T R e_\Omega + \hat{R}^T R \dot{e}_\Omega), \quad (\text{A.9})$$

$$\implies J\dot{\tilde{e}}_\Omega = J\dot{\hat{e}}_\Omega - J(-\tilde{e}_\Omega^\times \hat{R}^T R e_\Omega + \hat{R}^T R \dot{e}_\Omega). \quad (\text{A.10})$$

From (3.19b) and (3.25b), we get

$$J\dot{\tilde{e}}_\Omega = \hat{\mu} - J(-\tilde{e}_\Omega^\times \hat{R}^T R e_\Omega + \hat{R}^T R J^{-1}[\mu + \theta]). \quad (\text{A.11})$$

From the control moments defined in (3.31) - (3.32) we have,

$$\begin{aligned} J\dot{\tilde{e}}_\Omega &= \hat{\mu} - J\left(-\tilde{e}_\Omega^\times \hat{R}^T R e_\Omega \right. \\ &\quad \left. + \hat{R}^T R J^{-1}[J R^T \hat{R} J^{-1}(\hat{\mu} + \tilde{k}_R \tilde{e}_R + \tilde{k}_\Omega \tilde{e}_\Omega) \right. \\ &\quad \left. - R^T \hat{R} \hat{\theta} + J R^T \hat{R} \tilde{e}_\Omega^\times \hat{R}^T R e_\Omega + \theta]\right) \end{aligned} \quad (\text{A.12})$$

$$= -\tilde{k}_R \tilde{e}_R - \tilde{k}_\Omega \tilde{e}_\Omega + J \hat{R}^T R J^{-1} R^T \hat{R} (\hat{\theta} - \hat{R}^T R \theta). \quad (\text{A.13})$$

Thus, we finally have,

$$J\dot{\tilde{e}}_\Omega = -\tilde{k}_R \tilde{e}_R - \tilde{k}_\Omega \tilde{e}_\Omega + P \tilde{\theta} \quad (\text{A.14})$$

where P and $\tilde{\theta}$ are defined in (3.34b), (3.29) respectively.

The time-derivative of the configuration error function $\tilde{\Psi}$ (see (3.28)) can be computed and simplified using (2.2) as follows,

$$\dot{\tilde{\Psi}} = -\frac{1}{2}Tr[R^T \hat{R}(\hat{\Omega}^\times - \hat{R}^T R \Omega^\times R^T \hat{R})] \quad (\text{A.15})$$

$$= -\frac{1}{2}Tr[R^T \hat{R}(\hat{\Omega} - \hat{R}^T R \Omega)^\times] \quad (\text{A.16})$$

$$= -\frac{1}{2}Tr[R^T \hat{R} \tilde{e}_\Omega^\times]. \quad (\text{A.17})$$

Then using (2.1) we get,

$$\dot{\tilde{\Psi}} = \frac{1}{2}\tilde{e}_\Omega^T (R^T \hat{R} - \hat{R}^T R)^\vee = \tilde{e}_\Omega^T \tilde{e}_R = \tilde{e}_\Omega \cdot \tilde{e}_R. \quad (\text{A.18})$$

Lyapunov Function Candidate

To show the *exponential input-to-state stability* of the attitude errors, $(\tilde{e}_R, \tilde{e}_\Omega)$, we will consider the following control Lyapunov candidate function,

$$V = \frac{1}{2}\tilde{e}_\Omega \cdot J\tilde{e}_\Omega + \tilde{k}_R\tilde{\Psi}(\hat{R}, R) + c\tilde{e}_R \cdot \tilde{e}_\Omega + \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta}, \quad (\text{A.19})$$

where the adaptation gain, Γ , is a symmetric, positive-definite matrix and \tilde{k}_R is a positive number. Taking the derivative of V gives,

$$\begin{aligned} \dot{V} = & \tilde{e}_\Omega \cdot J\dot{\tilde{e}}_\Omega + \tilde{k}_R\dot{\tilde{\Psi}}(\hat{R}, R) \\ & + c\dot{\tilde{e}}_R \cdot \tilde{e}_\Omega + c\tilde{e}_R \cdot \dot{\tilde{e}}_\Omega + \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}}. \end{aligned} \quad (\text{A.20})$$

Substituting equations (A.6), (A.14), and (A.18) for $\dot{\tilde{e}}_R$, $\dot{\tilde{e}}_\Omega$ and $\dot{\tilde{\Psi}}$ respectively, we get,

$$\begin{aligned} \dot{V} = & \tilde{e}_\Omega \cdot (-k_R\tilde{e}_R - k_\Omega\tilde{e}_\Omega + P\tilde{\theta}) \\ & + k_R\tilde{e}_\Omega \cdot \tilde{e}_R + cC(R^T\hat{R})\tilde{e}_\Omega \cdot \tilde{e}_\Omega \\ & + c\tilde{e}_R \cdot J^{-1}(-k_R\tilde{e}_R - k_\Omega\tilde{e}_\Omega + P\tilde{\theta}) \\ & + \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}}. \end{aligned} \quad (\text{A.21})$$

