Carnegie Mellon University
Department of Mechanical Engineering

Safety-critical Geometric Control Design with Application to Aerial Transportation

Date of Submission: November 27th, 2017

SUBMITTED BY: Guofan Wu

SUPERVISOR: Koushil Sreenath
Department of Mechanical Engineering
Carnegie Mellon University

COMMITTEE MEMBERS: Mark Bedillion
Department of Mechanical Engineering
Carnegie Mellon University

Nathan Michael
Department of Computer Science, the Robotics Institute
Carnegie Mellon University

Bruno Sinopoli
Department of Electrical & Computer Engineering
Carnegie Mellon University

Abstract

Safety constraint are ubiquitous in many robotic applications. For instance, aerial robots such as quadrotors or hexcoptors need to realize a fast collision-free flight, and bipedal robots have to choose their discrete footholds properly to gain the desired friction and pressure contact forces. In this thesis, we address the safety critical control problem for fully-actuated and under-actuated mechanical systems. Since many mechanical systems evolve on nonlinear manifolds, we extend the concept of Control Barrier Function to a new concept called geometric Control Barrier Function which is specifically designed to handle safety constraints on manifolds. This type of Control Barrier Function stems from geometric control techniques and has a coordinate free and compact representation. In a similar fashion, we also extend the concept of Control Lyapunov Function to the concept of geometric Control Lyapunov Function to realize tracking on the manifolds. Based on these new geometric versions of CLF and CBF, we propose a general control design method for fully-actuated systems with both state and input constraints. In this CBF-CLF-QP control design, the control input is computed based on a state-dependent Quadratic Programming (QP) where the safety constraints are strictly enforced using geometric CBF but the tracking constraint is imposed on using relaxation. Through this type of relaxation, the controller could still keep the system state safe even in the cases when the reference is unsafe during some time period. For a single quadrotor, we propose the concept of augmented Control Barrier Function specifically to let it avoid external obstacles. Using this augmented CBF, we could still utilize the idea of CBF-CLF-QP controller in a sequential QP control design framework to let this quadrotor remain safe during the flight. In meantime, we also apply the geometric control techniques to the aerial transportation problem where a payload is carried by multiple quadrotors through cable suspension. This type of transportation method allows multiple quadrotors to share the payload weight, but introduces internal safety constraints at the same time. By employing both linear and nonlinear techniques, we are able to carry the payload pose to follow a pre-defined trajectory.

Keywords: Safety critical geometric control, Geometric Control Lyapunov Function, Control Barrier Function, Geometric Control Barrier Function, Aerial transportation.

Contents

List of Figures

List of Tables

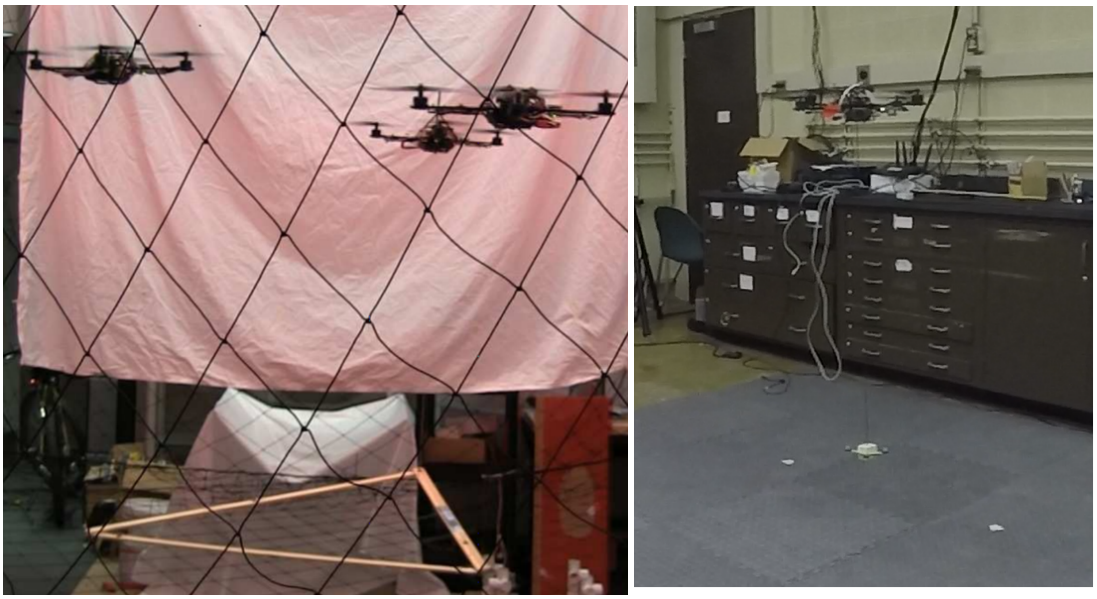
Chapter 1

Introduction

1.1 Motivation

Every robot has some safety constraints to consider during the operation process. Some of these safety constraints arise from the internal hardware limitations such as input saturations and torque limits. For instance, a humanoid robot has angle limits on its internal joints. Exceeding this limit would damage or potentially break the mechanical structure. More imporantly, since many joints are actuated by an electric motor, the maximum torque that can be applied is also limited. Reaching above this torque limit would result in overheating of the motor. Strictly enforcing these contraints would protect the internal hardware of the robot from malfunctioning and thus keep them safe. Other types of safety constraints come from interaction with the environment and collaboration with other robots: a drone has to stay far enough from the wall or other drones to prevent collision; a mobile rover has to pick a proper path to move along so that it won't get trapped in the soil; a bipedal robot has to choose a good place to step on to balance its torso. Failing to satisfy these constraints would result in failures of the robots' missions and cause serious damage. We call the constraints mentioned previously as "safety constraints" since they are highly related to the safety of each robot. Based on this concept of safety constraint, we define the **safety-critical control problem** as *choosing a proper feedback law such that the trajectory of the closed-loop system satisfies the safety constraint all the time.*

In this thesis, we propose a general control design idea for the safety-critical control problem for a particular type of systems, which comprises a payload suspended by a single or multiple quadrotors through cable shown in Fig. 1.1. The main reasons to choose this class of systems are twofold. On one hand, due to the rising of commercial UAVs such as quadrotors or hexcopters, we are able to employ small-sized drones for various tasks in surveillance, building health inspection and item delivery. A traditional way to deliver item is to employ a single, fairly large quadrotor to carry the item through direct attachment. This method



(a) Rigid body payload carried by multiple quadrotors, courtesy of [1].

(b) Point mass payload carried by a single quadrotor in HDR lab.

Figure 1.1: Aerial transportation systems under study.

is easy to implement and test, but this advantage comes at the cost of large increase in the inertia of the quadrotor, making it sluggish. As observed from past research, better agility and manueverability is always desired for such aerial drones as quadrotor. Faced with this major drawback, we find that transporting through cable suspensions is a method to maintain the agility and manueverability of the drone. Also, employing multiple quadrotors could help carry a heavier load through the coordination of these quadrotors. Therefore, the systems we study serve as accurate models for this type of transportation, and could be directly applied in practice. On the other hand, similar safety-crictical control problem has been well-studied in the past for simple systems where only few safety constraints have been considered. However, these methods do not scale very well as the safety constraints increase. For the systems under study, the existences of cables and multiple quadrotors would introduce many safety constraints, making these methods intractable. Hence, we treat them as a good platform to test the scalability of our method.

1.2 Potential Challenges

Safety-critical control problem for mechanical systems is fairly challenging. Take the aerial transportation system shown in Fig. 1.1 for example. This type of mechanical system evolves on a complex nonlinear manifold and has the issue of *singularity* when we employ local coodinates. Representing the dynamics of a complicated system in a compact way could be fairly challenging. Also, these systems are all underactuated,

which makes it impossible to control all the states simultaneously. Addressing the coupling effects in the system dynamics using linearization is another potential challenge. The most important challenge is how to construct proper safety constraints to guarantee that the cable doesn't entangle with the quadrotor and each quadrotor remains far away from each other.

1.3 Thesis Contributions

Compared to the past research on safety-critical control, the main contribution of this thesis could be briefly summarized into the following aspects:

1. We apply the variation idea in [2, 3] to the aerial transportation systems and derive the corresponding dynamics in a compact and coordinate-free way.

2. We propose several initial control designs for different aerial transportation systems: a geometric controller to carry a rigid body payload with more than 6 quadrotors; a variation-based linearization method, using which we could control the position of a point mass payload carried by a single quadrotor using linear quadratic regulator. These controllers could realize stable tracking, but do not strictly enforce the safety constraints.

3. Based on recent development on Control Barrier Function(CBF), we propose the concept of geometric CBF on a general manifold, which is used to address state constraints on the manifold.

4. Combining the original geometric control idea and Control Lyapunov Function(CLF), we propose a general CLF candidate called geometric-CLF for simple mechanical systems, which can realize tracking of any feasible trajectory.

5. Using the concept of geometric CLF and geometric CBF, we propose a safety-critical CBF-CLF-QP control design which utilizes the conditions imposed by CLF and CBF, and computes the control input based on a state-dependent Quadratic Programming(QP). When the online QP is feasible, we are able to satisfy all the safety constraints.

1.4 Thesis Organization

The rest of this thesis is organized in the following way: in Chap. 2, we present a comprehensive review on the past work, relevant to the aerial transportation problem, the safety-critical control problem on various systems, the geometric control design and the recent development of CLF and CBF; in Chap. 3, the necessary mathematical details of each aspect are elaborated on, and we also show the models utilized throughout this

thesis. in Chap. 4, we propose a new linearization technique called geometric linearization using variation on manifolds; in Chap. 5, we show the detailed derivations of two tracking control design for the point mass load and the rigid body load; in Chap. 6, we give the rigorous definition of geometric CLF and geometric CBF on a general manifold, based on which the CLF-CBF-QP control design has been illustrated. This type of CBF-CLF-QP control design only works for fully-actuated mechanical systems; in Chap. 7, we extend the concept of geometric CBF to a specific underactuated system, the quadrotor with different applications. Through the construction of augmented CBFs, we are able to reuse the idea of CBF-CLF-QP control design in a new framework as sequential QP control, which can avoid collision of a single quadrotor with its environment. Through the construction of visual CBFs, we are able to let a single quadrotor track a moving ground object; Chap. 8 presents the results of two groups of aerial transportation experiments. The former group contains a point mass payload carried by a single quadrotor, and the latter group involves a rigid-body payload carried by three quadrotors. Chap. 9 summarizes the thesis contents and provides some potential future directions. Chap. 10 includes some mathematical proofs of theorems used in the thesis.

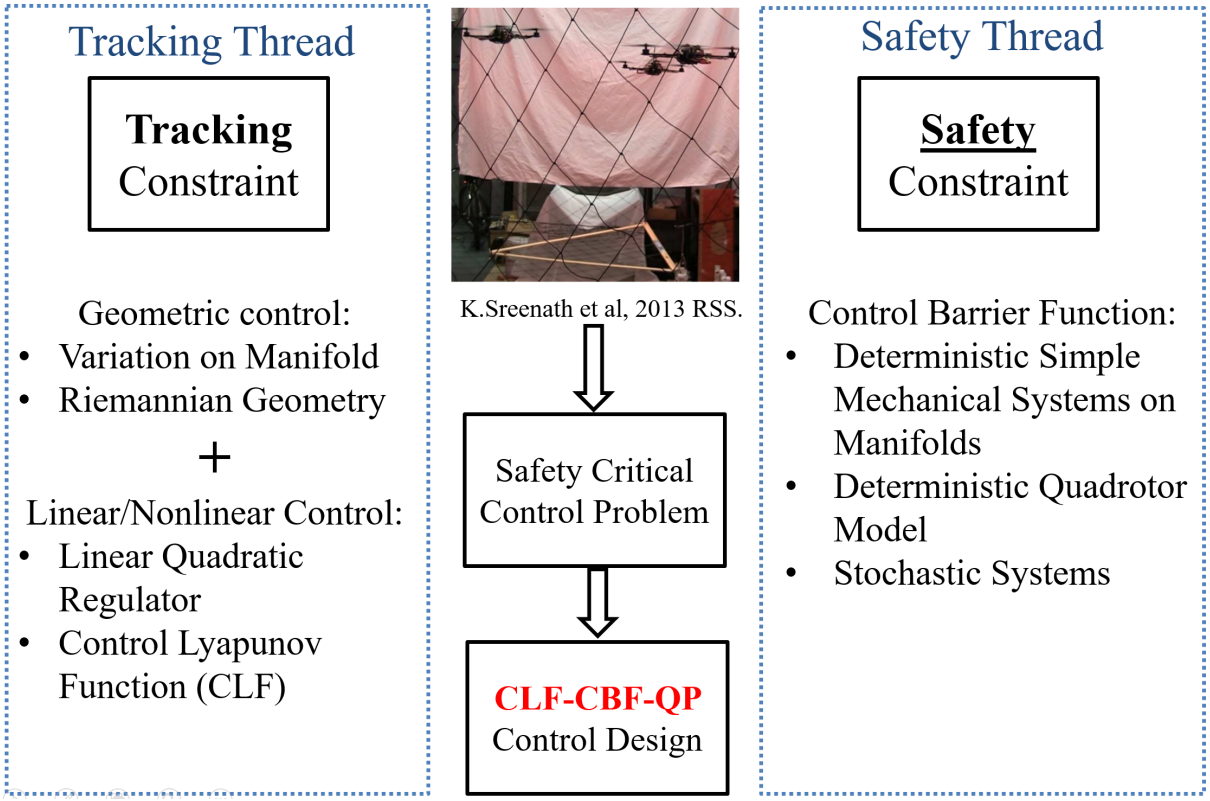


Figure 1.2: This figure summarizes the main methodology in the thesis.

Chapter 2

Literature Survey

In this chapter, we provide a detailed description on past methods in the areas of aerial transportation, control design techniques subject to constraints and geometric control theory designed for mechanical systems. The main reasons for choosing such a wide range of research is that transportating a payload carried by multiple quadrotors is a challenging yet interesting topic. Not only do we have to avoid the collision between each quadrotor which creates a lot of constraints in the state, but also the coupling effects between load and quadrotor so that the pose of load could be adjusted. Hence, coordination control in aerial robotics community and control design subject to constraints can provide a lot of inspirations for us to come up with feasible solutions. In the meantime, as the number of quadrotors increase, the system model becomes very complicated, making it hard to be checked and analyzed in the Cartesian state space representation. However, the recent development in geometric control design has provided a compact way to present the dynamics of mechanical systems, and we want to utilize the same methodology in this research as well.

2.1 Nonlinear Control Design in Aerial Transportation

Transportation using unmanned aerial vehicle (UAV) has become an active research topic in recent years. The rise of this particular problem is due to the emergence of commercialized UAVs such as quadrotors with sophisticated control algorithms and the need of item delivery for areas where ground transportation is expensive or unavailable. Depending on the specific applications, researchers have proposed methods to realize an efficient transportation of the payload in different situations. Relevant research work could be roughly categorized based on the following criteria: the method of attachment, the shape of the load and the number of quadrotors involved in the transportation.

For cases when the payload has a simple geometric shape or is small enough compared to the character-

istic length of the problem, a point mass model is sufficient to capture its dynamics. Research in [4, 5] have studied the problem of point mass load transportation with helicopters. Bernard et al.[4] address the problem of load transportation with multiple helicopters through ropes. To better compensate for the coupling effect between the load and the helicopter, the authors add in a sensor-based observer of the rope's orientation and successfully reduced the error in the load's position. They also provide some experimental results where a point mass payload is transported using a single and three helicopters, where a detailed description of the control architecture can be found in [5]. Since this method only considers the case when the payload is held static, it cannot account for the case when the load's trajectory is time-varying.

Furthermore, there's a substantial amount of research work which has been focused on the transportation of a point mass load using quadrotors from different perspectives in [6, 7, 3, 8, 9, 10]. Palunko et al.[7] studies the problem of planning out a swing free trajectory for the point mass load. Their methodology is to perform Jacobian linearization around an equilibrium point and get the trajectory candidates using Dynamic Programming(DP) approach. Based on this trajectory generation method, the authors in [7] propose an adaptive control scheme where the coupling effect of the cable is modeled as uncertainty applied to the quadrotor. Due to this existence of this amount of uncertainty, we could see a relatively large tracking error of the load's position in experiments. Also, the problem of dynamic load transportation remains. To get a good tracking of load position for dynamic trajectories, we need a clear understanding of the coupling effect between load and quadrotor.

To better understand this coupling effect, ideas of studying quadrotor's dynamics have been further extended to load transportation[8, 9]. A key property discovered for a quadrotor is the differential flatness of the corresponding system dynamics[11]. Sreenath et al.[8] has proven that a single quadrotor with a massless cable suspended load is also differentially flat, whose dynamics is intrinsically hybrid. The authors also develop a geometric controller for this system to let the load track simple a time-varying circular trajectory. To better explore the hybrid part of this system, Tang et al.[9] has proposed a more elaborate planning algorithm based on the previous method in quadrotor planning [12]. By representing the whole trajectory of the load using piecewise polynomials with proper continuity, [9] is able to let the cable switch between staying tout and slack to handle safety constraints. The experimental results given are only limited to planar case and could be extended to three dimensional space.

For cases when the payload admits complicated shape or is relatively large, point mass model is no longer appropriate, and we need to treat it as a rigid body. Research in [13, 14, 15, 16] have studied the problem of transporting a rigid body load using multiple quadrotors from the control aspect. By adding an onboard gripper, Mellinger et al.[13] is able to attach the quadrotor to a rigid body payload. This rigid attachment greatly simplifies the control design but sacrifices the agility of the small quadrotor. Faced with this trade-off, some research utilize cable suspension as an alternative. Wu et al.[14] and Lee et al.[3]

propose different geometric control methods to coordinate multiple quadrotors to carry a rigid body load under the assumption that the cable is massless. The work in [16] models the cable as a chain of pendulum and proposes a linear controller using linearization. Despite the successful simulation results provided in [16], no successful experimental results have been given so far for cable-suspended transportation using multiple quadrotors.

At one end are aerial robots equipped with fixed grippers, where the payload is rigidly attached to the aerial robot through the gripper, and the same control technique for flying without a load is used. These robots are typically characterized by slow, quasi-static motions for hovering and picking up objects [17]. Moreover, carrying a heavy load so close to the body, increases the inertia of the system considerably and thereby makes the system's attitude response very sluggish, significantly degrading performance. Cooperative aerial manipulation using multiple aerial robots equipped with grippers for aerial transportation of loads has also been carried out, [18]. However, once again, the motions are slow, as the load inertia becomes even more significant.

An alternative is to suspend the load through a cable, thereby retaining the agility of the aerial vehicle while still achieving the task of transportation of the suspended load. Although this preserves the fast attitude response of the aerial robot, it introduces additional degrees of underactuation at the cable suspension point. There are several control approaches for the underactuated cable-suspended system in literature. However, early work is split into controllers that rapidly stabilize load swing [19, 20], and/or trajectory generation schemes that achieve fast motion of the load with minimal swing through preshaping Moreover, cooperative load transportation with multiple quadrotors is useful for manipulating large and heavy loads through constrained urban spaces where additional safety is required through redundancy. This is challenging due to the dynamic coupling between the quadrotors through the load. Contrast this to existing results on formation control of multi-agent systems which are not only dynamically decoupled, but are also not subject to switching dynamics and unilateral constraints.

2.2 Control of Systems Subject to Input and State Constraints

Over the past few decades, there have been several approaches towards solving the constrained control problem. Model predictive control (MPC) is a popular technique for constrained control problems that solves an online optimal control problem over a finite-horizon to handle input and state constraints in control design [21, 22, 23, 24, 25]. MPC has been applied to diverse applications from automotive cruise control [26, 27] to legged locomotion [28]. But instead of using the constraints based on CBF and CLF, MPC imposes the actual constraints directly on the optimization problem and includes additional cost for future state. The number of states to look ahead upon is determined by the horizon which needs to be tuned for

stability issue. To boost computational performance and guarantee stability, many variants of MPC have been proposed for specific robotic systems. To get more details about relevant topic, we refer to [21, 22, 29]. A detailed overview of MPC is presented in [21, 22, 23]. The key ideas underlying many of them is to take advantage of the studied systems' structure, and simplify the overall control design. Furthermore, sufficient conditions for the stability of the MPC for a general nonlinear system are presented in [29, 30]. while [28] applies MPC to legged locomotion with a cost computed based on zero moment point (ZMP). In [26, 27], MPC is specifically chosen to realize adaptive cruise control. [24] adds barrier functions to the objective function to handle constrained MPC problems. Despite the fact that designing a MPC controller is fairly straightforward, the control performance is determined by the size of the finite-horizon and the cost function for optimization, and choosing good values of these is challenging. Moreover, for high-dimensional systems with fast dynamics, solving a nonlinear programming problem at real-time speeds is not always feasible leading to latency and lag issues that could lead to instability. Typical solutions to this involve simplifying the dynamics by considering a linear approximation.

Another approach to address constraints point-wise in time is to adjust the reference command using a pre-filter called reference governor [31]. However this typically requires online forward integration of the dynamics which is computationally expensive. Recent results in reduced-order reference governors [32] decompose the states into slow and fast dynamics and implement the reference governor based on the slow states variables only thereby reducing computational complexity, however, this comes at the cost of a smaller domain of attraction.

The safety-critical control problem has also been addressed through reachability analysis, with the core underlying idea being propagating the unsafe region backwards in time based on a worst case analysis [33, 34, 35, 36]. For uncertain systems, the control process can be treated as a two-player differential game where the uncertainty always tries its best to push the system trajectory towards the unsafe region. The evolving dynamics of this unsafe region is determined by a time-invariant Hamilton-Jacobi-Issacs (HJI) partial differential equation [34]. While the method of reachability analysis provides formal guarantees, its computationally expensive and does not scale well for high-dimensional systems.

Motivated by the idea of forward invariance in stability theory, the concept of Barrier functions (BF) have been established to realize safety-critical control [37, 38, 39, 40, 41, 42], wherein the unsafe set is outside a level set of the Barrier function. Logarithmic barrier functions have also been considered in [43, 39] to handle input and output constraints, alongside safety constraints. Barrier functions have also been used to re-target the pose of satellites subject to cone inclusion and exclusion constraints [44, 42, 41]. Barrier functions have also been added to the objective functions to handle constrained MPC problems [24]. However, the concept of the barrier function considered here is limited since the imposed condition, on the time-derivative of the barrier function being non-negative, does not allow the state to transverse the boundary of a level set

of the barrier function. This results in an extremely conservative trajectory for the corresponding closed-loop system, even when the system state is far away from the unsafe region. Furthermore, finding a barrier function for a particular safe set is challenging.

The search for a barrier function with desirable properties on its time-derivative is simplified by the introduction of the concept of a control Barrier function (CBF), where the control inputs explicitly appear in the derivative of the barrier function [45]. Furthermore, to alleviate the conservative condition of the time-derivative of the barrier function being non-negative, [46] relaxes and proposes a new condition for the time-derivative of the barrier function to be negative while still guaranteeing the enforcement of the safety constraint. Similar to the concept of Control Lyapunov Function (CLF) in [47, 48], we are able to generate the control input which can account for the safety constraints through CBF-based optimization. [49] proposes the concept of Control Lyapunov Barrier Function (CLBF) which accounts for both safety constraints and tracking goal at the same time. [46] imposes the conditions of CLF and CBF separately as the constraints of a state-dependent quadratic programming (QP) through relaxation of the CLF condition allowing the reference trajectory to be tracked to be unsafe.

2.3 Geometric Control Methods for Mechanical Systems

Typical mechanical systems have dynamics that evolve on non-Euclidean manifolds. Traditional dynamical models and controllers for such systems are constructed using local parametrizations, such as Euler angles, resulting in dynamics with singularities and controllers that are not valid globally [50]. Although many powerful methods have been developed on Cartesian spaces, most mechanical systems admit non-flat configuration spaces. These non-flat properties come from the orientation of a specific component or certain mechanical joint such as a spherical joint. To apply these methods mentioned previously, we need to construct local coordinates such as Euler angles, which will result in singularities. In many real applications, the existence of singularity could cause serious numerical problems,and thus need to be addressed accordingly. To circumvent this dilemma, geometric control methods have been introduced to obtain almost-global controllers on manifolds [51, 52, 53, 54, 1]. Geometric controllers have been widely applied to fully actuated mechanical systems [51] and even underactuated system such as quadrotors [52, 54, 1]. However, these geometric controllers typically do not take into account input, state, or safety-critical constraints. Recent results in [55, 56] address geometric constraints on SO(3) using geometric reference governors and model predictive control. However, these methods require the discretization of the system and variation integration of the discrete dynamics, which are hard for high-dimensional or coupled nonlinear systems.

2.4 Comparison between Constrained Control Techniques

The previous sections provide a detailed summary on the past research methodologies from different aspects in control theory. To make a clear comparison with past methods, we briefly summarize the advantages and disadvantages of these methods alongside with our solution in Table. 2.1 Note that this summary is based on tutorial papers for each method regardless of the latest research which might have fixed some of the disadvantages mentioned.

Table 2.1: Comparison between different control methods which can handle constraints.

Control Method	Methods/Advantages	Disadvantages
Input Shaping Method[57]	Removes specific frequency oscillation, model-free, easy to implement as a discrete time filter.	Oscillation components useful for tracking or fast convergence are also suppressed
Reference Governor[58]	Serves as a prefilter for a closed-loop system, and is able to handle both state and input constraints. Utilize optimization to compute the adjusted reference.	Needs estimation of the feasibility set which might be intractable for computation for highly nonlinear systems. Faster estimation method could lead to conservative behavior.
Model Predictive Control (MPC) [29]	Handles both state constraints and input saturations through optimization. For specific systems, the control input can be computed from Quadratic Programming efficiently. Could also improve long-term performance with specific horizon.	For general nonlinear systems, the overall optimization is hard to solve, and thus linearization techniques is required for simplification. Also, the horizon and the final-state cost should be carefully selected to ensure stability.
CBF-CLF-QP Control for Control-affine System [59]	Handles state and input constraints using Control Barrier Function with respect to the system dynamics. The control input is always computed based on a CBF-CLF-QP control which can be computed efficiently.	Stability property is still not well-understood, and selecting a specific shape of CBF depends on some intuitive heuristics. For systems with nonlinearity and constraints with higher relative degree, constructing CBF is not trivial.
Geometric CBF-CLF-QP Control [60]	Handles state and input constraints for a special type of fully-actuated mechanical system with a general expression of CLF and CBF. Could also be solved using QP efficiently.	Extension to the general underactuated systems is not studied yet. For environment with complicated geometry, approximation by simple shapes is required.

This chapter provides a high-level summary of the previous research in relevant areas. The next chapter

would cover the models to be studied in this thesis.

Chapter 3

Dynamic Models in the Thesis

In this chapter, we list all the models utilized in the thesis, which can be classified into two categories: single quadrotor model and aerial transportation model. The complexities of these models vary from the simplest planar model case to a fairly complicated system with more than thirty states. In later chapters, the models of a single quadrotor are studied for augmented CBF constructions to handle safety constraints, and the models of transportation systems are investigated for linear and geometric control development. Since we develop one of the transportation model using principles of dynamics, qualitative analysis and discussion of this system is also given.

3.1 Single Quadrotor Models

Quadrotor model and its propeller aerodynamics have been well-studied and identified in the past [13]. When the propeller is rotating at a specific speed, the local air flow would result in a lift force and a torque along the vertical direction as below:

$$f = k_f \omega^2, \quad M = k_M \omega^2, \tag{3.1}$$

where the coefficients k_f, k_M could be determined through experiments.

From this principle, we are able to directly control the external force and the moment along yaw direction for each motor. For the quadrotor with four motors, adjusting the motor speeds would provide a net force

along the yaw direction and a general moment in the body frame in [13, 12] as:

$$\begin{bmatrix} f \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ 0 & k_f L & 0 & -k_f L \\ -k_f L & 0 & k_f L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = K_W \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix},$$

where we ignore the coupling effects of aerial flows between each motor.

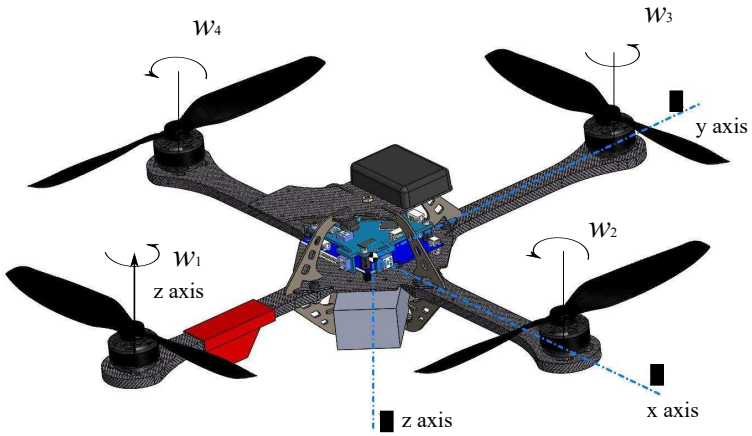


Figure 3.1: Diagram of a hummingbird quadrotor: note that the rotating directions of diagonal pair of motors are the same but the directions of the adjacent motors are opposite. This arrangement is to provide controllability over the yaw direction or otherwise net moment along the yaw direction would always be nonzero. Photo courtesy:`http://wiki.asctec.de/display/AR/CAD+Models`.

Note that the matrix K_W maps a vector of four motor rotating speeds to a subset of wrench to the quadrotor, and it is invertible. Based on this mapping, we would treat the thrust force and the moment as control inputs for later models, denoted as f and M respectively.

3.1.1 Planar Quadrotor Model

Planar quadrotor model is a simple abstraction of the quadrotor system in Fig. 3.1, where we confine a quadrotor to move within a virtual plane as shown in Fig. 3.2a. The corresponding dynamics could be given as:

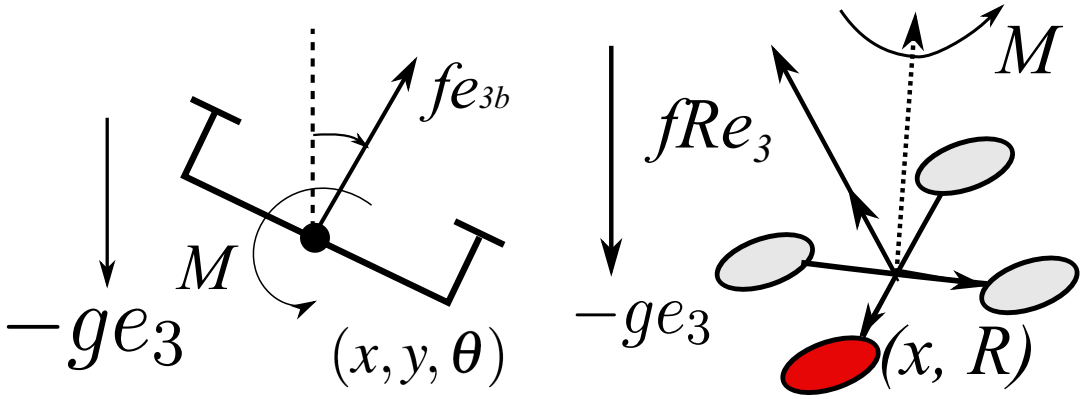
$$\begin{aligned} m\ddot{x} &= f \sin\theta, \\ m\ddot{y} &= f \cos\theta - mg, \\ J\ddot{\theta} &= -M, \end{aligned} \tag{3.2}$$

where g is the acceleration due to gravity, m, J are the mass and inertia of the planar quadrotor, and $f \in \mathbb{R}$, $M \in \mathbb{R}^2$.

We could rewrite this dynamics in state space as:

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

where $\mathbf{x} = \begin{bmatrix} x & y & \theta & \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$ and $\mathbf{u} = \begin{bmatrix} f & M^T \end{bmatrix}^T$ are the state and control input respectively.



(a) Diagram of a Planar Quadrotor

(b) Diagram of a 3D-moving Quadrotor

Since for the planar case, the effect of orientation is only reflected in the pitching angle θ which lies in the group $\mathbb{S}^1$. The geometric effect does not affect control design too much, and thus we could simply design linear control based on Jacobian linearization. In particular, nonlinear control design such as backstepping is employed by *picking the orientation as the feedback variable to be adjusted [61]*, which could be extended to the 3D case.

3.1.2 3D Quadrotor Model

Without the support of external device, planar quadrotor model is unrealistic in robotic applications. So we should proceed to the full 3D model shown in Fig. 3.2b, which captures the behaviors of a real quadrotor like Hummingbird. The corresponding dynamics coud be given as the Newton-Euler equation:

$$\begin{aligned} m\ddot{x} &= fRe_3 - mge_3, \\ \dot{R} &= R\hat{\Omega}, \\ J\dot{\Omega} &= (J\Omega) \times \Omega + M, \end{aligned} \tag{3.3}$$

where $x \in \mathbb{R}^3$ and $R \in SO(3)$ represent the CoM position and the rotation matrix of the body frame as shown in Fig. 3.1, $m > 0, J \in \mathbb{R}^{3\times 3}$ are the total mass and the inertia matrix with respect to the body frame, and

$f, M \in \mathbb{R}^3$.

Remark* 3.1.** *As explained later in Chap.5, T. Lee[62] is the first one to extend the geometric control techniques, which only work for fully-actuated systems, to the quadrotor system, which has two degrees of under-actuation. This control scheme allows the quadrotor to follow a desired time-varying or static reference, which can be decomposed into two parts: the first part treats the quadrotor as a moving point mass, and obtains a desired force based on position error; the second part converts this desired force to a desired orientation and utilize the geometric control on $SO(3)$ group to track the desired orientation with high gain values.* ***The key essence is to design the controller directly using the rotation matrix itself rather than setting up local coordinates*. The underlying methodology is to derive dynamics of complicated systems also in terms of the rotation matrices, and* ***discover useful structures based on this type of representations****.*

3.2 Aerial Transportation Models

In this part, we shift from a single quadrotor to a payload-carrying model. In particular, these two models resemble the real systems shown in Fig.1.1a and Fig. 1.1b respectively.

3.2.1 Point Mass Payload Carried by a Single Quadrotor

Let's first consider the system where a point mass payload is carried by a single quadrotor shown in Fig. 3.3. As studied in [1], the dynamics of this system could be given as:

$$\begin{aligned} \dot{x}_L &= v_L, \\ (m_L + m_Q)(\dot{v}_L + g e_3) &= (q \cdot f R e_3 - m_Q l (\dot{q} \cdot \dot{q}))q, \\ \dot{q} &= \omega \times q, \\ m_Q L \dot{\omega} &= -q \times f R e_3, \\ \dot{R} &= R\hat{\Omega}, \\ J_Q \dot{\Omega} + \Omega \times J_Q \Omega &= M, \end{aligned} \tag{3.4}$$

This type of equation is very convenient for control design since the tension force is implicitly contained in the homogeneous constraint. But due to the unilateral constraint of the cable, to maintain the state where the cable remains taut, we have to express the tension in an explicit way as:

$$T = \frac{m_L}{m_L + m_Q}(m_Q l (\dot{q} \cdot \dot{q}) - q \cdot f R e_3), \tag{3.5}$$

which we would utilize to design controller and test the control performance.

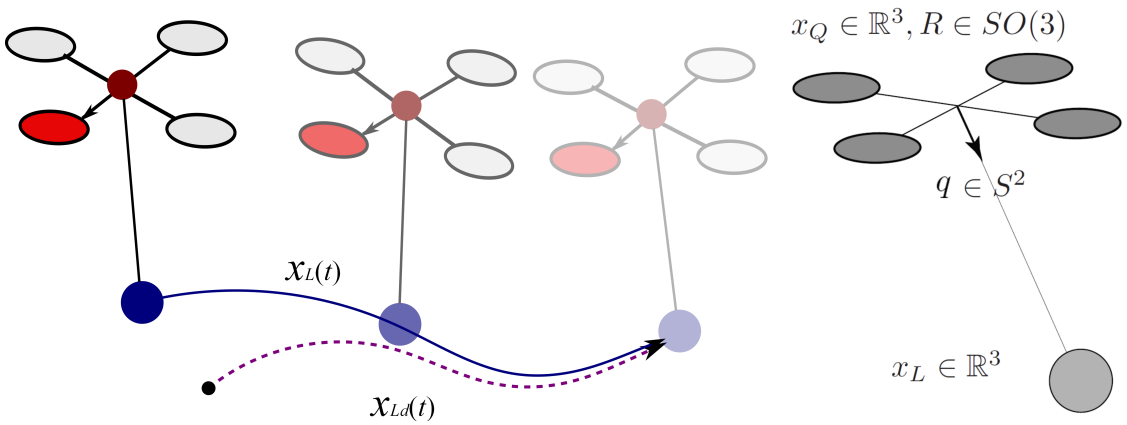


Figure 3.3: Illustration of a single point mass payload carried by a single quadrotor.

We assume a sufficiently smooth trajectory of the payload position, given as $x_{Ld}(t)$. Since this system[8] is differentially flat, all the references could be computed using $x_{Ld}(t)$ and its higher order derivatives including $q_d(t), \omega_d(t), R_d(t), \Omega(t)$ with the corresponding reference input f_d, M_d.

Control Goal: Design a control feedback law for f and M such that the actual load position x_L would track its reference x_{Ld} in a stable way based on noisy measurement of the system state. The steady state error should be small compared to the initial error.

3.2.2 Rigid Body Payload Carried by Multiple Quadrotors

The second model has a rigid body payload carried by multiple quadrotors shown in Fig. 3.4. As shown, $x_L \in \mathbb{R}^3$ is the position of center of mass of the load, $R_L \in SO(3)$ is the rotation matrix of the load from the body frame to the inertial frame, $R_i \in SO(3)$ is the rotation matrix of the i^{th} quadrotor from the body-fixed frame to the inertial frame, where i lies in the set $\{1, 2, \cdots, n\}$, $q_i \in S^2$ is the unit vector, in the inertial frame, from the i^{th} quadrotor to its attachment point on the load, and r_i is the vector from the center-of-mass (CoM) of the load to the attachment point of the i^{th} cable, expressed in the body-fixed frame of the load. Thus, the configuration space of the system is given by $Q = SE(3) \times (S^2 \times SO(3))^n$, and the position of the i^{th} quadrotor given by the following kinematic relation,

$$x_i = x_L + R_L r_i - L_i q_i.$$

We derive the corresponding dynamics in [14]. In this work, the method of Lagrange is used to develop the dynamical equations of motion. The Lagrangian of the system, $\mathcal{L} : TQ \to \mathbb{R}$, is defined by $\mathcal{L} = \mathcal{T} - \mathcal{U}$, where the kinetic and potential energies are denoted as $\mathcal{T} : TQ \to \mathbb{R}$, and $\mathcal{U} : TQ \to \mathbb{R}$, respectively, and are

given by,

$$\begin{aligned}\mathcal{T} &= \frac{1}{2}\{m_L||v_L||^2+\Omega_L^T J_L\Omega_L+ \\ &\quad \sum_{i=1}^{n}(m_i||v_L+R_L\hat{\Omega}_L r_i-L_i\dot{q}_i||^2+\Omega_i^T J_i\Omega_i)\}, \\ \mathcal{U} &= m_L x_L\cdot ge_3+\sum_{i=1}^{n}(m_i(x_L+R_L r_i-L_i q_i)\cdot ge_3).\end{aligned}$$

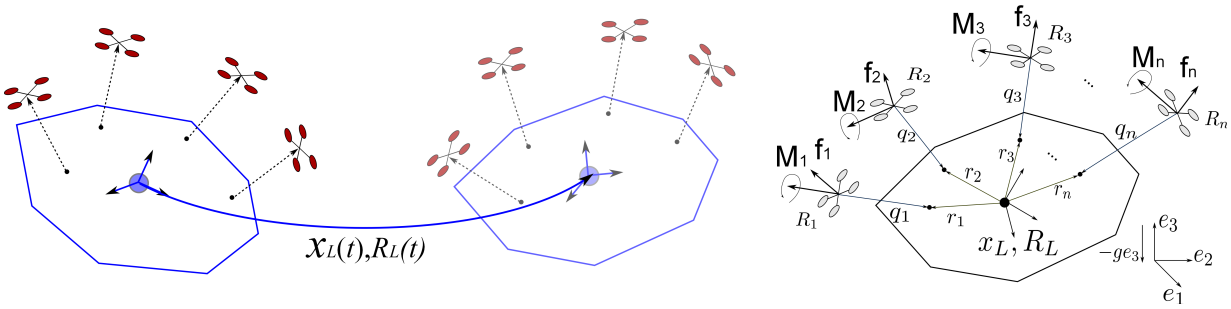


Figure 3.4: Illustration of a rigid body payload carried by multiple quadrotor.

From [69], the dynamics of the system then satisfy the Lagrange-d'Alembert principle,

$$\delta\int_0^\tau \mathcal{L}\,dt+\int_0^\tau\sum_{i=1}^{n}(\langle W_i^a,\hat{M}_i\rangle+W_i^b\cdot f_i R_i e_3)\,dt=0, \tag{3.6}$$

where δ indicates *taking the variation of the total action with respect the actual trajectory in the configuration space*, in other words, the infinitesimal change rate of the value in action equals zero if we perturb the actual trajectory a little bit.

In this equation, f_i is the thrust magnitude of the i^{th} quadrotor, M_i is the moment vector of the i^{th} quadrotor, $\langle\cdot,\cdot\rangle:\mathfrak{so}(3)\times\mathfrak{so}(3)\to\mathbb{R}$ is the inner product on $\mathfrak{so}(3)$, the *hat map* $\hat{}:\mathbb{R}^3\to\mathfrak{so}(3)$ is defined such that $\hat{x}y=x\times y,\forall x,y\in\mathbb{R}^3$, and $W_i^a=R_i^T\delta R_i$, $W_i^b=\delta x_i=\delta x_L+\delta R_L r_i-L_i\delta q_i$ are variational vector fields [63], with the infinitesimal variations satisfying [64, 65, 52],

$$\begin{gathered}\delta q_i=\xi_i\times q_i,\ \xi_i\in\mathbb{R}^3 \text{ s.t. } \xi_i\cdot q_i=0,\\ \delta\dot{q}_i=\dot{\xi}_i\times q_i+\xi_i\times\dot{q}_i,\\ \delta R_i=R_i\hat{\eta}_i,\ \eta_i\in\mathbb{R}^3,\\ \delta\Omega_i=\hat{\Omega}_i\eta_i+\dot{\eta}_i\end{gathered}$$

with δq_i a variation on S^2, and δR_i a variation on $SO(3)$.

