UNIVERSITY OF CALIFORNIA

Los Angeles

Independent Position and Attitude Control on Multirotor Aerial Platforms

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mechanical Engineering

by

Lecheng Ruan

2020

ABSTRACT OF THE DISSERTATION

Independent Position and Attitude Control on Multirotor Aerial Platforms

by

Lecheng Ruan
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2020
Professor Tsu-Chin Tsao, Chair

Multirotor aerial platforms have obtained growing attentions in industry and academia, for its simplicity in mechanical structure, agility in maneuverability and ability for vertical take-off and landing (VTOL). Conventional multirotor has underactuated dynamics, and can not be fully controlled in 6 Degree-of-Freedom (DoF). In fact, only its three-dimensional position and yaw angle, called the flat outputs, can be controlled independently. However, for certain applications, such as perching on a vertical vertical wall or flying in a narrow space, the the non-flat outputs, the roll and pitch angles, are independently specified from the position requirements at some particular time. These tasks require the independent control of position and attitude at least partially for certain instants, and are generally challenging for multirotor platforms.

This dissertation addresses this issue in two aspects. Firstly, an algorithm is designed for the conventional quadcopter platforms to generate trajectories for tasks with requirements on both position and attitude. It is formulated as an optimization, and converted into a series of convex problems to solve. Constraints on dynamics, space limitations, inputs and states are explicitly included. The algorithm is verified numerically on the task of quadcopter perching at the specified location on a vertical wall.

Secondly, a fully actuated multirotor aerial platform is proposed. Commercial quadcopters and passive hinges are used to generate tiltable thrust vectors during flight. This platform has a salient feature for mechanical simplicity, as it does not require additional ac-

tuators to control the directions of thrust vectors. A controller for the proposed multirotor platform is designed to enable independent control of position and attitude.

The proposed multirotor platform has overactuation in dynamics, which renders a redundancy of 2 DoF for inputs. A new controller is proposed, under which the input allocation scheme searches within this redundancy for smaller thrust forces required to hover at different attitudes. The range of achievable attitudes is enlarged under this new scheme compared with the previously proposed controller, under the same thrust saturation limit for the platform actuators.

These controllers are validated with both simulation and experiments and demonstrated by the proposed multirotor aerial platform hovering at non-horizontal attitudes, or tracking independent trajectories for position and attitude simultaneously.

The dissertation of Lecheng Ruan is approved.

James Steven Gibson

Tetsuya Iwasaki

Lieven Vandenberghe

Tsu-Chin Tsao, Committee Chair

University of California, Los Angeles

2020

To my parents and grandparents,
my teachers and friends,
the ones I love and love me . . .
Life is becoming so charming
for your presence.

Table of Contents

List of Figures

List of Tables

Acknowledgments

My first and greatest appreciation shall be of no doubt credited to my advisor, Professor Tsu-Chin Tsao, for his guidance, support, understanding and patience during my Ph.D. career. I learned so much from him in the past half decade, not only repetitive control and ILC, but to be creative in thinking, brave on attempting, serious to commitments, passionate to work, persistent in belief and nice to people. I am deeply honored and grateful to be his student.

I am very experienced of being a student. Among this long journey for over two decades, I feel so grateful for all my teachers, in elementary school, high school, HIT and UCLA. I would particularly appreciate Professor Song-Chun Zhu, Professor Xiuheng Wang, Professor Tetsuya Iwasaki and Professor Jason Speyer, for showing me the fantasies of this world.

I want to give special credit to Mr. Han Wang, my buddy and roommate, for all the happy moments we shared: the interesting chats we had, the workout class at night, the restaurant hunting in LA, and so many interesting stories with Ferrari the big dog.

I am very grateful for my close friends and collaborators, Mr. Yao Su, Mr. Pengkang Yu and Mr. Chen-Huan Pi, for their assistance and dedication in our research project, in addition to the kindness and brightness they brought to my life.