The resulting \dot{V} can be separated into two parts, one with terms containing $\tilde{\theta}$ and the other without $\tilde{\theta}$,

$$\dot{V} = \dot{V}_{\tilde{\eta}} + \dot{V}_{\tilde{\theta}}, \quad (\text{A.22})$$

$$\begin{aligned} \dot{V}_{\tilde{\eta}} := & \tilde{e}_\Omega \cdot (-\tilde{k}_R\tilde{e}_R - \tilde{k}_\Omega\tilde{e}_\Omega) \\ & + \tilde{k}_R\tilde{e}_\Omega \cdot \tilde{e}_R + cC(R^T\hat{R})\tilde{e}_\Omega \cdot \tilde{e}_\Omega \\ & + c\tilde{e}_R \cdot J^{-1}(-\tilde{k}_R\tilde{e}_R - \tilde{k}_\Omega\tilde{e}_\Omega), \end{aligned} \quad (\text{A.23})$$

$$\dot{V}_{\tilde{\theta}} := \tilde{e}_\Omega \cdot P\tilde{\theta} + c\tilde{e}_R \cdot J^{-1}P\tilde{\theta} + \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}}, \quad (\text{A.24})$$

Simplify (A.23) to get,

$$\begin{aligned} \dot{V}_{\tilde{\eta}} = & -k_\Omega\|\tilde{e}_\Omega\|^2 - ck_R\tilde{e}_R \cdot J^{-1}\tilde{e}_R \\ & + cC(\hat{R}^TR)\tilde{e}_\Omega \cdot \tilde{e}_\Omega - ck_\Omega\tilde{e}_R \cdot J^{-1}\tilde{e}_\Omega. \end{aligned} \quad (\text{A.25})$$

This has the same form as [33, (58)] and thus we have,

$$\dot{V}_{\tilde{\eta}} \leq -\tilde{\eta}^TW\tilde{\eta}, \quad (\text{A.26})$$

where

$$\tilde{\eta} = [\|\tilde{e}_R\| \quad \|\tilde{e}_\Omega\|]^T \quad (\text{A.27})$$

and W is,

$$W = \begin{bmatrix} \frac{ck_R}{\lambda_M(J)} & -\frac{ck_\Omega}{2\lambda_m(J)} \\ -\frac{ck_\Omega}{2\lambda_m(J)} & k_\Omega - c \end{bmatrix}, \quad (\text{A.28})$$

where $\lambda_m(A)$ and $\lambda_M(A)$ are the minimum and maximum eigenvalues of the matrix A , respectively. The constant c is chosen such that W is positive definite resulting in $\dot{V}_{\tilde{\eta}} \leq 0$. From (A.19), let V_1 be,

$$V_1 = \frac{1}{2}\tilde{e}_\Omega \cdot J\tilde{e}_\Omega + \tilde{k}_R\tilde{\Psi}(\hat{R}, R) + c\tilde{e}_R \cdot \tilde{e}_\Omega \quad (\text{A.29})$$

such that $V = V_1 + \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta}$. Then, as shown in detail in [13], we can show V_1 satisfies,

$$\tilde{\eta}^TW_1\tilde{\eta} \leq V_1 \leq \tilde{\eta}^TW_1\tilde{\eta}, \quad (\text{A.30})$$

where,

$$W_1 = \frac{1}{2} \begin{bmatrix} \tilde{k}_R & -c \\ -c & \lambda_m(J) \end{bmatrix}, \quad W_2 = \frac{1}{2} \begin{bmatrix} \frac{2\tilde{k}_R}{2-\psi} & c \\ c & \lambda_M(J) \end{bmatrix} \quad (\text{A.31})$$

with,

$$\tilde{\Psi}(\hat{R}(t), R(t)) \leq \psi < 2, \quad \text{for any } t,$$

and $\tilde{\eta}$ defined in (A.27). V_1 , and $\dot{V}_{\tilde{\eta}}$ are bounded as

$$\lambda_m(W_1)\|\tilde{\eta}\|^2 \leq V_1 \leq \lambda_M(W_2)\|\tilde{\eta}\|^2, \quad (\text{A.32})$$

$$\dot{V}_{\tilde{\eta}} \leq -\lambda_m(W)\|\tilde{\eta}\|^2, \quad (\text{A.33})$$

we can further show that,

$$\dot{V}_{\tilde{\eta}} \leq -\beta V_1, \quad \beta = \frac{\lambda_m(W)}{\lambda_M(W_2)}. \quad (\text{A.34})$$