The equations of motion are then obtained by ensuring (3.6) is satisfied for all possible variations. These equations comprise of the load pose dynamics, cable attitude dynamics, and the attitude dynamics for each quadrotor:

Load pose dynamics:

$$\begin{aligned}
\dot{x}_L &= v_L, \\
\dot{R}_L &= R_L \hat{\Omega}_L, \\
\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \dot{v}_L + g e_3 \\ \dot{\Omega}_L \end{bmatrix} &= \sum_{i=1}^{n} \begin{bmatrix} b_{i1} \\ b_{i2} \end{bmatrix} (q_i \cdot \frac{f_i}{m_i L_i} R_i e_3 - \omega_i \cdot \omega_i) + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix},
\end{aligned}$$

where each term is defined as,

$$\begin{aligned}
A_{11} &= m_L I_3 + \sum_{i=1}^{n} m_i q_i q_i^T, \quad A_{12} = -\sum_{i=1}^{n} m_i q_i q_{ib}^T \hat{r}_i, \\
A_{21} &= \sum_{i=1}^{n} m_i \hat{r}_i q_{ib} q_i^T = A_{12}^T, \\
A_{22} &= J_L + \sum_{i=1}^{n} m_i (\hat{r}_i q_{ib})(\hat{r}_i q_{ib})^T, \\
b_{i1} &= m_i L_i q_i, \quad b_{i2} = m_i L_i \hat{r}_i q_{ib}, \\
c_1 &= -\sum_{i=1}^{n} m_i (q_{ib}^T \hat{\Omega}_L^2 r_i) q_i, \\
c_2 &= -(\Omega_L \times J_L \Omega_L + \sum_{i=1}^{n} m_i (q_{ib}^T \hat{\Omega}_L^2 r_i) \hat{r}_i q_{ib}).
\end{aligned}$$

Cable attitude dynamics for the i^{th} quadrotor:

$$\dot{q}_i = \omega_i \times q_i, \quad \dot{\omega}_i = q_i \times \left(-\frac{f_i}{m_i L_i} R_i e_3 + \frac{1}{L_i} (\dot{v}_L + g e_3) + \frac{1}{L_i} R_L (\hat{\Omega}_L^2 r_i + \dot{\hat{\Omega}}_L r_i) \right).$$

Attitude dynamics for the i^{th} quadrotor:

$$\begin{aligned}
\dot{R}_i &= R_i \hat{\Omega}_i, \\
\dot{\Omega}_i &= J_i^{-1} (-\Omega_i \times J_i \Omega_i + M_i).
\end{aligned}$$

We can convert these equations into the following compact form as

$$\underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_{A} \begin{bmatrix} \ddot{x}_L + ge_3 \\ \dot{\Omega}_L \end{bmatrix} = \underbrace{\begin{bmatrix} G_1 & G_2 & \cdots & G_n \end{bmatrix}}_{G} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} + \underbrace{\begin{bmatrix} d_1 \\ d_2 \end{bmatrix}}_{d}, \tag{3.7}$$

$$\begin{aligned} \dot{R}_L &= R_L \hat{\Omega}_L, \\ \dot{q}_i &= \omega_i \times q_i, \\ \dot{\omega}_i &= -\hat{q}_i u_i + q_i \times \frac{1}{L_i}(\ddot{x}_L + ge_3 + R_L(\hat{\Omega}_L^2 + \dot{\hat{\Omega}}_L) r_i), \\ \dot{R}_i &= R_i \hat{\Omega}_i, \\ \dot{\Omega}_i &= J_i^{-1}(-\Omega_i \times J_i \Omega_i + M_i), \end{aligned}$$

where

$$G_i = \begin{bmatrix} m_i L_i q_i q_i^T \\ m_i L_i \hat{r}_i q_{ib} q_i^T \end{bmatrix}, \quad u_i = \frac{f_i}{m_i L_i} R_i e_3,$$

and

$$\begin{aligned} d_1 &= -\sum_{i=1}^{n} m_i (q_{ib}^T \hat{\Omega}_L^2 r_i + L_i(\omega_i \cdot \omega_i)) q_i, \\ d_2 &= -\Omega_L \times J_L \Omega_L - \sum_{i=1}^{n} m_i (q_{ib}^T \hat{\Omega}_L^2 r_i + L_i(\omega_i \cdot \omega_i))(\hat{r}_i q_{ib}). \end{aligned}$$

The system considered has $5n+6$ degrees-of-freedom (DoFs) and $4n$ actuators, thereby having $n+6$ degrees-of-underactuation. Similarly for later analysis, the tension in the the i^{th} cable can be computed as

$$T_i = q_i \cdot [\ddot{x}_L + ge_3 + R_L(\dot{\hat{\Omega}}_L + \hat{\Omega}_L^2) r_i - f_i R_i e_3] + m_i L_i (\omega_i \cdot \omega_i), \tag{3.8}$$

and we would consider the control problem for this system as:

Control Goal: Design a control feedback law for f_i and M_i for the $i^{(th)}$ quadrotor such that the actual load position x_L and load orientation R_L would track their references $x_{Ld}(t), R_{Ld}(t)$ in a stable way subject to noisy measurement.

3.2.3 Qualitative Analysis of System Dynamics

From the dynamics of the load, we see that the matrix G is a *projection operator* that maps the external force being applied on the system by the quadrotors to a *wrench* that's applied to the rigid-body payload.

Simplfying further, the load pose dynamics can be written as,

$$A \begin{bmatrix} \ddot{x}_L + g e_3 \\ \dot{\Omega}_L \end{bmatrix} = G^* \begin{bmatrix} f_1 R_1 e_3 \cdot q_1 \\ f_2 R_2 e_3 \cdot q_2 \\ \vdots \\ f_n R_n e_3 \cdot q_n \end{bmatrix} + d,$$

where the new matrix

$$G^* = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \\ \hat{r}_1 q_{1b} & \hat{r}_2 q_{2b} & \cdots & \hat{r}_n q_{nb} \end{bmatrix}.$$

Moreover, the matrix A on the left hand side can also be represented as,

$$A = M_L + G^* M_Q G^{*T}$$

where

$$M_L = \begin{bmatrix} m_L I_3 & 0_{3\times 3} \\ 0_{3\times 3} & J_L \end{bmatrix}, \quad M_Q = diag[m_1, m_2, \cdots, m_n].$$

***Remark* 3.2.** *This concise expression for A clearly illustrates its physical meaning. In particular, A is the* state-dependent generalized inertia *of the system, where the term M_L is the intrinsic inertia of the load and the term $G^* M_Q G^{*T}$ is the induced inertia due to the attitude change of each cable.*

Next, similar to the case in [66], we decompose the control input into the following two parts,

$$u_i^{\parallel} = (q_i \cdot u_i) q_i = q_i q_i^T u_i,$$
$$u_i^{\perp} = (I_3 - q_i q_i^T) u_i = -\hat{q}_i^2 u_i,$$

corresponding to parallel and perpendicular components of u_i with respect to the cable attitude, q_i. Further, noting that $-\hat{q}_i u_i = -\hat{q}_i u_i^{\perp} = \hat{q}_i^3 u_i$, we have,

$$G \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = G \begin{bmatrix} u_1^{\parallel} \\ u_2^{\parallel} \\ \vdots \\ u_n^{\parallel} \end{bmatrix}.$$

***Remark* 3.3.** *From the above equation, it is evident that the load pose dynamics is* only *affected by the net effect of all the $u_i^{\parallel}$. In particular, $u_i^{\perp}$ does not affect the load pose.*

***Remark* 3.4.** *Furthermore, note that both $u_i^{\parallel}$ and $u_i^{\perp}$ affect the attitude of the i^{th} cable, q_i. The difference lies*

in the fact that $u_i^{\perp}$ *appears* explicitly *in the dynamics of* q_i*, where as* $u_i^{\parallel}$ *affects* q_i *only* implicitly *through the load attitude* R_L.

As we shall see in later chapters, all of the system models presented here would be analyzed for different tasks in aerial transportation.

Chapter 4

Variation on Manifolds and Geometric Linearization Methods

This chapter presents another linearization techniques to realize tracking on manifolds. This method is called geometric linearization, where we utilize the variation instead of the difference in local coordinates as a measure between the desired trajectory and the actual trajectory. The main reason for introducing such a linearization method is **not to replace Jacobian linearization method, but rather serve as a complement.** As seen in previous chapter, nearly all the models provided are given in terms of the rotation matrices, rather than the local coordinates such as Euler angles. The existence of rotation matrices makes Jacobian linearization infeasible since we cannot determine the distance between two rotation matrices by direct subtraction. To resolve this dilemma, we convert the idea in [67, 68] where the error dynamics is given in terms of variations to the method of geometric linearization. The first two sections would be focused on introducing some mathematical details, and the third section provides a detailed description of linearization process. Then the fourth section goes through this process intuitively for some mechanical systems and the last section presents the corresponding simulation results. In particular, we include the point mass load carried by a single quadrotor as one example.

4.1 Variations on Manifolds

To linearize along a reference trajectory on a manifold, we need to take variations with respect to the reference trajectory. We do this through variational vector fields [63], such that the perturbed trajectory is also on the manifold. 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. For a more formal

explanation, we refer to Chapter 5-7 in [69].

By exploiting the specific variation expressions on $SO(3)$ and $\mathbb{S}^2$, we are able to parameterize the error directly based on rotation matrices and unit vectors. In this specific error representation, the variation-based linearized dynamics can be derived from the original nonlinear dynamics. We can then design a controller based on the variation-based linearized dynamics which is locally valid on the nonlinear system.

Thus, we will address the following three questions below: what are the error states, what type of constraints should be satisfied on these error states, and finally in order to perform state feedback, how can we design a controller based on the current error state. We will address the first question below, based on [66], and the other questions in the subsequent sections.

4.1.1 Variation in $SO(3)$

In $SO(3) := \{R \in \mathbb{R}^{3\times 3} \mid R^T R = I, \det(R) = +1\}$, the infinitesimal variation with respect to a reference $R_d(t) \in SO(3)$ is given by [66],

$$\delta R(t) = \frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} R_d \exp(\varepsilon\hat{\eta}) = R_d(t)\hat{\eta}(t),$$

where $\eta \in \mathbb{R}^3$. The corresponding infinitesimal change in body angular velocities can be given as:

$$\delta\Omega(t) = \hat{\Omega}_d(t)\eta(t) + \dot{\eta}(t). \tag{4.1}$$

So if we assume that the actual rotation matrix $R(t)$ is close enough to the desired rotation $R_d(t)$, the state $s = [\eta(t), \delta\Omega(t)]^T$ can treated as a linear approximation of the errors between the desired and actual state of the system on $SO(3)$. Note that $\eta(t)$ and $\delta\Omega(t)$ are dynamically coupled through the relationship specified in (4.1).

Given $R(t), R_d(t)$ as the actual and reference trajectory, the actual errors between these two trajectories are given below [70] :

$$\begin{aligned} e_R(t) = &\quad \tfrac{1}{2}(R_d^T(t)R(t) - R^T(t)R_d(t))^\vee, \\ e_\Omega(t) = &\quad \Omega(t) - (R^T(t)R_d(t))\Omega_d(t), \end{aligned}$$

where

$$\hat{\Omega}(t) = R^T(t)\dot{R}(t), \quad \hat{\Omega}_d(t) = R_d^T(t)\dot{R}_d(t),$$

and the map $\Omega_d \mapsto R^T R_d \Omega_d$ is called the transport map which allows comparison between tangent vectors at different points. Here we assume that this linear error s is small enough so that they coincide with the

actual error. Thus it holds that

$$s = \begin{bmatrix} \eta \\ \delta\Omega \end{bmatrix} \approx \begin{bmatrix} e_R \\ e_\Omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(R_d^T(t)R(t) - R^T(t)R_d(t))^\vee \\ \Omega(t) - (R^T(t)R_d(t))\Omega_d(t) \end{bmatrix}, \tag{4.2}$$

which we would use for feedback control on $SO(3)$ in later sections.

4.1.2 Variation in $\mathbb{S}^2$

In $\mathbb{S}^2 := \{q \in \mathbb{R}^3 \mid q \cdot q = 1\}$, the infinitesimal variation with respect to a reference $q_d(t) \in \mathbb{S}^2$ is given by [66],

$$\delta q(t) = \frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} \exp(\varepsilon\hat{\xi}(t))q_d(t) = \widehat{\xi(t)}q_d(t),$$

where $\xi \in \mathbb{R}^3$, s.t., $\xi \cdot q_d = 0$. The corresponding infinitesimal change in angular velocity is denoted as $\delta\omega(t)$, with the angular velocity defined as $\omega = q \times \dot{q}$.

From [66], the constraints imposed on ξ and $\delta\omega$ can be given below as:

$$\xi \cdot q_d = 0, \quad (\xi \times q_d) \cdot \omega_d + q_d \cdot \delta\omega = 0, \tag{4.3}$$

The first constraint comes from the variation expression. The second constraint can be derived based on the fact that $\omega \cdot q = 0$. Then the variation of this term $\delta(\omega \cdot q) = \delta\omega \cdot q_d + \omega_d \cdot \delta q = 0$. It follows that $\omega_d \cdot (\xi \times q_d) + q_d \cdot \delta\omega = 0$.

Assuming that the actual direction $q(t)$ is close enough to thedesired direction $q_d(t)$ again, the approximated error can be specified as $s = [\xi(t), \delta\omega(t)]^T$ under constraint (4.3).

For $\mathbb{S}^2$, the actual error between $q(t), q_d(t)$ is given as

$$e_q(t) = \widehat{q_d}(t)q(t), \quad e_\omega = \omega(t) - (-\hat{q}^2)\omega_d(t), \tag{4.4}$$

with $\omega_d \mapsto -\hat{q}^2\omega_d$ being the transport map.

Applying the same assumption, we would use the following formula for feedback control on $\mathbb{S}^2$ in later sections:

$$s = \begin{bmatrix} \xi \\ \delta\omega \end{bmatrix} \approx \begin{bmatrix} e_q \\ e_\omega \end{bmatrix} = \begin{bmatrix} \widehat{q_d}(t)q(t) \\ \omega(t) - (-\hat{q}^2)\omega_d(t) \end{bmatrix}. \tag{4.5}$$

4.2 Controllability of Linear Time-Varying Systems

To establish some qualitative result on the variation system, the following theorem about controllability is very useful. Given a general linear time-varying system as,

$$\dot{x} = A(t)x + B(t)u,$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathcal{C}^\infty(\mathbb{R}^{n\times n})$ and $B \in \mathcal{C}^\infty(\mathbb{R}^{m\times n})$ Let's define a linear operator $\mathscr{A}^k = (\frac{d}{dt} + A(t))^k$ (similar to the right operator in [71]) as follows,

$$\begin{aligned}
\mathscr{A}^0(B(t)) &:= B(t), \\
\mathscr{A}(B(t)) &:= \frac{d}{dt}B(t) + A(t)B(t), \\
\mathscr{A}^k(B(t)) &:= \mathscr{A}(\mathscr{A}^{k-1}(B(t)).
\end{aligned}$$

Then a condition that implies controllability of the above linear time-varying system is, [72],

$$\forall t, \quad \text{rank}(B, \mathscr{A}B, \cdots, \mathscr{A}^{n-1}B) = n.$$

4.3 Variation-based Linear Quadratic Regulator for Reference Tracking

Given a system with configuration space $X = X_1 \times X_2 \times \cdots \times X_n$ where X_i $(i = 1, 2, \cdots, n)$ is one of the following three manifolds $\mathbb{R}^3, \mathbb{S}^2, SO(3)$. We list its state variable in three separate groups,

$$\begin{aligned}
\mathbb{R}^3 &: (x_1, \dot{x}_1, \cdots, x_q, \dot{x}_q) && \text{(translational dynamics)} \\
\mathbb{S}^2 &: (q_1, \omega_1, \cdots, q_k, \omega_k) && \text{(joint dynamics)} \\
SO(3) &: (R_1, \Omega_1, \cdots, R_l, \Omega_l) && \text{(orientational dynamics)}
\end{aligned}$$

where each $x_i, \dot{x}_i \in \mathbb{R}^3$ represents a rigid body CoM's position and velocity, each $q_i \in \mathbb{S}^2, \omega_i \in \mathbb{R}^3$ reflects the relative position and angular velocity of the link connecting each pair of adjacent rigid bodys, and each $R_i \in SO(3), \Omega_i \in \mathbb{R}^3$ are the rotation matrix and body-fixed angular velocity of each rigid body.

The control input for this system is denoted by $u \in \mathbb{R}^m$. Given a dynamically-feasible reference trajectory $x_d(t)$ with the reference input $u_d(t)$ to follow this trajectory, the following paragraph shows how to derive the variation dynamics around it in a symbolic way.

Assumptions of Variation-based Methods:

1. The system dynamics is of 2nd order: This means that the control input u would only appear in the time derivative of $\dot{x}_i, \omega_j, \Omega_p$ where $i \in [1,q],\ j \in [1,k],\ p \in [1,l]$.

2. The system is control affine with respect to u.

3. The dynamic model of the system only consists of vector addition, dot product, cross product and matrix multiplication with a matrix or vector.

According to Assumption (1),(2), the system dynamics could be written out explicitly as:

$$\begin{aligned} \frac{d}{dt}x_i &= \dot{x}_i, & \frac{d}{dt}\dot{x}_i &= f_i + A_i u, \quad i = 1,2,\cdots,q, \\ \frac{d}{dt}q_j &= \omega_j \times q_j, & \frac{d}{dt}\omega_j &= g_j + B_j u, \quad j = 1,2,\cdots,k, \\ \frac{d}{dt}R_p &= R_p \hat{\Omega}_p, & \frac{d}{dt}\Omega_p &= h_p + C_p u, \quad p = 1,2,\cdots,l, \end{aligned}$$

where each $f_i, g_j, h_p \in \mathbb{R}^3,\ A_i, B_j, C_p \in \mathbb{R}^{3\times m}$ that are state-dependent vector and matrix functions.

Steps for Generating the Variation-based Linear Dynamics:

- Step 1: We start by taking variation on both sides of the above system dynamics. On the left hand side, we simply add a δ in front of each time-derivative symbol. On the right hand side, we apply the following formulas recursively to get the variation of the functions f_i, g_j, h_p:

$$\begin{gathered} \delta(x+y) = \delta x + \delta y,\ \delta(x \times y) = \delta x \times y_d + x_d \times \delta y, \\ \delta(x \cdot y) = \delta x \cdot y_d + x_d \cdot \delta y,\ \delta(R_1 x) = \delta R_1 x_d + R_{1d}\delta x, \\ \delta(R_1 R_2) = \delta R_1 \cdot R_{2d} + R_{1d}\delta R_2, \end{gathered}$$

  where $x, y \in \mathbb{R}^3$ and $R_1, R_2 \in \mathbb{R}^{3\times 3}$, and δ represents the variation. For example, the variation of $\dot{x} = x \times (Ry)$ will result in a variation dynamics as:

$$\begin{aligned} \delta\dot{x} = \delta(x \times (Ry)) &= x_d \times \delta(Ry) + \delta x \times R_d y_d \\ &= x_d \times (\delta R y_d + R_d \delta y) + \delta x \times R_d y_d. \end{aligned}$$

- Step 2: For the control input u, the formulas below are applied to obtain:

$$\begin{aligned} \delta(A_i u) &= \delta A_i u_d + A_{id}\delta u, \quad \delta(B_j u) = \delta B_j u_d + B_{jd}\delta u, \\ \delta(C_p u) &= \delta C_p u_d + C_{pd}\delta u, \end{aligned}$$

where $i \in [1,q],\ j \in [1,k],\ p \in [1,l]$, and the variation $\delta A_i, \delta B_j, \delta C_k$ are according to Step 1. Here the variable δu is the control input for which a linear controller will be designed in later sections.

***Remark* 4.1.** *Note that we use the symbol δ to represent the 1st order approximation of the actual error on manifold. The formulas presented here are just based on* Chain rule *for matrix-valued functions. Recall that from Taylor's formula, the difference of a function $f : \mathbb{R}^n \to \mathbb{R}$ has the expression:*

$$f(x) - f(x_d) = \nabla f(x_d) \cdot \underbrace{(x - x_d)}_{\delta x} + o(||x - x_d||) \implies \delta f = \nabla f(x_d) \cdot \delta x,$$

which is exactly the case when the variable is a vector.

But here what we consider as variables here could also include matrix.

- Step 3: The resulting variation terms will be in terms of $\eta_j, \delta\Omega_j$ for $SO(3)$, and $\xi_p, \delta\omega_p$ for $\mathbb{S}^2$, as shown in Sec. 4.1. These can be rearranged into a linear system, which forms the variation-based linear dynamics.

***Remark* 4.2.** *It must be noted that this process is carried out symbolically, without specifying an explicit reference trajectory. As we will see, the resulting linear dynamics will be in terms of a symbolic reference trajectory, enabling making conclusions on the controllability properies of the linear system without explicitly choosing a specific reference trajectory.*

4.4 Examples of Variation-Based Linearization

The method described in the previous section develops the variation-based linearization of a nonlinear system about a reference trajectory. Here we will illustrate the method through three concrete examples: A 3D pendulum, a spherical pendulum, and a quadrotor with a cable-suspended load (see Fig. 6.5 and Fig. 3.3). In general, the variation dynamics can be written as the linear system below:

$$\dot{s} = A(x_d(t))s + B(x_d(t))\delta u, \tag{4.6}$$

$$C(x_d(t))s = 0, \tag{4.7}$$

where $s = \{\delta x_i, \delta \dot{x}_i, \xi_j, \delta\omega_j, \eta_p, \delta\Omega_p\}$ is the variation for each component introduced in Section 4.1, $\delta u \in \mathbb{R}^m$ is the linear control input, and $C \in \mathbb{R}^{k \times N}$ reflects any constraint that is introduced due to the geometric structure of the manifold.

The linear system produced by the variation-based linearization method represents the linearized dynamics of the errors with respect to the desired reference trajectory about which the nonlinear system was

linearized. As can be seen above, the linear system depends on the desired reference trajectory. As we will see, we will design linear controllers for this linearized system to drive the error states $s(t)$ to zero, which in turn will result in tracking the desired reference trajectory for the nonlinear system. However, before we do that, since the system matrices and thus the controllable subspace depend on the desired reference trajectory, we need to answer a fundamental question: For what desired trajectories is the variation-based linearization of the nonlinear system controllable? Or, in other words, do there exist desired reference trajectories that render the the above linear time-varying system uncontrollable?

Finally, if there exists constraints, i.e., $C \not\equiv 0$ in (4.7), then we need to check if state trajectories that respect the constraints are controllable. To do this we introduce the concept of controllability under state constraints.

***Definition* 1.** State-Constrained Controllability*: A linear system $\dot{s} = A(t)s + B(t)u$ with state constraints $C(t)s = 0$ is said to be state-constrained controllable, i.e., controllable under the state constraints, if its constraint subspace is invariant and is covered by the controllable subspace.*

If a system that is state-constrained controllable as per the definition above, then any state that respects the constraints can be driven to the origin while guaranteeing that the constraint will be feasible for all time. Note that this definition is different to the *constrained controllability* definition in literature, [73, 74], wherein the input (and not the state) is constrained. We will see the above better through the examples below.

4.4.1 3D Pendulum

A 3D pendulum comprises of a rigid body that is attached to a frictionless pivot and subject to gravity. With the state variable $x = (R, \Omega) \in SO(3) \times \mathbb{R}^3$, the system dynamics can be shown as (see [66] for details):

$$\dot{R} = R\hat{\Omega},$$
$$J\dot{\Omega} + \Omega \times J\Omega = -mg\rho \times R^T e_3 + u.$$

Here $J \in \mathbb{R}^{3\times 3}$ is the inertia matrix of the pendulum about the pivot, $R \in SO(3)$ the rotation matrix of this body representing its orientation with respect to the inertial frame, g is the scalar gravity constant and $\rho \in \mathbb{R}^3$ the displacement vector from the pendulum's pivot to its center of mass in its body-fixed frame.

Taking variation on both sides about the desired reference trajectory (R_d, Ω_d), we get the following

variation dynamics:

$$\begin{aligned}\delta\dot{R} &= \delta R\hat{\Omega}_d + R_d\delta\hat{\Omega},\\ J\delta\dot{\Omega} &= -\delta\Omega\times J\Omega_d - \Omega_d\times J\delta\Omega - mg\rho\times\delta R^T e_3 + \delta u,\end{aligned}$$

Note that since the model here is simple enough, Steps 1,2, of Section 4.1 are combined together. Using the variation in $SO(3)$ from Section 4.1 as $\delta R = R_d\hat{\eta},\ \eta\in\mathbb{R}^3$, and substituting for δR, $\delta\dot{R}$ in the above dynamics, we can simplify the above equation into a linear system below:

$$\frac{d}{dt}\begin{bmatrix}\eta\\ \delta\Omega\end{bmatrix} = \begin{bmatrix}-\hat{\Omega}_d & I_3\\ -mg\hat{\rho}\widehat{R_d^T e_3} & J^{-1}(\widehat{J\Omega_d} - \hat{\Omega}_d J)\end{bmatrix}\begin{bmatrix}\eta\\ \delta\Omega\end{bmatrix} + \begin{bmatrix}0_{3\times 3}\\ I_3\end{bmatrix}\delta u,$$

which is of the form (4.6)-(4.7), with A & B obtained from above, with $C=0$, and the state s as defined in (4.2).

***Remark* 4.3.** *Note that this is an implicitly time-varying linear system, with the system matrices (A,B,C) only dependent on the desired reference trajectory. As we will see, the system properties, like controllability, and invariant subspaces (with respect to the dynamics) are also dependent on the desired reference trajectory.*

We have the following result.

***Proposition* 4.1.** *The linear time -varying system obtained as the linearization of the nonlinear 3D pendulum system about a desired reference trajectory is controllable for all desired trajectories.*

Proof. See Appendix 10.1 □

***Remark* 4.4.** *It's remarkable that we can analytically verify the controllability of a time-varying system resulting from linearization of a nonlinear system along a trajectory, especially without explicitly specifying the trajectory as a function of time. This is only possible because of the coordinate-free formulation and the variation-based linearization.*

4.4.2 Spherical Pendulum

A spherical pendulum comprises of a mass attached to a fixed point through a suspended cable. The state of this system is $x=(q,\omega)\in\mathbb{S}^2\times\mathbb{R}^3$ with the dynamics:

$$\begin{aligned}\dot{q} &= \omega\times q,\\ ml\dot{\omega} &= q\times(f - mge_3).\end{aligned}$$

Here $q \in \mathbb{S}^2$ is a unit vector that specifies the attitude of the spherical pendulum, $\omega \in \mathbb{R}^3$ is the angular velocity of the spherical pendulum, m is the mass, l is the length, g is the scalar acceleration due to gravity, e_3 is the third directional vector, and f is the controlled force exerted on the spherical pendulum.

Suppose we are given a smooth reference trajectory (q_d, ω_d) to track. Based on Chain rule, taking variation around the reference q_d, ω_d yields the following variation dynamics,

$$\delta\dot{q} = \delta\omega \times q_d + \omega_d \times \delta q,$$
$$ml\delta\dot{\omega} = \delta q \times (f_d - mge_3) + q_d \times \delta f.$$

Substituting for the variation expression $\delta q = \xi \times q_d, \xi \in \mathbb{R}^3, \xi \cdot q_d = 0$ on $\mathbb{S}^2$ (from Section 4.1), and its time-derivative $\delta\dot{q}$ yields the following error dynamics:

$$\frac{d}{dt}\begin{bmatrix}\xi\\ \delta\omega\end{bmatrix} = \begin{bmatrix} q_d q_d^T \hat{\omega}_d & I_3 - q_d q_d^T \\ \widehat{(f_d - mge_3)}\hat{q}_d/ml & 0_{3\times 3}\end{bmatrix}\begin{bmatrix}\xi\\ \delta\omega\end{bmatrix} + \begin{bmatrix}0_{3\times 3}\\ \hat{q}_d/ml\end{bmatrix}\delta f,$$

with the constraint on the states,

$$\begin{bmatrix} q_d^T & 0_{1\times 3} \\ -\omega_d^T \hat{q}_d & q_d^T\end{bmatrix}\begin{bmatrix}\xi\\ \delta\omega\end{bmatrix} = 0.$$

This is once again of the form (4.6)-(4.7), with A, B, and C defined from above and with the state s as defined in (4.5).

Note that the above system is a constrained linear time-varying system. Traditional analysis would require forming the zero dynamics of this system, as in [71], which is fairly involved. As we will see next, the variation-based linear system obtained by the linearization along a desired reference trajectory is controllable for any reference trajectory that satisfies the constraints.

***Proposition* 4.2.** *The linear-time varying system obtained as the linearization of the nonlinear spherical pendulum system about a desired reference trajectory is* state-constrained controllable*, i.e., it's controllable for all desired trajectories that respect the constraints.*

Proof. We will demonstrate this by establishing that the system is state-constrained controllable. We will do this by showing that the constraint space is time invariant, i.e., $\frac{d}{dt}(Cs(t)) \equiv 0$, and that the controllable subspace covers the constraint space, i.e., $\mathcal{R}(\begin{bmatrix}B & \mathscr{A}B & \cdots & \mathscr{A}^{n-1}B\end{bmatrix}) \supset \mathcal{N}(C)$. We will do this through the following lemmas. □

Lemma 4.3. *The constraint space of the variation-based linearized error dynamics of the spherical pendulum is time invariant, i.e.,* $\frac{d}{dt}(Cs(t)) \equiv 0$.

Proof. See Appendix 10.2. □

***Remark* 4.5.** *From the above lemma, the value $Cs(t)$ is conserved, i.e., if the initial condition satisfies $Cs(t_0) = 0$, then $Cs(t) = Cs(t_0) = 0, \quad \forall t \geq t_0$. Thus, as long as the initial condition starts in the constraint space, the system's trajectory is the same as the unconstrained one that evolves according to $\dot{s} = A(x_d(t))s + B(x_d(t))\delta u$. So we could put the constraint aside and treat this system as an unconstrained one.*

Lemma 4.4. *The Nullspace of the constraint matrix is given by the column span of the matrix N, and the orthogonal complement of the nullspace of the constraint matrix is given by the column span of $N^{\perp}$, where*

$$N = \begin{bmatrix} 0 & 0 & \tilde{\omega} & \hat{q}_d\tilde{\omega} \\ \tilde{\omega} & \hat{q}_d\tilde{\omega} & 0 & -q_d \end{bmatrix}, \; N^{\perp} = \begin{bmatrix} q_d & \hat{q}_d\tilde{\omega} \\ 0 & q_d \end{bmatrix},$$

where,

$$\tilde{\omega} = \begin{cases} \omega_d, & \omega_d \neq 0, \\ \kappa, \; s.t. \; q_d^T\kappa = 0, & \omega_d = 0. \end{cases}$$

Proof. We can check that $CN = 0$, $CN^{\perp} \neq 0$, and $N^T N^{\perp} = 0$. In these computations, we make use of the fact that $q_d^T\omega_d = 0$. Also note that N is a 6×4 matrix and $N^{\perp}$ is a 6×2 matrix. In particular, due to the above identities, the columns of $N, N^{\perp}$ form a full set of basis for $\mathbb{R}^6$, i.e., colspan($\begin{bmatrix} N & N^{\perp} \end{bmatrix}$) $= \mathbb{R}^6$. □

***Remark* 4.6.** *It's remarkable that we can analytically write down the nullspace of a time-varying matrix, that resulted from linearization along a trajectory, without specifying the trajectory explicitly as a function of time.*

Lemma 4.5. *The controllable subspace of the linearization of the nonlinear spherical pendulum system includes the nullspace of the constraint matrix for all desired trajectories.*

Proof. See Appendix 10.3. □

***Remark* 4.7.** *Since the linearized error should always stay within the nullspace of the constraint matrix according to our derivation, results established by the previous lemma guarantee that the origin can be reached from any point that stays in the nullspace, i.e dynamically feasible. Thus the variational linearization of the spherical pendulum is controllable under state-constraints.*

4.4.3 Single Quadrotor UAV with a Cable-Suspended Load

After applying this technique to two very simple mechanical systems, we now consider a slightly more complicated system that comprises a quadrotor UAV with a cable-suspended pointmass load, with dynamics

shown in Eq. (3.4). The variation-based linearized error dynamics can be derived as (see Appendix. 10.4):

$$\frac{d}{dt}\begin{bmatrix}\delta x_L\\ \delta v_L\\ \xi\\ \delta\omega\\ \eta\\ \delta\Omega\end{bmatrix}=\begin{bmatrix}0 & I_3 & 0 & 0 & 0 & 0\\ 0 & 0 & A_{23} & A_{24} & A_{25} & 0\\ 0 & 0 & A_{33} & A_{34} & 0 & 0\\ 0 & 0 & A_{43} & 0 & A_{45} & 0\\ 0 & 0 & 0 & 0 & -\hat{\Omega}_d & I_3\\ 0 & 0 & 0 & 0 & 0 & A_{66}\end{bmatrix}\begin{bmatrix}\delta x_L\\ \delta v_L\\ \xi\\ \delta\omega\\ \eta\\ \delta\Omega\end{bmatrix}+\begin{bmatrix}0 & 0\\ b_{21} & 0\\ 0 & 0\\ b_{41} & 0\\ 0 & 0\\ 0 & B_{62}\end{bmatrix}\begin{bmatrix}\delta f\\ \delta M\end{bmatrix}, \tag{4.8}$$

with the constraint matrix

$$C=\begin{bmatrix}0_{1\times 6} & q_d^T & 0_{1\times 3} & 0_{1\times 6}\\ 0_{1\times 6} & -\omega_d^T\hat{q}_d & q_d^T & 0_{1\times 6}\end{bmatrix},$$

where each term is defined as,

$$\begin{aligned}
A_{23} =& -\frac{1}{m_Q+m_L}[(q_d\cdot f_d R_d e_3 - m_Q L(\dot{q}_d\cdot\dot{q}_d))I_3\\
&+ f_d q_d (R_d e_3)^T]\hat{q}_d\\
A_{24} =& \frac{2m_Q L}{m_Q+m_L}q_d\dot{q}_d^T\hat{q}_d,\ A_{25} = -\frac{f_d}{m_L+m_Q}q_d q_d^T R_d\hat{e}_3\\
A_{33} =& q_d q_d^T\hat{\omega}_d,\ A_{34} = I_3 - q_d q_d^T\\
b_{21} =& \frac{1}{m_L+m_Q}q_d q_d^T R_d e_3,\ A_{43} = -\frac{f_d}{m_Q L}\widehat{R_d e_3}\hat{q}_d\\
A_{45} =& \frac{f_d}{m_Q L}\hat{q}_d R_d\hat{e}_3,\ b_{41} = -\hat{q}_d R_d e_3\\
A_{66} =& J_Q^{-1}(\widehat{J_Q\Omega_d} - \hat{\Omega}_d J_Q),\ B_{62} = J_Q^{-1}.
\end{aligned}$$

Proving the constrained controllability of this system using direct method is intractable since we need to take higher order time-derivatives for both A and B.

4.5 Simulation Results on Several Mechanical Systems

We use the variation-based linearization and controllers presented in the previous sections to perform several simulations to test the effectiveness of our proposed method. Additional results are also provided for comparison to show the robustness of this method. An interesting fact about all the systems studied here is that all of them are *differentially flat*, [76], wherein knowing the time-varying trajectory of a set of flat outputs enables us to analytically compute the time trajectories of the entire state as well as the control input that results in this state trajectory through higher order time derivatives of the flat output. This enables

planning dynamically-feasible reference trajectories very easily. It must be noted that the flat outputs need not be just a subset of the states, but rather can be a function of the states, the inputs, and higher order time derivatives of the inputs. We now present simulation results for these three systems presented in the paper.

4.5.1 3D Pendulum

The rotation matrix, R_d, corresponding to the rigid body orientation, forms a set of flat outputs for this system since the angular velocity can be computed from $\dot{R}_d$ and the control can be computed from the time derivative of the angular velocity and the system parameters. We use the system parameters $l = 1,\ \rho = 0.5e_3, J = diag(0.1006, 0.1006, 0.0127), m = 0.2827$, and choose the flat output as,

$$R_d(t) = \begin{bmatrix} \cos\omega_0 t & \sin\omega_0 t & 0 \\ -\sin\omega_0 t & \cos\omega_0 t & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \omega_0 = 1.5.$$

The following matrices are used to design the LQR controller, $Q_1 = \mathrm{diag}(2,2,2,5,5,5),\ Q_2 = 0.5I_3,\ P_T = 2.5I_6$. The initial condition for simulation is specified as

$$R_0 = \mathrm{diag}(1,-1,-1), \quad \Omega_0 = [-1.5, 0.8, 1.0]^T.$$

This corresponds to a maximum possible error in orientation. As we will see, the linear controller has a large domain of attraction. Figure 4.1 shows the simulation results for the 3D pendulum, illustrating tracking of the reference trajectory. The errors e_R, e_Ω in the figure are errors on $SO(3)$, computed as, [62],

$$e_R = \frac{1}{2}(R_d^T R - R^T R_d)^\vee, \quad e_\Omega = \Omega - R^T R_d \Omega_d.$$

The configuration error is computed as,

$$\Psi_R = \mathrm{trace}(I - R_d^T R)/2,$$

with $\Psi_R = 0$ when $R = R_d$, and $\Psi_R = 2$ when there is a 180° error between R and R_d. As can be seen from the figure, the controller is able to stabilize large initial errors in attitude. Here we demonstrate the controller recovering from the largest possible attitude error with $\Psi_R = 2$. However, we must note that although the linear controller results in exponential stability on the nonlinear system for small (local) errors, the controller only results in asymptotic stability for large (global) errors. This is evident in the plot of the configuration error Ψ_R in Figure 4.1. Further, note that the initial configuration error $\Psi_R = 2$ corresponds to one of the equilibrium points for the open-loop 3D pendulum with $e_R = 0$. The controller is able to recover from this

initial error, albeit slowly, due to the feedforward component of the control.

4.5.2 Spherical Pendulum

The spherical pendulum system we are considering consists of a mass attached to a fixed point through a suspended cable. It turns out that the flat output for this system is the tension force vector in the cable. This flat output is a function of both the state and the control input on the system. The tension vector is a flat output since the cable orientation specified by $q_d \in S^2$ can be obtained from the tension vector, the angular velocity can be obtained from q_d and it's time-derivative, and finally the control input can be obtained from the time-derivative of the angular velocity and the system properties. We specify the following flat output to generate the reference trajectory for tracking:

$$T_d(t) = 5[\cos\frac{\pi}{6}\cos 1.5t, \cos\frac{\pi}{6}\sin 1.5t, \sin\frac{\pi}{6}]^T,$$

with the initial conditions $q_0 = [-\sqrt{2}/2, 0, -\sqrt{2}/2]^T$, $\omega_0 = [0, 1.5, 0]^T$. For the simulation, we select the following gain matrices are used, $Q_1 = diag(50, 50, 50, 15, 15, 15), Q_2 = 0.25I_3$ and $P = I_6$. The error functions e_q, e_ω, Ψ_q are errors on $\mathbb{S}^2$, computed as, [77],

$$e_q = \hat{q}_d q, \quad e_\omega = \omega - (-\hat{q}^2)\omega_d, \quad \Psi_q = 1 - q \cdot q_d.$$

Fig. 4.2 shows relevant tracking results of the LQR controller designed on the variation-based linearization and applied to the spherical pendulum system. As can be seen from the figure, stability can be guaranteed using our method even for the cases with very large initial error.

4.5.3 Single Quadrotor UAV with a Cable-Suspended Load

This quadrotor with a cable-suspended load system is also differentially flat, with the load position x_L and quadrotor yaw ϕ as flat outputs, see [75] for more details. We thus specify the following flat output to generate the reference trajectory and nominal input,

$$x_{Ld}(t) = [\cos t, \sin t, 0.5t]^T, \ \phi_d(t) \equiv 0.$$

For the LQR control design, the weighing matrices are set as,

$$Q_1 = \begin{bmatrix} Q_{11} & 0 & 0 \\ 0 & Q_{22} & 0 \\ 0 & 0 & Q_{33} \end{bmatrix},$$

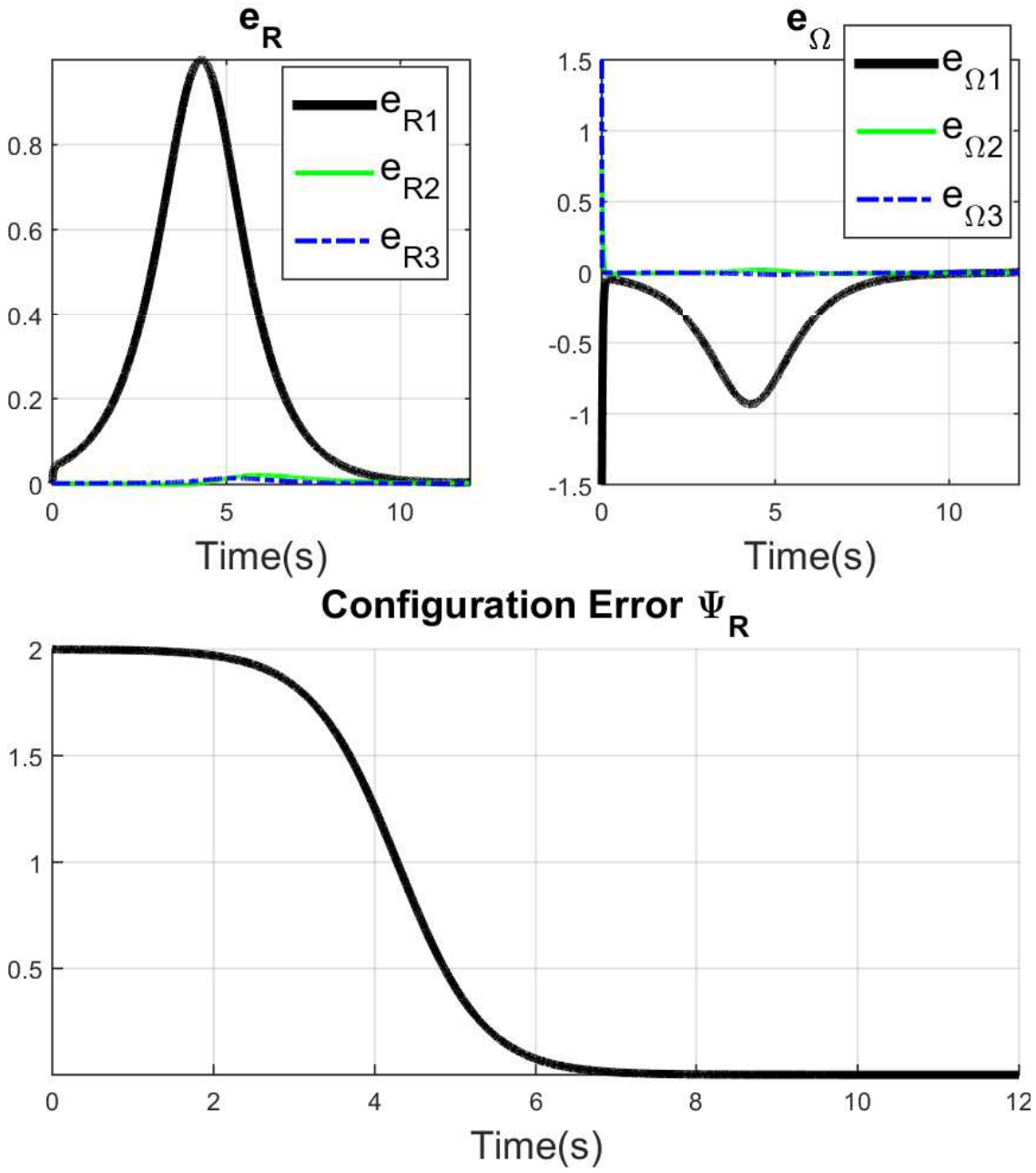


Figure 4.1: Tracking errors obtained by simulating the variation-based linearization controller for the 3D pendulum system described in Section 4.5.1. In particular, the vector error function for the position and velocities, e_R, e_Ω, and the configuration error function, Ψ_R, are shown as functions of time. Here we set the initial configuration error $\Psi_R = 2$, the maximum possible orientation error, and the linear controller is still able to asymptotically track the reference trajectory.