These results are used later in the proof. Next, from (A.24), we have,

$$\dot{V}_{\tilde{\theta}} = \tilde{\theta}^T(P^T\tilde{e}_\Omega + cP^TJ^{-T}\tilde{e}_R) + \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}}. \quad (\text{A.35})$$

Taking derivative of (3.29) we have,

$$\dot{\tilde{\theta}} = \dot{\tilde{\theta}} + \tilde{e}_\Omega^\times(\hat{R}^TR)\theta - \hat{R}^TR\dot{\theta}. \quad (\text{A.36})$$

Therefore (A.35) can be updated as,

$$\begin{aligned}\dot{V}_{\tilde{\theta}} &= \tilde{\theta}^T [\mathbf{P}^T \tilde{e}_{\Omega} + c\mathbf{P}^T J^{-T} \tilde{e}_R] \\ &\quad + \tilde{\theta}^T \Gamma^{-1} [\dot{\tilde{\theta}} + \tilde{e}_{\Omega}^{\times} (\hat{R}^T R) \theta - \hat{R}^T R \dot{\theta}].\end{aligned}\tag{A.37}$$

This is rewritten as,

$$\begin{aligned}\dot{V}_{\tilde{\theta}} &= \underbrace{\tilde{\theta}^T [\mathbf{P}^T \tilde{e}_{\Omega} + c\mathbf{P}^T J^{-T} \tilde{e}_R + \Gamma^{-1} \dot{\tilde{\theta}}]}_{\triangleq \dot{V}_{\tilde{\theta}_a}} \\ &\quad + \underbrace{\tilde{\theta}^T \Gamma^{-1} [\tilde{e}_{\Omega}^{\times} (\hat{R}^T R) \theta - \hat{R}^T R \dot{\theta}]}_{\triangleq \dot{V}_{\tilde{\theta}_b}}.\end{aligned}\tag{A.38}$$

We use a property of the Γ -projection operator as shown in [53],

$$\tilde{\theta}^T (\Gamma^{-1} Proj_{\Gamma}(\hat{\theta}, y) - y) \leq 0,\tag{A.39}$$

with the projection operator defined in (3.33) and the projection function y defined in (3.34a), we have,

$$\dot{V}_{\tilde{\theta}_a} \leq 0.\tag{A.40}$$

We assume that the uncertainty θ and its time derivative $\dot{\theta}$ are bounded. Furthermore, the projection operator in (3.33) will also keep $\tilde{\theta}$ bounded (see [46] for a detailed proof about these properties.) Thus, we consider the following bounds,

$$\|\tilde{\theta}\| \leq \tilde{\theta}_b \quad \& \quad \|\dot{\tilde{\theta}}\| \leq \dot{\theta}_b \quad \& \quad \|\theta\| \leq \theta_b.\tag{A.41}$$

From (A.38), $\dot{V}_{\tilde{\theta}_b}$ can be bounded as follows,

$$\begin{aligned}\dot{V}_{\tilde{\theta}_b} &= \tilde{\theta}^T \Gamma^{-1} (\tilde{e}_{\Omega}^{\times} (\hat{R}^T R) \theta - \hat{R}^T R \dot{\theta}) \\ &\leq \|\tilde{\theta}\| \|\Gamma^{-1}\| \left(\|\tilde{e}_{\Omega}\| \|\hat{R}^T R\| \|\theta\| + \|\hat{R}^T R\| \|\dot{\theta}\| \right) \\ &\leq \tilde{\theta}_b \|\Gamma^{-1}\| (\|\tilde{e}_{\Omega}\| \theta_b + \dot{\theta}_b).\end{aligned}\tag{A.42}$$

From (A.14), we can show that the \tilde{e}_{Ω} is decreasing for a right choice of \tilde{k}_R , \tilde{k}_{Ω} (since $\tilde{\theta}$ is bounded). Also, from (3.36), $\tilde{e}_{\Omega}(0)$ is bounded. Initial value of \tilde{e}_{Ω} being bounded and \tilde{e}_{Ω} decreasing, implies \tilde{e}_{Ω} is bounded for all time. Therefore, let $\|\tilde{e}_{\Omega}\|$ be bounded by $\tilde{e}_{\Omega b}$. Then,

$$\dot{V}_{\tilde{\theta}_b} \leq \tilde{\theta}_b (\tilde{e}_{\Omega b} \theta_b + \dot{\theta}_b) \|\Gamma^{-1}\|,\tag{A.43}$$

Choosing a large adaptation gain, Γ , would result in a very small Γ^{-1} and thus the right side can be bounded to a small neighborhood δ . Then,

$$\dot{V}_{\tilde{\theta}_b} \leq \delta,\tag{A.44}$$

and thus from (A.40) and (A.44), we have,

$$\dot{V}_{\tilde{\theta}} \leq \delta. \quad (\text{A.45})$$

Substituting (A.34) and (A.45) in (A.22) we get,

$$\dot{V} \leq -\beta V_1 + \delta \quad (\text{A.46})$$

$$\leq -\beta \underbrace{(V_1 + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta})}_V + \beta (\frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}) + \delta \quad (\text{A.47})$$

$$\leq -\beta V + \beta \frac{\tilde{\theta}_b^2}{2} \|\Gamma^{-1}\| + \delta \quad (\text{A.48})$$