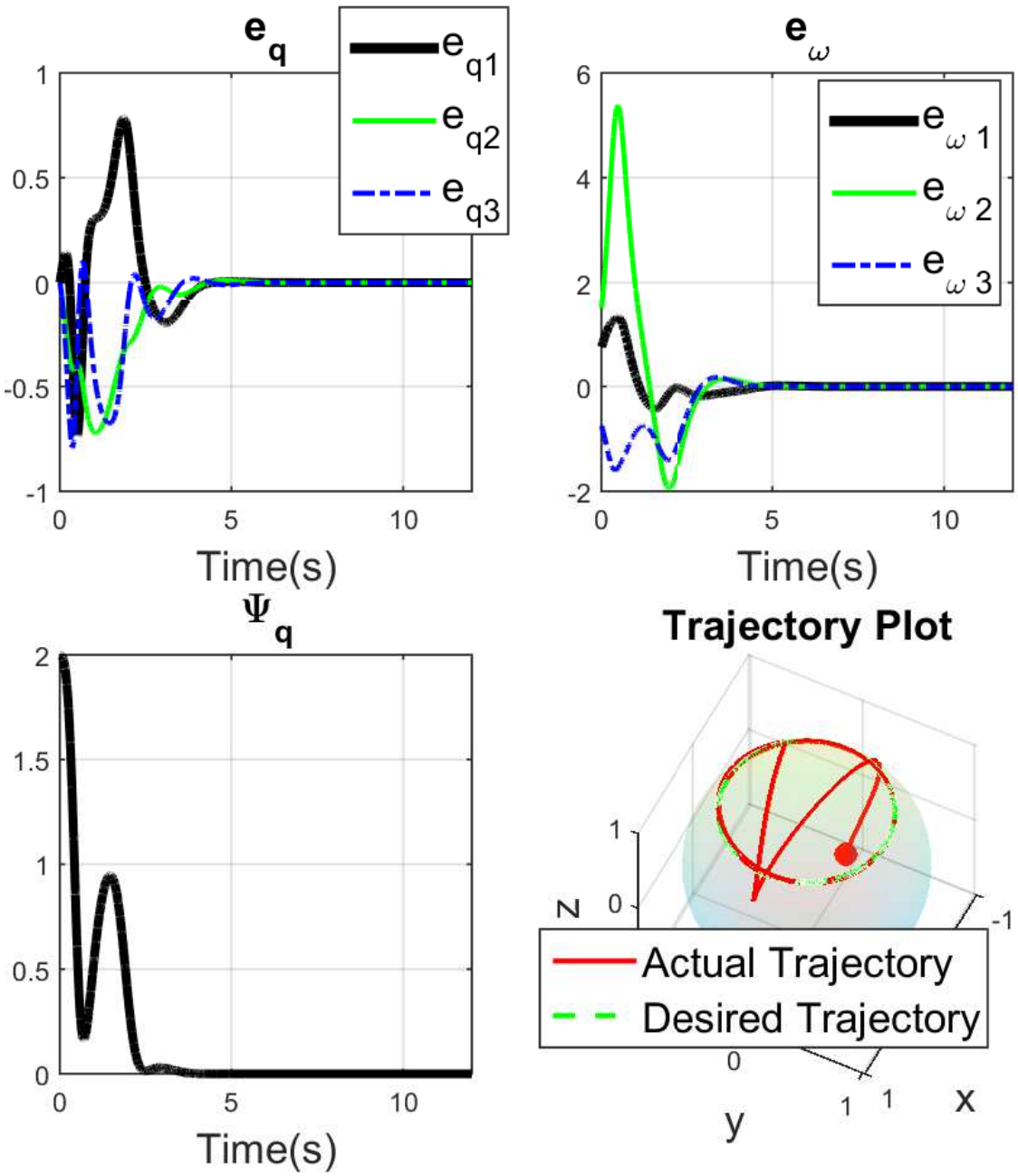


Figure 4.2: Tracking errors obtained by simulating the variation-based linearization controller for the spherical pendulum system, as described in Section 4.5.2, is shown. In particular, the vector error function for the position and velocities, e_q, e_ω, and the configuration error function, Ψ_q, are shown as functions of time. In addition, the desired trajectory is shown on the unit sphere with the actual trajectory tracking the desired.

where the matrix blocks are given as,

$$Q_{11} = 0.5 \cdot I_6,\ Q_{22} = I_3,\ Q_{33} = 0.75 I_9,$$
$$Q_2 = 0.2 I_4,\ P_T = 0.01 \cdot I_{18}.$$

We evaluate the performance of the controller in three trials by specifying different initial conditions for the 18-dimensional state, as tabulated in Table. 4.1. Figure 4.3 illustrates the convergence to the reference trajectory starting at the three initial conditions. Note that, for Trial 1, the quadrotor is initially inverted, and the controller is still able to track the specified reference trajectory, illustrating that the linear controller has a large domain of attraction. Figure 4.4 depicts the tracking performance of the controller for the three initialconditions. As can be seen, in all the cases, the translational error in the quadrotor position, and the rotational errors for both the quadrotor orientation and the cable attitude go to zero. This illustrates the validity of the proposed method for higher-dimensional systems and demonstrates the large domain of attraction that is possible through a linear controller.

So we could see that the controller is able to track the reference with respect to different initial conditions through we haven't provided a rigorous proof for the stability of this point-mass payload system. This shows that the geometric-linearization techniques can be applied to stabilize the transportation systems with linear control design methods. In next chapter, we would employ the geometric control techniques as another alternative to the transportation of a rigid-body payload.

Table 4.1: Initial Conditions for Simulation

Initial State	Trial 1	Trial 2	Trial 3
x_{L0}	$[0,5,-1.5]^T$	$[0,0,0]^T$	$[-1,0,1]^T$
v_{L_0}	$[0,0,0]^T$	$[0,-1,-1]^T$	$[0,0,0]^T$
q_0	$[0,0,-1]^T$	$[0;\sqrt{3}/2,-0.5]^T$	$[\sqrt{2}/4,\sqrt{6}/4,\sqrt{2}/2]^T$
ω_0	$[0,0,0]^T$	$[-0.5,0,0]^T$	$[0,0,0]^T$
R_{Q0}	$diag(1,-1,-1)$	I_3	I_3
Ω_{Q0}	$[0,0,0]^T$	$[1.5,0,1]^T$	$[0,0,0]^T$

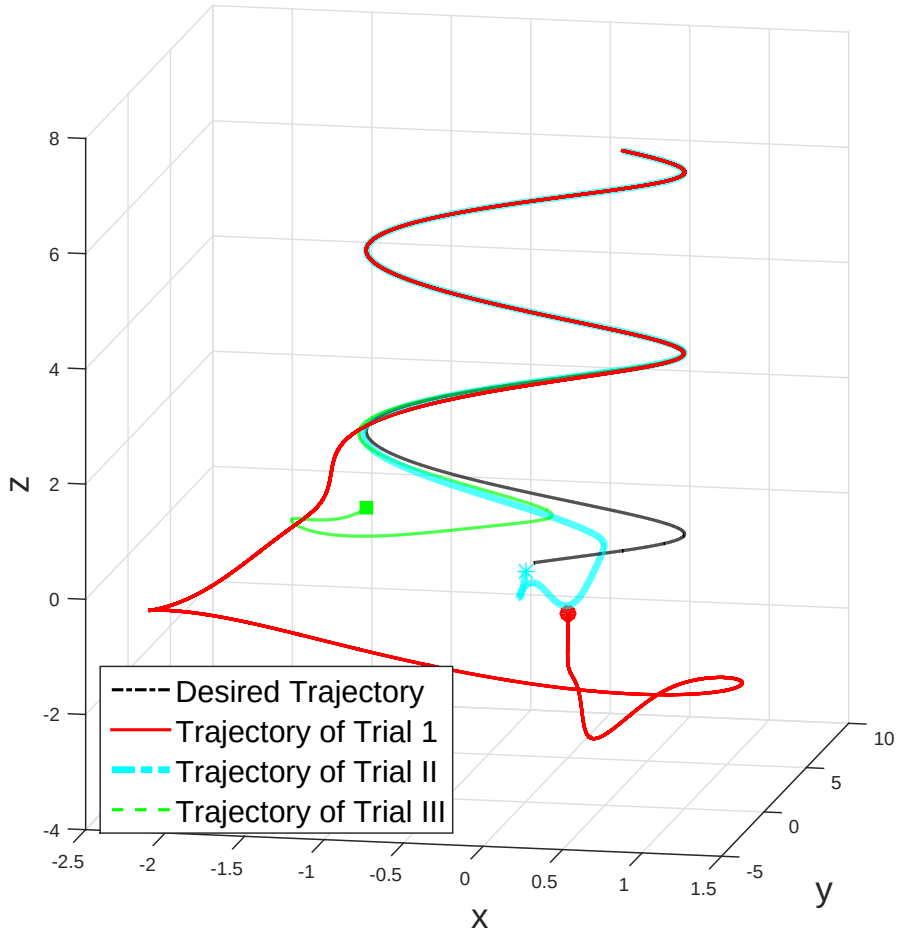


Figure 4.3: Load trajectory plot obtained by simulating the variation-based linearization controller for the quadrotor with a cable suspended load, as described in Section 4.5.3, for the three initial conditions is shown. The initial conditions are denoted by bold points in the figure. For each initial condition, the variation-based linearization controller is able to drive the load to the desired trajectory.

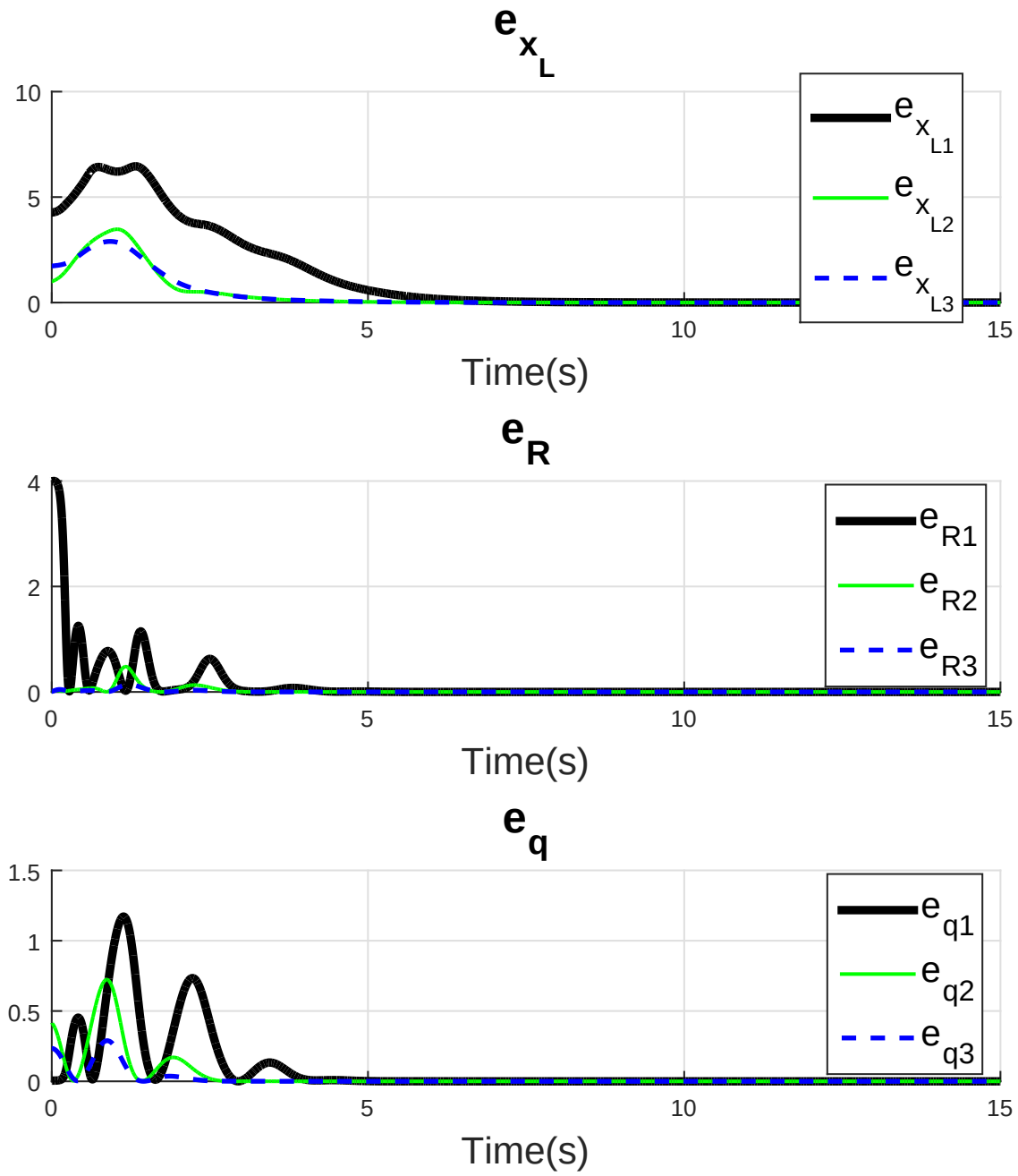


Figure 4.4: Tracking errors obtained by simulating the variation-based controller for the quadrotor with a cable suspended load, as described in Section 4.5.3, for the initial condition specified by Trial III is shown. In particular, the translational position errors, e_{x_L}, and vector errors for the quadrotor orientation, e_R, and load orientation, e_q, are shown as functions of time.

Chapter 5

Geometric Control Design for Rigid-body Payload Transportation

In this chapter, we proceed to the control design of a team of quadrotors transporting rigid-body payload, where the geometry of the payload can not be neglected. The main methodology to be used is the geometric control design idea employed in [2, 77, 1, 78] based on the model in Eq. (3.7). We provide a stabilizing controller based on the rigorous stability analysis using **Singular Perturbation** and the corresponding simulation result of the closed-loop system.

5.1 Rigid Payload Pose Tracking using Inertial Geometric Control

As mentioned previously, we want to solve the tracking control problem for the rigid body payload in Sec. 3.2.2. The overall controller could be decomposed into two parts: wrench control for the rigid body payload and the orientation control of each individual quadrotor. Since the system has a high-degree of underactuation, the controller is formulated as an output tracking controller with the goal being to track a set of outputs.

In particular, the control goal is that given a bounded and sufficiently smooth reference output for the system, defined as $r_d(t) = [x_{Ld}(t), R_{Ld}(t), q_{id}(t), \theta_{id}(t)]$, for $i = 1, 2, \cdots, n$, where $t \in [t_0, \infty]$ and θ_{id} is the desired yaw angle for each quadrotor, design a state feedback law

$$\begin{aligned} \Gamma : [t_0, \infty] \times TM &\to (\mathbb{R}^4)^n, \\ (t, x) &\mapsto (f_1, M_1, f_2, M_2, \cdots, f_n, M_n), \end{aligned}$$

such that the system trajectory tracks the outputs exponentially. As defined in Tab. 6.1, M is the configuration space of the system, TM is *the tangent bundle of* M, and $x \in TM$ is the system state.

To achieve the control goal just defined, we draw inspiration from the "inertial controller" design in [79]. In particular, we assume the number of quadrotors employed for the load transportation problem is large enough so that the column vectors of G *span* $\mathbb{R}^6$ for all time. This enables the application of an arbitrary wrench to the load, subject to the dynamics of the quadrotors being sufficiently fast.

Our control design can be divided into two parts shown in the following subsections: (a) control design for each individual quadrotor to track a virtual force input and desired yaw angle sufficiently fast, and (b) control design for specifying these virtual forces to enable exponential tracking of the load pose and cable attitudes.

5.1.1 Quadrotor force and yaw tracking control

We consider the problem of tracking the force and yaw angle of the i^{th} quadrotor, i.e given a smooth, bounded time-varying force $v_i : [t_0, \infty] \to \mathbb{R}^3$, and quadrotor yaw angle $\theta_{id} : [t_0, \infty] \to \mathbb{S}^1$, design the quadrotor control input $(f_i, M_i) \in \mathbb{R} \times \mathbb{R}^3$ such that the force being applied by the quadrotor $f_i R_i e_3$ tracks the specified force v_i, and the quadrotor body yaw angle θ_i tracks θ_{id} for all time.

To do this, we begin by defining the unit vector $e_3^{ic} = v_i / ||v_i||$, along with another unit vector that represents the yaw attitude to track as $e_1^{id} = [\cos\theta_{id} \;\; \sin\theta_{id} \;\; 0]^T$. Next we define a rotation matrix based on these, as,

$$R_{ic} = \begin{bmatrix} e_1^{ic} & e_3^{ic} \times e_1^{ic} & e_3^{ic} \end{bmatrix}, \text{where } e_1^{ic} = -\frac{e_3^{ic} \times (e_3^{ic} \times e_1^{id})}{||e_3^{ic} \times (e_3^{ic} \times e_1^{id})||}.$$

Then, we define the orientation error for geometric control as,

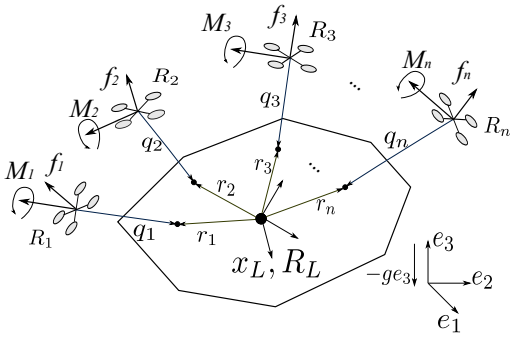
$$e_{R_i} = \frac{1}{2}(R_{ic}^T R_i - R_i^T R_{ic})^\vee, \quad e_{\Omega_i} = \Omega_i - R_i^T R_{ic} \Omega_{ic}$$

where $\Omega_{ic} = (R_{ic}^T \dot{R}_{ic})^\vee$, and the *vee map* $^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ is the reverse of the hat map $\hat{\cdot}$.

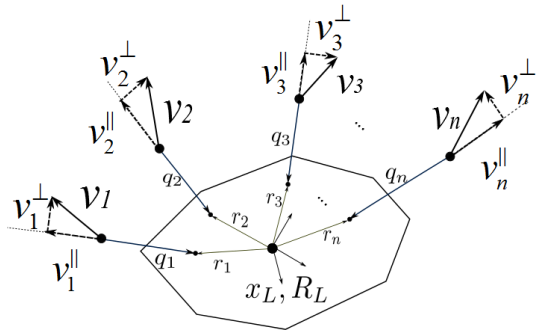
***Proposition* 5.1.** (Force and Yaw Tracking for a Single Quadrotor) *Consider a desired force v_i to be applied by a quadrotor with a desired quadrotor yaw θ_{id}. Also consider the following quadrotor inputs,*

$$f_i = v_i \cdot R_i e_3, \quad M_i = -\frac{k_{R_i}}{\varepsilon^2} e_{R_i} - \frac{k_{\Omega_i}}{\varepsilon} e_{\Omega_i} + \Omega_i \times J_i \Omega_i - J_i(\hat{\Omega}_i R_i^T R_{ic} \Omega_{ic} - R_i^T R_{ic} \dot{\Omega}_{ic}),$$

with $R_{ic}, \Omega_{ic}, e_{R_i}, e_{\Omega_i}$ as defined earlier, then there exists parameters k_{R_i}, k_{Ω_i} and $\bar{\varepsilon} > 0$ such that the errors $v_i - f_i R_i e_3$, $\theta_i - \theta_{id}$ tend to zero exponentially for any $i \in \{1, 2, \cdots, n\}$ when $\varepsilon < \bar{\varepsilon}$. Moreover, the convergence rate can be increased by reducing the value of ε.



(a) Diagram of the original system.



(b) Diagram showing the reduced system.

Proof. See. Appendix 10.5. □

***Remark* 5.1.** *For the extreme case when $\varepsilon = 0$, the force applied by the quadrotor $f_i R_i e_3$ can be controlled directly. In this case, the quadrotor attitude dynamics is entirely omitted, and the original system is reduced to the simplified one shown in Figure 5.1b.*

5.1.2 Load pose and cable attitude tracking control

We first consider the case when $\varepsilon = 0$, and design a control that tracks a desired load pose and cable attitude for the *reduced system* in Figure 5.1b. Next, we extend this control to the *full system* dynamics that includes the quadrotor dynamics.

We begin by designing a feedback law for the reduced system so as to specify the virtual force v_i to be applied by each quadrotor for tracking a desired load pose and cable attitude. To do this, we make the following assumptions:

- The rank of the matrix G is 6 for all time, i.e G has full rank all the time.
- There's no cable that becomes slack during the control process.

Assumption 1 is equivalent to the *column span* of G being $\mathbb{R}^6$, as discussed in the previous section. From the expression of G it follows that we need at least 6 quadrotors in order to realize this. Moreover, an equivalent condition is that G has Moore-Penrose pseudoinverse, denoted as $G^\dagger$. The second assumption guarantees that our model is valid for the control design, as the system is actually hybrid due to unilateral tension constraints. As we will see, our results validate that the assumption holds for a large class of trajectories.

In order to distinguish the reduced system and the full system, we add the subscript "r" to the states of the reduced system. Our control design is split into the following steps:

- Step 1: Compute a feedback wrench applied to the load as:

$$W_d = \begin{bmatrix} \ddot{x}_{Ld} + ge_3 \\ -\hat{\Omega}_L R_{Lr}^T R_{Ld} \Omega_{Ld} + R_{Lr}^T R_{Ld} \dot{\Omega}_{Ld} \end{bmatrix} - \begin{bmatrix} k_{x_L} e_{x_L r} + k_{v_L} e_{v_L r} \\ k_{R_L} e_{R_L r} + k_{\Omega_L} e_{\Omega_L r} \end{bmatrix}$$

  where the load's translational error is defined as

$$e_{x_{Lr}} = x_{Lr} - x_{Ld}, \quad e_{v_{Lr}} = \dot{x}_{Lr} - \dot{x}_{Ld},$$

  and its orientation error is defined as

$$e_{R_{Lr}} = \frac{1}{2}(R_{Ld}^T R_{Lr} - R_{Lr}^T R_{Ld})^\vee, \; e_{\Omega_{Lr}} = \Omega_{Lr} - R_{Lr}^T R_{Ld} \Omega_{Ld}.$$

- Step 2: Use the above expression to obtain $u_{iv}^{\parallel}$ which is the *parallel component of v_i along q_{ir}* as the following

$$\begin{bmatrix} u_{1v}^{\parallel} \\ u_{2v}^{\parallel} \\ \vdots \\ u_{nv}^{\parallel} \end{bmatrix} = u_v^{\parallel} = G^{\dagger}(-d + AW_d),$$

  where the terms G, d, A are computed using the dynamics based on states of reduced system.

- Step 3: Based on this, we are able to cancel out the effects of load's accelerations on the cable attitude using the perpendicular part of v_i as:

$$u_{iv}^{\perp} = \hat{q}_{ir}((q_{ir} \cdot \omega_{id})\dot{q}_{ir} - \hat{q}_{ir}^2 \dot{\omega}_{id} - (k_{q_i} e_{q_{ir}} + k_{\omega_i} e_{\omega_{ir}})) - \frac{1}{L_i}\hat{q}_{ir}^2(\ddot{x}_{Lbr} + ge_3 + R_{Lr}(\hat{\Omega}_{Lr}^2 + \dot{\hat{\Omega}}_{Lbr})r_i)$$

  where $i = 1, 2, \cdots, n$ and

$$e_{q_{ir}} = q_{id} \times q_{ir}, \quad e_{\omega_{ir}} = \omega_{ir} + \hat{q}_{ir}^2 \omega_{id},$$

  are the position and velocity error functions in $\mathbb{S}^2$.

We then have the following proposition:

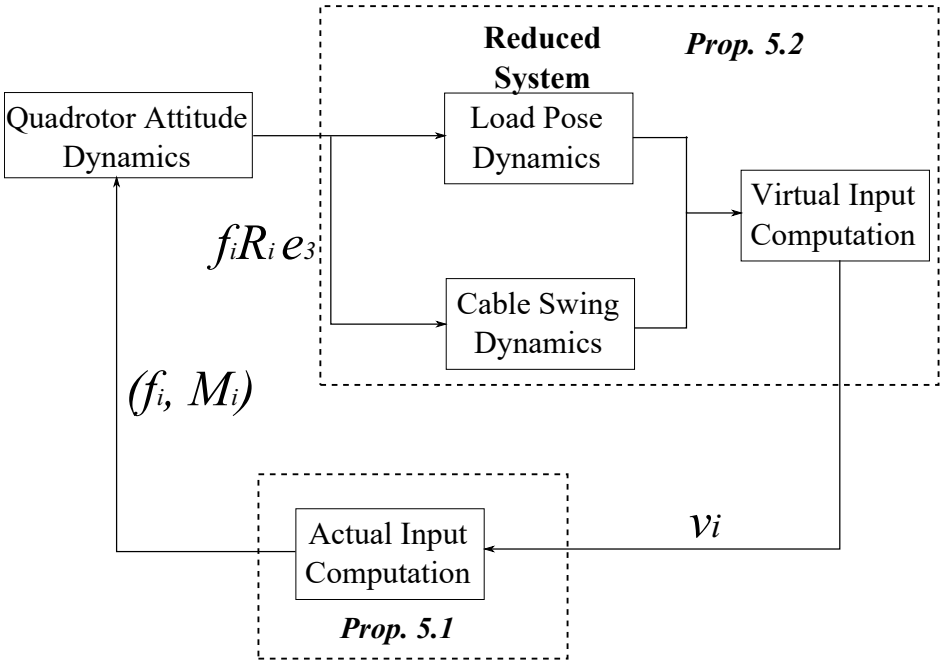


Figure 5.2: Block diagram of the geometric controller's structure.

***Proposition* 5.2.** (Almost Global Exponential Tracking of the Reduced System) *Consider the reduced system shown in Figure 5.1b. Also consider the desired force to be applied by each quadrotor as,*

$$v_i = m_i L_i (u_{iv}^{\parallel} + u_{iv}^{\perp}),$$

where $u_{iv}^{\parallel}, u_{iv}^{\perp}$ are as defined above. Then, there exist gain parameters $k_{x_L}, k_{v_L}, k_{R_L}, k_{\Omega_L}$ and $k_{q_i}, k_{\omega_i}, k_{R_i}, k_{\Omega_i}$ $i = 1, 2, \cdots, n$ such that the reduced system tracks the reference output $(x_{Ld}(t), R_{Ld}(t), q_{id}(t))$ exponentially.

Proof. See Appendix 10.6. □

Then by selecting the virtual control v_i as specified by Prop. 5.2 and the reference yaw angle θ_{id} and inputs (f_i, M_i) for each quadrotor as specified by Proposition 5.1, we have completed the full controller design. The stability properties are established by the following proposition.

***Proposition* 5.3.** (Exponential Tracking for the Full System) *Consider the full model of the system which includes the quadrotor dynamics, and consider the virtual control v_i as specified by Prop. 5.2, along with the reference yaw angle θ_{id} and the quadrotor inputs (f_i, M_i) as specified by Prop. 5.1. Then there exists $\varepsilon^* > 0$ such that $\forall \varepsilon < \varepsilon^*$, the reference outputs $(x_{Ld}, R_{Ld}, q_{id}, \theta_{id})$ is exponentially tracked for the closed-loop full system.*

Proof. See Appendix 10.7. □

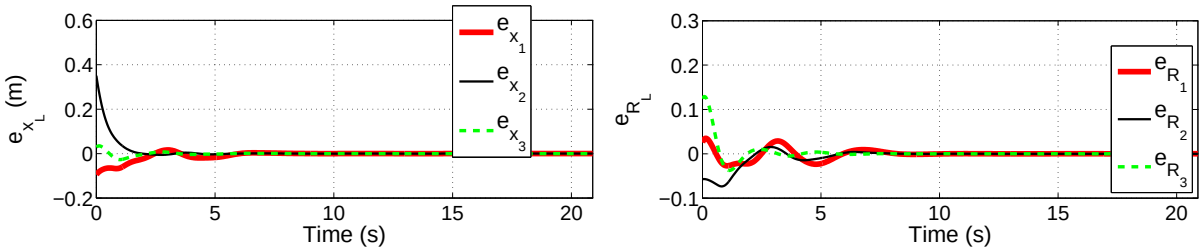


Figure 5.3: Position error (e_{x_L}) and orientation error (e_{R_L}) of the load with respect to time.

Table 5.1: System Parameters for Simulation (Units in SI)

Parameter	Value	Parameter	Value
n	7	L	1.2
m_L	3.0	m_i	0.55
J_L	$diag[3,3,6]$	J_Q	$diag[2.32,2.32,7.6]\times 10^{-3}$
r_{bi}	$[2\cos\frac{k\pi}{3}, 2\sin\frac{k\pi}{3}, 0.05]$ $(i=1,2,\cdots,6)$		
r_{b7}	$[0,0,0.05]$	ε	0.005

5.2 Simulation Results and Discussion

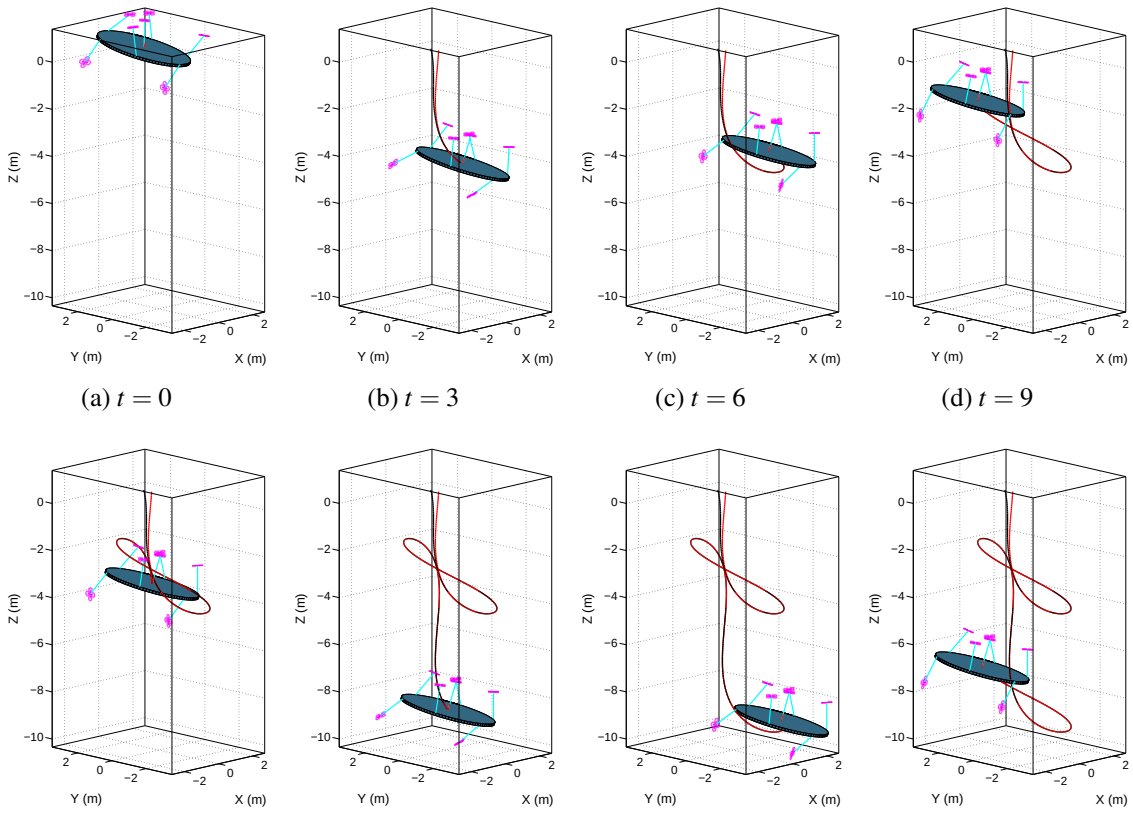
In order to validate the stability of our controller and shed some light on its limitations, we perform a numerical simulation in Matlab. We consider a cylinder-shaped load suspended with cables of equal length from 7 identical quadrotors. The values of the important system parameters are shown in Table 5.1.

Since the system is shown to be differentially flat in [1], we are able to generate a smooth reference based on the flat outputs that consist of the the load's pose, the tensions for the last four quadrotors, and the yaw angles for all quadrotors. These flat outputs are listed below,

$$
\begin{aligned}
x_{Ld} &= [1.2\cos t, 2.0\sin(0.5t+\frac{\pi}{6}), 2.5\cos(0.5t+\frac{\pi}{4})-0.4t]^T \\
R_{Ld} &= e^{\hat{\Omega}_0}, \quad \text{where} \quad \Omega_0 = [0.2,0.2,0.5]^T, \\
T_4 &= [2,2,5]^T,\ T_5 = [-2,2,5]^T,\ T_6 = [2,-2,5]^T \\
T_7 &= [0,0,8]^T
\end{aligned}
$$

where the four tensions ($T4-T7$) are specified as constants with respect to the load's body frame shown in Figure 5.5, and the quadrotor yaw angles are specified as zero.

The simulation is performed using Matlab, with the full system state vector containing 144 variables (18 states for the load and each of the seven quadrotors). We also consider two quadrotors in an inverted



(a) $t = 0$ (b) $t = 3$ (c) $t = 6$ (d) $t = 9$

(e) $t = 12$ (f) $t = 15$ (g) $t = 18$ (h) $t = 21$

Figure 5.4: Snapshots of a dynamic load manipulation trajectory. The load, cables, and quadrotors are shown at intervals of 3 seconds. The black line is the nominal trajectory to be followed and red line is actual trajectory for the load's CoM.

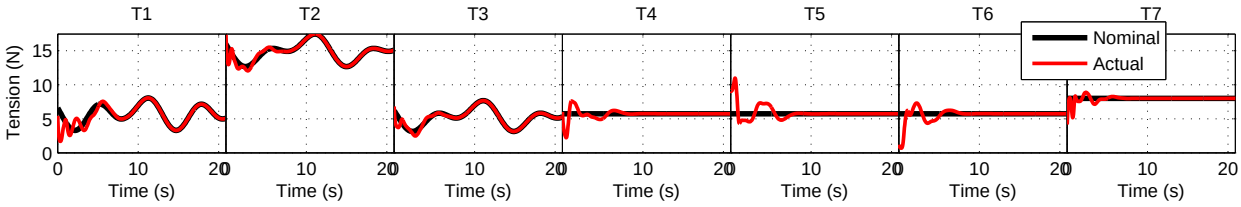


Figure 5.5: Plots of the tension in each of the cables that are between the suspended load and the quadrotors. As is clearly seen, the tension is strictly positive indicating that the controller did not cause any of the cables to go slack.

configuration that pull the load down to illustrate the flexibility of the proposed controller.

Simulation results are presented in Figures 5.3-5.5. Figure 5.3 illustrates the load position and orientation errors going down to zero exponentially. Figure 5.4 illustrates snapshots of the system comprising of the load, suspension cables, and quadrotors, as the controller drives the system to follow the reference output (black and red traces represent the nominal and actual trajecotries for the load's CoM respectively.) A large initial position error along with an orientation error for the load are specified and the controller drives them to zero. Finally, in order to test the validness of our assumption that the cables do not go slack, each cable tension is computed from the simulation and illustrated in Figure 5.5. As can be seen, the tensions in all cables are strictly positive.

***Remark* 5.2.** *Having presented numerical results demonstrating the performance of the proposed controller, we now discuss some of its shortcomings. Our geometric control design requires a perfect model, which could cause problems on real-world systems with uncertainties. Moreover, although the controller demonstrates good performance with cable tensions that are strictly positive, the controller does not explicitly guarantee this. There potentially exists initial conditions that could cause the cables to go slack. Furthermore, the controller does not place any state or input constraints, such as collision constraints between the various quadrotors and constraints for actuator limits. Finally, the experimental implementation of the proposed controller will be potentially hard due to noisy state estimates and limited communication bandwidth.*

Both the theoretical proofs and simulations have shown the geometric controller is stable. What we have delivered so far is mainly focused on the tracking of the payload's pose, without considering many constraints of the transportation systems, especially the safety constraints which require collision avoidance of each quadrotor with its environment. The next two chapters would elaborate on how to handle such safety constraints for different mechanical systems, including a single quadrotor.

Chapter 6

Safety-critical Tracking Control Design for Mechanical Systems Evolving on Manifolds

While the previous chapters are focused on control design for the transportation systems, we propose some general control methods which can handle constraints in this chapter. In particular, we would define the concepts of geometric Control Lyapunov Function and Control Barrier Function for systems evolving on manifolds. Using these concepts, we are able to propose a general control design method to make a balance between tracking a reference and enforcing safety constraints.

6.1 Mathematical Preliminaries

Before we dive into the details of the specific concepts, a few necessary concepts need to be introduced. Since lots of symbols are introduced, a summary of the notation of each symbol is introduced in Table 6.1.

6.1.1 Geometric Control Fundamentals with Application to Mechanical Systems

Given a mechanical system evolving on a sufficiently smooth manifold M, we denote its configuration variable as $q \in M$, the tangent space at q as T_qM, the tangent bundle as $TM = \cup T_qM$, and state-space representation as $(q, \dot{q}) \in TM$.

Further, a vector field is a mapping from each point $q \in M$ to a vector in T_qM. While, an one-form $\omega : T_qM \to \mathbb{R}$ defines a mapping from the tangent space at each point $q \in M$ to the real number. A common

Table 6.1: Symbolic Notations for Chap. 6.

M	A smooth finite-dimensional manifold.	T_qM	The tangent space at a point $q \in M$.
T_q^*M	The cotangent space at a point $q \in M$.	TM	Tangent bundle representing state space.
g	Smooth function defined in M.	dg	An one-form in M representing g's differential.
X, Y	Smooth vector fields of M.	$\langle dg, X\rangle_q$	The value of one-form dg of the vector X at point q.
$\nabla_X Y$	Riemann connection of X, Y, i.e, the covariate derivative of Y with respect to X.	M_q	The inertia tensor which maps from T_qM to T_q^*M
$\langle\langle \dot{q}_1, \dot{q}_2\rangle\rangle$	The metric induced by M_q, i.e, $\langle\langle \dot{q}_1, \dot{q}_2\rangle\rangle = \langle M_q\dot{q}_1, \dot{q}_2\rangle$	$\mathcal{V}_q$	The potential function of a mechanical system.
F_i	The $i^{(th)}$ basis resultant force	F	The total external force as a linear combination of basis forces
u^j	The j^{th} control input.	$u \in \mathbb{R}^m$	The control input vector.
h_i	A smooth function in TM whose level set defines an admissible region.	$\mathcal{C}_i$	A region defined by $\{(q, \dot{q}) : h_i(q, \dot{q}) \geq 0\}$ in the state space TM.
B_i	The corresponding Control Barrier Function(CBF) of $\mathcal{C}_i$.	V	The Control Lyapunov Function.

one-form is *the differential* of a smooth function $f : M \to \mathbb{R}$ is denoted as df and given by

$$\langle df, X\rangle_q = \lim_{t\to 0} \frac{f(\alpha(t)) - f(\alpha(0))}{t},$$

where the curve $\alpha : [-1, 1] \to M$ satisfies $\alpha(0) = q \in M$ and $\alpha'(0) = X \in T_qM$. For example, suppose we define a smooth function on the unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$ as $f(q) = q \cdot n$, with $q \in \mathbb{S}^2$, $n \in \mathbb{R}^3$ and the $\cdot$ operator serving as the inner product in $\mathbb{R}^3$. Then, from multi-variate calculus, its differential with a tangent vector $\dot{q} \in T_qM$ is given by $\langle df, \dot{q}\rangle_q = \dot{q} \cdot n$.

In general, we denote the value of the one-form ω with tangent vector $\dot{q}$ at point q as $\langle \omega, \dot{q}\rangle_q$. Based on the notion of the one-form, we are able to define a two-form which is the exterior derivative of the one-form. Similarly, we are able to get a two-form from the differential of a function, which reflects the second order derivative of this function on a manifold. In particular, when a smooth function $f : M_1 \times M_2 \to \mathbb{R}$ is multivariate, we will denote its second order differential as:

$$d_i d_j f = d_i(d_j f), \quad i, j \in \{1, 2\} \tag{6.1}$$

where d_i is the exterior derivative of $d_j f$ with respect to M_i.

Since mechanical systems are governed by Newton's law with dynamics represented as second order differential equations, we need a way to describe how $\dot{q}$ will evolve along the system trajectory, namely the acceleration term on a manifold. This requires the introduction of the notions of a Riemannian metric and a Riemannian connection. A Riemannian metric $\langle\langle\cdot,\cdot\rangle\rangle_q : T_qM \times T_qM \to \mathbb{R}$ is an inner product, defined on the tangent space T_qM, which changes smoothly as the tangent space is shifted from one point to another. Through this metric, every element in the space of linear functionals on the tangent space T_qM (denoted as T_q^*M) could be uniquely identified with an element in T_qM by the Riesz representation theorem. Intuitively, due to this metric, we can treat every vector in T_qM as *either a tangent vector or a linear functional in* T_q^*M.

For example, consider a point mass with mass $M_q > 0$ in $M = \mathbb{R}^3$, then a candidate metric could be given as

$$\langle\langle\dot{q}_1, \dot{q}_2\rangle\rangle_q = (M_q\dot{q}_1)\cdot\dot{q}_2, \quad \dot{q}_1, \dot{q}_2 \in T_qM.$$

From another perspective, the mass M_q actually maps a vector in T_qM to a linear functional in T_q^*M. Based on this metric, we are able to generate a Riemannian connection which is *compatible with it and torsion-free*. Intuitively, given two smooth vector fields X, Y, the connection $\nabla_X Y$ in M describes how the vector of Y at q would change if we move the point q along the direction of X. Given a Riemannian metric, a torsion-free, compatible connection could be uniquely determined, which we call the Riemannian connection.

***Remark* 6.1.** *For simplicity of notation, we will drop the explicit dependence of $\langle\langle\cdot,\cdot\rangle\rangle_q$ on $q \in M$ by dropping the subscript q to obtain $\langle\langle\cdot,\cdot\rangle\rangle$, with the expectation that the tangent vectors provides this information.*

Now we are able to describe a simple, fully-actuated mechanical system and the corresponding geometric control law. If the configuration manifold M admits the following structure:

1. A metric $M_q : T_qM \to T_q^*M$ which represents the kinetic energy by $\langle\langle\dot{q},\dot{q}\rangle\rangle = \langle M_q\dot{q},\dot{q}\rangle$.
2. A connection ∇ which is compatible with M_q that serves to describe how a tangent vector or the velocity $\dot{q}$ changes along the manifold M.
3. A collection of one-forms $F_i : T_qM \to \mathbb{R}$ representing the external force applied on the system, where $i \in \{1, 2, \cdots, m\}$. The span of these one-forms is the entire cotangent space T_q^*M at each point q.
4. A smooth potential energy function $V_q : M \to \mathbb{R}$.
5. A configuration error $\Psi : M \times M \to [0, \infty)$ that serves as a measure of distance between the two points $q, q_d \in M$. We also require $\Psi(q, q_d)$ to be quadratic as defined in [51]. Denote the differential of the configuration error with respect to the i^{th} argument as $d_i\Psi$ $(i = 1, 2)$, and define the position error as $e_q = d_1\Psi$.

6. A transport map $\mathscr{T}_{(q,q_d)} : T_{q_d}M \to T_qM$ which maps a tangent vector at q_d to one at q with the compatible condition,

$$d_2\Psi = -\mathscr{T}^*_{(q,q_d)} d_1\Psi, \tag{6.2}$$

where $\mathscr{T}^*_{(q,q_d)}$ is the dual map of $\mathscr{T}_{(q,q_d)}$. In this way, we are able to compare tangent vectors in different tangent spaces through the velocity error defined as

$$e_{\dot{q}} = \dot{q} - \mathscr{T}_{(q,q_d)}\dot{q}_d.$$

Then, a system evolving on M with the above structure is called a simple mechanical system with dynamics given by,

$$\nabla_{\dot{q}}\dot{q} = M_q^{-1}(-dV_q(q) + \sum_{i=1}^{m} F_i(q,\dot{q})u_i), \tag{6.3}$$

where $u_i \in \mathbb{R}$. See [70] for more details on the above points.

For this type of system, a general PD-type feedback law could be set up which guarantees *exponential stability*. In order to track a dynamically feasible smooth reference $q_d(t) : [0,\infty) \to M$, a geometric control input could be expressed as

$$u = \underbrace{u_{ff}}_{\text{feedforward term}} + \underbrace{u_{fb}}_{\text{feedback term}},$$

where u_{fb} is a linear combination of the position and velocity errors, e_q and $e_{\dot{q}}$, and serves as a generalization of a PD control on the manifold M.

6.1.2 Time-varying Exponentially Stabilizing Control Lyapunov Function (CLF)

In the sections below, we will introduce the concepts of control Lyapunov functions (CLFs) and control Barrier functions (CBFs). Though originally introduced for systems in Cartesian spaces, these concepts have been generalized to the case of simple mechanical systems (6.3) on manifolds, as studied in [60]. Consider a control affine system in $\mathbb{R}^n$ of the form,

$$\dot{x} = f(x) + g(x)u, \qquad x(t_0) = x_0, \tag{6.4}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$.

For system (6.4), a continuously differentiable function $V : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}$ is called Exponentially

Stabilizing Lyapunov Function (ES-CLF) if there exist constants $c_1, c_2, \eta > 0$ such that,

$$c_1||x||^2 \leq V(t,x) \leq c_2||x||^2,$$
$$\inf_{u\in\mathbb{R}^m}\{\frac{\partial V}{\partial t} + L_f V + L_g V u + \eta V\} \leq 0,$$

holds for every $(t,x) \in [0,\infty)\times\mathbb{R}^n$ where $L_f V = \frac{\partial V}{\partial x} f$, $L_g V = \frac{\partial V}{\partial x} g$ are the Lie derivatives of V with respect to f and g.

***Remark* 6.2.** *CLFs give a qualitative analysis of the stability of the origin. If such a function exists, we could determine the control input analytically through the Sontag control or the min-norm control, or directly through a state-based optimization pointwise in time. For many control applications, a closed-loop Lyapunov function could be directly employed as a candidate CLF.*

6.1.3 Geometric Control Lyapunov Function

We consider the problem of tracking a desired reference state $(q_d, \dot{q}_d) \in TM$.

***Definition* 1.** *A smooth function $V : [0,\infty)\times TM \to \mathbb{R}$ is called a geometric ES-CLF for the system in* (6.3) *if there exist constants $c_1, c_2, \eta > 0$ such that*

$$\begin{aligned}
&V(t,q,\dot{q}) \geq c_1(\Psi(q,q_d) + \langle\langle e_{\dot{q}}, e_{\dot{q}}\rangle\rangle),\\
&V(t,q,\dot{q}) \leq c_2(\Psi(q,q_d) + \langle\langle e_{\dot{q}}, e_{\dot{q}}\rangle\rangle),\\
&\inf\nolimits_{u\in\mathbb{R}^m}\{\tfrac{\partial V}{\partial t} + \underbrace{\langle d_1 V, \dot{q}\rangle - \langle d_2 V, M_q^{-1} dV_q\rangle}_{\textit{equivalent to } L_f V}\\
&+ \underbrace{\sum_{i=1}^{m}\langle d_2 V, M_q^{-1} F_i\rangle u_i}_{\textit{equivalent to } L_{g_i} V} + \eta V\} \leq 0,
\end{aligned} \tag{6.5}$$

holds for all $(t,q,\dot{q}) \in [0,\infty)\times TM$ where $e_{\dot{q}} = \dot{q} - \mathscr{T}_{(q,q_d)}\dot{q}_d$ and $u = [u_1, u_2, \cdots, u_m]^T$ is the vector containing all control inputs.

Note that the CLF $V(t,q,\dot{q})$ depends on the reference trajectory $q_d(t), \dot{q}_d(t)$ implicitly through the time t. Also, this generalized definition coincides with the previous one if the manifold is chosen to be $\mathbb{R}^{2n}$ and the reference point is chosen to be $(0,0)$. The concept of geometric CLF is specifically useful for control design for systems whose configuration spaces are nonlinear manifolds, with dynamics given by (6.3).

6.1.4 Time-varying Control Barrier Function (CBF)

Control Barrier functions (CBFs) are defined with respect to a region in the state space. For the control affine system (6.4), suppose we have a continuously differentiable function $h : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}$ with its super level set $\mathcal{C}_t = \{x \in \mathbb{R}^n : h(t,x) \geq 0\}$. If this set admits a non-empty interior at each time in $[0,\infty)$, then a smooth function $B : [0,\infty) \times \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ is called a CBF of $\mathcal{C}_t$ if there exist two class $\mathcal{K}$ function α_1, α_2 and $\mu > 0$ such that

$$\begin{aligned} &\frac{1}{\alpha_1(h(x))} \leq B(t,x) \leq \frac{1}{\alpha_2(h(x))}, \\ &\inf_{u \in \mathbb{R}^m} \{\frac{\partial B}{\partial t} + L_f B + L_g B u - \frac{\mu}{B}\} \leq 0, \end{aligned} \tag{6.6}$$

holds for any $t \in [0,\infty)$ and any $x \in \mathcal{C}_t^\circ$ which is the interior of $\mathcal{C}_t$.

***Remark* 6.3.** *The idea of a CBF is based on invariance analysis of a set for a dynamic system. To stay safe, the system trajectory should always remain within the safe set, $\mathcal{C}_t$, which entails that the safe set should be forward invariant for the closed loop system. By imposing condition* (6.6)*, we are able to set up a positive lower bound for the value of $h(t,x)$ that holds globally, which means that safety is guaranteed. Note that, the range of the CBF is chosen to be the extended real line which allows for the Barrier $B(t,x)$ to go to infinity as $h(t,x)$ goes to zero.*

6.1.5 Geometric Control Barrier Function

Similar to the case of CLF, we extend the concept of CBF to simple mechanical systems (6.3) evolving on manifolds. Because we can only analyze a set with respect to a specific topology, the topology on the configuration manifold M is considered to be the relative topology with respect to the smallest Cartesian space in which M can be embedded smoothly.

***Definition* 2.** *Suppose there exist a smooth function $h : [0,\infty) \times TM \to \mathbb{R}$ such that the safety region is defined by $\mathcal{C}_t = \{(q,\dot{q}) \in TM : h(t,q,\dot{q}) \geq 0\}$ which has nonempty interior for any $t \in [0,\infty)$. Then a smooth function $B : [0,\infty) \times \mathbb{R} \to \mathbb{R} \cup \{\pm\infty\}$ is called a geometric CBF of $\mathcal{C}_t$ if there exist two class $\mathcal{K}$ functions α_1, α_2 and a*

constant $\mu > 0$ *such that*

$$\begin{aligned}
&\tfrac{1}{\alpha_1(h(t,q,\dot{q}))} \leq B(h(t,q,\dot{q})) \leq \tfrac{1}{\alpha_2(h(t,q,\dot{q}))}, \\
&\inf_{u\in\mathbb{R}^m}\{\tfrac{\partial B}{\partial t} + \underbrace{B'(h)\left(\langle d_1 h, \dot{q}\rangle - \langle d_2 h, M_q^{-1} dV_q\rangle\right)}_{\text{equivalent to } L_f B} \\
&\underbrace{+ B'(h)\left(\sum_{i=1}^{m}\langle d_2 h, M_q^{-1}F_i\rangle\right)}_{\text{equivalent to } L_{g_i} B} u_i - \tfrac{\mu}{B}\} \leq 0,
\end{aligned} \tag{6.7}$$

for any $t \in [0,\infty)$ *and* $x \in \mathcal{C}_t^\circ$.

***Remark* 6.4.** *Note that the time-varying CBF, as a scalar function, is purely based on the region function h. Both time and system state are implicitly contained within the CBF through the scalar function h. Thus, to design a CBF in this type, we just need to choose a suitable shape of this scalar function.*

The following Lemma provides a formal guarantee of safety.

Lemma 6.1. *(Safety Guarantee of Time-varying Geometric CBF in* (6.7)*) For the system* (6.4)*, if the control input u satisfies the condition* (6.7) *at each time* $t \in [0,\infty)$*, then the set* $\{x \in \mathbb{R}^n : h(t,x) > 0\}$ *is forward invariant, i.e, the system trajectory* $(q(t),\dot{q}(t))$ *would always remain within* $\mathcal{C}_t^\circ$ *if* $(q(0),\dot{q}(0)) \in \mathcal{C}_{t=0}^\circ$.

Proof. See Appendix. 10.8. □

6.2 Safety Critical Control Design on Riemmanian Manifold

In this section, we will propose a general method to extend configuration constraints, given in terms of only the configuration variable q, to the whole state space $(q,\dot{q})$ so as to enforce the configuration constraints thereby maintaining safety. Following this, a candidate geometric CBF will be constructed and combined with a candidate geometric CLF. Based on the combination of the geometric CBF and CLF, we will propose a feedback controller through a state-dependent quadratic program that is solved point-wise in time.

6.2.1 Control Problem Formulation

Given a fully actuated simple mechanical system (6.3) and a list of time-varying safety constraints in terms of the configuration variables q, written as $g_i(t,q) \geq 0$ for $i \in \{1,2,\cdots,k\}$, where

$$g_i(t,q) := (-1)^{\delta_i}(b_i(t) - \Psi(q, q_i(t))), \tag{6.8}$$

with $\delta_i \in \{0,1\}$, $b_i(t) > 0$ representing the radius, $q_i(t) \in M$ representing the center and $\Psi(q,q_i(t))$ being the configuration error between the current configuration q. We can then define a time-varying safe set,

$$\mathcal{B}_{i,t} := \{(q,\dot{q}) \in TM : \ g_i(t,q) \geq 0, \quad q_i \in M\}, \tag{6.9}$$

satisfying the condition that

$$(\cap_{i=1}^{k} \mathcal{B}_{i,t})^\circ =: (\mathcal{B}_t)^\circ = \mathcal{B}_t^\circ \neq \emptyset \tag{6.10}$$

where the set $\mathcal{B}_{i,t}$ is the safety region for constraint g_i, and $\mathcal{B}_t^\circ$ denotes the interior of the set $\mathcal{B}_t$ with respect to the topology of the manifold M.

***Remark* 6.5.** *The constraints presented here are in terms of the configuration error. The value of δ_i is to indicate whether the inside or outside of the region $\mathcal{B}_{i,t}$ is safe or not. The center point $q_i : [0,\infty) \to M$ and the radius $b_i : [0,\infty) \to (0,\infty)$ are both sufficiently smooth. In real applications, these regions could be constructed through methods in computational geometry. By picking specific configuration errors, we are able to approximate the actual safe region in a proper way. For example, level sets of L_2, L_1, L_∞ norms on $\mathbb{R}^2$ could be a circle or squares with different orientations. We can choose a specific norm that could best fit the actual safe region through optimization.*

The above compatible condition in (6.10) *is to ensure that the set of all specified constraints* can be satisfied at the same time. *The geometric intuition is that the corresponding free space is nonempty.*

With the above definitions, we are now in a position to state the control problem as follows.
Geometric constrained control problem: Given a smooth reference curve $q_d(t) \in \mathcal{B}_{\sqcup t}^\circ$ for any $t \in [0,\infty)$, design the feedback control input $u = u(t,q,\dot{q})$ so that the following conditions are satisfied:

$$\begin{aligned} &q(t) \in \mathcal{B}_t, \ \forall t \in [0,\infty) && \text{(Safety Constraints)} \\ &q(t) \to q_d(t) \in \mathcal{B}_t \text{ as } t \to +\infty && \text{(Stability Constraints)} \end{aligned} \tag{6.11}$$

6.2.2 Geometric CBF Candidate

Having formulated the control problem, we now construct a geometric CBF to enforce safety. In particular, based on the configuration constraint functions g_i in (6.8), we propose a general method which can expand the safety region in configuration space M to the state space TM. We choose a smooth class $\mathcal{K}$ function $\beta : [0,\infty) \to \mathbb{R}$ and define a new constraint function in terms of t and $(q,\dot{q})$ as:

$$h_i(t,q,\dot{q}) = \gamma_i \beta(g_i(t,q)) + \langle d_1 g_i, \dot{q} \rangle + \frac{\partial g_i}{\partial t}. \tag{6.12}$$

Note that h_i is well-defined since dg_i is an one-form on M and thus it's a *linear functional* of the tangent space at each point q, and by the Chain rule, additional partial derivatives with respect to time can be expressed as:

$$\frac{\partial g_i}{\partial t} = (-1)^{\delta_i}(\dot{b}_i - \langle d_2 \Psi_{q,q_i}, \dot{q}_i \rangle).$$

Here, we have rewritten the configuration error $\Psi(q,q_i)$ as Ψ_{q,q_i} for symbolic simplicity.

We can then define a new expanded safety region in TM as:

$$\mathcal{C}_t := \cap_i^m \mathcal{C}_{i,t} := \cap_i^m \{(q,\dot{q}) \in TM : h_i(t,q,\dot{q}) \geq 0\} \tag{6.13}$$

where the parameters γ_i and $\beta(\cdot)$ are chosen to make sure the interior of $\mathcal{C}_t$ nonempty.

Question arises as whether the construction of $\mathcal{C}_t$ is meaningful or not. To be more precise, we expect the set $\mathcal{C}_t$ to be a nonempty set with some good properties for future analysis. We have listed the properties of $\mathcal{C}_t$ in the following lemmas.

Lemma 6.2. *(Nonemptyness of $\mathcal{C}_t$ for the General Case)*

Suppose that the free space $\mathcal{B}_t$ is nonempty for each time $t \geq 0$. Also, there exist parameters $c_1, c_2, \cdots, c_k > 0$ such that $c_i g_i(t,q) \geq |\partial_t g_i(t,q)|$ for each $t \geq 0$ and each $q \in \mathcal{B}_t$, then there exist a Class $\mathcal{K}$ function $\beta(\cdot)$ and parameters $\gamma_1, \gamma_2, \cdots, \gamma_k > 0$ in Eq. (6.12) *which makes the corresponding $\mathcal{C}_t$ nonempty.*

Proof. See Appendix 10.9. □

Lemma. 6.2 provides a sufficient condition for the derivatives of $\dot{b}_i$ in order to make the set $\mathcal{C}_t$ nonempty. In particular, for the case when b_i is a constant, the condition of Lemma. 6.2 is automatically satisfied, and thus we have created a nonempty set $\mathcal{C}_t$ to work on. Moreover, from this lemma, we know that for each $q \in \mathcal{B}_t$, all the feasible tangent vectors $\dot{q}$ form a polyhedron in the linear space $T_q M$, which is based on the linear constraint in Eq. (10.1). The following lemma is about the relationship between the connectedness of $\mathcal{B}_t$ and $\mathcal{C}_t$. In particular, we will be focused on the time-invariant case for simplicity.

Lemma 6.3. *(Connectedness of $\mathcal{C}_t$ for the Static Case)*

If the following conditions are satisfied:

- *There exist parameters γ_i and function $\beta(\cdot)$ such that $\mathcal{C}_t$ is nonempty.*
- *Both q_i and b_i are time-invariant.*
- *The set $\mathcal{B}_t$ has nonempty interior $\mathcal{B}_t^\circ$, and $\mathcal{B}_t^\circ$ is path connected.*

Then a subset $\mathcal{C}_t^B$ of $\mathcal{C}_t$, defined as $\{(q,\dot{q}) \in \mathcal{C}_t : q \in \mathcal{B}_t^\circ\}$ is also path connected.

Proof. See Appendix 10.10. □

The previous lemma establishes a relationship between the connectedness of the original constraint set $\mathcal{B}_t$ and part of the expanded set denoted as $\mathcal{C}_t^B$. In summary, the previous two lemmas justify the construction method of $\mathcal{C}_t$ in Eq. (6.12). The argument provided does not include the general time-varying case, but sheds some light on the construction of the set $\mathcal{C}_t$. The previous argument has set up some relationship between $\mathcal{B}_t$ and $\mathcal{C}_t$ which are defined in terms of configuration variable and state variable respectively. As already noticed in the proof, when $(q,\dot{q}) \in \mathcal{C}_t$, it's possible that $g(q)$ could be negative and consequently $q \notin \mathcal{B}_t$. Moreover, taking the dynamics into account, we could infer the safety properties of $\mathcal{B}_t$ and $\mathcal{C}_t$ in the lemma below.

***Proposition* 6.4.** *(Forward Invariance Preservation of the Feasible Region $\mathcal{B}_t$)*

Suppose the region $\mathcal{C}_t$ is forward invariant for the system (6.3)*, then the region $\mathcal{B}_t$ is also forward invariant whenever initially $q_0 \in \mathcal{B}_{t=0}$ and $(q_0,\dot{q}_0) \in \mathcal{C}_{t=0}$.*

Proof. See Appendix. 10.11. □

***Remark* 6.6.** *This proposition guarantees that if we could enforce the forward invariance of region $\mathcal{C}_t$, the safety constraints $(g_i(t,q,\dot{q}) \geq 0)$ are satisfied automatically. Equivalently, the previous constrained control problem* (6.11) *could be converted to a new problem below:*

$$(q(t),\dot{q}(t)) \in \mathcal{C}_t,\ \forall t \in [0,\infty)$$
$$(q(t),\dot{q}(t)) \to (q_d(t),\dot{q}(t)) \in \mathcal{C}_t \text{ as } t \to +\infty$$

Based on the new region $\mathcal{C}_{i,t}$ in (6.13), we propose the following CBF candidate for each constraint in terms of h_i:

$$B_i(q,\dot{q}) = \frac{1}{h_i(t,q,\dot{q})}, \quad (q,\dot{q}) \in \mathcal{C}_t^\circ$$

Differentiating each B_i with respect to time yields,

$$\begin{aligned}\dot{B}_i(t,q,\dot{q}) = & -\frac{1}{h_i^2}\dot{h}_i(q,\dot{q}) \\ = & -\frac{1}{h^2}(\gamma_i\beta'(g_i)\langle d_1 g_i,\dot{q}\rangle + \langle \nabla_{\dot{q}} d_1 g_i,\dot{q}\rangle) \\ & + \frac{1}{h^2}\langle d_1 g_i, M_q^{-1} dV_q\rangle - \frac{1}{h^2}\sum_{i=1}^{m}\langle d_1 g_i, M_q^{-1}F_i\rangle u_i \\ & + \frac{(-1)^{\delta_i+1}}{h^2}(\ddot{b}_i - d_1 d_2\Psi_{q,q_i}(\dot{q},\dot{q}_i) \\ & - d_2 d_2\Psi_{q,q_i}(\dot{q}_i,\dot{q}_i) - \langle d_2\Psi_{q,q_i}, \nabla_{\dot{q}_i}\dot{q}_i\rangle).\end{aligned}$$

where M_q is the inertia metric, $d_1 d_2\Psi$ is the differential 2-form as introduced in (6.1), and the last term is the equivalent of second derivative of $\partial h/\partial t$ on the manifold.

In order to make sure each B_i is a CBF, the following condition should be satisfied:

$$\inf_{u\in\mathbb{R}^m}\{\dot{B}_i - \frac{\mu_i}{B_i}\} \le 0, \quad \forall (q,\dot{q}) \in \mathcal{C}_t,\ \forall j \in \{1,2,\cdots,k\}$$

which is equivalent to

$$\inf_{u\in\mathbb{R}^m}\{-\dot{h}_i - \mu_i h_i^3\} \le 0, \quad \forall (q,\dot{q}) \in \mathcal{C}_t,\ \forall j \in \{1,2,\cdots,k\} \tag{6.14}$$

where $\mu_i > 0$ is the increasing rate of the value of B_i as the state moves closer to the boundary.

This condition would fail when the m dimensional vector

$$[\langle dg_i, M_q^{-1}F_1\rangle, \langle dg_i, M_q^{-1}F_2\rangle, \cdots, \langle dg_i, M_q^{-1}F_m\rangle]^T$$

is a zero vector in Cartesian space because otherwise, we could always use a nonzero control input to cancel out all the other terms in 6.14. Using the fact that system (6.3) is *fully-actuated* and that the inertia metric M_q is *non-degenerate*, the condition is equivalent to that $dg_i = (-1)^{\delta_i+1} e_q(q,q_i) = 0$ in the cotangent space.

By the definition of the position error e_q, the set $\mathcal{D}_{i,t}^B = \{(q,\dot{q}) \in \mathcal{C}_t : e_q(q,q_i) = 0\}$ has *measure zero* in M for any $t \in [0,\infty)$. Then by sub-additivity of measure, it follows that

$$\mathcal{D}_t^B := \cup_{i=1}^m \mathcal{D}_{i,t}^B \Rightarrow \mu_L(\mathcal{D}_t^B) \le \sum_{i=1}^{m}\mu_L(\mathcal{D}_{i,t}^B) = 0, \tag{6.15}$$

where $\mu_L(\mathcal{T})$ is the Lebesgue measure of a measurable set $\mathcal{T}$.

Hence, the CBFs defined here hold everywhere except for a set with measure zero. So the candidate

function proposed is an almost globally valid CBF.

***Remark* 6.7.** *Although we haven't given a rigorous definition of Lebesgue measure on manifolds, it can be roughly treated as an estimation of the area of the manifold under study. Thus it is intuitive that any lower dimensional compact submanifold should have measure 0 since we could cover it using a strip with infinitesimal area. In fact, if we want to sample points uniformly from the configuration manifold, the probability of getting points on the lower dimensional compact submanifold is always zero. We refer to [80] for a more formal introduction.*

In order to present the feedback controller in a more concise manner, we denote the terms in $\dot{h}_i$ which multiply u as

$$\phi_0^i(t,q,\dot{q}) = [\langle dg_i, M_q^{-1}F_1\rangle, \cdots, \langle dg_i, M_q^{-1}F_m\rangle]^T,$$

and the terms in $\dot{h}_i$ which are independent of u as $\phi_1^i(t,q,\dot{q})$. In this way, the condition (6.14) which the CBF must satisfy could be reformulated as finding a control input u s.t.,

$$-(\phi_0^i \cdot u + \phi_1^i + \mu_i h_i^3) \leq 0, \quad i \in \{1,2,\cdots,k\}. \tag{6.16}$$

6.2.3 Geometric CLF Candidate

From [70], we could get a candidate CLF with the expression as:

$$V(t,q,\dot{q}) = \alpha\Psi(q,q_d) + \frac{1}{2}\langle\langle e_{\dot{q}}, e_{\dot{q}}\rangle\rangle + \varepsilon\langle e_q, e_{\dot{q}}\rangle$$

where the coefficients $\alpha, \varepsilon > 0$ are chosen specifically to make V quadratic in terms of e_q and $e_{\dot{q}}$.

Differentiating it with respect to t gives us an expression which includes the control input explicitly as:

$$\begin{aligned}
\dot{V} =& \alpha\langle d_1\Psi, e_{\dot{q}}\rangle - \langle\langle e_{\dot{q}}, [\frac{d}{dt}\Big|_{q \text{ fixed}}(\mathscr{T}\dot{q}_d) + (\nabla_{\dot{q}}\mathscr{T})\dot{q}_d]\rangle\rangle \\
&+ \varepsilon[\langle \nabla_{e_{\dot{q}}}(e_q), e_{\dot{q}}\rangle - \langle d_1\Psi, ((\nabla_{e_{\dot{q}}}\mathscr{T})e_{\dot{q}})\rangle] \\
&- \varepsilon\langle e_q, [\frac{d}{dt}\Big|_{q \text{ fixed}}(\mathscr{T}\dot{q}_d) + (\nabla_{\dot{q}}\mathscr{T})\dot{q}_d]\rangle \\
&- [\varepsilon\langle d_1\Psi, M_q^{-1}dV_q\rangle + \langle\langle e_{\dot{q}}, M_q^{-1}dV_q\rangle\rangle] \\
&+ \sum_{i=1}^{m}[\varepsilon\langle d_1\Psi, M_q^{-1}F_i u^i\rangle + \langle\langle e_{\dot{q}}, M_q^{-1}F_i u^i\rangle\rangle]
\end{aligned}$$

Applying a similar argument to this CLF candidate, as we did for the geometric CBF, we have that only

when the vector

$$\psi_0(t,q,\dot{q}) := \begin{bmatrix} [\varepsilon\langle e_q, M_q^{-1}F_1\rangle + \langle\langle e_{\dot{q}}, M_q^{-1}F_1\rangle\rangle] \\ [\varepsilon\langle e_q, M_q^{-1}F_2\rangle + \langle\langle e_{\dot{q}}, M_q^{-1}F_2\rangle\rangle] \\ \vdots \\ [\varepsilon\langle e_q, M_q^{-1}F_n\rangle + \langle\langle e_{\dot{q}}, M_q^{-1}F_m\rangle\rangle] \end{bmatrix}$$

becomes zero, the CLF condition (6.5) fails to hold. By the definition of the position and velocity errors, $e_q, e_{\dot{q}}$, the region

$$\mathcal{D}_t^V = \{(q,\dot{q}) \in \mathcal{C}_t : \psi_0(q,\dot{q}) = 0\} \tag{6.17}$$

has measure zero for any $t \in [0,\infty)$. Hence almost global property also holds for this type of CLF. By denoting the net term which doesn't depend on u as $\psi_1(t,q,\dot{q})$, the condition imposed by this CLF could be written as:

$$\psi_0 \cdot u + \psi_1 + \eta V \leq 0 \tag{6.18}$$

where $\eta > 0$ indicates the convergent rate of CLF.

So far we know the condition of CLF and CBF would only fail to be satisfied for the set $\mathcal{D}_t = \mathcal{D}_t^B \cup \mathcal{D}_t^V$, with $\mathcal{D}_t^B, \mathcal{D}_t^V$ as defined in (6.15), (6.17), where at least one of the vectors ϕ_0^i in (6.16), ψ_0 in (6.18) become zero. We call it *singularity set* of Ψ which depends on the specific configuration error chosen. Note that the singularity here isn't referring to that of the local coordinate, but rather it is related to the specific properties of CBF and CLF.

6.2.4 CBF-CLF-QP Control Design for Fully Actuated, Simple Mechanical Systems

The previous subsections have introduced geometric CBFs and CLFs for the general mechanical systems which hold almost globally. Now we are able to combine them together in the control design using optimization.

First, decompose the total control input into two parts, the feed-forward and feedback components:

$$u = u_{ff} + u_{fb}$$

where the feed-forward term is directly computed as the solution of the linear equation below

$$\sum u_{ff}^i F_i = dV_q(q) + M_q[\frac{d}{dt}\Big|_{q \text{ fixed}} (\mathscr{T}\dot{q}_d) + (\nabla_{\dot{q}}\mathscr{T})\dot{q}_d]$$

which comes from [70] in geometric control.

Then compute the feedback term u_{fb}^i based on the following state-dependent optimization problem.

(CBF-CLF-QP Control Design):

Formulate the feedback control problem based on a state dependent Quadratic Programming (QP):

$$
\begin{aligned}
[u_{fb}, \delta^*] = \underset{v \in \mathbb{R}^m, \delta \in \mathbb{R}}{\operatorname{argmin}} \; & \frac{1}{2} v^T H v + \frac{1}{2} \lambda \delta^2 \\
\text{s.t.} \quad & \psi_0 \cdot v + [\psi_0 \cdot u_{ff} + \psi_1 + \eta V] \leq \delta, \\
& -(\phi_0^i \cdot v + \phi_0^i \cdot u_{ff} + \phi_1^i + \mu_i h_i^3) \leq 0,
\end{aligned} \tag{6.19}
$$

where the weight matrix $H \in \mathbb{R}^{m \times m}$ is positive definite, and $\lambda > 0$ is the penalty weight for the relaxation parameter δ.

***Remark* 6.8.** *As discussed in [46], the constraints imposed by CBFs are treated as Hard Constraints that must be satisfied during the whole control process, while the constraint imposed by the CLF is a Soft Constraint with the relaxation parameter δ. The hard constraints are the safety-critical constraints. Also recall that $\alpha, \varepsilon, \eta > 0$ are scalars that are relevant to the CLF V while $\beta : \mathbb{R} \to \mathbb{R}, \gamma_i, \mu_i > 0$ are function and parameters that are relevant to each CBF B_i. These parameters can be tuned to improve the controller's performance in numerical implementations.*

***Remark* 6.9.** *In order to guarantee the existence and uniqueness of system trajectory, we require the control input should be at least piecewise continuous. We refer to [81, 82] for detailed discussion on the continuity of the solution of state dependent QPs. The solution is guaranteed to be Lipschitz continuous under certain conditions, which can be satisfied for our problem.*

The proposition below shows the safety guarantee of CBF-CLF-QP controller.

***Proposition* 6.5.** *(Safety Property of CBF-CLF-QP Controller)*

If the following conditions are satisfied:

- *There exist proper parameters for both V and B_i, $i \in \{1, 2, \cdots, k\}$ such that the initial condition $(q_0, \dot{q}_0)$ stays within the expanded safety region $\mathcal{C}_t^\circ$.*
- *The singularity set $\mathcal{D}_t$ where CBF or CLF fail has measure zero.*
- *The online QP* (6.19) *has feasible solution for the set $\mathcal{C}_t \backslash \mathcal{D}_t$.*

then the trajectory of the closed loop system (6.3) *is safe for all $t \in [0, \infty)$.*

Proof. See Appendix 10.12. □

Having presented the general safety-critical control design using geometric CBFs and geometric CLFs, we next specialize the controller to some simple mechanical systems and present numerical results.

6.3 Simulation Examples of CBF-CLF-QP Control

In this part, we will present simulation results on several typical simple mechanical systems including (a) a single point mass in $\mathbb{R}^3$, (b) a spherical pendulum in S^2 and (c) a 3D pendulum in $SO(3)$, which are common fully-actuated, simple mechanical systems.

6.4 Safety Critical Control for a 3D Point Mass

We begin by specializing the general construction presented in the previous section to a 3D-moving point mass in $\mathbb{R}^3$ by developing the associated geometric CLF and CBF, and presenting numerical results for both time-invariant and time-varying safety-critical constraints. In subsequent sections, this formulation is extended to a spherical pendulum in S^2 and a 3D pendulum in $SO(3)$.

Consider a single 3D point mass, with configuration space $\mathcal{Q} = \mathbb{R}^3$ with Cartesian position $q = [x, y, z]^T \in \mathcal{Q}$, velocity $\dot{q} \in T_q\mathcal{Q} = \mathbb{R}^3$, input $u = [u_1, u_2, u_3]^T \in \mathbb{R}^3$, and mass m, such that the system dynamics is $\ddot{q} = m^{-1}u$ with Riemannian metric $\langle\langle \dot{q}_1, \dot{q}_2 \rangle\rangle = m\dot{q}_1 \cdot \dot{q}_2$. Given a smooth reference trajectory $q_d \in \mathbb{R}^3$, the configuration error can be written as,

$$\Psi(q, q_d) = ||q - q_d||^2 = (q - q_d) \cdot (q - q_d),$$

which is equipped with the differential,

$$e_q = d_1\Psi(q, q_d) = 2(q - q_d), \quad d_2\Psi(q, q_d) = -2(q - q_d).$$

The compatible transport map satisfying (6.2) is given by,

$$\mathscr{T}_{(q,q_d)}\dot{q}_d = \dot{q}_d \implies e_{\dot{q}} = \dot{q} - \dot{q}_d,$$

since we can directly compare two tangent vectors in $\mathbb{R}^3$.

Next, consider a list of safety constraints $g_i(q) = (-1)^{\delta_i}(b_i - \Psi(q, q_i)) \geq 0$ with $i \in \{1, 2, \cdots, k\}, b_i > 0$, wherein with our chosen configuration error Ψ, each constraint $g_i(q)$ represents the safe set as either a sphere centered at q_i or its complement, depending on the choice of $\delta_i \in \{0, 1\}$ respectively. The corresponding

CLF and CBF candidates can be respectively constructed as:

$$
\begin{aligned}
V &= \frac{1}{2}m(||\dot{q}-\dot{q}_d||^2) + \frac{\alpha}{2}(||q-q_d||^2) \\
&\quad + \varepsilon(q-q_d)\cdot(\dot{q}-\dot{q}_d), \\
h_i &= \gamma_i g_i + (-1)^{\delta_i}[\dot{b}_i - 2(q-q_i)\cdot(\dot{q}-\dot{q}_i)], \\
B_i &= 1/h_i,
\end{aligned} \tag{6.20}
$$

where $\alpha > 0$ and $\gamma_i > 0$. Note that the feedforward input is given by $u_{ff} = m\ddot{q}_d$. We incorporate the above into the online QP in (6.19). Also, using the definitions in (6.15) and (6.17), we can specify the singular sets as

$$
\begin{aligned}
\mathcal{D}_t^V &= \{(q,\dot{q}) \in \mathbb{R}^6 : me_{\dot{q}} + \varepsilon e_q = 0\}, \\
\mathcal{D}_t^B &= \cup_{i=1}^k \{(q,\dot{q}) \in \mathbb{R}^6 : q = q_i\},
\end{aligned}
$$

which are the union of several curves in $T\mathcal{Q} \simeq \mathbb{R}^6$ and thus have measure zero.

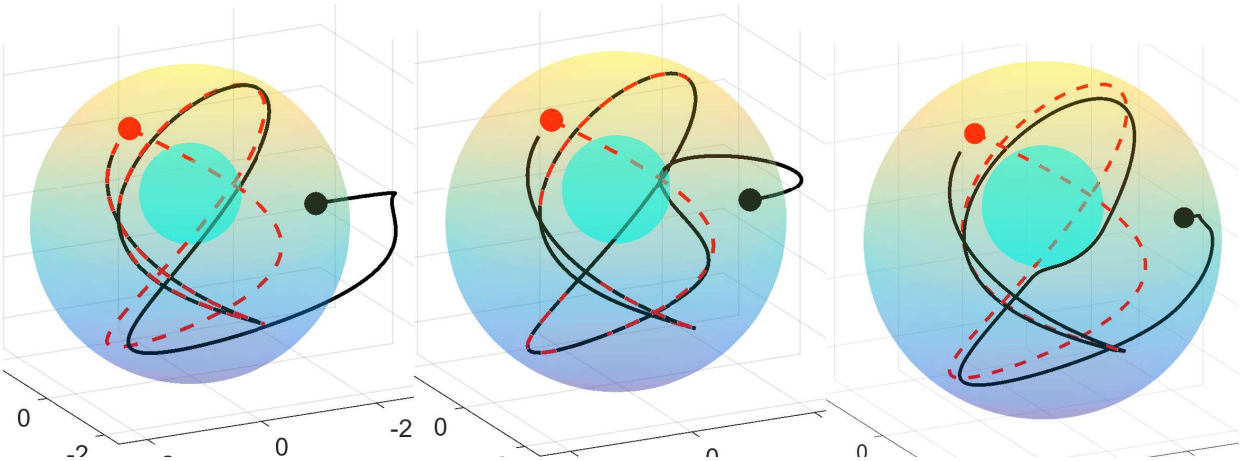
6.4.1 3D Point Mass with Time-Invariant Safety Constraints

Based on the previous construction, we now numerically validate our proposed controller. We consider a point mass with $m = 2.5kg$ moving strictly inside a large safety region ($g_1(q) \geq 0$) and avoiding a small spherical obstacle inside the safety region ($g_2(q) \geq 0$) while tracking a desired reference trajectory $q_d(t)$, which in the extreme case, passes directly through the obstacle (see red dashed line in Fig. 6.1). These safety constraints are

$$
\begin{aligned}
g_1 &= (-1)^0(2.5^2 - \Psi(q,[0,0,0]^T)) \geq 0, \\
g_2 &= (-1)^1(0.85^2 - \Psi(q,[0,0,0.5]^T)) \geq 0,
\end{aligned} \tag{6.21}
$$

while the reference trajectory is $q_d(t) = [-0.5\sin 1.5t, 1.25\cos(1.25t+\pi/4), 0.75\cos(0.75t+\pi/6)]^T$.

We compare the performance of three different controllers, a geometric CLF min-norm controller, traditional geometric controller from [70], and our proposed geometric CBF-CLF-QP controller. The results in Fig. 6.1 show that the CBF-CLF-QP controller keeps the point mass away from boundary of the unsafe regions to strictly enforce the safety constraints encoded by the non-negativity of g_1, g_2 as seen in Fig. 6.2. We also plot the geometric CLF in Fig. 6.2 while highlighting the durations when the reference becomes unsafe. It can be seen that the actual trajectory exponentially tracks the reference when the reference is safe.



(a) Min-Norm (b) Geometric Control (c) CBF-CLF-QP

Figure 6.1: *(3D Point Mass with Time-Invariant Safety Constraints)*: Simulation of various controllers on the point mass system on $\mathbb{R}^3$, which is required to track a desired trajectory while being restricted to move within the region between two spheres. As can be seen for (a) min-norm, and (b) geometric control, the system trajectory (black solid) exits the outer sphere as well as enters the inner sphere, violating critical safety region constraints. However, for (c) CLF-CBF-QP controller, the critical safety constraint is enforced while still following the desired trajectory (red dashed).

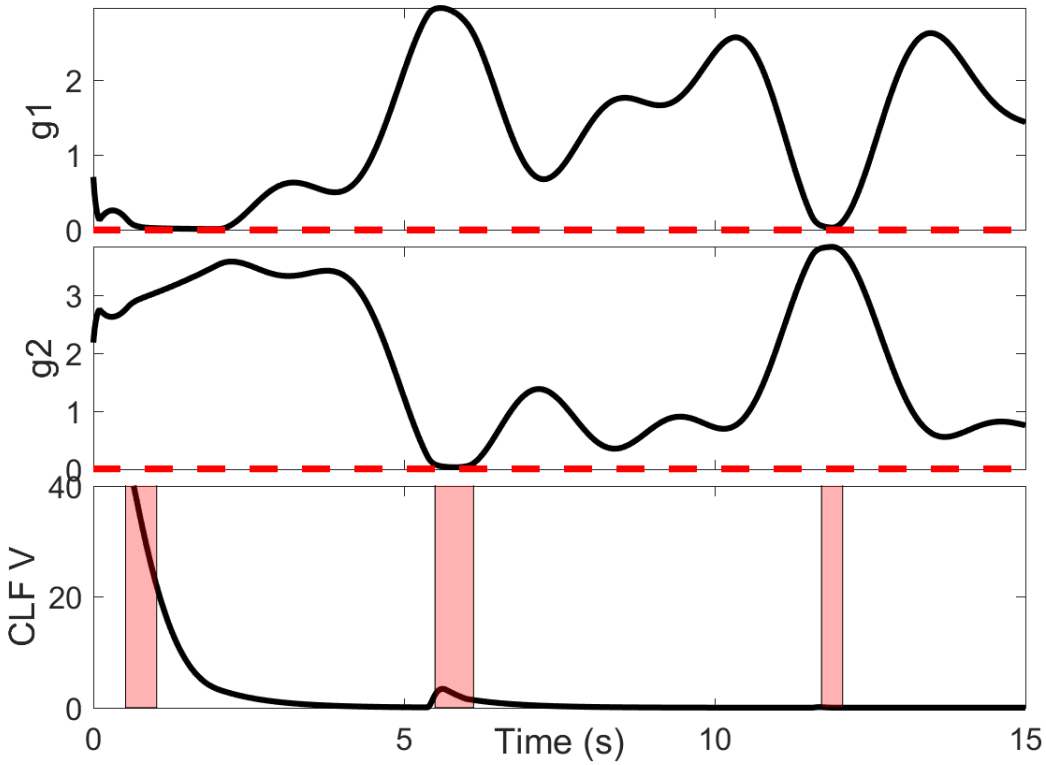


Figure 6.2: *(3D Point Mass with Time-Invariant Safety Constraints)*: The proposed controller enforces the constraints $g_i \geq 0$. Durations when the reference trajectory is unsafe are highlighted in red, during which the geometric CLF could increase due to the relaxation δ in (6.19).

6.4.2 3D Point Mass with Time-Varying Safety Constraints

We next consider time-varying safety constraints by introducing time-varying obstacles that the point-mass needs to avoid. In particular, we consider a desired reference trajectory $q_d(t) = [-\sin 1.25t, \cos 1.25t, 0]^T$, which is a circle in the XY plane centered at the origin. We require the point mass to stay within a safety region given by a ball of radius $r_0(t) \equiv 3.0$ centered at the origin (large transparent outer sphere in Fig.6.3) while avoiding the two sphere-shaped obstacles of radii $r_1(t) = r_2(t) \equiv 0.15$ (two small spheres in Fig.6.3) that move along the circular reference trajectory in opposite directions with angular velocities $\omega_1 = 0.5,\ \omega_2 = 0.8$. To be precise, the corresponding safety constraints are given below as:

$$\begin{aligned} g_1 &= (-1)^0(r_0^2 - \Psi(q,0)) \geq 0, \\ g_2 &= (-1)^1(r_1^2 - \Psi(q,q_1(t))) \geq 0, \\ g_3 &= (-1)^1(r_2^2 - \Psi(q,q_2(t))) \geq 0, \end{aligned} \tag{6.22}$$

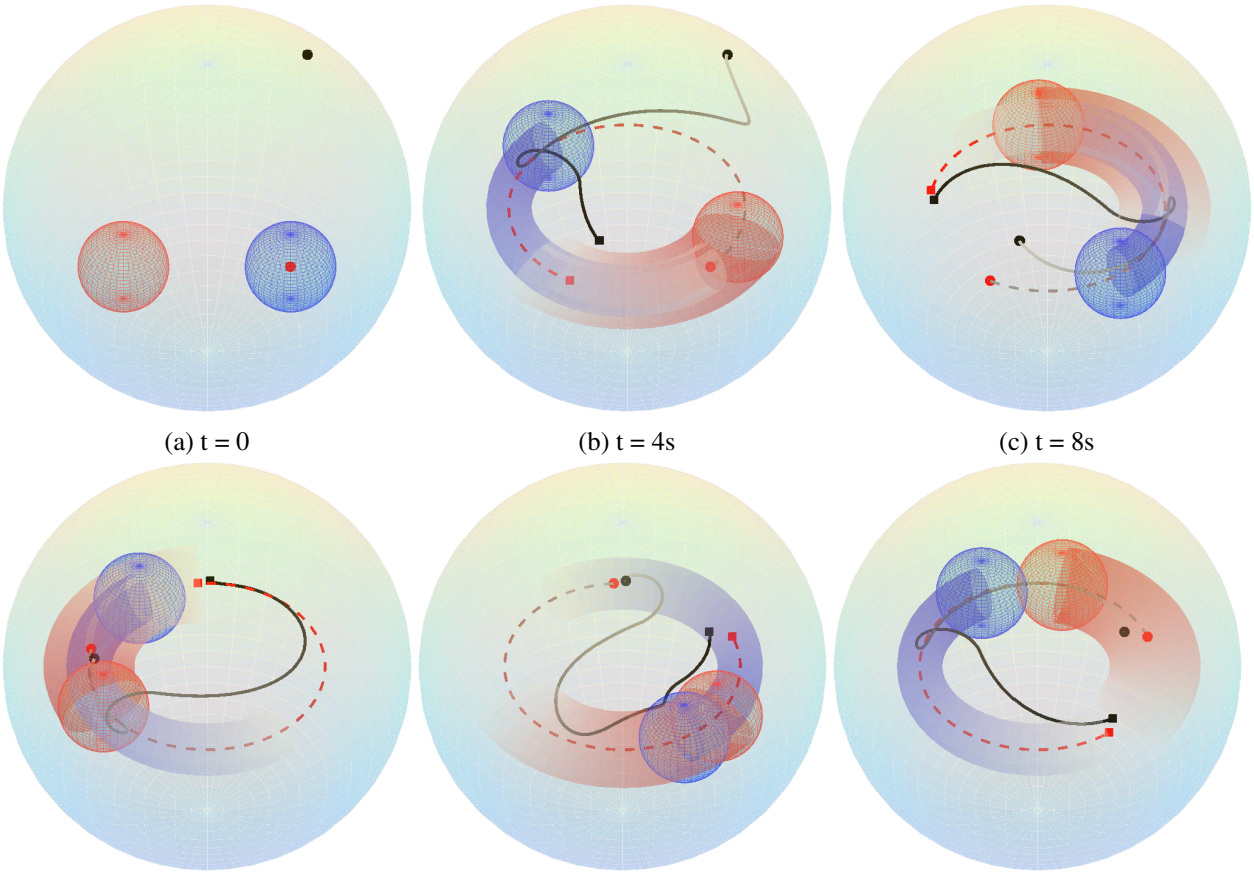
where $q_1(t) = [\cos(\omega_1 t), \sin(\omega_1 t), 0]^T, q_2(t) = [\sin(\omega_2 t), \cos(\omega_2 t), 0]^T$.