We finally have,

$$\dot{V} + \beta V \leq \beta \delta_V \quad (\text{A.49})$$

with $\delta_V \triangleq \frac{\tilde{\theta}_b^2}{2} \|\Gamma^{-1}\| + \frac{\delta}{\beta}$. In (A.49), if $V \geq \delta_V$ it results in $\dot{V} \leq 0$. As a result, by choosing a sufficiently large adaptation gain Γ , the Control Lyapunov Candidate function V , decreases exponentially to result in $V \leq \delta_V$, an arbitrarily small neighborhood δ_V . As shown in (A.49), V is an *exponential Input-to-State Stable* Lyapunov function [52], and thus the attitude errors $(\tilde{e}_R, \tilde{e}_\Omega)$ are exponential input-to-state stable.

A.2 Proof of Lemma 7.1

Proof. Illustration of the differential-flatness is shown in Figure 7.2. For the purpose of proving differential-flatness we redefine the dynamics of the system using tensions in the cable links as given below,

$$\overline{m}_0 \ddot{x}_0 = T_1 - \overline{m}_0 g \mathbf{e}_3, \quad (\text{A.50})$$

$$\overline{m}_a \ddot{x}_a = T_{a+1} - T_a - \overline{m}_a g \mathbf{e}_3, \quad (\text{A.51})$$

$$\overline{m}_b \ddot{x}_b = T_{b+1} - T_b - \overline{m}_a g \mathbf{e}_3 + f_b R_b \mathbf{e}_3, \quad (\text{A.52})$$

$$\overline{m}_b \ddot{x}_n = -T_n - \overline{m}_n g \mathbf{e}_3 + f_n R_n \mathbf{e}_3, \quad (\text{A.53})$$

where $\forall a \in \mathcal{S} \setminus \{0, \mathcal{I}\}$ i.e., all the points excluding the starting point of the cable and those connected to the quadrotors and $\forall b \in \mathcal{I} \setminus \{n\}$ and the quadrotor attitude dynamics are as given in (7.15). Also note $n \in \mathcal{I}$, i.e., end the cable is attached to the quadrotor. Number of inputs in the system are $4n_Q$ corresponding to the thrust and moment of the quadrotors. Number of flat outputs are 3 (for position x_0) + $3(n_Q - 1)$ (3 for each tension $T_{k+1} \forall k \in \mathcal{I} \setminus \{n\}$) and n_Q (for each quadrotor yaw $\psi_j = 4n_Q$).

Making use of the these dynamics we prove the flatness as follows.

- (i) Given, x_0 is a flat-output and therefore we have the cable start position and its derivatives as shown,

$$\{x_0, \dot{x}_0, \ddot{x}_0, x_0^{(3)}, \dots, x_0^{(2n+4)}\}. \quad (\text{A.54})$$

- (ii) Taking derivatives of (A.50) and making use of (A.54) we have tension vector T_1 in the first link and its derivatives,

$$\{T_1, \dot{T}_1, \ddot{T}_1, T_1^{(3)} \dots, T_1^{(2n+2)}\}. \quad (\text{A.55})$$

- (iii) Attitude of the first link is then determined from the tension vector in (A.55) as

$$q_1 = T_1 / \|T_1\| \quad (\text{A.56})$$

and its higher derivatives,

$$\{\dot{q}_1, \dots, q_1^{(2n+2)}\}, \quad (\text{A.57})$$

are computed by taking derivatives of (A.56) and using (A.55).

- (iv) Position and its derivatives of the next link point-mass \bar{m}_1 is computed using (7.1),

$$\{x_1, \dot{x}_1, \ddot{x}_1, \dots, x_1^{(2n+2)}\}. \quad (\text{A.58})$$

- (v) Repeating the steps (ii)-(iv), we can compute the link attitudes, tensions and the positions iteratively till x_b .
- (vi) Using (A.52) and the fact that T_{b+1} is a flat-output (note $b \in \mathcal{I} \setminus \{n\}$) we can compute the thrust in the quadrotor $f_b R_b \mathbf{e}_3$.
- (vii) From x_b , $f_b R_b \mathbf{e}_3$ and their derivatives, the quadrotor attitude, angular velocity R_j, Ω_j and moment M_j can be computed as shown in [26].
- (viii) Rest of the states and inputs for the *multiple quadrotors carrying a flexible cable* segments can be iteratively determined as described above.

□

Proof for Corollary 7.1

.

Proof. For tethered system we have $x_0 \equiv 0$ and T_1 is known since it is a flat-output, i.e., steps (i)-(ii) (see (A.54)-(A.55)). Rest of the proof follows from Lemma 7.1. □

Appendix B

Derivations

B.1 Mapping Operators for Variation-based Extended Kalman Filter

$$\ominus : TS^2 \times TS^2 \rightarrow \mathbb{R}^6:$$

Variation, x_k , between states \mathcal{X}_{k-1} and \mathcal{X}_k on S^2 , ($x_k = \mathcal{X}_k \ominus \mathcal{X}_{k-1}$) The cable attitude and angular velocity corresponding to the states \mathcal{X}_{k-1} and \mathcal{X}_k are given below,

$$\mathcal{X}_{k-1} = \begin{bmatrix} q_{k-1} \\ \omega_{k-1} \end{bmatrix}, \quad \mathcal{X}_k = \begin{bmatrix} q_k \\ \omega_k \end{bmatrix}. \quad (\text{B.1})$$

In order to transform q_{k-1} to q_k while preserving the unit-length of q , we rotate q_{k-1} to q_k as shown in (2.19), where the rotation matrix is represented using exponential map ($\exp[\xi_k^\times]$), (ξ is the angle-axis representation of the rotation). The amount of rotation between q_{k-1} and q_k is given by the angle between the vectors q_{k-1} and q_k ,

$$\theta = \arccos [(q_{k-1} \cdot q_k)]. \quad (\text{B.2})$$

Since q_{k-1} and q_k are unit-vectors, a vector perpendicular to the two unit vectors q_{k-1} , q_k is given by,

$$\vec{\xi}_k = \pm \frac{(q_{k-1} \times q_k)}{\|(q_{k-1} \times q_k)\|}. \quad (\text{B.3})$$

Therefore, axis of rotation is found by finding the unit vector along $\vec{\xi}_k$ resulting in the least amount of rotation i.e.,

$$q_{(k)+} = \exp[(\theta \vec{\xi}_k)^\times] q_{k-1}, \quad (\text{B.4})$$

$$q_{(k)-} = \exp[(-\theta \vec{\xi}_k)^\times] q_{k-1}, \quad (\text{B.5})$$

$$\xi_k = \begin{cases} 0_{3 \times 1} & \text{if } \theta = 0, \\ -\theta \vec{\xi}_k & \text{else if} \\ \theta \vec{\xi}_k & \text{else.} \end{cases} \quad (\text{B.6})$$

where, ξ_k is the infinitesimal variation between q_{k-1} and q_k . The infinitesimal variation in the angular velocity, $(\delta\omega_k)$, is similarly calculated using (2.19) as shown below. Taking time-derivative of (2.19) would result in,

$$\dot{q}_k = \exp[(\xi_k)^\times](\dot{\xi}_k)^\times q_{k-1} + \exp[(\xi_k)^\times]\dot{q}_{k-1}. \quad (\text{B.7})$$

Let $R = \exp[(\xi_k)^\times]$, therefore we have,

$$\dot{q}_k = R(\dot{\xi}_k)^\times q_{k-1} + R\dot{q}_{k-1}, \quad (\text{B.8})$$

$$R^T \dot{q}_k - \dot{q}_{k-1} = -(q_{k-1})^\times \dot{\xi}_k, \quad (\text{B.9})$$

From (4.18), we have, $\dot{\xi}_k = \left(q_{k-1} q_{k-1}^T (\omega_{k-1})^\times \xi_k + (I_{3 \times 3} - q_{k-1} q_{k-1}^T) \delta\omega_k \right)$ and substituting it in (B.9) results in,

$$R^T \dot{q}_k - \dot{q}_{k-1} = -(q_{k-1})^\times \left(q_{k-1} q_{k-1}^T (\omega_{k-1})^\times \xi_k + (I_{3 \times 3} - q_{k-1} q_{k-1}^T) \delta\omega_k \right). \quad (\text{B.10})$$

Note $a^\times a a^T = 0_{3 \times 3}$, for any $a \in \mathbb{R}^3$. Therefore, (B.10) is simplified to,

$$R^T \dot{q}_k - \dot{q}_{k-1} = -(q_{k-1})^\times \delta\omega_k \quad (\text{B.11})$$

Simplifying the above equation and using (2.22), $q \cdot \delta\omega_k = -(\xi_k^\times q) \cdot \omega$ results in,

$$\delta\omega_k = -(R^T \dot{q}_k - \dot{q}_{k-1})^\times (q_{k-1}) - (\omega_{k-1} \cdot (\xi_k^\times q_{k-1})) q_{k-1}. \quad (\text{B.12})$$

Also note that $\dot{q} = \omega \times q$ from (4.15). Finally, variation between \mathcal{X}_{k-1} and \mathcal{X}_k is given by (B.6) and (B.12), i.e., $x_k = \begin{bmatrix} \xi_k \\ \delta\omega_k \end{bmatrix}$.