Fig.6.3 illustrates snapshots in time, wherein the motion of the obstacles are depicted through shaded regions and time is conveyed through change in transparency from light to dark. As can be seen, the actual trajectory (black solid) avoids the moving obstacles while trying to follow the desired trajectory (red dashed). The values of each constraint function g_i and the geometric CLF are plotted in Fig. 6.4. As can be seen, the controller ensures the non-negativity of the constraints. The periodic fluctuation is due to the reference trajectory encountering a moving obstacle every three seconds. When the reference trajectory becomes unsafe (indicated by red shared region), the controller relaxes tracking while ensuring safety.

***Remark* 6.10.** *As can be seen from Fig. 6.4, the value of the CLF could potentially increase in the red shaded regions, where the reference is unsafe. This is natural, since to remain safe we have to keep the system state away from the unsafe reference. Typically this is realized either through re-planning of the desired reference or through switched control schemes which are turned on when the actual state is close to the unsafe region. In contrast, due to the existence of the relaxation parameter δ in the online QP* (6.19)*, we are able to smoothly mediate between stable tracking of the reference and staying safe when the reference is unsafe.*

6.5 Safety Critical Control for a Spherical Pendulum

We next consider a spherical pendulum system that comprises of a point mass connected to a pivot through a mass-less rod as shown in Fig. 6.5. The configuration of this system is given by the unit sphere



(a) t = 0 (b) t = 4s (c) t = 8s

(d) t = 12s (e) t = 16s (f) t = 20s

Figure 6.3: *(3D Point Mass with Time-Varying Safety Constraints)*: Snapshots of a 3D moving point mass subject to following a desired trajectory while avoiding dynamic obstacles are shown. Each snapshot illustrates the past four seconds of the actual trajectory of the point-mass (black solid) and the desired trajectory (red dashed), with start and end positions marked by small circle and square markers respectively. Increasing time is conveyed by a change in transparency from light to dark.

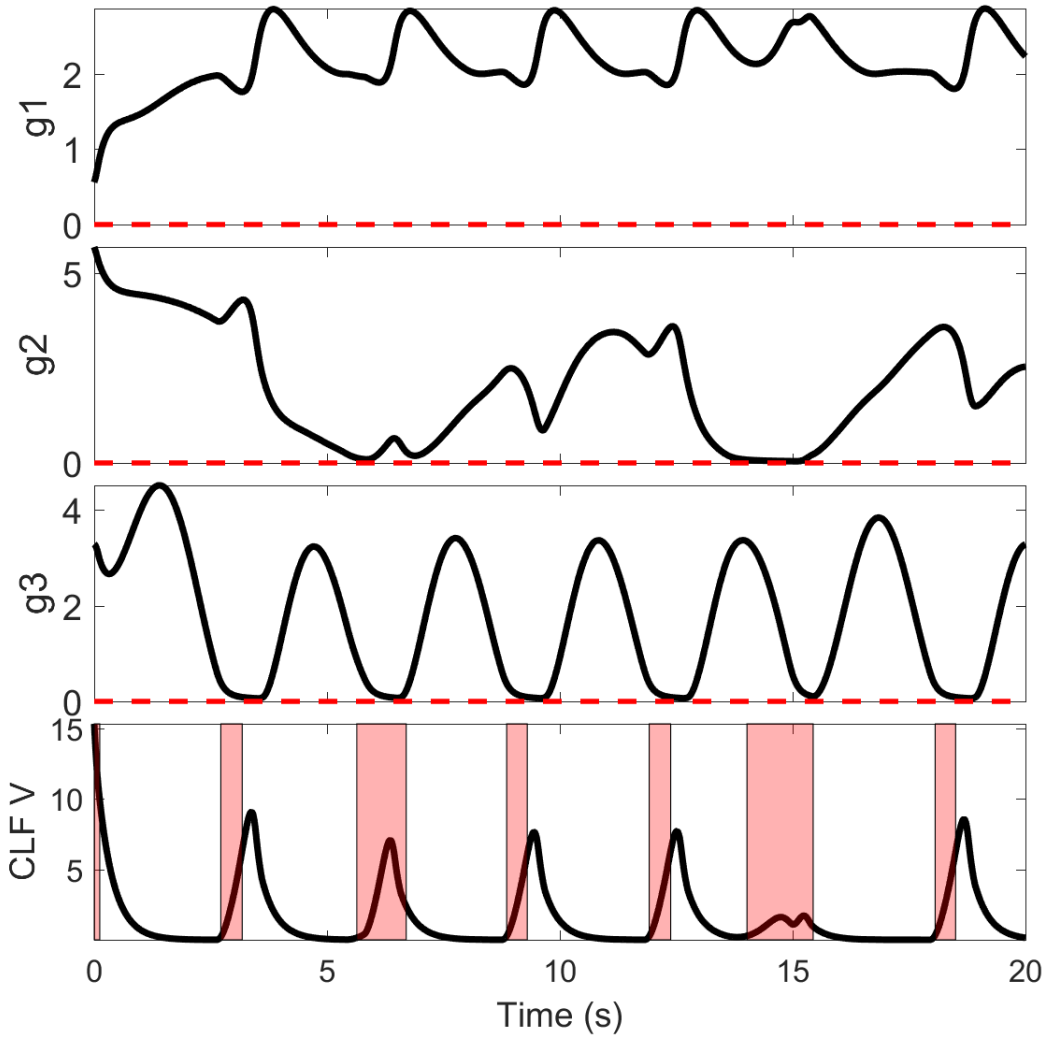


Figure 6.4: *(3D Point Mass with Time-Varying Safety Constraints)*: Plots of constraint functions g_i and the geometric CLF V. The proposed controller ensures $g_i \geq 0$. Durations when the reference trajectory is unsafe are highlighted in red.

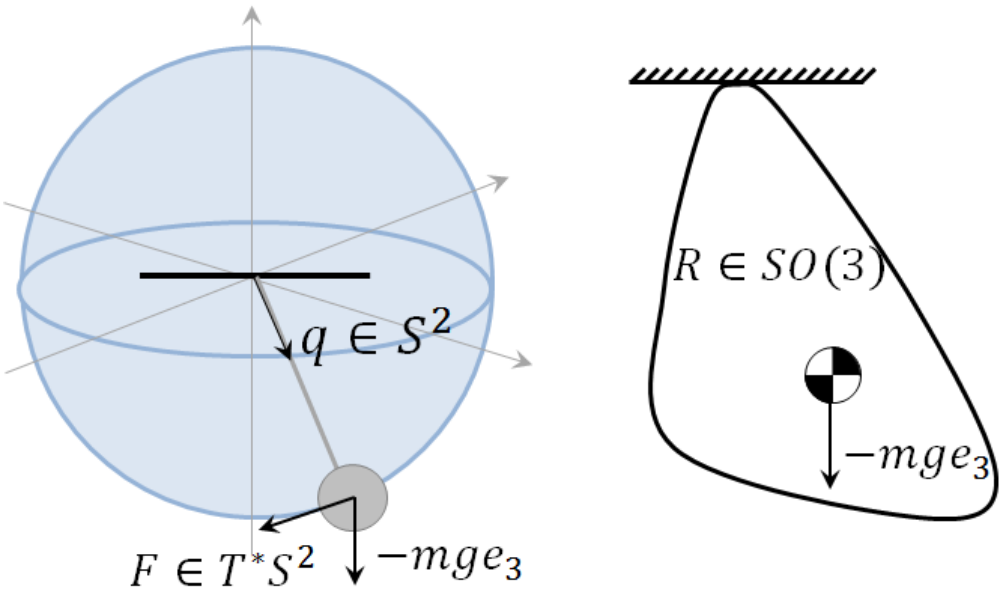


Figure 6.5: Diagram of the spherical and 3D pendulums.

$\mathbb{S}^2$. Using the directional vector $q \in \mathbb{S}^2$ corresponding to the unit vector from the pendulum pivot to the point mass, we have the dynamical equation given as,

$$\begin{aligned} \dot{q} &= \omega \times q, \\ \dot{\omega} &= q \times (\tfrac{F}{ml} - \tfrac{g}{l} e_3), \end{aligned} \qquad \text{or} \quad \ddot{q} + (\dot{q} \cdot \dot{q}) q = -\hat{q}^2 (\frac{F}{ml} - \frac{g}{l} e_3),$$

which is a fully-actuated simple mechanical system. The hat map $\hat{\cdot}: \mathbb{R}^3 \to \mathfrak{so}(3)$ converts a three-dimensional vector to a skew-symmetric real matrix as follows:

$$\hat{x} = \begin{bmatrix} \hat{x_1} \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \implies x \times y = \hat{x} y$$

Denote a normalized force $u := (F/ml - g/le_3)$ and thus the system dynamics could be simplified as:

$$\ddot{q} + (\dot{q} \cdot \dot{q}) q = -\hat{q}^2 u. \tag{6.23}$$

For this system, the Riemmanian metric degenerates to the normal inner product $\langle\langle \dot{q}_1, \dot{q}_2 \rangle\rangle = \dot{q}_1 \cdot \dot{q}_2$. The configuration error can be defined as:

$$\Psi(q, q_d) = 1 - q \cdot q_d, \quad e_q = d_1 \Psi = \hat{q}^2 q_d, \quad d_2 \Psi = \hat{q}_d^2 q.$$

The compatible transport map and velocity error are given by:

$$\mathscr{T}_{(q,q_d)}\dot{q}_d = (q_d \times \dot{q}_d) \times q \implies e_{\dot{q}} = \dot{q} - (q_d \times \dot{q}_d) \times q.$$

Next, given a smooth reference trajectory $q_d \in \mathbb{S}^2$ and a list of constraints $g_i = (-1)^{\delta_i}(b_i - \Psi(q, q_i)) \geq 0$ where $b_i \in (-1, 1)$, the corresponding CLF and CBF are given as

$$\begin{aligned} V &= \frac{1}{2} e_{\dot{q}} \cdot e_{\dot{q}} + \frac{1}{2}\alpha(1 - q \cdot q_d) + \varepsilon e_q \cdot e_{\dot{q}}, \\ h_i &= \gamma_i g_i + (-1)^{\delta_i}(q_i \cdot \dot{q} + \dot{q}_i \cdot \dot{q} - \dot{b}_i), \\ B_i &= 1/h_i, \end{aligned} \tag{6.24}$$

where $\delta_i \in \{0, 1\}$, and $\gamma_i > 0$.

The constraint defined here could be visualized as a cone centered at q_i bounded by the radius b_i. Depending on the value of δ_i, the cone is treated as either unsafe or safe region. The corresponding feedforward input is $u_{ff} = (q_d \times \ddot{q}_d) \times q - \hat{q}^2(q_d \times \dot{q}_d) \times \dot{q}$, which is used by the controller (6.19).

In order to analyze the singularity set $\mathcal{D}_t$, we write out the vectors in (6.15) and (6.17) explicitly as

$$\phi_0^i = (-1)^{\delta_i} q_i, \quad \psi_0^i = (e_{\dot{q}} + \varepsilon e_q).$$

The corresponding singularity set for the spherical pendulum could be defined as:

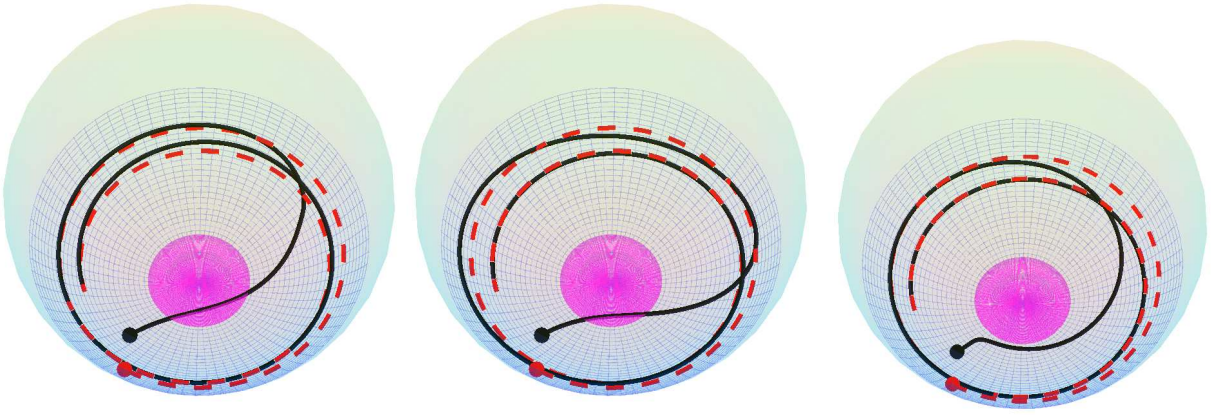
$$\begin{aligned} \mathcal{D}_t^V &= \{(q, \dot{q}) \in T\mathbb{S}^2 : e_{\dot{q}} + \varepsilon e_q = 0\}, \\ \mathcal{D}_t^B &= \cup_{i=1}^k \{(q, \dot{q}) \in T\mathbb{S}^2 : q = \pm q_i\}, \end{aligned}$$

which also has measure zero in the state-space.

6.5.1 Spherical Pendulum with Time-Invariant Safety Constraints

Using the above formulations for the spherical pendulum, we compare the performance of the geometric min-norm, geometric controller in [70], and the proposed geometric CBF-CLF-QP controller for the normalized system (6.23). In this scenario, the safety region is defined to be the difference of two cone regions on the unit sphere $\mathbb{S}^2$. The corresponding safety constraints are given as:

$$\begin{aligned} g_1 &= (-1)^0(\cos(\pi/12) + 1 - \Psi(q, q_n)) \geq 0, \\ g_2 &= (-1)^1(\cos(\pi/4) + 1 - \Psi(q, q_n)) \geq 0, \end{aligned} \tag{6.25}$$



(a) Min-Norm (b) Geometric Control (c) CBF-CLF-QP

Figure 6.6: *(Spherical Pendulum with Time-Invariant Safety Constraints)*: Simulation of various controllers on the spherical pendulum system on $\mathbb{S}^2$, restricted to remain between two cones in a unit sphere. The inner (magenta) cone represents the unsafe region while the outer (blue) cone represents the safe region. For (a) min-norm, and (b) geometric control, the system trajectory enters the unsafe inner cone area, whereas for (c) CBF-CLF-QP, the controller ensures the trajectory remains within the safe set while converging to the desired trajectory.

which represent the outer and inner cones respectively in Fig. 6.6, with the static center point given by $q_n = [0,0,-1]^T$.

The results in Fig. 6.6 illustrate that our proposed controller is able to keep the pendulum outside the inner (red) unsafe cone and inside the outer (blue) safe cone, thereby guaranteeing the safety constraints encoded by the non-negativity of g_1, g_2 as seen in Fig. 6.7. The tracking convergence is illustrated by the CLF plot in Fig. 6.7.

6.5.2 Spherical Pendulum with Time-Varying Safety Constraints

We next consider time-varying safety constraints on $\mathbb{S}^2$ by introducing two time-varying cone constraints, where the cone axis is specified through time-varying trigonometric functions with the cone radii held constant. In particular, the safety constraints are given as:

$$\begin{aligned} g_1 &= (-1)^0(\cos(\pi/12)+1-\Psi(q,q_n(t))) \geq 0, \\ g_2 &= (-1)^1(\cos(\pi/4)+1-\Psi(q,q_n(t))) \geq 0, \end{aligned} \tag{6.26}$$

which represent the outer cone and inner cone separately, with the time-varying axis given by $q_n(t) = [\sin(\pi/5)\cos(0.25t), \sin(\pi/5)\sin(0.25t), -\cos(\pi/5)]^T$.

Fig. 6.8 illustrates snapshots in time, where in the motion of the cones are depicted through shaded

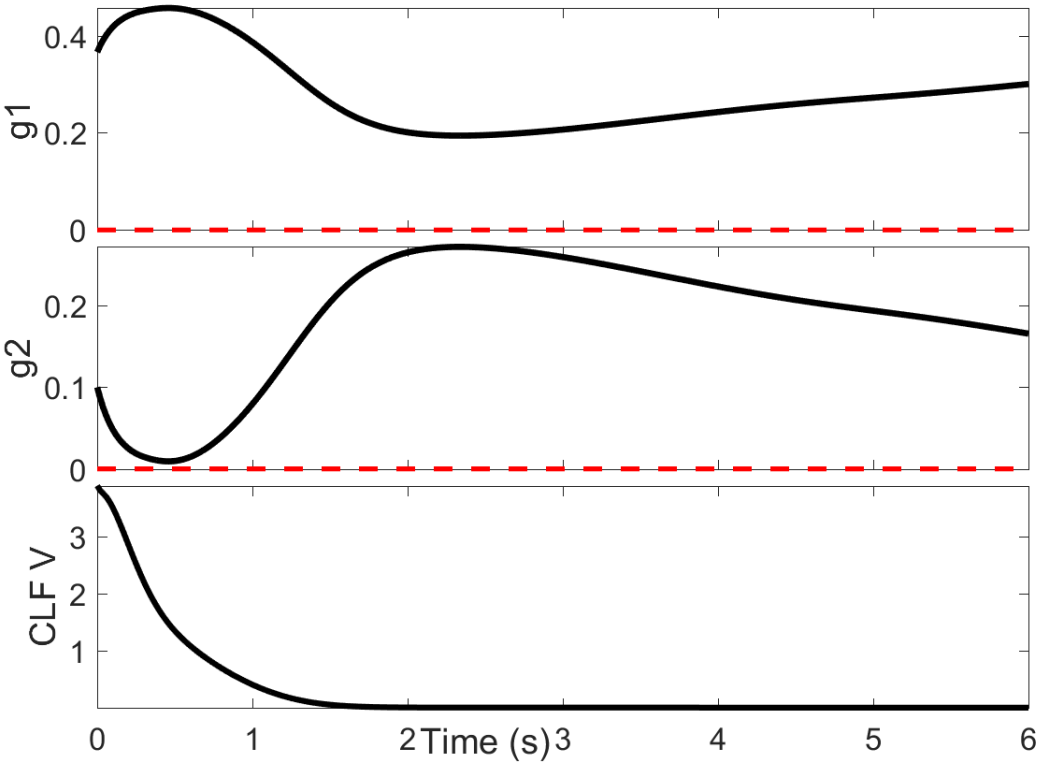
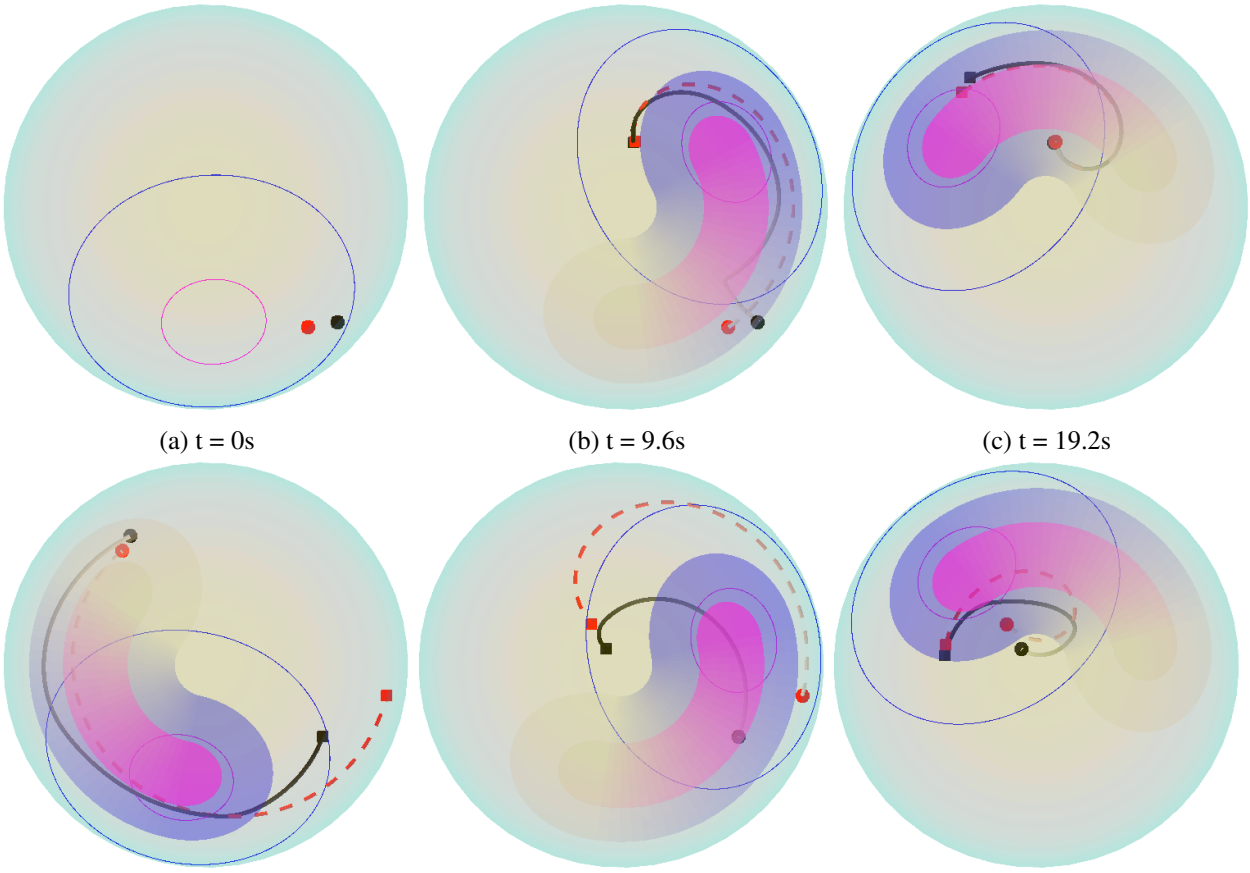


Figure 6.7: *(Spherical Pendulum with Time-Invariant Safety Constraints)*: Plots of constraint functions g_i and the geometric CLF V. The proposed controller ensures $g_i \geq 0$.



(a) t = 0s (b) t = 9.6s (c) t = 19.2s

(d) t = 28.8s (e) t = 38.4s (f) t = 48s

Figure 6.8: *(Spherical Pendulum with Time-Varying Safety Constraints)*: Snapshots of a single spherical pendulum subject to two time-varying cone constraints are shown. The inner (red) cone is an obstacle which should be avoided while the outer (blue) cone is the safe region that the pendulum should remain within. The black solid line is the actual trajectory of the pendulum's point mass and the red dashed line represents the reference trajectory. The first snapshot shows the initial position and shape of the safety set. The rest of the snapshots show the inner and outer cones with changing transparency to indicate the progress of time. Each snapshot also shows the boundary of the cones at the ending time.

regions and time is conveyed through change in transparency from light to dark. The outer (blue) cone is safe, while the inner (red) cone is unsafe. The reference trajectory for the spherical pendulum is shown in red dashed line while the actual trajectory is drawn in black solid line. As seen in Fig. 6.9, the controller enforces the safety constraints while tracking the desired trajectory by maintaining the non-negativity of g_1, g_2. Fig. 6.9 also shows the plot of geometric CLF for the spherical pendulum. As can be seen, the value of CLF keeps decreasing when the reference is in the safe region and potentially increases when the reference is unsafe (highlighted by red region).

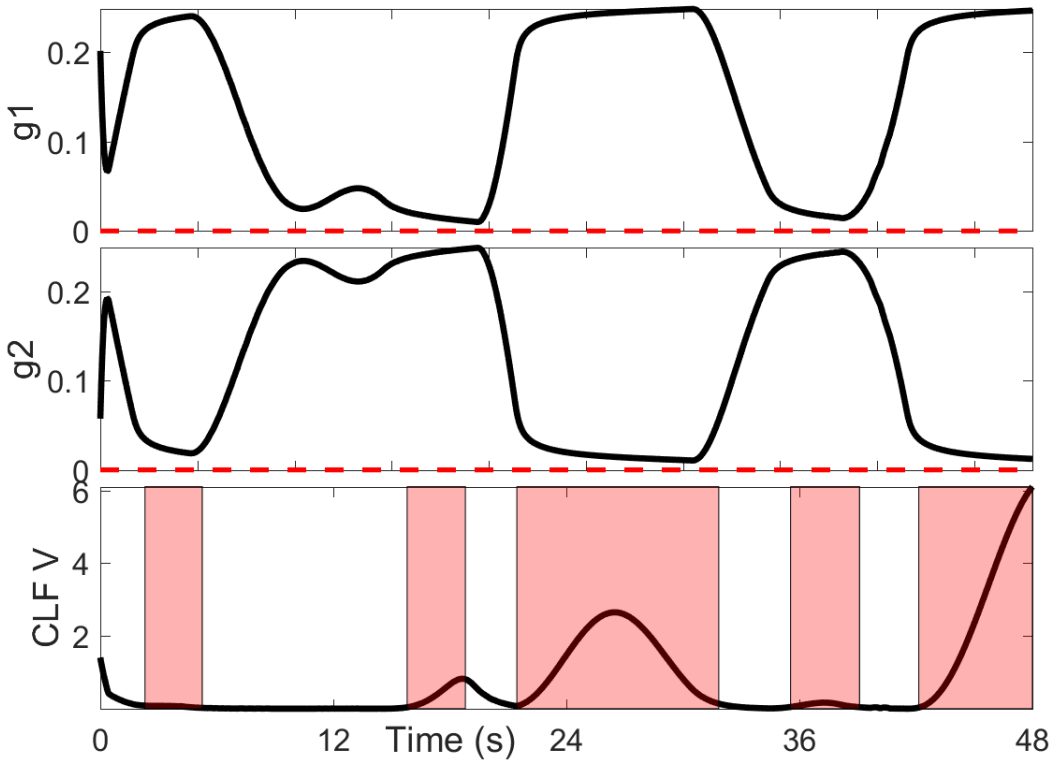


Figure 6.9: *(Spherical Pendulum with Time-Varying Safety Constraints)*: Plots of constraint functions g_i and geometric CLF V. The proposed controller ensures $g_i \geq 0$.

6.5.3 Safety Critical Control for a 3D Pendulum

The mechanical system considered next is a 3D pendulum attached to a pivot, also shown in Fig. 6.5. The orientation of this rigid body is controlled by a torque exerted at the pivot. To be consistent with the symbolic annotation of previous literature, we will use R and Ω to represent the orientation configuration and the angular velocity of the 3D pendulum, instead of $q, \dot{q}$ as used for the spherical pendulum. Note that the configuration space of the 3D Pendulum is $SO(3)$. The system dynamics are given in [53] as:

$$\dot{R} = R\Omega,\ J\dot{\Omega} = J\Omega \times \Omega + \tau,$$

where τ represents the torque at the pivot.

Since the configuration space is a Lie group, it is naturally a *smooth Riemannian manifold* and admits more elaborate algebraic and topological structure. As a group, the manifold allows for multiplication and the existence of an identity. The tangent vector at the identity is called Lie algebra which can be identified with the tangent space at any point through left and right multiplication. More information about the properties of Lie group can be found in [51]. The metric between two elements $R, R_d \in SO(3)$ here is

given by the *right attitude error* as:

$$\Psi(R,R_d) = \frac{1}{2}\mathrm{Tr}(I - R_d^T R).$$

To represent the tangent vector at R, we use the *body-fixed angular velocity* Ω that's related to $\dot{R}$ through the equation $\dot{R} = R\hat{\Omega}$. In this representation of body angular velocity, the corresponding position error could be given by:

$$e_R = d_1\Psi(R,R_d) = \frac{1}{2}(R_d^T R - R^T R_d)^\vee,\ d_2\Psi(R,R_d) = -e_R$$

where we have identified the tangent space with the corresponding angular velocity space $\mathbb{R}^3$.

Note that the vector e_R is an element in the cotangent space, and here its value with any angular velocity is given by the normal inner product in $\mathbb{R}^3$, as detailed explained in [70]. A compatible transport map is given by $\mathscr{T}_{(R,R_d)}(\Omega_d) = R^T R_d \Omega_d$ where $\Omega_d = (R_d^T \dot{R}_d)^\vee$, which is the desired angular velocity. Thus the velocity error has the expression $e_\Omega = \Omega - R^T R_d \Omega_d$.

Given a smooth reference trajectory $R_d(t) \in SO(3)$ and a list of constraints $g_i = (-1)^{\delta_i}(b_i - \mathrm{Tr}(I - R_i^T R)/2) \geq 0$, the corresponding geometric CLF and CBF candidates are chosen as:

$$\begin{aligned} V &= \frac{1}{2}e_\Omega^T J e_\Omega + \frac{1}{2}\alpha \cdot \mathrm{Tr}(I - R_d^T R) + \varepsilon e_R \cdot e_\Omega, \\ h_i &= \gamma_i g_i + (-1)^{\delta_i}[\frac{1}{2}v_i \cdot (\Omega - \Omega_i) + \dot{b}_i], \\ B_i &= 1/h_i, \end{aligned} \tag{6.27}$$

where

$$v_i = \begin{bmatrix} (R_i^T R)_{23} - (R_i^T R)_{32} \\ (R_i^T R)_{31} - (R_i^T R)_{13} \\ (R_i^T R)_{12} - (R_i^T R)_{21} \end{bmatrix},\ \Omega_i = (R_i^T \dot{R}_i)^\vee,\ b_i \in (0,2),$$

and the feedforward term is given as:

$$\tau_{ff} = \Omega \times J\Omega + J(-\hat{\Omega}R^T R_d \Omega_d + R^T R_d \dot{\Omega}_d).$$

Following the similar derivation in $\mathbb{S}^2$, we have the singularity set $\mathcal{D}_t$ on $SO(3)$ as below:

$$\begin{aligned} \mathcal{D}_t^V &= \{(R,\Omega) \in TSO(3) : Je_\Omega + \varepsilon e_R = 0\}, \\ \mathcal{D}_t^B &= \cup_{i=1}^k \{(R,\Omega) \in TSO(3) : v_i = 0\}, \end{aligned}$$

which is also the union of several curves in the state space $TSO(3)$ and thus has measure 0.

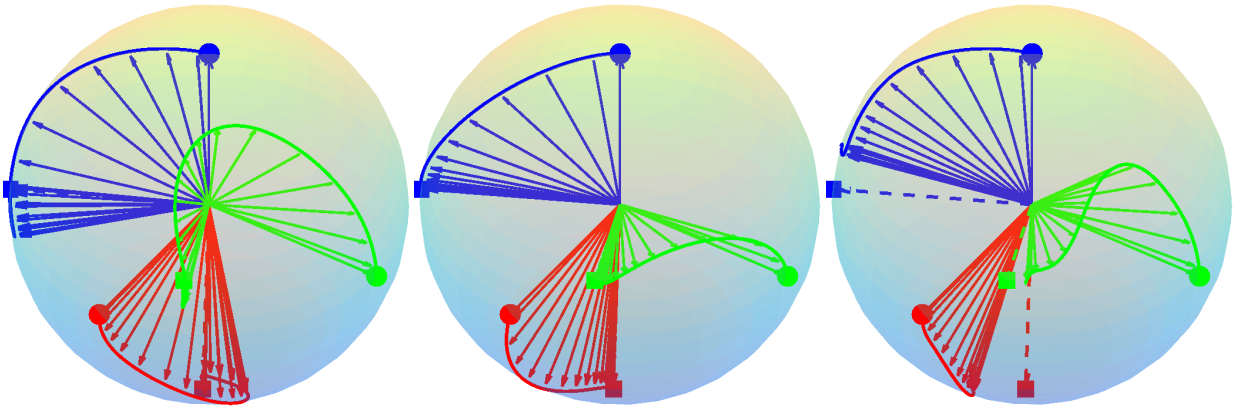
We next consider numerical validation of our proposed controller on the 3D pendulum system. In this scenario, the 3D pendulum is required to move from an initial orientation to a final orientation. The safety constraint is given by

$$g = (-1)^1 \left(\frac{1}{2}\text{Tr}(I - R) - 0.75 \right) \geq 0 \tag{6.28}$$

with the inertia matrix $J = diag(0.1, 0.2, 0.5)$ and goal location $R_d(t) = \exp(\hat{\xi})$ where $\xi = [0.5, 1.5, 0]^T$.

The corresponding simulation results are shown in Fig. 6.10, Fig. 6.11 and Fig. 6.12. In Fig. 6.10, we depict the trajectories of each unit vector of the 3-axis frame on a unit sphere $\mathbb{S}^2$ to compare the performance of the previous three controllers, geometric min-norm, geometric controller in [70], and the geometric CBF-CLF-QP controller. The circles on the sphere indicate the initial positions of each axis of the reference R_d while the squares represent the goal positions of each axis of the reference R_d. From Fig. 6.10, we see that both the geometric controller and the CLF-minnorm controller converge to the goal, while the CBF-CLF-QP controller stops at a distance from the final goal, since the final goal configuration is unsafe. We can also verify safety by looking at the trajectory of g shown in Fig. 6.11. The fact that g is always positive establishes the safety guarantee of our proposed CBF-CLF-QP controller on a Lie group. On the other hand, since the goal is always unsafe, the whole control process is highlighted as unsafe in Fig. 6.12. Note that the value of geometric CLF would decrease to a nonzero constant, which indicates that the actual trajectory would reach to a point which is the minimum of CLF in the safety region.

This chapter formulate CBF-CLF-QP controller for the simple, fully-actuated mechanical systems, as studied in geometric control literature [70]. In particular, we numerically demonstrate its safety-critical performance on three different mechanical systems with configuration spaces as, $\mathbb{R}^3, \mathbb{S}^2$, and $SO(3)$, to satisfy both time-invariant and time-varying safety constraints. Using the same type of CBF-CLF-QP idea, we would investigate on constructing CBF for a single quadrotor system in the next chapter.



(a) Min-Norm (b) Geometric Control (c) CBF-CLF-QP

Figure 6.10: *(3D Pendulum in $SO(3)$ with static safety constraints)*: Comparison between various controllers for the 3D pendulum on $SO(3)$. The trajectories of all three directional vectors are plotted out on a sphere for better visualization. Here we use the dashed line to the static reference and set up a safe region as $\Psi(R,I) \leq 0.75$. As can be seen for (a) min-norm, and (b) geometric control, the actual rotation matrix would tend to desired one. However, for (c) CBF-CLF-QP controller, the actual rotation is forced a distance away from it due to the imposed constraint.

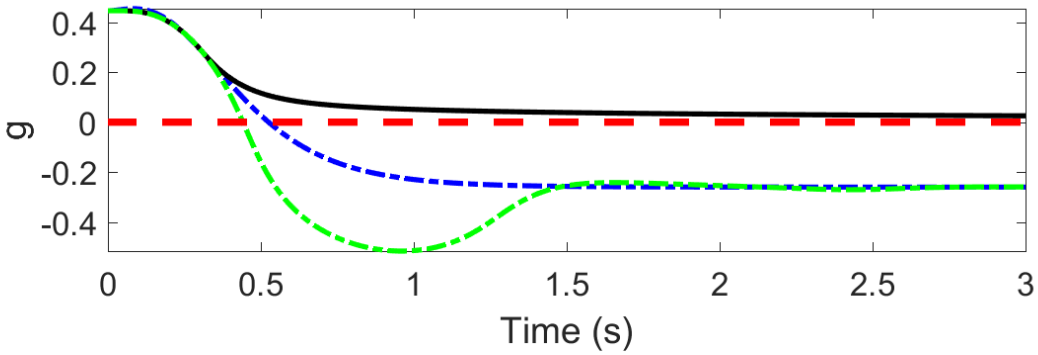


Figure 6.11: The value change of the constraint function g in (6.28) for 3D pendulum: This shows that safety is also guaranteed for mechanical systems on Lie group.

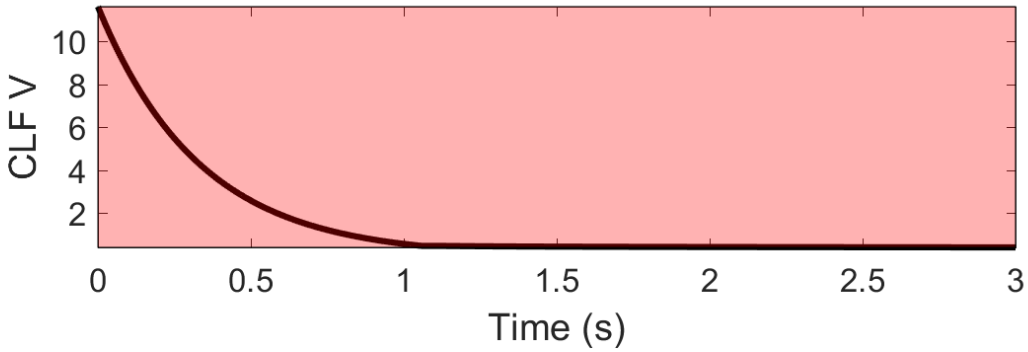


Figure 6.12: The value of geometric Lyapunov function defined in (6.27) where $\alpha = 10, \varepsilon = 1.5$. Note that the reference is a static point staying outside the safety region. In the control process, the reference is always unsafe which makes the value of Lyapunov function stay above a certain threshold.

Chapter 7

Extension of Control Barrier Function to Quadrotor Control Subject to Constraints

Since CBF-CLF-QP control design in the previous chapter only applies to fully-actuated systems, we want to extend this control idea to the underactuated system of a single quadrotor in Chap. 3. Depending on the applications, we could impose various constraints on the quadrotor. Among these constraints, the most important constraint is **the collision avoidance of the quadrotor with its environment**, which is directly related to safety. Also, visual tracking of a quadrotor using onboard camera is of great interest in field tasks such as UAV-UGV cooperation. Considering these scenarios, we include three sections in this chapter. The first two sections illustrate a way of constructing CBF for both planar and 3D quadrotor. Using this new type of augmented CBF, we could still satisfy this type of safety constraints in a similar matter as CBF-CLF-QP contdsrol. The last section is focused on visual tracking of a single quadrotor for a ground object, where we can still utilize CBF to satisfy constraint.

7.1 Augmented Control Barrier Function for a Planar Quadrotor

Consider the planar quadrotor system in Eq. (3.2) first. Assume we are given a smooth reference trajectory $(x_d(t), y_d(t), \theta_d(t))$ for the quadrotor to track, along with a list of potentially time-varying safe sets $\mathcal{C}_{i,t} = \{(x,y) \mid g_i(t,x,y) \geq 0\}$, as determined by the functions $g_i(t,x,y) = (x - x_i(t))^2 + (y - y_i(t))^2 - b_i(t)^2,\ b_i \geq 0, i = 1,2,\cdots,k$. Define the overall safe set in state-space as $\mathcal{C}_t = \cap_{i=1}^{k} \mathcal{C}_{i,t}$, and assume that the interior $\mathcal{C}_t^{\circ}$ is nonempty for any $t > 0$. The control goal then is to design a feedback law for $F, M : \mathbb{R}^2 \times \mathbb{S}^1 \to \mathbb{R}$ such that

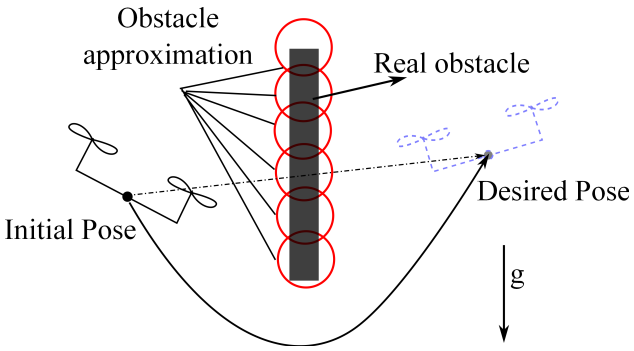


Figure 7.1: Constrained control problem of planar quadrotor. A reference trajectory (straight line) and a list of safe sets are provided (exterior of the red circles). The overall safe region is the intersection of these safe sets, which serves as an approximation of the real obstacle (solid black rectangle). The control goal is to track the reference trajectory while simultaneously strictly enforcing that the state remains in the safe region.

it satisfies the following:

$$\begin{aligned} &(x,y) \to (x_d,y_d) \in \mathcal{C}_t, \text{as } t \to \infty && \text{(Position tracking)} \\ &0 \le F \le F_{\max}, \quad |M| \le M_{\max} && \text{(Input saturation)} \\ &(x(t),y(t)) \in \mathcal{C}_t, \forall t \in [0,\infty) && \text{(Safety constraint)} \end{aligned}$$

As can be seen in Fig. 7.1, the obstacle's shape is approximated as a union of several circles with center $(x_i(t), y_i(t)$, with the exteriors of the circles representing the safe regions $\mathcal{C}_{i,t}$. As long as the reference trajectory belongs to the overall safe region, $\mathcal{C}_t$, asymptotic stability should be attained.

For fully-actuated simple mechanical systems with position-based constraints, a general construction of CBFs can be found in [60]. The process is to expand the safety region, specified in position space, to the whole state-space as follows:

$$h_i(x,y,\dot{x},\dot{y}) := \gamma_i \alpha_i(g_i(x,y)) + \dot{g}_i(x,y),$$

where $\gamma_i > 0$ and α_i is a class $\mathcal{K}$ function. Furthermore, for brevity, we have dropped the explicit dependence on time. Then, as shown in [60, Prop. 1], guaranteeing the state constraint $h_i(x,y,\dot{x},\dot{y}) \ge 0$ implies the position constraint $g_i(x,y) \ge 0$.

However, for (3.2), this method does not work. The major challenge lies in the fact that the derivatives of the normal CBF are not dependent on all the control inputs, or equivalently, $h_i(x,y,\dot{x},\dot{y})$ *lacks a well-defined vector relative degree.* We can select a candidate

$$B_i(\mathbf{x}) = \frac{1}{h_i(x,y,\dot{x},\dot{y})},$$

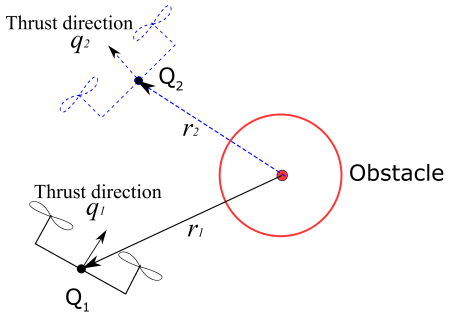


Figure 7.2: Motivation for an augmented CBF: The Quadrotor Q_1 (black solid) is less capable than the Quadrotor Q_2 (blue dashed) to avoid the red circular obstacle, as quadrotors can only apply a positive thrust. A positive thrust on Quadrotor Q_1 causes it to move closer to the obstacle. This can be captured through the fact $q_1 \cdot r_1 < 0 < q_2 \cdot r_2$, where q_i, r_i are the thrust and position orientation vectors respectively.

and check whether

$$\inf_{\mathbf{u}\in U}\{L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} - \frac{\eta}{B(\mathbf{x})}\} \leq 0. \tag{7.1}$$

holds for all $\mathbf{x}$.