$$\oplus : TS^2 \times \mathbb{R}^6 \rightarrow TS^2:$$

Calculating the state \mathcal{X}_k by transforming a state \mathcal{X}_{k-1} through a variation x_k ($\mathcal{X}_k = \mathcal{X}_{k-1} \oplus x_k$) Variation x_k and state \mathcal{X}_{k-1} are decomposed as,

$$x_k = \begin{bmatrix} \xi_k \\ \delta\omega_k \end{bmatrix}, \quad \mathcal{X}_{k-1} = \begin{bmatrix} q_{k-1} \\ \omega_{k-1} \end{bmatrix}. \quad (\text{B.13})$$

The transformed q can be calculated similar to (2.19) as follows,

$$q_k = \exp[(\xi_{k-1})^\times] q_{k-1}, \quad (\text{B.14})$$

and the updated angular velocity is calculated using (B.11) as shown below,

$$\dot{q}_{k-1} = \omega_{k-1} \times q_{k-1}, \quad (\text{B.15})$$

$$\dot{q}_k = R_2(\dot{q}_{k-1} - (q_{k-1})^\times \delta \omega_k), \quad (\text{B.16})$$

$$\omega_k = q_k \times \dot{q}_k, \quad (\text{B.17})$$

where $R_2 = \exp[(\xi_k)^\times]$. Therefore, perturbed state is given as,

$$\mathcal{X}_k = \begin{bmatrix} q_k \\ \omega_k \end{bmatrix}. \quad (\text{B.18})$$

B.2 Dynamics for Multiple Quadrotors Carrying a Flexible Cable

In this section, we present the detailed derivation of the equations of motion, (7.13)-(7.15), for the given system. Starting with the principle of least action in (7.6) and substituting for Lagrangian and virtual work from (7.4), (7.5) and (7.7), we have,

$$\begin{aligned} \delta \int \left(\sum_{i \in \mathcal{S}} \frac{1}{2} \bar{m}_i \langle v_i, v_i \rangle - \bar{m}_i g \mathbf{e}_3 \cdot x_i + \underbrace{\sum_{j \in \mathcal{I}} \frac{1}{2} \langle \Omega_j, J_j \Omega_j \rangle}_{\text{rot. energy}} \right) + \\ \int \left(\sum_{j \in \mathcal{I}} \underbrace{\langle W_{1j}, \hat{M}_j \rangle}_{\text{rot. work}} + \langle W_{2j}, f_j R_j \mathbf{e}_3 \rangle \right) dt = 0. \end{aligned} \quad (\text{B.19})$$

Separating and solving the rotational components, we have the following rotational dynamics

$$J_j \dot{\Omega}_j = M - \Omega_j \times J_j \Omega_j, \quad \forall j \in \mathcal{I} \quad (\text{B.20})$$

Taking variation on rest of the equation results in,

$$\int \sum_{i \in \mathcal{S}} \left[\bar{m}_i \langle \delta v_i, v_i \rangle + \delta x_i \cdot \underbrace{(-\bar{m}_i g \mathbf{e}_3 + f_i R_i \mathbf{e}_3 \mathbf{1}_i)}_{u_i} \right] dt = 0.$$

Expanding the summation,

$$\int (\bar{m}_0 \langle \delta v_0, v_0 \rangle + \delta x_0 \cdot u_0 + \bar{m}_1 \langle \delta v_1, v_1 \rangle + \delta x_1 \cdot u_1 + \dots + \bar{m}_n \langle \delta v_n, v_n \rangle + \delta x_n \cdot u_n) dt = 0$$

Replacing the variations $\delta v_i, \delta x_i$ with their expansions (7.10), (7.11), we get,

$$\int \left(\begin{aligned} &(\bar{m}_0 \langle \delta v_0, v_0 \rangle + \delta x_0 \cdot u_0) + \\ &(\bar{m}_1 \langle \delta v_0 - l_1 (q_1^\times \dot{\xi}_1 + \dot{q}_1^\times \xi_1), v_1 \rangle \\ &\quad + (\delta x_0 - l_1 q_1^\times \xi_1) \cdot u_1) + \\ &\quad \vdots \\ &(\bar{m}_n \langle \delta v_0 - \sum_{k=1}^n l_k (q_k^\times \dot{\xi}_k + \dot{q}_k^\times \xi_k), v_n \rangle \\ &\quad + (\delta x_0 - \sum_{k=1}^n l_k q_k^\times \xi_k) \cdot u_n) \end{aligned} \right) dt = 0 \quad (\text{B.21})$$