For this case, $L_g B$ has the expression $[\bullet, 0]$. Then (7.1) depends only on the thrust part of input $\mathbf{u} = [F, M]^T$, which can only be nonnegative. Thus, for some cases, there may not exist a feasible $\mathbf{u} \in U$ such that (7.1) is satisfied.

The key idea to tackle this difficulty is to augment the original position region function g_i by adding another term that is dependent on the orientation θ. This idea comes from the fact that *the capability of a quadrotor to move away from an obstacle* varies as its orientation changes. For example, as shown in Fig. 7.2, the red circle represents an obstacle. It is obvious that Quadrotor Q_2 (blue dashed) is more capable of avoiding the obstacle than Quadrotor Q_1 (black solid), since the quadrotor's thrust direction is fixed and since the thrust magnitude can only be positive. Hence intuitively, the CBF B can work for the former case but probably would fail for the latter, since positive thrust would bring the quadrotor closer to the obstacle.

To address this, our CBF-based controller should be able to adjust the orientation of the quadrotor, and thus the moment should be included in the derivatives of the CBF. Based on this argument, we propose a detailed construction of an *augmented CBF* for (3.2). We begin by considering the safety region defined by $\mathcal{D} = \{(x,y) : (x-x_o)^2 + (y-y_o)^2 \geq b^2\}$. First, we reduce the size of the original safe region by a factor β, resulting in,

$$g(x,y) = (x-x_o)^2 + (y-y_o)^2 - \beta b^2,\ \beta > 1,$$

where x_o, y_o, b are smooth time-varying functions. We then augment the value of b based on the current orientation as following:

$$\hat{g}(x,y,\theta) := g(x,y) - \sigma(s),$$

where $s := \sin\theta(x-x_o) + \cos\theta(y-y_o)$ and $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth function whose properties will be determined later.

***Remark* 7.1.** *For convenience, we define the direction of thrust F as $q = (\sin\theta, \cos\theta)$ and the distance vector from the obstacle center to the current location of the quadrotor as $r = (x-x_o, y-y_o)$. Then the argument s in $\sigma(s)$ equals $r \cdot q$, as shown in Fig. 7.2, and can be treated as a* signed measure of the quadrotor's ability to escape from the obstacle from the current pose.

Next, following the same construction procedure in [60], we can expand the safety set to the whole state-space:

$$\hat{h} := \gamma\alpha(\hat{g}) + \dot{\hat{g}} = \gamma\alpha(\hat{g}) + \dot{g}(x,y) - \sigma'(s)(p\dot{\theta} + v),$$

where $p := \cos\theta(x-x_o) - \sin\theta(y-y_o) = r \times q$ and $v := \sin\theta(\dot{x}-\dot{x}_o) + \cos\theta(\dot{y}-\dot{y}_o)$ and $\alpha : \mathbb{R} \to \mathbb{R}$ is a class $\mathcal{K}$ function.

Note that enforcing $\hat{h} \geq 0$ will guarantee $\hat{g} \geq 0$, as proved in [60]. To simplify the computation of $\dot{\hat{h}}$, first express the derivative of p, v as:

$$\dot{p} = -s\dot{\theta} + w, \quad \dot{v} = w\dot{\theta} + \sin\theta(\ddot{x}-\ddot{x}_o) + \cos\theta(\ddot{y}-\ddot{y}_o), \tag{7.2}$$

with $w := \cos\theta(\dot{x}-\dot{x}_o) - \sin\theta(\dot{y}-\dot{y}_o)$.

Substituting (7.2) into $\dot{\hat{h}}$ yields

$$\begin{aligned}\dot{\hat{h}} = {} & \gamma\alpha'(\hat{g}) \cdot \dot{\hat{g}} - \sigma''(s)(p\dot{\theta} + v)^2 - \sigma'(s)\dot{\theta}(2w - s\dot{\theta}) \\ & + \sigma'(s)(\sin\theta\ddot{x}_o + \cos\theta\ddot{y}_o) \\ & - p\sigma'(s)\ddot{\theta} - \sigma'(s)(\sin\theta\ddot{x} + \cos\theta\ddot{y}) + \ddot{g}(x,y).\end{aligned}$$

Plugging the system dynamics (3.2) into $\dot{\hat{h}}$ yields

$$\dot{\hat{h}} = \frac{p\sigma'(s)}{J}M + \frac{1}{m}(2s - \sigma'(s))F + \Gamma.$$

where F, M are the thrust and moment part of u and the term Γ is independent of u.

A candidate augmented CBF $\hat{B} = 1/\hat{h}^a$ with $a > 0$ is selected and we have the following lemma:

Lemma 7.1. (Safety guarantee of augmented CBF): *Suppose the scalar function $\sigma(s)$ satisfies the following*

properties:

$$\begin{aligned}
&\sigma'(s) < 0,\ \forall s \in \mathbb{R} && \textit{(Strictly decreasing)}\\
&|\sigma(s)| \leq (\beta-1)b^2,\ \forall s \in \mathbb{R} && \textit{(Global boundedness)}\\
&2s - \sigma'(s) > 0, \quad \forall s \in (-\sqrt{\beta}b, 0), && \textit{(Bounded derivative)}
\end{aligned}$$

Then, the candidate function $\hat{B}$ is an almost global CBF for (3.2)*. Thus, the safe set $\mathcal{D} = \{(x,y)|\ g(x,y) \geq 0\}$ will be forward invariant for* (3.2)*, provided that $\hat{h}\big|_{t=0} \geq 0$ and $F_{\max}, M_{\max}$ are large enough.*

***Remark* 7.2.** *The role of the scalar function $\sigma(s)$ is to adjust the radius b using the value of s. A candidate function that satisfies the conditions of Lemma 7.4 is the sigmoid function shown below,*

$$\sigma_1(s) = k_1 \frac{\exp(-k_2 s + k_3) - 1}{\exp(-k_2 s + k_3) + 1}, \tag{7.3}$$

where the constants $k_1, k_2, k_3, a_1, a_2, a_3$ are positive parameters chosen to satisfy the requirements of Lemma 7.4.

The CBF construction of a planar quadrotor is helpful and inspiring for us to get an augmented CBF for the 3D quadrotor in the next section.

7.1.1 Augmented Control Barrier Function to a 3D Quadrotor

The quadrotor UAV is an underactuated system with 6 degrees of freedom and only 4 control inputs. Typically the attitude is controlled by the moment inputs while the position is controlled through adjusting the attitude. Thus, the controller has to consider both orientation and position at the same time to avoid an obstacle (see Fig. 7.3.) Since the thrust can only be positive, larger thrust can help Quadrotor Q_2 escape the obstacle while it would deteriorate the situation for Quadrotor Q_1. Hence we need a larger safety margin for Quadrotor Q_1 so that it can adjust its orientation before colliding with the obstacle. To capture this, we augment the definition of $g_i(t,x)$ to depend on the orientation of the quadrotor as well to construct the augmented safety function $\hat{g}_i(t,x,R)$. The intuition is to actively adjust the size of the safe set based on the quadrotor's orientation and it's ability to prevent exiting the safe set. In the following, we present a CBF construction method for a 3D quadrotor system.

We will illustrate the construction of the augmented CBF for a 3D quadrotor through a simple example. Given a safety region for position, $\mathcal{B} = \{x \in \mathbb{R}^3 : g(x) \geq 0\}$, determined by an implicitly time-varying smooth configuration safety function $g(x)$ below(we've dropped the explicit dependence of time on x, x_0, b for simplicity):

$$g(x) = ||x - x_o||^2 - b \geq 0, \quad b > 0,$$

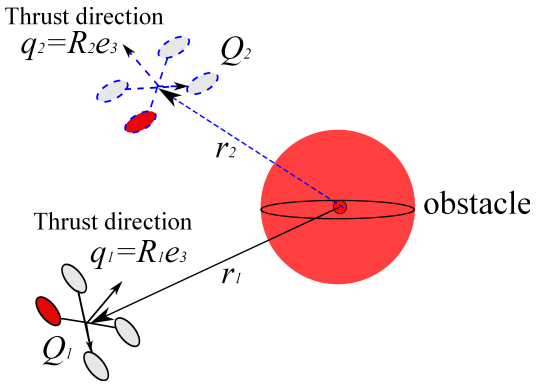


Figure 7.3: The ability of a quadrotor to avoid an obstacle depends both on its position and orientation. In particular, the Quadrotor Q_1 is less capable of avoiding the obstacle than Quadrotor Q_2. This can be captured through $s_1 := r_1 \cdot R_1 e_3 < 0 < s_2 := r_2 \cdot R_2 e_3$.

we expand it to the whole configuration space by defining the augmented configuration safety function,

$$\hat{g}(x,R) := ||x - x_o||^2 - \beta b - \sigma(s), \tag{7.4}$$

where $s = r \cdot q,\ r = x - x_o,\ q = Re_3,\ \beta > 1$, and $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth function satisfying the following conditions:

$$\sigma'(s) < 0, \qquad \text{(Strictly decreasing)} \tag{7.5}$$

$$|\sigma(s)| \leq \sigma_0 < (\beta - 1)b,\ \forall s \in \mathbb{R}, \qquad \text{(Boundedness)} \tag{7.6}$$

$$2s - \sigma'(s) > 0, \forall s \in (-\sqrt{\beta}b, 0), \qquad \text{(Local quadratic)} \tag{7.7}$$

***Remark* 7.3.** *Note that s is the inner product of the direction of the quadrotor thrust and the distance vector between the quadrotor and the center of the unsafe set $\mathbb{R}^3 \backslash \mathcal{B}$. Intuitively, it reflects the quadrotor's ability to avoid a certain obstacle as shown in Fig. 7.3.*

Lemma 7.2. *If the system state always stays within $\hat{\mathcal{B}} = \{(x,R) \in SE(3) : \hat{g} \geq 0\}$, then the position state x would always remain within $\mathcal{B}$.*

Proof. It is obvious that whenever $(x,R) \in \hat{\mathcal{B}}$, $||x - x_o|| \geq \beta b - \sigma(s) \geq \beta b - \sigma_0 > \beta b - (\beta - 1)b = b$, based on (7.6). This implies that $x \in \mathcal{B}$, which guarantees safety in the position space. □

The augmented safety function $\hat{g}$ above is a function of the configuration space and does not have relative

degree 1. To address this we construct a new safety constraint function $\hat{h} : TSE(3) \to \mathbb{R}$ as:

$$\begin{aligned}\hat{h} &:= \gamma\alpha(\hat{g}) + \dot{\hat{g}} \\ &= \gamma\alpha(\hat{g}) + 2(\dot{x} - \dot{x}_o) \cdot (x - x_o) - \beta\dot{b} - \sigma'(s)\dot{s},\end{aligned} \tag{7.8}$$

where $\dot{s} = (\dot{x} - \dot{x}_o) \cdot Re_3 + (x - x_o) \cdot R(\Omega \times e_3)$. We can also define a new safety region in the state space as

$$\mathcal{C} = \{(x, R, \dot{x}, \Omega) \in TSE(3) : \hat{h}(x, R, \dot{x}, \Omega) \geq 0\}.$$

Lemma 7.3. *If the safety region $\mathcal{C}$ is forward invariant, then the safety region $\mathcal{B}$ is also forward invariant.*

Proof. The proof follows from [60, Prop. 1] and Lemma 7.2. Suppose $(x(0), R(0), \dot{x}(0), \Omega(0)) \in \mathcal{C}$, then by forward invariance of $\mathcal{C}$, $(x(t), R(t), \dot{x}(t), \Omega(t)) \in \mathcal{C}, \forall t \geq 0$, i.e, $\hat{h}(x(t), R(t), \dot{x}(t), \Omega(t)) = \gamma\alpha(\hat{g}(x(t), R(t))) + \dot{\hat{g}}(x(t), R(t), \dot{x}(t), \Omega(t)) \geq 0$.

Then we could proceed through contradiction. Assume the configuration variable at time $t_2 > 0$ lies outside the region $\hat{\mathcal{B}}$. Then since the function $\hat{g}(x, R)$ is smooth, by intermediate value theorem, there exists $0 < t_1 < t_2$ such that $\hat{g}(x(t_1), R(t_1)) = 0$, $\dot{\hat{g}}(t_1) < 0$, or equivalently, the state would escape $\mathcal{B}$ at t_1. However, due to the above inequality, a contradiction arises since $\dot{\hat{g}}(t_1) \geq -\alpha(\hat{g}(x(t_1), R(t_1))) = 0$. Thus the assumption is not true, which implies that the region $\hat{\mathcal{B}}$ is also forward invariant. Applying Lemma 7.2 yields that the region $\mathcal{B}$ is forward invariant. □

Then select a candidate CBF $\hat{B} := 1/\hat{h}^a$ with $a > 0$. We have the following lemma regarding the safety:

Lemma 7.4. *If the scalar function σ satisfies the conditions* (7.5)*,* (7.6) *and* (7.7)*. Then the candidate function $\hat{B}$ is an almost global CBF for* (3.2)*. Moreover, we can guarantee safety for the trajectory of* (3.2)*, provided that $\hat{h}(0) \geq 0$ and the thresholds $F_{\max}, M_{i,\max}$ are sufficiently large.*

***Remark* 7.4.** *Note that in the proof, we have not considered input saturation. In particular, if the input constraints are too stringent, its possible for the QP to become infeasible. If we assume the relative velocity of the obstacle entering the sensing range is bounded, then we can choose the constant γ and the function σ to still guarantee safety. In general, incorporating input saturation into the construction of a Barrier is an open research problem.*

7.1.2 Safety-Critical Control Problem for 3D Quadrotor with Range-Limited Sensing

Consider a 3D quadrotor system shown in Fig. 7.4 with dynamics in Eq. (3.3).

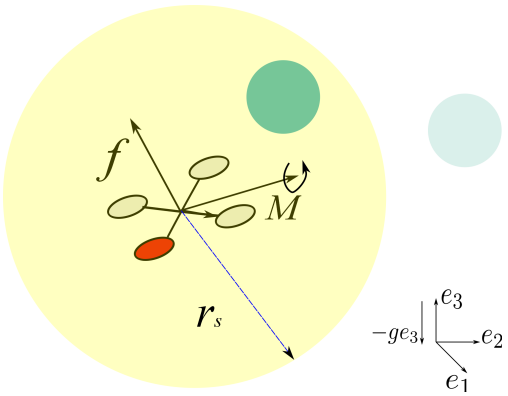


Figure 7.4: Thrust f and moment M produced by a 3D quadrotor, along with a range-limited omnidirectional sensor model with range r_s are shown. Two obstacles are shown (green circles), with the detected obstacle that is within the sensing range highlighted.

The safety-critical control problem for the 3D quadrotor is stated next. First, we assume the following are given:

1. A smooth reference trajectory $(x_d(t), R_d(t))$ for (3.2), with $x_d \in \mathbb{R}^3$ and $R_d \in SO(3)$ representing the desired position and orientation of the quadrotor.

2. A list of time-varying safe regions in the position configuration space, $\mathcal{B}_{i,t} = \{x : g_i(t,x) \geq 0\}$, where x is the position of the quadrotor and $g_i(t,x) = ||x - x_i(t)||^2 - b_i(t),\ b_i(t) > 0, i = 1,2,\cdots,k$, such that $\mathcal{B}_i$ defines the space outside a closed ball centered about $x_i(t) \in \mathbb{R}^3$ of radius $\sqrt{b_i(t)}$, with the overall safe region in position configuration-space denoted as $\mathcal{B}_t = \cap_{i=1}^m \mathcal{B}_{i,t}$ with non-empty interior.

3. A finite limited sensing range r_s exists, such that the obstacle $\mathcal{B}_{i,t}$ is detected whenever $||x - x_i(t)|| \leq r_s$.

***Remark* 7.5.** *Note that in A.1, since the system* (3.2) *is differentially flat [11], we are able to obtain a smooth reference trajectory easily by choosing a set of flat outputs. Also, note that in A.3, a sensor such as lidar, an omnidirectional camera, 3D IR proximity sensor, or even multiple pairs of stereo vision can provide a full* 360° *sensing.*

Constrained Control Problem: The control goal is then to design a feedback law for the inputs f, M for the system (3.2) such that the following constraints are satisfied:

$$
\begin{aligned}
&(x,R) \to (x_d, R_d), \text{ when } x_d \in \mathcal{B}_t, \text{as } t \to \infty && \text{(Stability constraint)} \\
&x(t) \in \mathcal{B}_t, \forall t \in [0,\infty) && \text{(Safety constraint)} \\
&0 \leq F \leq F_{\max},\ \ |M_j| \leq M_{j,\max},\ \ j = 1,2,3 && \text{(Input constraint)}
\end{aligned}
$$

***Remark* 7.6.** *Note that this is a challenging control problem for several reasons:*

1. *We are considering the 3D quadrotor dynamics without typical local Euler angle parametrizations, requiring the control to be directly computed on manifolds.*
2. *We can not assume the desired trajectory is safe, requiring the controller to enforce strict safety constraints even when the desired trajectory violates them.*
3. *Since the safety constraints are in terms of position constraints and the quadrotor can not arbitrarily change it's position (due to underactuation), respecting these position safety constraints is non-trivial.*
4. *The time-varying safety constraints and limited sensing range introduce additional challenges.*
5. *Requiring strict enforcement of actuator limits in addition to all the above challenges makes this a hard control problem.*

Having formulated the safety-critical control problem for 3D quadrotor with range-limited sensing, we will next see how the position-based safety constraints defined by the set $\mathcal{B}_{i,t}$ are extended to all of the configuration through the construction of an augmented CBF.

7.2 Sequential CBF-CLF-QP Control with Limited Sensing Range

Based on the augmented CBF constructed in Sec. 7.1.1, we propose a cascade optimization scheme for the CBF-CLF-QP control design. The underlying idea is inspired by the backstepping method in geometric control [1], which makes a singular perturbation argument to separate the fast orientation dynamics from the slow translational dynamics. Similar to this, the scheme here comprises of two levels: the first level is called *position level QP* and the second level is called *orientation level QP*.

First, we construct an augmented CBF $\hat{B}_i$ for each safety region $\mathcal{B}_{i,t}$ as indicated in Eq.(7.4). The corresponding expanded safe set is $\mathcal{C}_{i,t} = \{(x, \dot{x}, R, \Omega) \in TSE(3) : \hat{h}_i \geq 0\}$. Then assume that the underactuated part is "fully-actuated" with the virtual dynamics:

$$v = \dot{x}, \quad m\dot{v} = f \tag{7.9}$$

Select a quadratic CLF for this virtual system as:

$$\hat{V}_x = \frac{1}{2} m e_v \cdot e_v + \frac{1}{2} k_1 e_x \cdot e_x + \varepsilon_1 e_x \cdot e_v \tag{7.10}$$

where $e_x = x - x_d, e_v = v - v_d$, and the value of $k_1, \varepsilon_1 > 0$ are chosen specifically to make $\hat{V}$ quadratic.

Then, we are able to compute a virtual force based on $\hat{V}$ through the following QP based on (7.10):

Position Level QP (virtual force computation)

$$f^* = \underset{f\in\mathbb{R}}{\operatorname{argmin}} \frac{1}{2} f^T Q f$$
$$\text{s. t. } \dot{\hat{V}}_x(f) + \eta_1 \hat{V}_x \leq 0,$$

---

where $\eta_1 > 0$. The solution f^* is computed as a *virtual force* and passed onto the lower orientation level of optimization as an input parameter,

Then we decompose this input f^* into its norm $F_c = |f^*|$ and direction $b_{3c} = f^*/F_c$. Then compute a desired rotation matrix as below: $R_c = [b_{1c}, b_{3c} \times b_{1c}, b_{3c}]$ where the unit vector $b_{1c} = -\hat{b}_{3c}^2 \times R_d e_1 / ||\hat{b}_{3c}^2 \times R_d e_1||$. In this way, we can construct a geometric CLF for the orientation part as:

$$\hat{V}_R = \frac{1}{2} e_R^T J e_R + \frac{1}{2} e_\Omega^T K_2 e_\Omega + \varepsilon_2 e_R \cdot e_\Omega \tag{7.11}$$

Then, the orientation level QP is constructed to obtain our actual control inputs F and M:

Orientation Level QP (virtual force tracking and safety guarantee):

$$[F^*, M^*] = \underset{F\in\mathbb{R}, M\in\mathbb{R}^3}{\operatorname{argmin}} \frac{1}{2}\lambda_1 (F - F_c)^2 + \frac{1}{2} M^2 + \frac{1}{2}\lambda_2 \delta^2$$
$$\text{subject to } \dot{\hat{V}}_R(M) + \eta_2 \hat{V}_R \leq \delta$$
$$\dot{\hat{B}}_j(F, M) \leq \frac{\gamma_j}{\hat{B}_j},\ j \in I_s(t)$$

---

where $I_s(t)$ is the collection of indices corresponding to obstacles that are detected by the quadrotor's onboard range-limited sensor, $\dot{\hat{B}}_j(F,M)$ is as computed in (10.7) and $\lambda_1, \lambda_2, \eta_2, \gamma_j$ are all positive gain parameters.

***Remark* 7.7.** *Based on analysis in [52], the stability of a geometric controller can be guaranteed roughly under the condition of a fairly large proportional gain in the orientation control. Since the convergence rate of CLF can be related to the proportional gain, we here entail that $\eta_2 \gg \eta_1$. This can be shown in the simulation parameters.*

Regarding the safety property of this controller, we have the following proposition:

***Proposition* 7.5.** *(Safety Guarantee of Sequential CBF-CLF-QP Controller)*

If the following assumptions are satisfied:

1. *Suppose at any time $t \in [0,\infty)$, the sequential QP controller always admits a solution F,M;*

2. *The conditions of Lemma 7.4 are satisfied;*

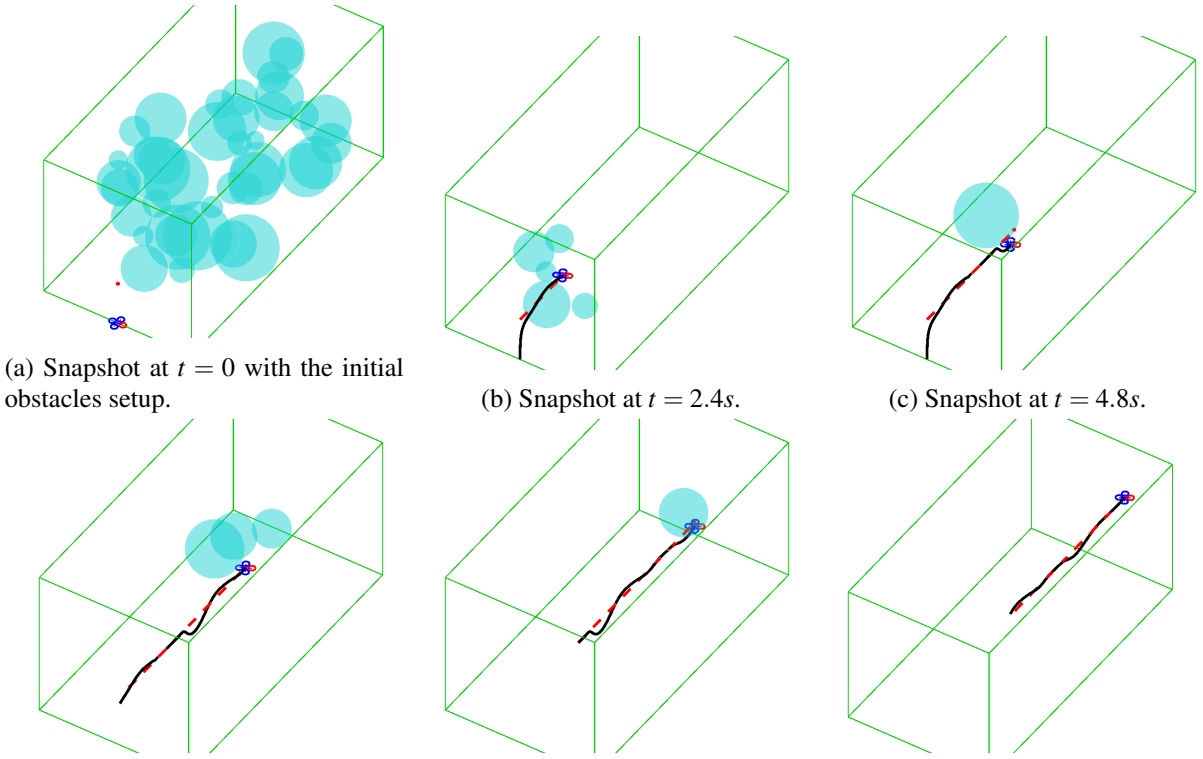
3. *There exists sequence $\{t_n,\ n \in \mathbb{N}\}$ such that I_s is constant between $[t_n,t_{n+1}])$ for each $n \in \mathbb{N}$, and $\hat{B}_i(t_n) \geq 0$ for all $i \in I_s$;*

then the system trajectory would always remain within $\mathcal{C}_t$.

Proof. First observe that only those obstacles within the sensing range will affect the trajectory's safety at each time $t > 0$. Fix $n \in \mathbb{N}$, consider the time interval $[t_n,t_{n+1}]$. By applying Lemma 7.4 to each function $\hat{B}_i$, the condition of every CBF can be enforced, and thus the set $\mathcal{C}_{i,t}$ is forward invariant for each $i \in I_s$. Thus, the system trajectory would remain within every region $\mathcal{C}_{i,t}$, and the position trajectory would remain within $\mathcal{B}_{i,t}$ by Lemma 7.3 for each $i \in I_s$. This indicates the system position would remain within $\mathcal{B}_t$ for $[t_n,t_{n+1})$ by the previous argument. Hence, over the entire interval $[0,\infty)$, the system trajectory would always remain within $\mathcal{C}_t$ and thus stay safe. □

7.2.1 Simulation Results

To numerically validate the performance of the proposed sequential QP controller, we have created a simulation framework in Matlab 2015b. The simulator utilizes ode15s as the solver since the problem is intrinsically stiff due to new obstacles being sensed. Each level of QP is solved by "interior point method" with convergence tolerances 10^{-4} and 10^{-6} respectively. The quadrotor model is an Asctec Hummingbird, with a two meter sensing range. The mass parameters of this quadrotor are provided by the company as $m = 0.52kg, J = \text{diag}[2.32, 2.32, 7.60] \times 10^{-3} kg \cdot m^2$. The function σ is given as $\sigma(s) = -a_1 \arctan(a_2 s + a_3)$. As mentioned in Section. 7.1, we will choose the positions x_d and yaw angle $\phi_d$as the flat outputs to generate the reference $(x_d(t), R_d(t))$. Given the above reference trajectory, we then solve the control problem as described in Section 7.1 of tracking the reference asymptotically when ever its safe while ensuring the state is in the safe set by avoiding obstacles and enforcing input constraints. We use the position and orientation level QP controllers described in Section 7.2, however we incorporate the input bounds, $0 \leq f \leq 50N$, and $|M| \leq 0.038Nm$. With this controller, we next present the results of two simulation experiments, where the quadrotor has to detect and avoid randomly generated static and dynamic obstacles respectively.



(a) Snapshot at $t = 0$ with the initial obstacles setup.

(b) Snapshot at $t = 2.4s$.

(c) Snapshot at $t = 4.8s$.

(d) Snapshot at $t = 7.2$.

(e) Snapshot at $t = 9.6s$.

(f) Snapshot at $t = 12s$.

Figure 7.5: Numerical validation of 3D quadrotor flight through a dense cluttered static obstacle field. Snapshots of the simulation for 15 seconds, where we show all the obstacles in the first snapshot and only plot out the obstacles that are detected by the quadrotor in subsequent snapshots. The black solid line is the actual trajectory and the red dashed line is the reference trajectory. The quadrotor strictly guarantees non-collision with the obstacles and tracks the desired reference trajectory when feasible, all without the need for any re-planning. Simulation video: `https://youtu.be/LHNesE603us`.

7.2.2 Trajectory Tracking with Randomly Generated Static Obstacles

In the first experiment, the reference of the flat outputs is given as a straight line $x_d(t) = [0, 0.75t, 1.5]^T$, $\phi_d(t) = 0$. This straight line will pass through a box $[-2.5, 2.5] \times [0, 10] \times [0, 5]$ where we put in randomly generated sphere-shaped obstacles. These obstacles are generated offline from uniform distribution. The positions of these obstacles are uniformly sampled within the box while the corresponding radii are taken from an uniform distribution over the interval $[0.25, 1]$. We show the simulation results graphically in Fig. 7.5. As shown, initially we plot out 40 obstacles to show the general setup in Fig. 7.5a. The rest of figures plot out the reference and actual trajectory of the quadrotor's CoM in red dashed and black solid line separately. As can be seen, the CBF-CLF-QP controller will help avoid the obstacles while tracking the reference trajectory when it's safe.

7.2.3 Trajectory Tracking with Randomly Generated Dynamic Obstacles

In the second experiment, the corresponding flat output is $x_d(t) = [-\sin(0.5t), 5+\cos(0.5t), 1.5]^T$, $\phi_d(t) = \frac{\pi}{6}$ which is a planar circle parallel to the XY plane. During the simulation process, we randomly generate 12 obstacles near the wall of a box and shoot them at the quadrotor with speed $1.5m/s$ every 2 seconds. The simulation results are shown in Fig. 7.6 and Fig.7.7. Fig. 7.6 shows the error plot in position and orientation with respect to reference. Since the obstacle will be constantly shot at the quadrotor, asymptotic stability for the trajectory tracking is only possible when there is no obstacle along the trajectory. The controller attempts to track the reference in a stable manner when the reference trajectory is safe, while the controller relaxes tracking the reference when it is no longer safe. This leads to the error to fluctuate from converging to zero to going to non-zero values. Fig. 7.7 also shows snapshots of the reference and actual trajectory, using the same line pattern as in Fig. 7.5. In the static case, only a few obstacles are sensed, however for the dynamic case, a lot more obstacles are sensed since the obstacles are shot towards the quadrotor. Due to the presence of numerous dynamic obstacles, the quadrotor has to constantly oscillate around the reference to avoid collision. Check the video link in the caption for a better illustration.

7.2.4 Discussion

In addition to the promising simulation results presented in Figs. 7.5,7.7, we also want to add in a few comments regarding advantages and drawbacks of our controller. In particular, although our controller can adapt to the changing environment rapidly to avoid obstacles while tracking a reference trajectory, there is a possibility that it can get trapped in nonconvex regions as it's only using local information. However, this can be easily addressed by combining this controller with a long range planner, thereby enabling safe and efficient flight. As mentioned in Remark 7.4, there is a possibility that the QP becomes infeasible for extreme situations with stringent torque saturation constraints. However, bounding the relative velocity of the obstacle on entering the sensing field and carefully selecting the constant γ and the function σ, in practice we can still retain the guarantees of safety. Finally, the presented σ function was chosen arbitrary to satisfy certain properties. We believe that finding a systematic way to search for a σ function would be an interesting future direction.

7.3 Visual Tracking for a Single Quadrotor using CBF

Now consider a quadrotor with a camera facing in the downwards direction shown in Fig. 7.8. For a moving sphered-shape object shown in Fig.7.9, we want to let the quadrotor track this object in a way so that the object is always within sight.

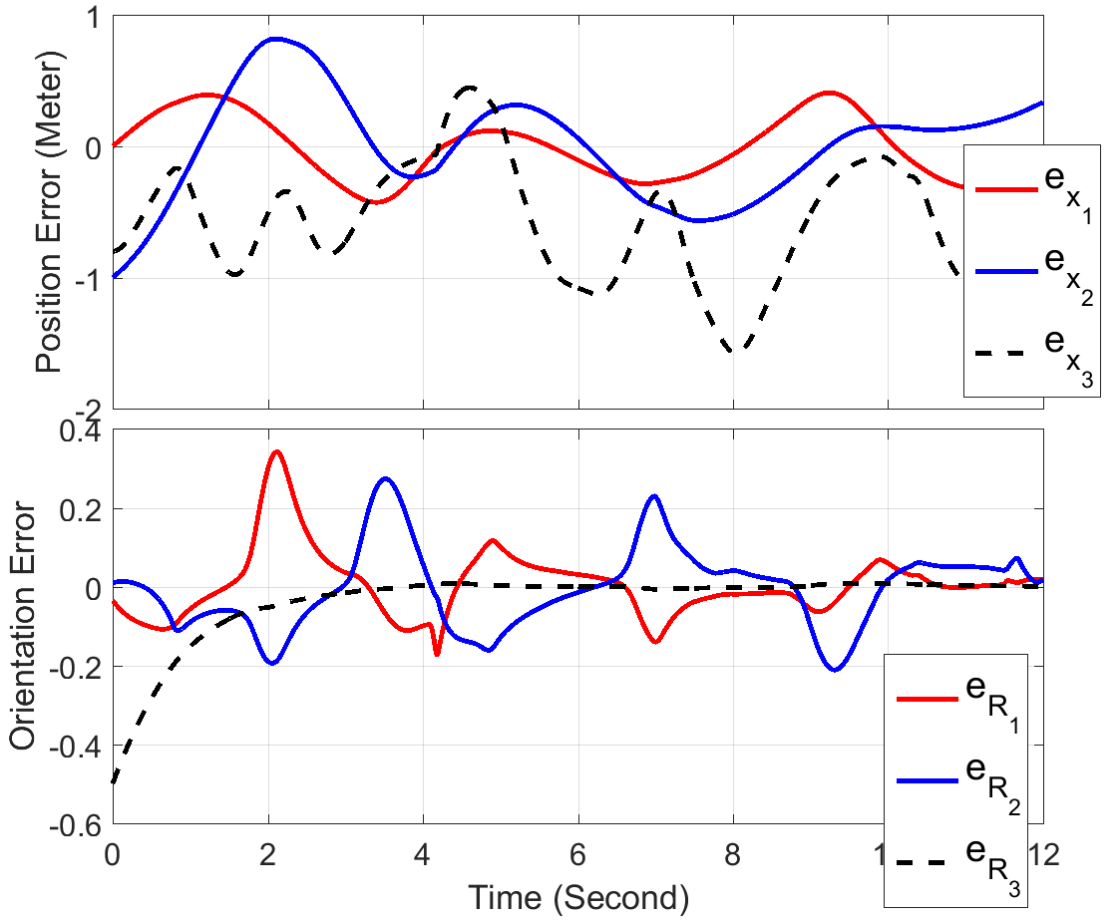


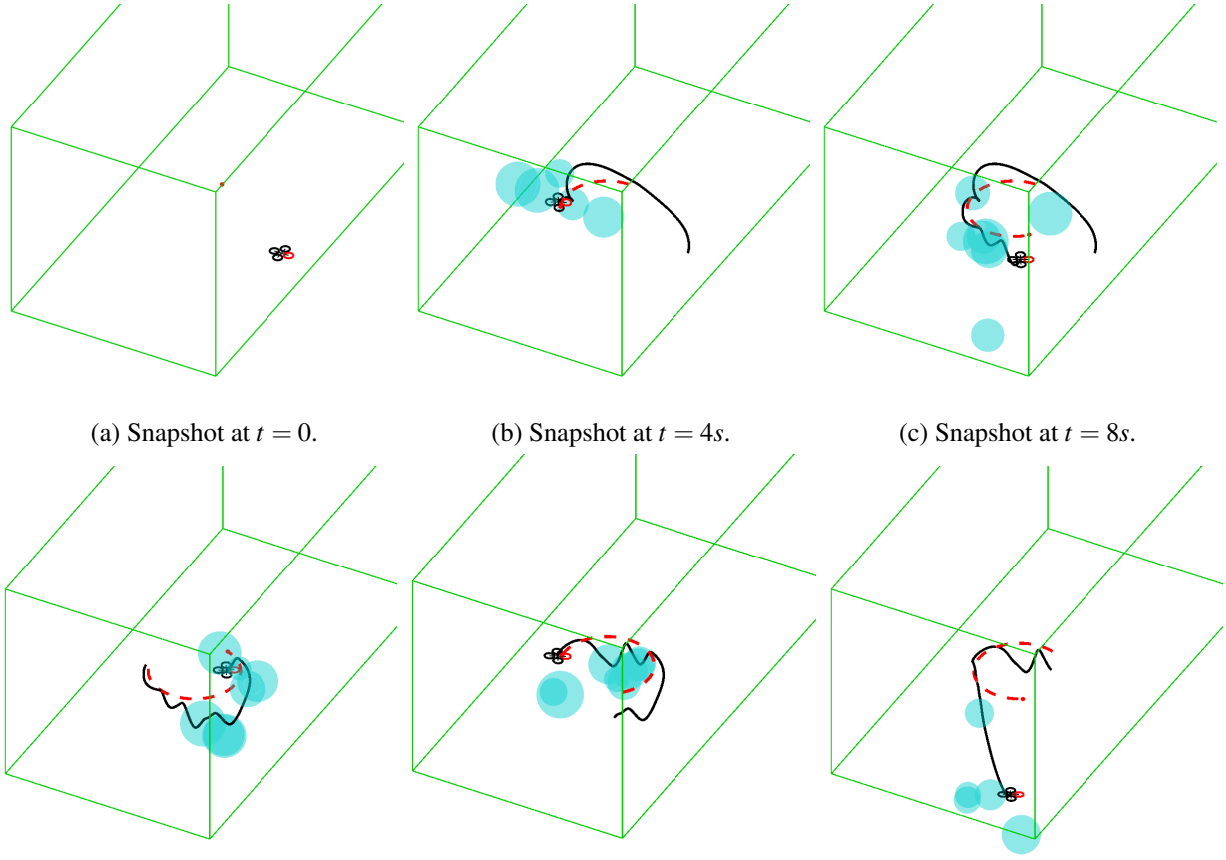
Figure 7.6: Position error e_x and orientation error e_R plot. The controller drives the error to zero when the reference trajectory is safe. When the reference trajectory violates the safety constraint (by passing through an obstacle), the controller automatically relaxes trajectory tracking to strictly enforce safety constraints. All this is done without the need for re-planning.

Denote the position of the geometric center of this object as x_o, the onboard camera lens x_l in the world frame, and the set of points in the object as $\mathcal{O} = \{x \in \mathbb{R}^3 : ||x - x_o||_2^2 \leq R^2\}$. Then the constraints to be satisfied could be expressed as:

$$\arccos(R_Q e_3 \cdot \frac{(x_l - x_o)}{||x_l - x_o||_2} \leq \theta_0, \quad \forall x \in \mathcal{O} \tag{7.12}$$

$$(x_l - x_o) \cdot R_Q e_3 \geq 0, \tag{7.13}$$

where we call the former (7.12) cone constraint, and the latter (7.13) direction constraint.



(a) Snapshot at $t = 0$.

(b) Snapshot at $t = 4s$.

(c) Snapshot at $t = 8s$.

(d) Snapshot at $t = 12s$.

(e) Snapshot at $t = 16s$.

(f) Snapshot at $t = 20s$.

Figure 7.7: Numerical validation of 3D quadrotor flight through a dense cluttered dynamic time-varying obstacle field. Snapshots of the simulation process for 12 seconds are shown. The red dashed line is the reference trajectory while the black solid line is the actual trajectory of the quadrotor's CoM. The obstacles are dynamic and move with randomly generated velocities. Simulation video: `https://youtu.be/LHNesE603us`.

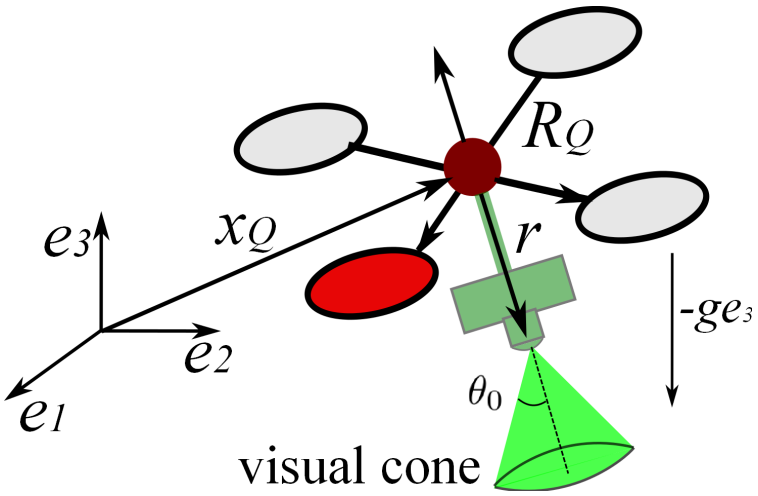


Figure 7.8: Diagram of a single quadrotor with camera facing downwards, where the cone represents the camera angle of view.

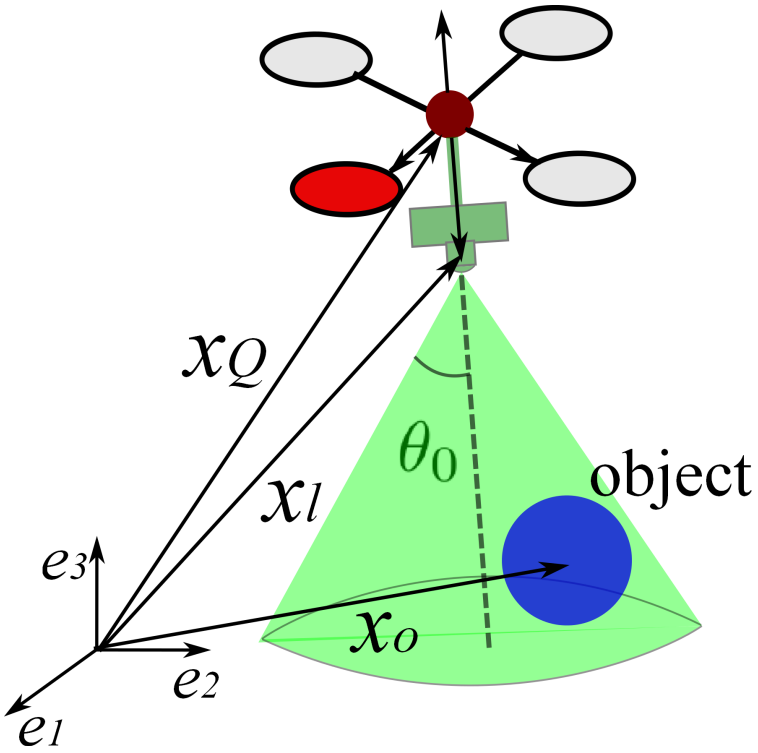


Figure 7.9: Visual tracking problem of an object for a single quadrotor with an onboard camera.

7.3.1 Enforcing Cone Constraint using CBF

Let's ignore the size of this object, and treat it is as a point mass. Then the set $\mathcal{O}$ is just a singleton $\{x_o\}$, and the cone constraint in Eq. (7.12) could be expressed equivalently as:

$$\frac{(x_l - x_o)}{||x_l - x_o||_2} \cdot R_Q e_3 \geq \cos\theta_0 = \beta \tag{7.14}$$

Taking square of both sides of this inequality yields

$$\begin{aligned} [(x_l - x_o)^T R_Q e_3]^2 &\geq \beta^2 (x_l - x_o) \cdot (x_l - x_o) \\ (x_l - x_o)^T (R_Q e_3 e_3^T R_Q^T - \beta^2 I)(x_l - x_o) &\geq 0 \end{aligned} \tag{7.15}$$

Based on Fig. 7.8, r is the distance vector from the center of mass of the quadrotor to the len's center in the quadrotor body-fixed frame and $x_l = x_Q + R_Q r$. Then we could further simplify (7.14) as belows,

$$\begin{aligned} g_o &= (x_Q + R_Q r - x_o)^T \underbrace{(R_Q e_3 e_3^T R_Q^T - \beta^2 I)}_{J(R_Q)} (x_l - x_o) \geq 0 \\ g_o &= (x_Q - x_o)^T J(R_Q)(x_Q - x_o) + (r \cdot e_3)^2 - \beta^2 (r \cdot r) \\ &\quad + 2(x_Q - x_o)^T \underbrace{J(R_Q) R_Q}_{F(R_Q)} r \geq 0 \end{aligned}$$

where $F(R_Q) = J(R_Q)R_Q = R_Q e_3 e_3^T - \beta^2 R_Q = R_Q(e_3 e_3^T - \beta^2 I)$.