Using the following simplifications in (B.21),

$$\begin{aligned}\langle -l_k(q_k^\times \dot{\xi}_k + \dot{q}_k^\times \xi_k), v \rangle &= l_k(q_k^\times v) \cdot \dot{\xi}_k + l_k(\dot{q}_k^\times v) \cdot \xi_k, \\ -l_k(q_k^\times \xi_k) \cdot u &= l_k(q_k^\times u) \cdot \xi_k,\end{aligned}$$

and regrouping the respective variations would result in,

$$\int \left(\begin{array}{c} \delta v_0 \cdot \sum_{k=0}^n \bar{m}_k v_k + \delta x_0 \cdot \sum_{k=0}^n u_k + \\ \left[\dot{\xi}_1 \cdot (l_1 q_1^\times \sum_{k=1}^n \bar{m}_k v_k) + \right. \\ \left. \xi_1 \cdot (l_1 \dot{q}_1^\times \sum_{k=1}^n \bar{m}_k v_k + l_1 q_1^\times \sum_{k=1}^n u_k) \right] + \\ \vdots \\ \left[\dot{\xi}_n \cdot (\bar{m}_n l_n (q_n^\times v_n)) + \right. \\ \left. \xi_n \cdot (\bar{m}_n l_n (\dot{q}_n^\times v) + l_n (q_n^\times u_n)) \right] \end{array} \right) dt=0. \quad (\text{B.22})$$

Integration by parts on the respective variation sets results in

$$\int \left(\begin{array}{c} -\delta x_0 \cdot (\bar{m}_0 \dot{v}_0 + \bar{m}_1 \dot{v}_1 + \dots \bar{m}_n \dot{v}_n) + \\ \delta x_0 \cdot (\sum_{k=0}^n u_k) + \\ -\xi_1 \cdot \left(\cancel{l_1 \dot{q}_1^\times \sum_{k=1}^n \bar{m}_k v_k} + l_1 q_1^\times \sum_{k=1}^n \bar{m}_k \dot{v}_k \right) \\ + \xi_1 \cdot \left(\cancel{l_1 \dot{q}_1^\times \sum_{k=1}^n \bar{m}_k v_k} + l_1 q_1^\times \sum_{k=1}^n u_k \right) \\ \vdots \\ -\xi_n \cdot \left(l_n q_n^\times \bar{m}_n \dot{v}_n \right) + \xi_n \cdot \left(l_n q_n^\times u_n \right) \end{array} \right) dt=0, \quad (\text{B.23})$$

and finally,

$$\int \left(\begin{array}{c} \delta x_0 \cdot (\sum_{k=0}^n -\bar{m}_k \dot{v}_k + u_k) + \\ \xi_1 \cdot l_1 q_1^\times \left(\sum_{k=1}^n -\bar{m}_k \dot{v}_k + u_k \right) \\ \vdots \\ \xi_n \cdot \left(-l_n q_n^\times \bar{m}_n \dot{v}_n + l_n q_n^\times u_n \right) \end{array} \right) dt=0. \quad (\text{B.24})$$

By principle of least action the above integral is valid $\forall \delta x_0, \xi_i, t$ and thus, to ensure the above equation to be zero for all time we have,

$$\begin{aligned} & \left(-(\bar{m}_0 \dot{v}_0 + \bar{m}_1 \dot{v}_1 + \dots \bar{m}_n \dot{v}_n) + \sum_{k=0}^n u_k \right) = 0 \\ q_1 \times, & \left(-l_1 q_1^\times \sum_{k=1}^n \bar{m}_k \dot{v}_k + l_1 q_1^\times \sum_{k=1}^n u_k \right) = 0, \\ & \vdots \\ q_n \times & \left(-l_n q_n^\times \bar{m}_n \dot{v}_n + l_n q_n^\times u_n \right) = 0. \end{aligned} \quad (\text{B.25})$$

Expanding the \dot{v}_i we have,

$$\begin{aligned}
& (\bar{m}_0 \dot{v}_0 + \bar{m}_1 (\dot{v}_0 + \sum_{k=1}^1 l_k \ddot{q}_k) + \dots + \bar{m}_n (\dot{v}_0 + \sum_{k=1}^n l_k \ddot{q}_k)) = \sum_{k=0}^n u_k, \\
& l_1 (q_1^\times)^2 (\bar{m}_1 (\dot{v}_0 + \sum_{k=1}^1 l_k \ddot{q}_k) + \dots + \bar{m}_n (\dot{v}_0 + \sum_{k=1}^n l_k \ddot{q}_k)) = l_1 (q_1^\times)^2 \sum_{k=1}^n u_k, \\
& \vdots \\
& l_n (q_n^\times)^2 (\bar{m}_n (\dot{v}_0 + \sum_{k=1}^n l_k \ddot{q}_k)) = l_n (q_n^\times)^2 \sum_{k=n}^n u_k.
\end{aligned}$$

Simplifying the above equations using (7.12) and the following relations,

$$\begin{aligned}
\dot{q} &= \omega \times q, \\
\ddot{q} &= \dot{\omega} \times q + \omega \times \dot{q} = \dot{\omega} \times q - \|\omega\|^2 q \\
(q^\times)^2 \ddot{q} &= q \times (q \times \ddot{q}) = (q \cdot \ddot{q})q - (q \cdot q)\ddot{q} = -(\dot{q} \cdot \dot{q})q - \ddot{q},
\end{aligned}$$

would result in (7.14).

B.3 Variation-based Linearized Dynamics for Multiple Quadrotors Carrying a Flexible Cable

Taking variations with respect to desired states for various states is as follows,

$$\delta q_i = \xi_i^\times q_{id} = -q_{id}^\times \xi_i \quad (\text{B.26})$$

$$\delta(\|\omega_i\|^2) = \delta(\omega_i^\top \omega_i) = 2\omega_{id}^\top (\delta \omega_i) \quad (\text{B.27})$$

$$\delta R_j = R_{jd} \eta_j^\times \quad (\text{B.28})$$

Taking variation on the first row of (7.14),

$$\delta \left(M_{00} I_3 \dot{v}_0 - \sum_{i=1}^n M_{0i} q_i^\times \dot{\omega}_i = \sum_{i=1}^n M_{0i} \|\omega_i\|^2 q_i + \sum_{k=0}^n u_k \right) \quad (\text{B.29})$$

$$\begin{aligned}
& M_{00} I_3 \delta \dot{v}_0 - \sum_{i=1}^n M_{0i} q_{id}^\times \delta \dot{\omega}_i = \\
& \sum_{i=1}^n \left(M_{0i} [(\dot{\omega}_{id}^\times - \|\omega_{id}\|^2 I_3) q_{id}^\times] \xi_i + M_{0i} (2q_{id} \omega_{id}^\top) \delta \omega_i \right) + \sum_{k=0}^n ((\delta f_k) R_{kd} e_3 \mathbf{1}_k + f_{kd} \delta R_{kd} e_3 \mathbf{1}_k) \quad (\text{B.30})
\end{aligned}$$

taking variation on rest of the equations,

$$\delta \left(M_{i0} q_i^\times \dot{v}_0 + M_{ii} I_3 \dot{\omega}_i - \sum_{j=1, j \neq i}^n M_{ij} q_i^\times q_j^\times \dot{\omega}_j = \sum_{k=1}^n (M_{ik} \|\omega_k\|^2 q_i^\times q_k) + l_i(q_i^\times) \sum_{k=i}^n u_k \right) \quad (\text{B.31})$$

$$\begin{aligned} & M_{i0} (\delta(q_i^\times) \dot{v}_{0d} + q_{id}^\times \delta \dot{v}_0) + M_{ii} I_3 \delta \dot{\omega}_i - \\ & \sum_{j=1, j \neq i}^n M_{ij} [\delta(q_i^\times) q_{jd}^\times \dot{\omega}_{jd} + q_{id}^\times \delta(q_j^\times) \dot{\omega}_{jd} + q_{id}^\times q_{jd}^\times \delta \dot{\omega}_j] \\ & = \sum_{k=1}^n (M_{ik} [2q_{id}^\times q_{kd} \omega_{kd}^\top \delta \omega_k + \|\omega_{kd}\|^2 \delta(q_i^\times) q_{kd} + \|\omega_k\|^2 q_i^\times \delta q_{kd}]) + \\ & \quad l_i \delta(q_i^\times) \sum_{k=i}^n u_k + l_i(q_i^\times) \sum_{k=i}^n \delta u_k \end{aligned} \quad (\text{B.32})$$

$$\begin{aligned} & M_{i0} q_{id}^\times \delta \dot{v}_0 + M_{ii} I_3 \delta \dot{\omega}_i - \sum_{j=1, j \neq i}^n M_{ij} [q_{id}^\times q_{jd}^\times \delta \dot{\omega}_j] \\ & = \left[M_{i0} \dot{v}_{0d}^\times - \sum_{j=1, j \neq i}^n M_{ij} ((q_{jd}^\times) \dot{\omega}_{jd})^\times + \|\omega_{jd}\|^2 q_{jd}^\times - l_i \left(\sum_{k=i}^n u_k^\times \right) \right] (-q_i^\times) \xi_i \\ & \quad + \sum_{j=1, j \neq i}^n \left[M_{ij} q_{id}^\times (\dot{\omega}_{jd}^\times - \|\omega_{jd}\|^2 I) q_{jd}^\times \right] \xi_j + \sum_{j=1, j \neq i}^n M_{ij} [2q_{id}^\times q_{jd} \omega_{jd}^\top] \delta \omega_j \\ & \quad + l_i(q_i^\times) \sum_{k=i}^n ((\delta f_k) R_{kd} e_3 \mathbf{1}_k + f_{kd} \delta R_{kd} e_3 \mathbf{1}_k) \end{aligned} \quad (\text{B.33})$$

Finally (B.30) & (B.33) gives the linearized dynamics in (7.19).