Hence, from the cone constraint, we could get a nonlinear constraint in terms of the state variable x_Q and R_Q. Following the same idea in [60], we could extend the constraint to the whole state space $TSO(3)$ as $h_o = \alpha g_o + \dot{g}_o$ with $\alpha > 0$. Then by choosing a candidate ECBF as $B_o = h_o$, we could select control input which satisfies the condition:

$$\dot{h}_o + \gamma h_o = \ddot{g}_o + (\alpha + \gamma)\dot{g}_o + \gamma\alpha g_o \geq 0, \tag{7.16}$$

To simplify the overall derivation of $\ddot{g}_o$, let's denote $S = e_3 e_3^T - \beta^2 I$, $x_r = R_Q^T(x_Q - x_o)$ and $b = (r \cdot e_3)^2 - \beta^2(r \cdot r)$. The original constraint function can be expressed as:

$$g_o = x_r^T S x_r + 2x_r^T S r + b \geq 0,$$

while by Chain rule of vector functions, the expanded constraint function is could be given as:

$$h_o = \alpha g_o + \dot{g}_o = \alpha g_o + 2(x_r + r)^T S \dot{x}_r.$$

Based on this, the CBF condition in 7.14 is equivalent to the following:

$$\begin{aligned} 2(x_r+r)^T S\ddot{x}_r + 2\dot{x}_r^T S\dot{x}_r + 2(\alpha+\gamma)(x_r+r)^T S\dot{x}_r \\ + \gamma\alpha g_o \geq 0. \end{aligned} \tag{7.17}$$

Similarly, we could get the higher order derivatives of x_r by applying Chain rule twice:

$$\begin{aligned} \dot{x}_r &= \dot{R}_Q^T(x_Q - x_o) + R_Q^T(\dot{x}_Q - \dot{x}_o), \\ \ddot{x}_r &= \ddot{R}_Q^T(x_Q - x_o) + 2\dot{R}_Q^T(\dot{x}_Q - \dot{x}_o) + R_Q^T(\ddot{x}_Q - \ddot{x}_o), \end{aligned}$$

Plugging these expressions back into Eq. (7.17) yields:

$$\underbrace{2m_Q^{-1}(x_r+r)^T Se_3}_{\psi_F} \cdot F + \underbrace{2(x_r+r)^T S\hat{x}_r J_Q^{-1}}_{\psi_M} M + \psi_0 \geq 0, \tag{7.18}$$

where ψ_0 is the sum of all terms that are independent of the control inputs F and M shown below:

$$\begin{aligned} \psi_0 = {} & 4(x_r+r)^T S\dot{R}_Q^T(\dot{x}_Q - \dot{x}_o) + \psi_M[(J_Q\Omega)\times\Omega] \\ & - 2(x_r+r)^T SR_Q^T(\ddot{x}_o + ge_3) - 2(x_r+r)^T S(R\hat{\Omega}^2)^T \\ & + 2\dot{x}_r^T S\dot{x}_r + 2(\alpha+\gamma)(x_r+r)^T S\dot{x}_r + \gamma\alpha g_o, \end{aligned}$$

Thus by the condition of ECBF, we are able to make sure that the constraint $g_o \geq 0$ is always satisfied. Regarding its property, we have the following proposition.

***Proposition* 7.6.** (Visual Tracking CBF Candidate)

The CBF candidate B_o in Eq.(10.6) *is indeed an ECBF with respect to the relaxed cone constraint in* (7.15) *for the single quadrotor system* (3.3) *where $F \in \mathbb{R}$, $M \in \mathbb{R}^3$.*

Proof. Denote that the visual tracking zone as $\mathcal{B} = \{(x_Q, R_Q, x_o) \in SE(3)\times\mathbb{R}^3 : (7.15) \text{ is satisfied}\}$. By the previous derivation, the CBF condition to check is given in (7.17) explicitly as:

$$\sup_{F\in\mathbb{R}, M\in\mathbb{R}^3} \{\psi_F F + \psi_M M + \psi_0\} \geq 0,$$

for all $x \in \mathcal{B}$.

Let's consider the case based on the value of ψ_M. When $\psi_M \neq 0$, we could always pick a candidate as

$$F^* = 0,\ M^* = -\frac{\psi_0 \psi_M^T}{||\psi_M||^2} \implies \psi_F \cdot F^* + \psi_M M^* + \psi_0 = 0$$

When $\psi_M = 0$, we could claim that $\psi_F > 0$ for all $x \in \mathcal{B}$. Note that $\psi_M = (x_r + r)^T S\hat{x}_r J_Q^{-1} = 0 \implies (x_r + r)^T S\hat{x}_r = (S^T(x_r + r))^T \hat{x}_r = 0$ because the inertia matrix J_Q is nonsingular. Then taking the transpose yields that $-\hat{x}_r S^T(x_r + r) = 0$, and thus there exists $\alpha \neq 0$ such that $S^T(x_r + r) = S(x_r + r) = \alpha x_r$. Assume that $\psi_F = 0$, then it holds that

$$(x_r + r)^T S e_3 = (x_r + r)^T \begin{bmatrix} -\beta^2 & 0 & 0 \\ 0 & -\beta^2 & 0 \\ 0 & 0 & 1-\beta^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

which further implies that $(x_r + r)_3 = 0$ since $1 - \beta^2 > 0$.

Now plugging this back into the previous inequality yields:

$$(S(x_r + r))_3 = (\alpha x_r)_3 \implies (1-\beta^2)(x_r + r)_3 = \alpha (x_r)_3 = 0$$

So we have both the fact that $(x_r + r)_3 = (x_r)_3 = 0$ which is impossible since $r = -ke_3$. Thus, combining the derivative condition proved here and [60, Prop. 2] together gives the final conclusion. □

***Remark* 7.8.** *This proposition implies that the relaxed cone constraint can be satisfied using the visual CBF proposed here. Yet two problems remain: whether the actual visual constraint in* (7.14) *could be enforced using this CBF; whether the CBF condition is feasible due to the constraint that* $F > 0$.

7.3.2 Sequential QP Controller for Visual Tracking Task

In the presence of a moving object to track, assume that we have provided a reference for the quadrotor state as $(x_d, v_d, R_d, \Omega_d)$ from differential flatness. Based on this visual tracking ECBF h_o, we still employ the structure of sequential QP in previous sections:

Position Level QP (virtual force computation)

$$\begin{aligned} f^* = \underset{f \in \mathbb{R}}{\operatorname{argmin}} \ & \frac{1}{2} f^T Q f \\ & \text{s. t. } \dot{\hat{V}}_x(f) + \eta_1 \hat{V}_x \leq 0, \end{aligned}$$

---

where $\eta_1 > 0$, and $\hat{V}_x$ is called virtual Control Lyapunov Function accordingly. The solution f^* is computed as a *virtual force* and passed onto the lower orientation level of optimization as an input parameter,

Then we decompose this input f^* into its norm $F_c = |f^*|$ and direction $b_{3c} = f^*/F_c$. Then compute a

desired rotation matrix as below: $R_c = [b_{1c}, b_{3c} \times b_{1c}, b_{3c}]$ where the unit vector $b_{1c} = -\hat{b}_{3c}^2 \times R_d e_1 / ||\hat{b}_{3c}^2 \times R_d e_1||$. In this way, we can construct a geometric CLF for the orientation part as:

$$\hat{V}_R = \frac{1}{2} e_R^T J e_R + \frac{1}{2} e_\Omega^T K_2 e_\Omega + \varepsilon_2 e_R \cdot e_\Omega \tag{7.19}$$

Then, the orientation level QP is constructed to obtain our actual control inputs F and M:

Orientation Level QP (virtual force tracking and safety guarantee):

$$\begin{aligned} [F^*, M^*] = \underset{F \in \mathbb{R}, M \in \mathbb{R}^3}{\operatorname{argmin}} \; & \frac{1}{2}\lambda_1 (F - F_c)^2 + \frac{1}{2} M^2 + \frac{1}{2}\lambda_2 \delta^2 \\ \text{subject to } \; & \dot{\hat{V}}_R(M) + \eta_2 \hat{V}_R \leq \delta \\ & -\psi_F F - \psi_M M \leq \psi_0 \end{aligned}$$

where $\hat{V}_R$ is called geometric Control Lyapunov Function, designed for orientation.

7.3.3 Visual Tracking Simulation using CBF-CLF-QP Controller

In this simulation, we are assuming that the kinetic information of this object is perfectly known. For this case, the object is moving along a planar circle on the ground as:

$$x_o(t) = [3.2\cos t, -3.2\sin t, 0]^T$$

with a $3.2m$ radius.

The desired reference for the quadrotor is computed based on the following flat ouputs:

$$x_{Qd}(t) = [0, 0, 1.5]^T, \quad \phi_{Qd} = 0$$

with the constraint angle $\theta_0 = \pi/4$.

Note that tracking the original refernce no longer satisfies the visual tracking constraint since $1.5 \cdot \tan(\pi/3) = 2.6 < 3.2$. It is not possible see the geometric center of this object, not to mention the onboard camera. The initial condition of the quadrotor is given as $x_0 = [0, 0, 5]^T$, $R_0 = I_3$, $v_0 = \Omega_0 = [0, 0, 0]^T$.

The corresponding simulation results are plotted in Fig. 7.10 and Fig. 7.11. Fig. 7.10 draws the position errors of the quadrotor with respect to its desired states. In order to satisfy the constraint, the quadrotor has

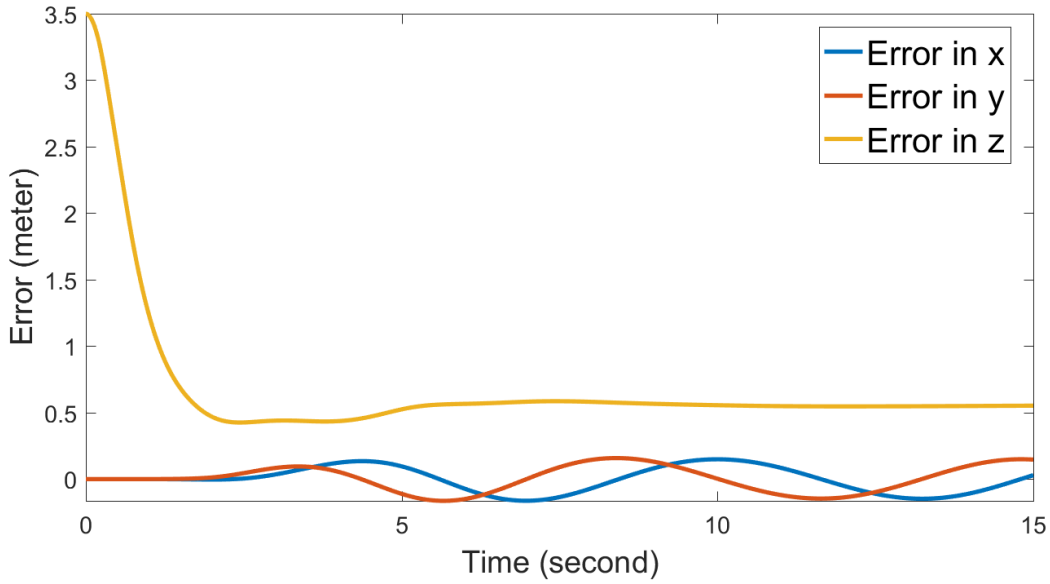


Figure 7.10: Position error of the quadrotor with respect to the desired hovering position. Check the video here: `https://youtu.be/vh7HZqa5FDk`

to stay a bit higher than expected while at the same time keeps oscillating around the vertical axis. Fig. 7.11 shows the change of the original constraint function g_0 defined below:

$$g_0 = \frac{x_l - x_o}{||x_l - x_o||_2} \cdot R_Q e_3 - \cos\theta_0 \tag{7.20}$$

which we could see that the value would remain very closed to zero as time increases.

This chapter includes the construction of various CBFs for the different tasks of a single quadrotor. As tested in numerical simulations with theoretical guarantee, it can be seen that CBF could be applied to satisfy safety constraints for underactuated systems with proper adjustments. Up to this chapter, we have only tested our controller performance in the simulation environment. In the next chapter, we would provide some experimental validation results for some controller given, especially **the geometric linearization techniques in Chap. 4**.

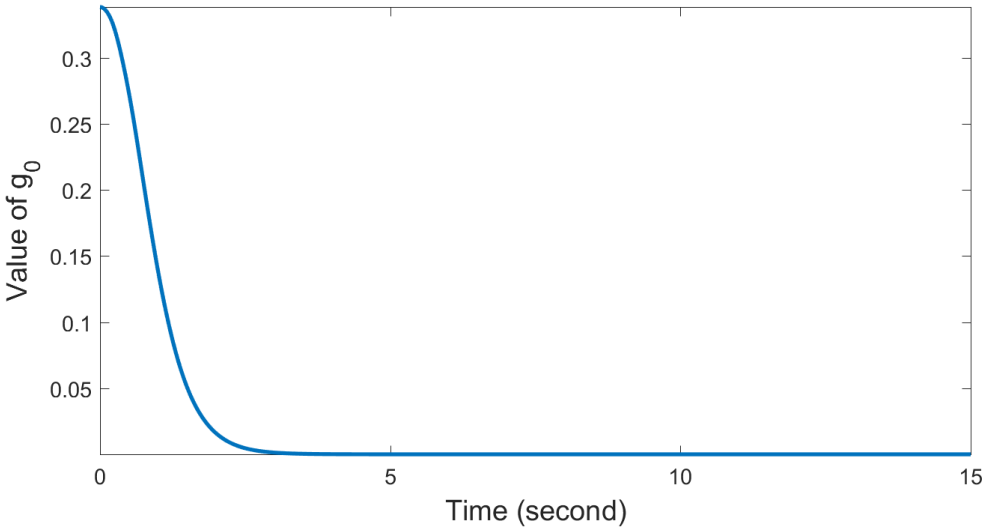


Figure 7.11: Value plot of the original constraint function g_0 defined in (7.20).

Chapter 8

Experimental Results with Payloads

In this chapter, we present several experimental results based on the control techniques for some initial payload transportation. The main technique employed is *geometric linearization* in Chap. 4, with some slight adjustments to fit hardware needs. The first section provides static hovering and tracking experiments of point mass payload carried by a single quadrotor, while the second section provides rigid-body load hovering and tracking experiments. Throughout this chapter, we would utilize Hummingbird in Fig. 3.1 as the experiment platform, and the values of parameters are given in several tables for better reference.

8.1 Experiments with Point-Mass Payload

In this part, we would first introduce a modified version of geometric-linearization based controller for the point mass load. Based on this method, we are able to perform multiple experiments for a single quadrotor to carry a small, cube-shaped payload in Fig. 8.1a.

(a) Experimental setup of point-mass load.

(b) Experimental setup of rigid-body load.

8.1.1 Modified Geometric Linearization Techniques for Point-Mass Payload

This part addresses the implementation details of our control method for the system in 3.2.1. For the same control problem, Chap. 5 has provided a linear controller using geometric linearization with direct force control in simulation. Unlike the models represented in Chap. 3, where we directly control the force based on position feedback, hardware implementations usually entails **the separation between position and orientation control[11, 12].** The onboard processor runs at 1kHz while due to the bandwidth limitation of wireless communication, the ground station can only send command up to 200Hz. Thus the quadrotor will hold a constant moment value for about 0.01 second in the force control. This amount of accumulated effect will change the quadrotor's orientation so aggressively that the ground station won't be able to correct it before failure happens. To deal with the challenges of implementation, we propose the control design with two steps.

Step 1: ignore the quadrotor orientation dynamics and treat its external force as a virtual control vector $F \in \mathbb{R}^3$. In this way, we could repeat the derivation process in Appendix 10.4, and a modified variation dynamics $\dot{s} = A(t)s + B(t)\delta F$. Using linear control design method in Chap. 4 to compute the virtual thrust as $F = F_d + \delta F$.

Step 2: convert the virtual thrust F into a desired orientation and thrust magnitude as below:

$$R_c = \begin{bmatrix} e_{1c} & e_{2c} & e_{3c} \end{bmatrix}, \tag{8.1}$$

where

$$e_{3c} = \frac{F}{||F||_2},\ e_{2c} = \frac{e_{3c} \times (R_d e_1)}{||e_{3c} \times (R_d e_1)||_2},\ e_{1c} = e_{2c} \times e_{3c},$$

Then we will send this desired rotation matrix R_c, reference angular velocity Ω_d to the quadrotor with the geometric controller onboard. Then this controller would convert to the corresponding angular velocity for each motor as specified in [12]. We show the control block diagram in Fig. 8.2 for better illustration.

8.1.2 Hovering and Tracking Experimental Results

Using the previous control algorithm, we conduct both hovering and tracking experiments for the point mass payload with parameters shown in Tab. 8.1. Note that the cable mass is extremely small compared to the quadrotor's and the payload's. Thus the assumption that it is massless is valid. To prevent overshooting in the quadrotor orientation and accord with input saturations, I have utilized a set of high gains for the virtual force in the x, y directions. Also, since we put small penalties on the cable's direction error and angular velocity, cable swing is not highly suppressed, and thus we are using the swing of the cable to realize faster tracking of the payload's position.

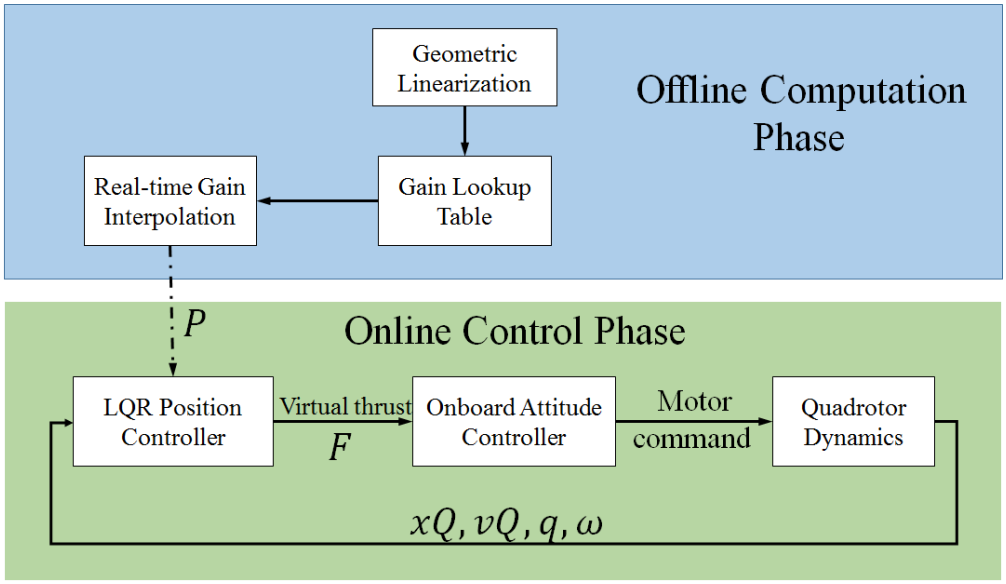


Figure 8.2: Block diagram showing the whole control architecture for both point-mass and rigid-body payloads: a combination of geometric LQR and attitude controllers.

Using the same set of parameters, we could generate the gain matrices for hardware implementations. The corresponding experimental results are shown in Tab. **??** with both hovering and tracking experiments.

Table 8.1: Experimental Parameters for Point-mass Payload Experiment

Mass Parameters			
Quadrotor Mass	0.544 kg	Load Mass	0.105 kg
Cable Mass	0.001 kg	Cable Length	0.80 m
LQR Weight Penalties			
Position Penalties	$diag(15,15,30)$	Velocity Penalties	$diag(5,5,10)$
Cable Direction Penalties	$diag(1.2,1.2,0.0)$	Cable Velocity Penalties	$diag(0.8,0.8,0.0)$
Force Penalties	$diag(50,50,15)$		

8.2 Experiments with Rigid-Body Payload

Similar to the point-mass payload case, we would introduce two modified control algorithms specifically for the rigid-body payload shown in Fig. 8.1b. Unfortunately, *due to recent breakdown of a router, I cannot show the corresponding experimental results for this draft.* I would add back the experimental results in later versions.

8.2.1 Geometric Linearization Method for Payload Hovering

Now consider a rigid-body payload carried by n **point masses where external control forces can be applied in any direction.** Then the dynamics of this virtual system could be given directly as:

$$\begin{bmatrix} m_L I_3 + \sum_{i=1}^n m_i q_i q_i^T & -\sum_{i=1}^n m_i q_i q_{ib}^T \hat{r}_i \\ \sum_{i=1}^n m_i \hat{r}_i q_{ib} q_i^T & J_L + \sum_{i=1}^n m_i (\hat{r}_i q_{ib})(\hat{r}_i q_{ib})^T \end{bmatrix} \begin{bmatrix} \ddot{x}_L + g e_3 \\ \dot{\Omega}_L \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} m_i L_i q_i \\ m i L_i \hat{r}_i q_{ib} \end{bmatrix} (q_i \cdot \frac{F_i}{m_i L_i} - \omega_i \cdot \omega_i - \frac{q_{ib}^T \hat{\Omega}_L^2 r_i}{L_i}) + \begin{bmatrix} 0_{3\times 1} \\ (J_L \Omega_L) \times \Omega_L \end{bmatrix}$$

For hovering task of the payload, we are given For this situation, the higher order term such as $\omega_i \cdot \omega_i$, $(J_L \Omega_L) \times \Omega_L$ would vanish as taking the variation. Thus, the overall system dynamics for geometric linearization could be simplified as:

$$\begin{bmatrix} m_L I_3 + \sum_{i=1}^n m_i q_i q_i^T & -\sum_{i=1}^n m_i q_i q_{ib}^T \hat{r}_i \\ \sum_{i=1}^n m_i \hat{r}_i q_{ib} q_i^T & J_L + \sum_{i=1}^n m_i (\hat{r}_i q_{ib})(\hat{r}_i q_{ib})^T \end{bmatrix} \begin{bmatrix} \ddot{x}_L + g e_3 \\ \dot{\Omega}_L \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} q_i \\ \hat{r}_i q_{ib} \end{bmatrix} (q_i \cdot F_i)$$

For the next step derivation, the gravity term $g e_3$ is ignored, and we would add it back later. Hence, taking variation on both sides of the equation **without considering the gravity term** yields that

$$\underbrace{\begin{bmatrix} m_L I_3 + \sum_{i=1}^n m_i q_{id} q_{id}^T & -\sum_{i=1}^n m_i q_{id} q_{ibd}^T \hat{r}_i \\ \sum_{i=1}^n m_i \hat{r}_i q_{ibd} q_{id}^T & J_L + \sum_{i=1}^n m_i (\hat{r}_i q_{ibd})(\hat{r}_i q_{ibd})^T \end{bmatrix}}_{M_L} \begin{bmatrix} \delta \ddot{x}_L \\ \delta \dot{\Omega}_L \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} q_{id} \\ \hat{r}_i q_{ibd} \end{bmatrix} [(\delta q_i \cdot F_{id}) + (q_{id} \cdot \delta F_i)] + (q_{id} \cdot F_{id}) \begin{bmatrix} \delta q_i \\ \hat{r}_i \delta R_L^T q_{id} + \hat{r}_i R_{Ld}^T \delta q_i \end{bmatrix}$$

where $q_{ibd} = R_{Ld}^T q_{id}$ is the relative direction of the i^{th} cable.

We could further simplify this equation as:

$$\begin{aligned} M_L \begin{bmatrix} \delta \ddot{x}_L \\ \delta \dot{\Omega}_L \end{bmatrix} &= \sum_{i=1}^n \begin{bmatrix} q_{id} F_{id}^T + (q_{id} \cdot F_{id}) I_3 \\ \hat{r}_i q_{ibd} F_{id}^T + (q_{id} \cdot F_{id}) \hat{r}_i R_{Ld}^T \end{bmatrix} \delta q_i + \begin{bmatrix} q_{id} q_{id}^T \\ \hat{r}_i q_{ibd} q_{id}^T \end{bmatrix} \delta F_i + (q_{id} \cdot F_{id}) \begin{bmatrix} 0_{3\times 1} \\ \hat{r}_i \delta R_L^T q_{id} \end{bmatrix} \\ &= \sum_{i=1}^n \begin{bmatrix} q_{id} F_{id}^T + (q_{id} \cdot F_{id}) I_3 \\ \hat{r}_i q_{ibd} F_{id}^T + (q_{id} \cdot F_{id}) \hat{r}_i R_{Ld}^T \end{bmatrix} \delta q_i + \begin{bmatrix} q_{id} q_{id}^T \\ \hat{r}_i q_{ibd} q_{id}^T \end{bmatrix} \delta F_i + (q_{id} \cdot F_{id}) \begin{bmatrix} 0_{3\times 1} \\ \hat{r}_i (R_{Ld} \hat{\eta}_L)^T q_{id} \end{bmatrix} \\ &= \sum_{i=1}^n \begin{bmatrix} q_{id} F_{id}^T + (q_{id} \cdot F_{id}) I_3 \\ \hat{r}_i q_{ibd} F_{id}^T + (q_{id} \cdot F_{id}) \hat{r}_i R_{Ld}^T \end{bmatrix} \xi_i \times q_{id} + \begin{bmatrix} q_{id} q_{id}^T \\ \hat{r}_i q_{ibd} q_{id}^T \end{bmatrix} \delta F_i + (q_{id} \cdot F_{id}) \begin{bmatrix} 0_{3\times 1} \\ \hat{r}_i \hat{q}_{ibd} \end{bmatrix} \eta_L \end{aligned}$$

Now we could add back in the variation term that depends on gravity as:

$$\Delta(g) = \delta\left(\begin{bmatrix} m_L I_3 + \sum_{i=1}^n m_i q_i q_i^T & -\sum_{i=1}^n m_i q_i q_{ib}^T \hat{r}_i \\ \sum_{i=1}^n m_i \hat{r}_i q_{ib} q_i^T & J_L + \sum_{i=1}^n m_i (\hat{r}_i q_{ib})(\hat{r}_i q_{ib})^T \end{bmatrix}\right)\begin{bmatrix} g e_3 \\ 0_{3\times 1} \end{bmatrix}$$

We could also simplify it as:

$$\Delta(g) = \begin{bmatrix} \sum_{i=1}^n m_i(\delta q_i q_{id}^T + q_{id}\delta q_i^T) \\ \sum_{i=1}^n m_i \hat{r}_i(\delta q_{ib} q_{id}^T + q_{ibd}\delta q_i^T) \end{bmatrix} g e_3 = \sum_{i=1}^n m_i g \begin{bmatrix} (q_{id}\cdot e_3) I_3 + q_{id} e_3^T \\ \hat{r}_i((q_{id}\cdot e_3) R_{Ld}^T + q_{ibd} e_3^T) \end{bmatrix} \delta q_i + m_i g (q_{id}\cdot e_3) \begin{bmatrix} 0_{3\times 3} \\ \hat{r}_i \hat{q}_{ibd} \end{bmatrix} \eta_L$$

Similarly, for each cable's direction, it holds that

$$\dot{\omega}_i = q_i \times \left(-\frac{F_i}{m_i L_i} + \frac{1}{L_i}(\ddot{x}_L + g e_3 + R_L(\hat{\Omega}_L^2 r_i + \dot{\hat{\Omega}}_L r_i))\right)$$

whose variation could be given as

$$\begin{aligned} \delta\dot{\omega}_i &= \delta q_i \times (\frac{m_i g e_3 - F_{id}}{m_i L_i}) + q_{id} \times \left(-\frac{\delta F_i}{m_i L_i} + \frac{1}{L_i}(\delta\ddot{x}_L + R_{Ld}\delta\dot{\hat{\Omega}}_L r_i)\right) \\ &= \frac{F_{id} \widehat{-m_i g e_3}}{m_i L_i}\delta q_i - \frac{\hat{q}_{id}}{m_i L_i}\delta F_i + \frac{1}{L_i}\begin{bmatrix} \hat{q}_{id} & -\hat{q}_{id} R_{Ld} \hat{r}_i \end{bmatrix}\begin{bmatrix} \delta\ddot{x}_L \\ \delta\dot{\Omega}_L \end{bmatrix} \end{aligned}$$

Then we could put them together and get the corresponding expression as:

$$\begin{aligned} \begin{bmatrix} \delta\ddot{x}_L \\ \delta\dot{\Omega}_L \end{bmatrix} &= M_L^{-1}\left(\sum_{i=1}^n - \begin{bmatrix} q_{id}\tilde{F}_{id}^T + (q_{id}\cdot\tilde{F}_{id}) I_3 \\ \hat{r}_i q_{ibd}\tilde{F}_{id}^T + (q_{id}\cdot\tilde{F}_{id})\hat{r}_i R_{Ld}^T \end{bmatrix}\hat{q}_{id}\xi_i + (q_{id}\cdot\tilde{F}_{id})\begin{bmatrix} 0_{3\times 1} \\ \hat{r}_i\hat{q}_{ibd} \end{bmatrix}\eta_L + \begin{bmatrix} q_{id} q_{id}^T \\ \hat{r}_i q_{ibd} q_{id}^T \end{bmatrix}\delta F_i\right) \\ \delta\dot{\omega}_i &= -\frac{\hat{\tilde{F}}_{id}}{m_i L_i}\hat{q}_{id}\xi_i + \frac{1}{L_i}\begin{bmatrix} \hat{q}_{id} & -\hat{q}_{id} R_{Ld}\hat{r}_i \end{bmatrix}\begin{bmatrix} \delta\ddot{x}_L \\ \delta\dot{\Omega}_L \end{bmatrix} - \frac{\hat{q}_{id}}{m_i L_i}\delta F_i \end{aligned}$$

where $\tilde{F}_{id} = F_{id} - m_i g e_3$ is the desired tension force within each cable.

From this variation dynamics, we are still able to compute the virtual force as before, and send this command to each quadrotor.

8.2.2 Optimization-based Control Law for Payload Tracking

Linearization-based control law for this system is the very canonical way of designing controllers for complicated systems. The potential challenges lie in the simplification of the linearization process. Once we get the correct expression, designing a LQR-based controller is a routine job. However, the major drawback is that it does not consider any constraint In this part, we are going to introduce another type of controller,

which is more intuitive.

Note that the key control of load transportation is to constantly adjust the cable's directions so that the corresponding wrench could be exerted on the payload. Following this direction, we would decompose the whole control design process into two parts: compute a set of desired cable directions given the current load pose; adjust the cable's directions while treating each quadrotor as an inverted pendulum. Here are the details for each step with justification.

Desired Cable Directions Computation: in the ideal case, if we could adjust the cable's direction instantly, then controlling the pose is just a direct application of geometric control by applying a feedback wrench W_d. However in reality, the set of feasible wrench forms a cone which is embedded in the range of the matrix $A = \begin{bmatrix} q_1 & q_2 & q_3 \\ \hat{r}_1 R_L^T q_1 & \hat{r}_2 R_L^T q_2 & \hat{r}_3 R_L^T q_3 \end{bmatrix}$. Let's first relax to the case when the tension can be negative, which means that $W_d \in \textbf{range}(A)$. However, this might not hold since W_d is six-dimensional, but $\textbf{range}(A)$ is just a three-dimensional subspace which cannot cover a lot of wrench vectors. Hence, to get the desired wrench, we would need to get a set of desired q_{1d}, q_{2d}, q_{3d} such that $W_d \in \textbf{range}(\begin{bmatrix} q_{1d} & q_{2d} & q_{3d} \\ \hat{r}_1 R_L^T q_{1d} & \hat{r}_2 R_L^T q_{2d} & \hat{r}_3 R_L^T q_{3d} \end{bmatrix})$ without **diverging too far from the current set of directions.** Based on this argument, we could formulate this searching process as an optimization as below:

$$(q_1^*, q_2^*, q_3^*) = \textbf{argmin}_{q_{1d}, q_{2d}, q_{3d}} p_1 ||q_{1d} - q_1||^2 + p_2 ||q_{2d} - q_2||^2 + p_3 ||q_{3d} - q_3||^2$$

$$\textbf{Subject to } W_d \in \textbf{range}(\begin{bmatrix} q_{1d} & q_{2d} & q_{3d} \\ \hat{r}_1 R_L^T q_{1d} & \hat{r}_2 R_L^T q_{2d} & \hat{r}_3 R_L^T q_{3d} \end{bmatrix})$$

$$q_{id} \textbf{ stays within a cone with respect to the payload}$$

where $p_1,\ p_2,\ p_3$ are the penalty parameters that would be discussed later.

To set up an optimization, we need to represent the object function and the constraints in a preciser way. Utilizing a similar idea in computer vision, let $x = [T_1 q_{1d}^T,\ T_2 q_{2d}^T,\ T_3 q_{3d}^T]^T$ as a free 9 dimensional vector where T_i is the corresponding coefficient for W_d. Then, it holds that $\underbrace{\begin{bmatrix} I & I & I \\ \hat{r}_1 R_L^T & \hat{r}_2 R_L^T & \hat{r}_3 R_L^T \end{bmatrix}}_{B} x = W_d$. In this way, x could be parametrized as $x = x_0 + \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$ where $Bx_0 = W_d$ is a particular solution, and $x_1, x_2, x_3 \in \textbf{null}(B)$. Now we could treat $\lambda_1, \lambda_2, \lambda_3$ as the optimization variables and convert the previous

optimization to the following one:

$$
\begin{aligned}
(\lambda_1^*, \lambda_2^*, \lambda_3^*) = \textbf{argmin}_{(\lambda_1,\lambda_2,\lambda_3)}\ & p_1(1 - \frac{a^T q_1}{||a||}) + p_2(1 - \frac{b^T q_2}{||b||}) + p_3(1 - \frac{c^T q_3}{||c||}) \\
\textbf{Subject to } & x = x_0 + \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 = \begin{bmatrix} a^T & b^T & c^T \end{bmatrix}^T \\
& \frac{a}{||a||} \cdot q_{1r} \geq \cos\theta_{1t} \\
& \frac{b}{||b||} \cdot q_{2r} \geq \cos\theta_{2t} \\
& \frac{c}{||c||} \cdot q_{3r} \geq \cos\theta_{3t}
\end{aligned}
$$

which looks simpler but there's still room for simplification since the cost function is highly nonlinear.

Now we could get the final optimization which is convex and reasonable intuitively:

$$
\begin{aligned}
(\lambda_1^*, \lambda_2^*, \lambda_3^*) = \textbf{argmin}_{(\lambda_1,\lambda_2,\lambda_3)}\ & a^T(I - q_1 q_1^T)a + b^T(I - q_2 q_2^T)b + c^T(I - q_3 q_3^T)c \\
\textbf{Subject to } & x = x_0 + \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 = \begin{bmatrix} a^T & b^T & c^T \end{bmatrix}^T \\
& a^T(q_{1r} q_{1r}^T - \cos^2\theta_{1t} I)a \geq 0, \quad a^T q_{1r} \geq 0 \\
& b^T(q_{2r} q_{2r}^T - \cos^2\theta_{2t} I)b \geq 0, \quad b^T q_{2r} \geq 0 \\
& c^T(q_{3r} q_{3r}^T - \cos^2\theta_{3t} I)c \geq 0, \quad c^T q_{3r} \geq 0
\end{aligned}
$$

where the conversion of constraints (second order cone) are equivalent but the conversion of cost function is not. But we could see solving this optimization is much simpler because it is a **convex optimization.** To relate this quadratic cost with the previous cost function, let's take a closer look: the previous cost function is given as $p_1(1 - \frac{a^T q_1}{||a||}) + p_2(1 - \frac{b^T q_2}{||b||}) + p_3(1 - \frac{c^T q_3}{||c||})$. If we pick $p_1 = ||a||,\ p_2 = ||b||,\ p_3 = ||c||$, then the cost function is reduced to the expression $(||a|| - a^T q_1) + (||b|| - b^T q_2) + (||c|| - c^T q_3)$ which is the sum of the direct difference between the magnitude and the cable's component. If we tweak this expression to the difference of quadratic term, then it becomes $(||a||^2 - a^T q_1 q_1^T a) + (||b||^2 - b^T q_2 q_2^T b) + (||c||^2 - c^T q_3 q_3^T c)$, which becomes the final quadratic object function.

Once we get a candidate solution x^* from this optimization, the corresponding cable directions would be given as $q_{1d} = a/||a||,\ q_{2d} = b/||b||,\ q_{3d} = c/||c||$. We would utilize these values to compute the corresponding force for each quadrotor.

Desired Force Computation: similar to the geometric control methods, we could decompose each force vector as $F_i = F_i^{\parallel} + F_i^{\perp}$, which are the components that is parallel and perpendicular to q_i respectively.

The parallel components are given as the solution of

$$\begin{bmatrix} F_1^{\parallel} \cdot q_1 \\ F_2^{\parallel} \cdot q_2 \\ F_3^{\parallel} \cdot q_3 \end{bmatrix} = \left(\begin{bmatrix} q_1 & q_2 & q_3 \\ \hat{r}_1 R_L^T q_1 & \hat{r}_2 R_L^T q_2 & \hat{r}_3 R_L^T q_3 \end{bmatrix}^{\dagger} W_d \right) \vee \begin{bmatrix} F_{\min} \\ F_{\min} \\ F_{\min} \end{bmatrix}$$

where $F_{\min} > 0$ is the minimum tension within each cable, and we put a lower threshold for each parallel force $F_i^{\parallel}$.

The perpendicular components are given as the geometric controller for the spherical pendulum as:

$$F_i^{\perp} = -k_{q_i} e_{q_i} - k_{\omega_i} e_{\omega_i}, \quad i = 1,2,3.$$

where the error is computed with respect to the static reference q_{id}.

Then we could put together the virtual force as $F_i = F_i^{\perp} + F_i^{\parallel}$ for $i = 1,2,3$, and send it to the onboard geometric control as before.

Chapter 9

Conclusion and Future Work

In summary, the main contributions in this thesis lie in four aspects: in the first part, we utilize the variation expressions on manifolds as a type of linearization to design linear controllers for tracking the position of a point-mass payload; in the second part, geometric control techniques are employed for reference tracking of a rigid-body payload's pose; while first two parts are focused on tracking control, the third part provides a way to handle safety constraints, where we extend the concept of Control Lyapunov Function and Control Barrier Function from Cartesian space to manifolds, which are called geometric CLF and geometric CBF. By selecting proper candidates of CLF and CBF, we are able to compute the control input through Quadratic Programming (QP), realizing a safety-critical control design for fully-actuated simple mechanical systems and single quadrotor systems; in the last part, we conduct several payload transportation experiments which is mainly based on a combination of geometric-linearization techniques and geometric controller. The corresponding experimental results are accurate for hovering case and slow tracking case, but deviates for fast tracking case.

Chapter 10

Appendix

10.1 Proof of Prop. 4.1

We need to show that rank $\begin{bmatrix} B & \mathscr{A}B & \cdots & \mathscr{A}^{n-1}B \end{bmatrix} = n$, where the linear operator $\mathscr{A}$ is as defined in Section 4.2, and $n = 6$. In particular, we have

$$\begin{bmatrix} B & \mathscr{A}B \end{bmatrix} = \begin{bmatrix} 0_{3\times 3} & I_3 \\ I_3 & J^{-1}(J\hat{\Omega}_d - \hat{\Omega}_d J) \end{bmatrix},$$

which has rank $n = 6$ for all desired trajectories (R_d, Ω_d), implying that the above system is controllable for all trajectories.

10.2 Proof of Lemma 4.3

We have,

$$\frac{d}{dt}(Cs) = (CA + \dot{C})s + CB\delta u.$$

Here, note that $CB \equiv 0$, since $q_d^T \hat{q}_d \equiv 0$. Moreover,

$$\begin{aligned}(CA + \dot{C})s &= \dot{q}_d^T \xi - q_d^T \delta\omega \\ &= (\omega_d \times q_d) \cdot \xi - q_d^T \delta\omega \\ &= -(-\omega_d^T \hat{q}_d \xi + q_d^T \delta\omega) \\ &= 0,\end{aligned}$$

where the last equality follows from the (second) constraint on the variational linearization of the spherical pendulum. It then follows that $\frac{d}{dt}(Cs) \equiv 0$.

10.3 Proof of Lemma 4.5

We need to show that $\mathcal{R}(\begin{bmatrix} B & \mathscr{A}B & \cdots & \mathscr{A}^{n-1}B \end{bmatrix}) \supset \mathcal{N}(C)$, where the linear operator $\mathscr{A}$ is as defined in Section 4.2, $n = 6$, and $\mathcal{N}(C) = \text{colspan}(N)$. We will show that $\mathcal{R}(\begin{bmatrix} B & \mathscr{A}B \end{bmatrix}) \supset \mathcal{N}(C)$. In particular, we have

$$\begin{bmatrix} B & \mathscr{A}B \end{bmatrix} = \begin{bmatrix} 0_{3\times 3} & -\hat{q}_d/ml \\ -\hat{q}_d/ml & -\widehat{\omega_d q_d}/ml \end{bmatrix}.$$

Furthermore, since $\text{colspan}(\begin{bmatrix} N & N^\perp \end{bmatrix}) = \mathbb{R}^6$, we have $\mathcal{R}(\begin{bmatrix} B & \mathscr{A}B \end{bmatrix}) = \text{colspan}(\begin{bmatrix} B & \mathscr{A}B \end{bmatrix} \begin{bmatrix} N & N^\perp \end{bmatrix})$. By carrying out the matrix multiplication, we can easily show that $\mathcal{R}(\begin{bmatrix} B & \mathscr{A}B \end{bmatrix}) \supset \mathcal{N}(C)$. In particular, we note that $\begin{bmatrix} B & \mathscr{A}B \end{bmatrix} N^\perp = 0$, and $\text{colspan}(\begin{bmatrix} B & \mathscr{A}B \end{bmatrix} N) \supset \mathcal{N}(C)$.

10.4 Error Dynamics Derivation for the Quadrotor with a Suspended Load

In Sections 4.4.1, 4.4.2, we derived the variation-based linearization of the 3D and spherical pendulums respectively. Here we provide a detailed derivation of the variation-based linearization of the quadrotor with a cable-suspended load system about a desired reference trajectory. First, symbolically taking the variation of the dynamics of the quadrotor with a cable-suspended load, specified in (4.4.3), yields,

$$\begin{aligned}
\delta \dot{x}_L &= \delta v_L, \\
(m_Q + m_L)\delta \dot{v}_L &= [\delta q \cdot f_d R_d e_3 + q_d \cdot (\delta f R_d e_3 + f_d \delta R e_3) \\
&\quad - 2m_Q L(\dot{q}_d \cdot \delta \dot{q})]q_d \\
&\quad + (q_d \cdot f_d R_d e_3 - m_Q L(\dot{q}_d \cdot \dot{q}_d))\delta q, \\
\delta \dot{q} &= \delta \omega \times q_d + \omega_d \times \delta q, \\
m_Q L \delta \dot{\omega} &= -\delta q \times f_d R_d e_3 - \hat{q}_d(\delta f R_d e_3 + f_d \delta R e_3), \\
\delta \dot{R} &= R_d \delta \hat{\Omega} + \delta R \hat{\Omega}_d, \\
J_Q \delta \dot{\Omega} &= \delta M - \delta \Omega \times J_Q \Omega_d - \Omega_d \times J_Q \delta \Omega.
\end{aligned}$$

Rearranging each term, the error dynamics can be simplified into:

$$\begin{aligned}
\delta\dot{x}_L &= \delta v_L, \\
(m_Q+m_L)\delta\dot{v}_L &= [(q_d\cdot f_d R_d e_3 - m_Q L(\dot{q}_d\cdot\dot{q}_d))I_3 \\
&\quad + q_d(f_d R_d e_3)^T]\delta q - 2m_Q L(q_d\dot{q}_d^T)\delta\dot{q} \\
&\quad + f_d\delta R e_3 + (q_d^T R_d e_3)\delta f, \\
\delta\dot{q} &= \hat{\omega}_d\delta q - \hat{q}_d\delta\omega, \\
m_Q L\delta\dot{\omega} &= \widehat{f_d R_d e_3}\delta q - \hat{q}_d f_d\delta R e_3 - \hat{q}_d R_d e_3\delta f, \\
\delta\dot{R} &= R_d\delta\hat{\Omega} + \delta R\hat{\Omega}_d, \\
J_Q\delta\dot{\Omega} &= (\widehat{J_Q\Omega_d} - \widehat{\Omega}_d J_Q)\delta\Omega + \delta M.
\end{aligned}$$

Now, reusing the conclusions already attained for the 3D pendulum and spherical pendulum, we can further simplify these equations to obtain,

$$\begin{aligned}
\delta\dot{x}_L &= \delta v_L, \\
(m_Q+m_L)\delta\dot{v}_L &= -[(q_d\cdot f_d R_d e_3 - m_Q L(\dot{q}_d\cdot\dot{q}_d))I_3 \\
&\quad + q_d(f_d R_d e_3)^T]\hat{q}_d\xi \\
&\quad - 2m_Q L q_d\dot{q}_d^T(-\hat{\omega}_d\hat{q}_d\xi - \hat{q}_d\delta\omega) \\
&\quad - f_d R_d\hat{e}_3\eta + (q_d^T R_d e_3)\delta f, \\
\dot{\xi} &= q_d q_d^T(\omega_d\times\xi) + (I_3 - q_d q_d^T)\delta\omega, \\
m_Q L\delta\dot{\omega} &= -\widehat{f_d R_d e_3}\hat{q}_d\xi + \hat{q}_d f_d R_d\hat{e}_3\eta - \hat{q}_d R_d e_3\delta f, \\
\dot{\eta} &= -\hat{\Omega}_d\eta + I_3\delta\Omega, \\
J_Q\delta\dot{\Omega} &= (\widehat{J_Q\Omega_d} - \widehat{\Omega}_d J_Q)\delta\Omega + \delta M.
\end{aligned}$$

Also, using the vector triple product and the fact that $\omega_d\cdot q_d = 0$, we have,

$$\hat{\omega}_d\hat{q}_d\xi = (\xi\cdot\omega_d)q_d - (\omega_d\cdot q_d)\xi = (\xi\cdot\omega_d)q_d.$$

Right multiply the above equation by $\dot{q}_d^T$ on both sides, and recognizing that $q_d\cdot\dot{q}_d = 0$, we have,

$$\dot{q}_d^T\hat{\omega}_d\hat{q}_d\xi = \dot{q}_d^T q_d(\xi\cdot\omega_d) = (q_d\cdot\dot{q}_d)(\xi\cdot\omega_d) = 0.$$

We can then finally write down the linearized error dynamics for the quadrotor with a cable-suspended load as the following:

$$\begin{aligned}
\delta\dot{x}_L &= \delta v_L, \\
(m_Q+m_L)\delta\dot{v}_L &= -\{[(q_d\cdot f_d R_d e_3 - m_Q L(\dot{q}_d\cdot\dot{q}_d))I_3 \\
&\quad + q_d(f_d R_d e_3)^T]\hat{q}_d\}\xi + (2m_Q L q_d\dot{q}_d^T\hat{q}_d)\delta\omega \\
&\quad - f_d R_d\hat{e}_3\eta + (q_d^T R_d e_3)\delta f, \\
\dot{\xi} &= (q_d q_d^T\hat{\omega}_d)\xi + (I_3 - q_d q_d^T)\delta\omega, \\
m_Q L\delta\dot{\omega} &= -\widehat{f_d R_d}e_3\hat{q}_d\xi + \hat{q}_d f_d R_d\hat{e}_3\eta - \hat{q}_d R_d e_3\delta f, \\
\dot{\eta} &= -\hat{\Omega}_d\eta + I_3\delta\Omega, \\
J_Q\delta\dot{\Omega} &= (\widehat{J_Q\Omega_d} - \widehat{\Omega}_d J_Q)\delta\Omega + \delta M,
\end{aligned}$$

which is of the form (4.6)-(4.7), with the state $s = \begin{bmatrix}\delta x_L & \delta v_L\xi & \delta\omega & \eta & \delta\Omega\end{bmatrix}^T$, and constraint $q_d\cdot\xi\equiv 0$. Recognizing that this constraint is equivalent to $q_d\cdot\xi = \frac{d}{dt}(q_d\cdot\xi) = 0$, we can write this in matrix form as $Cs = 0$, where

$$C = \begin{bmatrix} 0_{1\times 6} & q_d^T & 0_{1\times 3} & 0_{1\times 6} \\ 0_{1\times 6} & -\omega_d^T\hat{q}_d & q_d^T & 0_{1\times 6}\end{bmatrix}.$$

10.5 Proof of Prop. 5.1

The key idea of proof comes from [75]. We first write out the closed-loop error dynamics for each quadrotor's orientation as

$$\varepsilon\begin{bmatrix}\dot{\tilde{e}}_{R_i} \\ \dot{e}_{\Omega_i}\end{bmatrix} = \begin{bmatrix}\frac{1}{2}(tr(R_i^T R_{ic})I_3 - R_i^T R_{ic})e_{\Omega_i} \\ -k_{R_i}\tilde{e}_{R_i} - k_{\Omega_i}e_{\Omega_i}\end{bmatrix}$$

where the new error function $\tilde{e}_{R_i} = e_{R_i}/\varepsilon$.

By *Prop. 1* in [75], there exists suitable values for $k_{R_i}, k_{\Omega_i}, \bar{\varepsilon}$ such that the quadrotor's orientation R_i could track the reference R_{ic} exponentially. In addition, the exponential rate is directly related to $\frac{1}{\varepsilon}$. On the other hand side, the error function for the external force

$$f_i R_i e_3 - v_i = (v_i\cdot R_i e_3)R_i e_3 - v_i = (I - e_3^i(e_3^i)^T)v_i$$

Then rewrite this expression in terms of e_3^{ic} as:

$$f_i R_i e_3 - v_i = [e_3^{ic} - (e_3^i \cdot e_3^{ic}) e_3^i] \cdot ||v||$$

Due to exponential tracking of attitude, it holds that e_{3i} tends to e_{3ic} exponentially. Based on the previous expression and boundedness of $||v||$, we have that the external force $f_i R_i e_3$ tends to v_i exponentially.

10.6 Proof of Prop. 5.2

For the reduced system, what we could directly control is the force v_i applied to each point-mass. Based on the previous discussion in 5.1, we have that only the parallel part of v_i denoted as $u_{iv}^{\parallel}$ could affect the load's pose dynamics while its perpendicular part $u_{iv}^{\perp}$ could change the cable's attitude.Thus, with the given feedback law:

$$\begin{bmatrix} u_{1v}^{\parallel} \\ u_{2v}^{\parallel} \\ \vdots \\ u_{nv}^{\parallel} \end{bmatrix} = u_v^{\parallel} = G^{\dagger}(-d + AW_d),$$

$$\begin{aligned} u_{iv}^{\perp} = \hat{q}_i((q_i \cdot \omega_{id})\dot{q}_i - \hat{q}_i^2 \dot{\omega}_{id} - (k_{q_i} e_{q_i} + k_{\omega_i} e_{\omega_i})) \\ - \frac{1}{L_i} \hat{q}_i^2 (\ddot{x}_{Lc} + g e_3 + R_L(\hat{\Omega}_L^2 + \dot{\hat{\Omega}}_{Lc}) r_i), \end{aligned}$$

we can compute the closed-loop accelerations of the load directly as:

$$\begin{aligned} A \begin{bmatrix} \ddot{x}_{Lr} + g e_3 \\ \dot{\Omega}_{Lr} \end{bmatrix} &= G(q) \begin{bmatrix} u_{1v}^{\parallel} \\ u_{2v}^{\parallel} \\ \vdots \\ u_{nv}^{\parallel} \end{bmatrix} + d, \\ &= G \cdot G^{\dagger}(AW_d - d) + d, \\ &= AW_d, \end{aligned}$$

where we have utilized the fact that $G \cdot G^{\dagger} = I$. Thus, the error dynamics of the load's pose for the reduced system is,

$$\begin{aligned} \ddot{e}_{x_{Lr}} = & -k_{x_L} e_{x_{Lr}} - k_{v_L} e_{v_{Lr}} \\ \dot{e}_{\Omega_{Lr}} = & -k_{R_L} e_{R_{Lr}} - k_{\Omega_L} e_{\Omega_{Lr}} \end{aligned}$$

Similarly, for the tracking of cable swing dynamics, we have:

$$\dot{e}_{\omega_{ir}} = -k_{q_i} e_{q_{ir}} - k_{\omega_i} e_{\omega_{ir}}$$

Thus, the errors of load's translational, rotational and cable attitude are totally decoupled from each other.Thus we could treat each of them as independent subsystems. As well-studied in the [70, 66], there exist suitable gains $k_{x_L}, k_{v_L}, k_{R_L}, k_{\Omega_L}$ and $k_{q_i}, k_{\omega_i}, k_{R_i}, k_{\Omega_i}$ $i = 1, 2, \cdots, n$ for the reduced system such that almost global exponential stability is guaranteed.

10.7 Proof of Prop. 5.3

We start the proof by first rearranging the terms in the hsystem dynamics:

$$\begin{aligned} \begin{bmatrix} \ddot{x}_L + g e_3 \\ \dot{\Omega}_L \end{bmatrix} = & -W_d + A^{-1} G \begin{bmatrix} f_1 R_1 e_3 - v_1 \\ f_2 R_2 e_3 - v_2 \\ \vdots \\ f_n R_n e_3 - v_n \end{bmatrix}, \\ \dot{\omega}_i = & -\hat{q}_i^2 [(q_i \cdot \omega_{id}) \dot{q}_i - \hat{q}_i^2 \dot{\omega}_{id} - (k_{q_i} e_{q_i} + k_{\omega_i} e_{\omega_i})] \\ & - \hat{q}_i (f_i R_i e_3 - v_i) \\ & + q_i \times \frac{1}{L_i} [\ddot{x}_L - \ddot{x}_{Lr} + R_L (\dot{\hat{\Omega}}_L - \dot{\hat{\Omega}}_{Lr}) r_i]. \end{aligned}$$

Next, rewriting the equation in terms of the errors for the full and reduced system, we have,

$$\begin{bmatrix} \ddot{e}_{x_L} \\ \dot{e}_{\Omega_L} \end{bmatrix} = \begin{bmatrix} \ddot{e}_{x_{Lr}} \\ \dot{e}_{\Omega_{Lr}} \end{bmatrix} + A^{-1} G \begin{bmatrix} \delta F_1 \\ \delta F_2 \\ \vdots \\ \delta F_n \end{bmatrix},$$

$$\begin{aligned} \dot{e}_{\omega_i} = \dot{e}_{\omega_{ir}} - \hat{q}_i(\delta F_i) \\ + \frac{1}{L_i} q_i \times [\ddot{e}_{x_L} - \ddot{e}_{x_{Lr}} - R_L \hat{r}_i (\dot{e}_{\Omega_L} - \dot{e}_{\Omega_{Lr}})], \end{aligned}$$

where $\delta F_i = f_i R_i e_3 - v_i$. Moreover, the fast-changing dynamics of quadrotor's orientation are given by,

$$\varepsilon \begin{bmatrix} \dot{\tilde{e}}_{R_i} \\ \dot{e}_{\Omega_i} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(tr(R_i^T R_{ic}) I_3 - R_i^T R_{ic}) e_{\Omega_i} \\ -k_{R_i} \tilde{e}_{R_i} - k_{\Omega_i} e_{\Omega_i} \end{bmatrix}.$$

By Prop. 5.1, for proper selected gain parameters k_{R_i}, k_{Ω_i}, there exists $\bar{\varepsilon}$ such that whenever $\varepsilon < \bar{\varepsilon}$, the *quadrotor's yaw angles* can be tracked exponentially. Next, treating ε as the perturbation parameter of the error dynamic system, we have, when $\varepsilon \to 0$, we have $f_i R_i e_3 \to v_i$ (i.e., $\delta F_i \to 0$.) In this case, the error of the full system becomes the error of the reduced system, and thus it will decay exponentially by Prop. 5.2. However, we know that for a physical system, ε can never become zero, but the method of *singular perturbations* indicates that for sufficiently small ε, the exponential stability of the reduced system can still be preserved for the full system under certain conditions [84, Thm. 11.4]. Now we are going to check these conditions step by step.

- It's obvious that zero is an isolated equilibrium for the error system.
- From Prop. 5.2, we know that the origin of the error dynamics for the reduced system is exponentially stable under a properly selected parameter set, $k_{x_L}, k_{v_L}, k_{R_L}, k_{\Omega_L}$ and $k_{q_i}, k_{\omega_i}, k_{R_i}, k_{\Omega_i}$.
- Since all the expressions involved in the dynamics are smooth, their partial derivatives are continuous functions. So *according to boundedness of state, we can conclude that all the partial derivatives up to the second order are bounded*.
- The fast dynamics, i.e quadrotor orientation dynamics, is also exponentially stable when $\varepsilon < \bar{\varepsilon}$ by Prop. 5.1.

Thus, applying Theorem 11.4 in [84] yields that there exists a $\tilde{\varepsilon}$ such that whenever $\varepsilon \leq \tilde{\varepsilon}$, the error for the full dynamical model would tend to zero exponentially. We now select the value $\varepsilon^* = \min\{\bar{\varepsilon}, \tilde{\varepsilon}\}$, and the conclusion of Prop. 5.3 holds accordingly.

10.8 Proof of Lemma 6.1

From condition (6.7) and Chain rule, the time derivative of B could be expressed as:

$$\dot{B} = B'(h) \cdot \dot{h} \leq \frac{\mu}{B}$$

Hence, by Lemma. 1 in [46], the set $\{(q,\dot{q}) \in TM : B(t,q,\dot{q}) > 0\}$ is forward invariant, which is C_t° by the first condition of CBF.

10.9 Proof of Lemma 6.2

We will first select the candidate function to be $\beta(x) = x$ and choose any parameters γ_i such that $\gamma_i \geq c_i$ for $i = 1,2,\cdots,k$. Then a candidate $\mathcal{C}_t$ is constructed.

Expanding out the expression of each h_i yields

$$h_i = (-1)^{\delta_i}(\dot{b}_i - \langle d_1\Psi_{q,q_i}, \dot{q}\rangle - \langle d_2\Psi_{q,q_i}, \dot{q}_i\rangle) + \gamma_i g_i$$

where the first bracket is evaluated at q, and the second is evaluated at q_i. Plugging in the property of transport map in (6.2), we have

$$h_i = (-1)^{\delta_i+1}\langle d\Psi_{q,q_i}, \dot{q} - \mathscr{T}_{q,q_i}\dot{q}_i\rangle + \gamma_i g_i + (-1)^{\delta_i}\dot{b}_i$$

Fix a time $t \geq 0$, since $\mathcal{B}_t$ is nonempty, it must contain a particular point $q \in \mathcal{B}_t$. Then for each $q \in \mathcal{B}_t$, if we want to find a tangent vector $\dot{q} \in T_qM$ such that $(q,\dot{q}) \in \mathcal{C}_t$, it must satisfy the conditions below for each $i = 1,2,\cdots,k$:

$$(-1)^{\delta_i+1}\langle d\Psi_{q,q_i}, \dot{q} - \mathscr{T}_{q,q_i}\dot{q}_i\rangle + \gamma_i g_i + (-1)^{\delta_i}\dot{b}_i \geq 0 \tag{10.1}$$

From the condition given, it holds that $\gamma_i g_i \geq c_i g_i \geq |\dot{b}_i| \geq -(-1)^{\delta_i}\dot{b}_i$, which implies $\gamma_i g_i + (-1)^{\delta_i}\dot{b}_i \geq 0$. Hence, by inspecting Eq. (10.1), we have $(q, \mathscr{T}_{q,q_i}\dot{q}_i) \in \mathcal{C}_t$ for each $q \in \mathcal{B}_t$ which makes $\mathcal{C}_t$ nonempty.

10.10 Proof of Lemma 6.3

Note that a set is path connected if every set of points in the set are connected by a piecewise continuously differentiable trajectory. To prove path connectedness of $\mathcal{C}_t^B$, we construct a path between any two points in $\mathcal{C}_t^B$ using geodesics and path-connectedness of $\mathcal{B}_t^\circ$. In particular, we want to propagate the initial state and

the end state through geodesic flows and connect the geodesics together through the path connectedness of $\mathcal{B}_t^\circ$.

Given two states $(q,\dot{q}),(\hat{q},\dot{\hat{q}}) \in \mathcal{C}_t^B$. We want to construct a piecewise smooth path $\Phi : [-1,1] \to \mathcal{C}_t^B$ such that $\Phi(-1) = (q,\dot{q}),\ \Phi(1) = (\hat{q},\dot{\hat{q}})$. Since the constraint function g_i is time-invariant, for any pair $(q,\dot{q}) \in \mathcal{C}_t^B$, it holds that

$$h_i = (-1)^{\delta_i+1}\langle d\Psi_{q,q_i},\dot{q}\rangle + \gamma_i g_i > 0, \tag{10.2}$$

For the case when $q = \hat{q}$, we know that both $\dot{q},\dot{\hat{q}}$ belong to the same tangent space T_qM, which satisfy the condition (10.2). Since T_qM is a Hilbert space equipped with the Riemannian metric, the collection of tangent vectors satisfying (10.2) is an open convex polygon. Thus a straight line can connect them together as $\Phi(t) = \left(q, \frac{1}{2}(t\dot{q} + (1-t)\dot{\hat{q}})\right)$ where $t \in [-1,1]$.

For the case when $q \neq \hat{q}$, a candidate path can be constructed in the following two steps. Since both $q,\hat{q} \in \mathcal{B}_t^\circ$, there exist two open neighborhoods U_1,U_2 around $q,\hat{q}$ such that $U_1,U_2 \subset \mathcal{B}_t^\circ$. Using the fact that M is a Riemannian manifold, there exists a unique geodesics passing through q, denoted as $\varphi_1 : [-1,1] \to M$, which satisfies that $\varphi_1(0) = q,\ \varphi_1'(0) = \dot{q}$. Similarly, there's another geodesics $\varphi_2 : [-1,1] \to M$ passing through $\hat{q}$ which satisfies $\varphi_2(0) = \hat{q},\ \varphi_2'(0) = \dot{\hat{q}}$. Also, since both U_1,U_2 are both nonempty open sets, they both contain closed subsets. Hence, there exist $\delta_1,\delta_2 > 0$ such that $\varphi_1([-\delta_1,\delta_1]) \subset U_1,\ \varphi_2([-\delta_2,\delta_2]) \subset U_2$. Considering the segments of geodesics $\varphi_1([-\delta_1,\delta_1]),\varphi_2([-\delta_2,\delta_2]) \subset \mathcal{B}_t^\circ$, we want to truncate these two geodesics to make sure the truncated parts with their derivatives are also contained in $\mathcal{C}_t^B$. Let $T_1 = \sup_{0\le s\le\delta_1}\{h_i(\varphi_1(s),\varphi_1'(s)) > 0 \text{ for every } i = 1,2,\cdots,k\}$, $T_2 = \inf_{-\delta_2\le s\le 0}\{h_i(\varphi_2(s),\varphi_2'(s)) > 0 \text{ for every } i = 1,2,\cdots,k\}$. Since the functions $h_i(\varphi_1(s),\varphi_1'(s))$ are smooth functions defined on $[-\delta_1,\delta_1]$, then it holds that $T_1 > 0$. Or otherwise, for each i, we could find a sequence $\{t_n\} \to 0$ such that $h_i(\varphi_1(t_n),\varphi_1'(t_n)) = 0$ which would imply that $h_i(q,\dot{q}) = 0$. This leads to a contradiction and that $T_1 > 0$. Applying a similar argument to φ_2 yields that $T_2 < 0$. Also, by the definition of T_1,T_2, we have the subset $\{(\varphi_1(t),\varphi_1'(t)) \in TM : 0 \le t \le T_1/2\},\ \{(\varphi_2(t),\varphi_2'(t)) \in TM : T_2/2 \le t \le 0\} \subset \mathcal{C}_t^B$. Since both $\varphi_1(T_1/2),\varphi_2(T_2/2) \in \mathcal{B}_t^\circ$, using the path connectedness of $\mathcal{B}_t^\circ$, there exists a piecewise smooth path $\varphi_3 : [-1,1] \to \mathcal{B}_t^\circ$ which satisfies $\varphi_3(-1) = \varphi_1(T_1/2),\ \varphi_3(1) = \varphi_2(T_2/2)$.

The next step is to scale the path φ_3 to make sure its derivative satisfies the condition (10.2) all the time. Define a new path as $\psi(t) = \varphi_3(t/K)$ whose derivative is given by $\psi'(t) = \varphi_3'(t/K)/K$ where $K > 0$. Note that for the scaled trajectory ψ, it is defined on $[-K,K]$. In order to make sure $(\psi(t),\psi'(t)) \in \mathcal{C}_t^B$ for every $t \in [-K,K]$, the following inequality should hold

$$\frac{1}{K}(-1)^{\delta_i+1}\langle d\Psi_{\varphi_3(t/K),q_i},\varphi_3'(t/K)\rangle + \gamma_i g_i(\varphi_3(t/K)) > 0$$

which can be converted to the inequality below,

$$K \geq \frac{(-1)^{\delta_i}\langle d\Psi_{\varphi_3(u),q_i}, \varphi_3'(u)\rangle}{\gamma_i g_i(\varphi_3(u))} = s_i(u) \tag{10.3}$$

for every $i = 1,2,\cdots,k$ and every $u \in [-1,1]$.

Since $\varphi_3(u) \in \mathcal{B}_t^\circ$, by definition $g_i(\varphi_3(u)) > 0$ for every $u \in [-1,1]$. Hence, function $s_i(u)$ on the right hand side of (10.3) is well-defined and piecewise continuous over the interval $[-1,1]$. Because there are only finite jumps of s_i in the whole interval $[-1,1]$, it has a maximum value over the compact interval $[-1,1]$ for each i. Let $K_0 = \max_{i=1,2,\cdots,k} \sup_{u\in[-1,1]} s_i(u)$ which is finite. Then, pick a value $K > K_0$ and we could make sure that $(\psi(t), \psi'(t)) \in \mathcal{C}_t^B$ for all $t \in [-K,K]$.

Now assemble the final path together as $\Phi(t) = (\varphi_1(t), \varphi_1'(t))$ when $0 \leq t \leq \frac{T_1}{2}$; $\Phi(t) = (\varphi_1(\frac{T_1}{2}), \frac{2(T_1-t)}{T_1}\varphi_1'(\frac{T_1}{2}) + \frac{2t-T_1}{T_1}\psi'(-K))$ when $\frac{T_1}{2} \leq t \leq T_1$; $\Phi(t) = (\psi(t-T_1), \psi'(t-T_1))$ when $T_1 \leq t \leq T_1+2K$; $\Phi(t) = (\varphi_2(\frac{T_2}{2}), \frac{2(T_1+2K-T_2/2-t)}{-T_2}\psi'(K)$ $\frac{2(T_1+2K-t)}{T_2}\varphi_2'(\frac{T_2}{2}))$ when $T_1+2K \leq t \leq T_1+2K-T_2/2$; $\Phi(t) = (\varphi_2(t+T_2-T_1-2K), \varphi_2'(t+T_2-T_1-2K))$ when $T_1+2K-T_2/2 \leq t \leq T_1+2K-T_2$. Note from the construction, the overall trajectory $\Phi(t)$ is continuous and piecewise smooth. Also, using the properties of $\varphi_1, \varphi_2, \psi$, it holds that $\Phi([0, T_1-T_2+2K]) \in \mathcal{C}_t^B$ which connects any pair of $(q,\dot{q}), (\hat{q},\dot{\hat{q}})$. Thus, the set $\mathcal{C}_t^B$ is path connected.

10.11 Proof of Proposition 6.4

Since $(q_0, \dot{q}_0) \in \mathcal{C}_0$, by forward invariance, it holds that $(q(t), \dot{q}(t)) \in \mathcal{C}_t$ which is equivalent to

$$\gamma_i \beta(g_i(q(t))) + \langle dg_i, \dot{q}(t)\rangle + \frac{\partial g_i}{\partial t}|_{t,q(t)} \geq 0 \tag{10.4}$$

for any $i \in \{1,2,\cdots,k\}$ and $t \in [0,\infty)$

Consider the extreme case when the system trajectory reaches the boundary of $\mathcal{B}_t$ at $t_1 > 0$, then there exists $j \in \{1,2,\cdots,k\}$ such that $g_j(t_1,t_1,q(t_1)) = 0$. By Chain rule and the previous inequality (10.4), it follows that

$$\left.\frac{dg_j}{dt}\right|_{t=t_1} = \langle dg_j, \dot{q}(t_1)\rangle + \frac{\partial g_j}{\partial t}|_{t_1,q(t_1)} \geq 0,$$

which means that the value of g_j would never decrease below zeros.

The previous argument indicates that the system trajectory would never escape $\mathcal{B}_{j,t}$ when it reaches the boundary of $\mathcal{B}_{i,t}$. This implies that the system trajectory would never escape the safety region $\mathcal{B}_t$, namely, $\mathcal{B}_t$ is forward invariant.

10.12 Proof of Proposition 6.5

Consider the quadratic programming in (6.19) with positive definite coefficient matrix $Q > 0$. Then if a solution exists for this QP which is strictly convex, this solution is unique according to proposition in convex optimization. So by assumption this control input is well-defined.

Since the set $\mathcal{D}$ has measure zero, it has an empty interior. Hence, the system's trajectory can only traverse it at discrete time points. For the time period (t_1, t_2) when $(q, \dot{q}) \in \mathcal{C} \backslash \mathcal{D}$, all the hard constraints are satisfied by the controller. Applying Theorem 2 in [46] yields that system trajectory would never escape $\mathcal{C} \backslash \mathcal{D}$. When $(q, \dot{q}) \in \mathcal{D}$, a solution might not exist but the current state lies in C° since $\mathcal{D} \in C^{\circ}$. For both cases, the system state always stay within $\mathcal{C}$. Thus it's forward-invariant for system (6.3). Using Proposition (6.4), it follows that the feasible region $\mathcal{B}$ is also forward-invariant.

10.13 Proof of Lemma 7.4

Proof. Assuming that the thresholds $F_{\max}, M_{\max}$ are unbounded, it holds that

$$\begin{aligned} \dot{\hat{B}} - \frac{\gamma}{\hat{B}} \leq 0 \Leftrightarrow & -\frac{1}{\hat{h}^{a+1}}(\dot{\hat{h}} + \gamma \hat{h}^{2a+1}) \leq 0, \\ & \Leftrightarrow \dot{\hat{h}} + \gamma \hat{h}^{2a+1} \geq 0. \end{aligned} \tag{10.5}$$

Substituting the expression of $\dot{\hat{h}}$ yields

$$\frac{p\sigma'(s)}{J} M + \frac{1}{m}(2s - \sigma'(s))F \geq -\Gamma + \gamma \hat{h}^{2a+1}. \tag{10.6}$$

If $p \neq 0$, then $p\sigma'(s) \neq 0$ which means that we could always select a moment to satisfy (10.6) with:

$$M = \frac{J(-\Gamma + \gamma \hat{h}^{2a+1})}{p\sigma'(s)}, \quad F = 0$$

If $p = 0$, then by definition it follows that r, q are parallel to each other, and $s = r \cdot q > 0$ which implies $2s - \sigma'(s) > 0$ since $\sigma'(s) < 0$ for $s \in \mathbb{R}$. Hence, we can use a large enough thrust to satisfy (10.7).

Otherwise for the case $p = 0$, the only situation when (10.7) may fail is given by $s \leq -\sqrt{\beta} b$ based on the third condition. This is actually two-dimensional compact manifold in $\mathbb{R}^2 \times \mathbb{S}^1$ and thus has Lebesgue measure zero in the state space. So $\hat{B}$ is an almost global CBF. For the case when $\hat{B}$ fails to work, we have that $s \leq -\sqrt{\beta} b$ which means that $|s| = \sqrt{(x - x_o)^2 + (y - y_o)^2} \geq \sqrt{\beta} b$ which implies that $(x - x_o)^2 + (y - y_o)^2 \geq \beta b^2 > b^2$. This implies that the trajectory would be safe even for the cases when $\hat{B}$ fails to work. When the condition imposed on $\hat{B}$ is satisfied, by *Lemma*. 1, the system trajectory would always remain within

the region $\mathcal{D}_1 = \{(x,y) : (x-x_o)^2 + (y-y_o)^2 \geq \beta b^2 + \sigma(s)\} \subset \{(x,y) : (x-x_o)^2 + (y-y_o)^2 \geq (\beta b^2 + (1-\beta)b^2\} = \{(x,y) : (x-x_o)^2 + (y-y_o)^2 \geq b^2\} = \mathcal{D}$ since $\sigma \geq -(\beta-1)b^2$. Hence, the system trajectory will stay within the safe set $\mathcal{D}$, defined by $g(x,y)$. □

10.14 Proof of Lemma 7.4

Proof. Assume $F_{\max}, M_{i,\max}$ are unbounded. Then it holds that

$$\begin{aligned} \dot{B} - \frac{\gamma}{B} \leq 0 \Leftrightarrow & -\frac{1}{\hat{h}^{a+1}}(\dot{\hat{h}} + \gamma \hat{h}^{2a+1}) \leq 0, \\ & \Leftrightarrow \dot{\hat{h}} + \gamma \hat{h}^{2a+1} \geq 0. \end{aligned} \tag{10.7}$$

Note that the time derivative of $\hat{h}$ defined in (7.8) can be written as

$$\begin{aligned} \dot{\hat{h}} &= \gamma \alpha'(\hat{g})\dot{\hat{g}} + 2(x-x_o)\cdot(\ddot{x}-\ddot{x}_o) \\ &+ 2(\dot{x}-\dot{x}_o)\cdot(\dot{x}-\dot{x}_o) - \beta\ddot{b} - \sigma''(s)\dot{s}^2 \\ &- \sigma'(s)[(Re_3)\cdot(\ddot{x}-\ddot{x}_o) + 2(\dot{x}-\dot{x}_o)\cdot R(\Omega \times e_3) \\ &+ (x-x_o)^T R\hat{\Omega}^2 e_3 - (x-x_o)^T R\hat{e}_3\dot{\Omega}] \\ &= \frac{1}{m}(2s-\sigma'(s))F + (\sigma'(s)(R^T r)^T \hat{e}_3 J^{-1})M + \Gamma_3 \\ &= \Gamma_1 F + \Gamma_2^T M + \Gamma_3 \end{aligned} \tag{10.8}$$

where the term Γ_3 is independent of $\ddot{x}$ and $\dot{\Omega}$ and Γ_1, Γ_2 are the coefficients of f and M respectively.

We can check several cases depending on the term Γ_3. If $\Gamma_3 \neq 0$, then we could assign the following control inputs:

$$F = 0, \quad M = \frac{-\Gamma_3 - \gamma \hat{h}^{2a+1}}{||\Gamma_2||^2} J\Gamma_2$$

which means that we could always avoid the obstacle by adjusting the altitude for this case.

When $\Gamma_2 = 0$, it holds that $r\hat{R}e_3 = 0$ since $\sigma'(s) < 0$ according to (7.5) and the matrix J is nonsingular. Further, it holds that $e_3 \times R^T r = 0$ which means that the vectors q and r are parallel. For the case when the directions of q and r coincide, we have $\Gamma_1 = 2s - \sigma'(s) > 2s = 2r \cdot q > 0$ using condition (7.5). This means that we could apply a large enough thrust to satisfy CBF condition as:

$$F = m\frac{-\Gamma_3 - \gamma \hat{h}^{2a+1}}{2s - \sigma'(s)}, \quad M = 0$$

When the directions q and r are contrary to each other, whether the CBF condition holds depends on the

range of s. The only condition when CBF would fail is that $s < -\sqrt{\beta}b$. Fortunately, the set $\{(x,R,\dot{x},\Omega) \in TSE(3) : s \leq -\sqrt{\beta}b\}$ is a submanifold in $TSE(3)$ and has Lebesgue measure zero. So B is an almost global CBF. Consider the case when CBF condition fails, we have that $s \leq -\sqrt{\beta}b$ which means that $-s = -q \cdot r = ||r|| \geq \sqrt{\beta}b$ by condition (7.7). This means that $g = ||r||^2 \geq \beta b > b$, and thus the system trajectory would remain within $\mathcal{B}$. When the condition of CBF is satisfied, the region $\mathcal{C}$ is forward invariant. By *Lemma* 7.3, we have the original safety region $\mathcal{B}$ is forward invariant. Combining the previous argument, the system trajectory will always remain within $\mathcal{B}$ for either case and thus stay safe. □

Bibliography

[1] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems (RSS)*, 2013.

[2] T. Lee, M. Leoky, and N. H. 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.

[3] T. Lee, K. Sreenath, and V. Kumar, "Geometric control of cooperating multiple quadrotor uavs with a suspended payload," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 5510–5515.

[4] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3258–3264.

[5] I. Maza, K. Kondak, M. Bernard, and A. Ollero, "Multi-uav cooperation and control for load transportation and deployment," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 417–449, 2010.

[6] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2691–2697.

[7] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation: Safe and efficient load manipulation with aerial robots," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[8] 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 *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4888–4895.

[9] S. Tang and V. 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.

[10] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011.

[11] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[12] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, p. 0278364911434236, 2012.

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

[14] G. Wu and K. Sreenath, "Geometric control of multiple quadrotors transporting a rigid-body load," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6141–6148.

[15] T. Lee, "Geometric control of multiple quadrotor uavs transporting a cable-suspended rigid body," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6155–6160.

[16] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric control of a quadrotor uav transporting a payload connected via flexible cable," *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1486–1498, 2015.

[17] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," in *Robotics: Science and Systems*, Los Angeles, CA, June 2011.

[18] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed Autonomous Robotic Systems*, Lausanne, Switzerland, November 2010, pp. 545–558.

[19] N. Yanai, M. Yamamoto, and A. Mohri, "Feedback Control for Wire-Suspended Mechanism with Exact Linearization," in *International Conference on Intelligent Robots and Systems*, no. October, 2002, pp. 2213–2218.

[20] J. Yu, F. L. Lewis, and T. Huang, "Nonlinear Feedback Control of a Gantry Crane," no. June, 1995.

[21] C. E. García, D. M. Prett, and M. Morari, "Model Predictive Control: Theory and Practice - A Survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[22] R. Findeisen and F. Allgower, "An Introduction to Nonlinear Predictive Control," in *21st Benelux Meeting on Systems and Control*, 2002, pp. 1–23.

[23] D. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=57020

[24] A. G. Wills and W. P. Heath, "Barrier function based model predictive control," *Automatica*, vol. 40, no. 8, pp. 1415–1422, Aug. 2004.

[25] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 2, pp. 267–278, 2010.

[26] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control systems technology," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[27] G. Naus, J. P. M. V. de Molengraft, W. Heemels, and M. Steinbuch, "Design and implementation of parametrized adaptive cruise control: An explicit model predictive control approach," *Control Engineering Practice*, vol. 18, no. 8, pp. 882–892, 2010.

[28] H. Takeuchi, "Real time optimization for robot control using receding horizon control with equal constraint," *Journal of Robotic Systems*, vol. 20, no. 1, pp. 3–13, 2003.

[29] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[30] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 50, no. 5, pp. 674–678, 2005.

[31] A. Bemporad, "Reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 415–419, March 1998.

[32] U. Kalabic, I. V. Kolmanovsky, and E. G. Gilbert, "Reduced order extended command governor," *Automatica*, vol. 50, no. 5, pp. 1466–1472, 2014.

[33] E. Asarin, O. Bournez, T. Dang, and O. Maler, "Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems," *Lecture Notes in Computer Science, Hybrid Systems: Computation and Control*, vol. 1790, pp. 20–31, 2000.

[34] J. Lygeros, G. Pappas, and S. Sastry, "An introduction to hybrid systems modeling, analysis and control," in *In Preprints of the First Nonlinear Control Network Pedagogical School*, 1999, pp. 307–329.

[35] I. Mitchell, a. Bayen, and C. Tomlin, "A time-dependent Hamiliton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, pp. 947–957, 2005.

[36] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," *IEEE Conference on Decision and Control*, 2008.

[37] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control*, 2004.

[38] R. Wisniewski and C. Sloth, "Converse barrier certificate theorem," in *IEEE Conference on Decision and Control*, 2013.

[39] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov Functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, Apr. 2009.

[40] K. P. Tee and S. S. Ge, "Control of nonlinear systems with full state constraint using a Barrier Lyapunov Function," *IEEE Conference on Decision and Control*, pp. 8618–8623, Dec. 2009.

[41] Y. Kim, M. Mesbahi, G. Singh, and F. Y. Hadaegh, "On the Convex Parameterization of Constrained Spacecraft Reorientation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp. 1097–1109, Jul. 2010.

[42] Y. Kim and M. Mesbahi, "Quadratically Constrained Attitude Control via Semidefinite Programming," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 731–735, May 2004.

[43] U. Lee and M. Mesbahi, "Spacecraft reorientation in presence of attitude constraints vs logarithmic barrier potentials," in *American Control Conference*, San Francisco, June 2011.

[44] A. Weiss, F. Leve, M. Baldwin, J. Forbes, and I. Kolmanovsky, "Spacecraft constrained attitude control using positively invariant sets on $so(3) \times r^3$," in *American Control Conference*, 2014, pp. 4955–4960.

[45] P. Wieland and F. Allgower, "Constructive safety using control barrier functions," in *Nonlinear Control Systems*, vol. 7, no. 1, 2007, pp. 462–467.

[46] A. Ames, J. Grizzle, and P. Tabuada, "Control Barrier Function based Quadratic Programs with Application to Adaptive Cruise Control," *ames.tamu.edu*, 2014. [Online]. Available: http://ames.tamu.edu/CLF_QP_ACC_final.pdf

[47] E. Sontag and H. J. Sussmann, "Nonsmooth control-lyapunov functions," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 3. IEEE, 1995, pp. 2799–2805.

[48] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs," *IEEE Access*, to appear, 2015.

[49] M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using control lyapunov-barrier function," *Automatica*, vol. 66, pp. 39–47, 2016.

[50] N. A. Chaturvedi, A. K. Sanyal, and H. McClamroch, "Rigid-body attitude control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 30–51, June 2011.

[51] F. Bullo and R. M. Murray, "Tracking for fully actuated mechanical systems: a geometric framework," *Automatica*, vol. 35, no. 1, pp. 17–34, Jan. 1999.

[52] T. Lee, M. Leok, and N. H. McClamroch, "Geometric Tracking Control of a Quadrotor UAV on SE (3)," in *IEEE Conference on Decision and Control*, no. 3, Atlanta, GA, 2010, pp. 5420–5425.

[53] ——, "Stable manifolds of saddle equilibria for pendulum dynamics on S2 and SO(3)," in *IEEE Conference on Decision and Control and European Control Conference*, no. 3. Orlando, FL: Ieee, Dec. 2011, pp. 3915–3921. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6160530

[54] 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 (CDC)*, Florence, Italy, Dec. 2013, pp. 5510–5515.

[55] U. V. Kalabic, R. Gupta, S. D. Cairano, A. M. Bloch, and I. V. Kolmanovsky, "Constrained spacecraft attitude control on so(3) using reference governors and nonlinear model predictive control," in *American Control Conference*, Portland, Oregon, 2014, pp. 5586–5593.

[56] R. Gupta, U. V. Kalabic, S. D. Cairano, A. M. Bloch, and I. V. Kolmanovsky, "Constrained spacecraft attitude control on so(3) using fast nonlinear model predictive control," in *American Control Conference*, 2015, pp. 2980–2986.

[57] W. Singhose, "Command shaping for flexible systems: A review of the first 50 years," *International Journal of Precision Engineering and Manufacturing*, vol. 10, no. 4, pp. 153–168, 2009.

[58] A. Bemporad, "Reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 415–419, 1998.

[59] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 6271–6278.

[60] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control lyapunov and control barrier functions for systems evolving on manifolds," in *American Control Conference (ACC)*, Chicago, IL, Jul. 2015, pp. 2038–2044.

[61] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall New Jersey, 1996, vol. 3.

[62] T. Lee, M. Leok, and N. McClamroch, "Control of complex maneuvers for a quadrotor uav using geometric methods on $se(3)$," *Asian Journal of Control*, vol. 15, no. 3, pp. 1–18, 2013.

[63] J. Milnor, *Morse Theory*. Princeton: Princeton University Press, 1963.

[64] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*. New York-Heidelberg-Berlin: Springer-Verlag, 2004.

[65] T. Lee, M. Leok, and N. H. McClamroch, "Discrete Control Systems," *Springer Encyclopedia of Complexity and Systems Science*, pp. 2002—-2019, 2008.

[66] T. Lee, M. Leok, and N. McClamroch, "Stable manifolds of saddle equilibria for pendulum dynamics on s^2 and $so(3)$," in *IEEE Conference on Decision and Control and European Control Conference*, December 2011, pp. 3915–3921.

[67] ——, "Dynamics and control of a chain pendulum on a cart," in *IEEE Conference on Decision and Control*, 2012, pp. 2502–2508.

[68] T. Lee, M. Leok, and N. H. McClamroch, "Stable manifolds of saddle equilibria for pendulum dynamics on s 2 and so (3)," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 3915–3921.

[69] J. Marsden and T. Ratiu, *Introduction to Mechanics and Symmetry*, 2nd ed. Springer Verlag Press, 2002.

[70] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*. New York-Heidelberg-Berlin: Springer-Verlag, 2004.

[71] A. Ilchmann and M. Mueller, "Time-varying linear systems: Relative degree and normal form," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 840–851, May 2007.

[72] P. Brunovsky, "A classification of linear controllable systems," *Kybernetika*, vol. 6, no. 3, pp. 173–188, 1970.

[73] R. V. Til and W. E. Schmitendorf, "Constrained controllability of discrete-time systems with additive disturbances," in *IEEE Conference on Decision and Control*, Fort Lauderdale, FL, December 1985, pp. 874–875.

[74] G. Peichl and W. Schappacher, "Constrained controllability in banach spaces," *SIAM Journal on Control and Optimization*, vol. 24, no. 6, 1986.

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

[76] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonliear systems: Introductory theory and examples," CAS, Internal Report A-284, January 1994.

[77] 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 (CDC)*, Florence, Italy, December 2013, pp. 5510–5515.

[78] 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 *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4888–4895.

[79] 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 (CDC)*, Florence, Italy, December 2013, pp. 5510–5515.

[80] F. Morgan, *Geometric measure theory: a beginner's guide*. Academic press, 2008.

[81] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," in *Analysis and Design of Hybrid Systems*, Atlanta, GA, October 2015, pp. 54–61.

[82] B. Morris, M. J. Powell, and A. D. Ames, "Sufficient conditions for the Lipschitz continuity of QP-based multi-objective control of humanoid robots," *IEEE Conference on Decision and Control*, pp. 2920–2926, Dec. 2013.

[83] G. Wu and K. Sreenath, "Variation-based linearization of nonlinear systems evolving on and," *IEEE Access*, vol. 3, pp. 1592–1604, 2015.

[84] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.