I would also like to thank all other members in UCLA MacLab, especially those senior ones, Dr. James Simonelli, Dr. Sandeep Rai, Dr. Niloufar Esfandi, Dr. Cheng-Wei Chen, Dr. Yu-Hsiu Lee and Dr. Matt Gerber, for their help, suggestions and supports through my study at UCLA.

I am so lucky to meet so many excellent friends, in China or in Los Angeles, in school or at work, for table tennis, travelling or creating cool stuffs together. I would particularly appreciate Dr. Hanzhang Pei, Mr. Chenxi Guo, Mr. He Mi, Mr. Xinyu Shi, all my buddies at HIT and UCLA, at HICA, PlusYoou and UCLA Table Tennis Team, for writing up an incredible chapter in my life. I feel not alone because of all of you.

The last paragraph is reserved for the most important people in my life: my family, my

mom, my dad and my grandparents. No matter how far I travel, I know you are always by my side. This is the source of all my love, passion and inspiration in life.

Vita

2017 – present	Ph.D. Candidate, Mechanical Engineering, University of California, Los Angeles.
2016 – 2018	Teaching Assistant/Associate/Fellow, University of California, Los Angeles.
2015 – 2017	Ph.D. Student, Mechanical Engineering, University of California, Los Angeles.
2011 – 2015	B.E. with Honor, Mechanical Design, Manufacture and Automation, Harbin Institute of Technology.

CHAPTER 1

Introduction

1.1 Unmanned Aerial Vehicles

An Unmanned Aerial Vehicle (UAV) is defined as a powered aerial platform which does not have a human pilot onboard. It usually flies automatically or is remotely piloted. It is obtaining growing attentions in industry and academia, and has been widely applied in various fields, including agriculture [TVM16], fire suppression [HSS18], surveillance [GSW18], search and rescue[NKO17], delivery [CKC18], exploration [NJL16], mobile sensor network [ZZZ17] and photography. High demands and increasing diversity in UAV market has pushed its development and extension in different shapes, sizes and functionalities.

Popular UAV configurations include fixed-wing, helicopter and multirotor. Fixed-wing UAV ([RTL18][PWM17][SSL17]) occupies a large portion of the current market, for its advantages of dynamic stability, high reliability, long fight duration and large load capacity. However, the long distance required for take-off and landing, in addition to its low maneuverability during flight, constrains its operation mostly within open-area, non-stopping scenarios.

Helicopter UAV ([OOP16][KPJ16][WZT18]) has the abilities of vertical take-off and landing(VTOL), hovering, in addition to high agility for maneuvering. While these features enable its operations in narrow space, the size and cost of helicopter UAV is hard to reduce due to its high sophistication in actuation mechanism.

Multirotor UAV is actuated by the thrusts forces generated from spinning propellers, the magnitudes of which are controlled directly with spinning speeds. This configuration inherits the functionality of VTOL, hovering and high maneuverability from helicopter, but

has much simpler mechanism and thus more possibility for smaller size and lower cost.

1.2 Underactuation of Multirotor

Multirotor UAV has advantages in tasks which require agile operations in complicated environments. The mechanical simplicity makes it easy to build and maintain. Intensive investigations have been conducted on its control and analysis, as surveyed in [MKC12] and [HHM13]. Applications of multirotor UAV include aerial manipulation, inspection, mobile network, delivery and rescue ([SNT19][LST18][KPL20][SKB16][AN19]).

Conventional multirotor is lifted with thrust forces against the air, which are generated by the spinning propeller-motor actuators. The thrust forces are usually aligned parallelly upwards to efficiently compensate for gravity. However, under this configuration, the actuators can only provide the total thrust force at one fixed direction for one specific pose. To provide thrust force in 3D space, the multirotor has to present different attitude. This is called the underactuation of multirotor dynamics, which shows that its position and attitude can not be controlled independently simultaneously. The motion outputs of multirotor usually consist of a 3D position vector and a set of Euler angles, which are usually named roll, pitch and yaw angles, to describe the orientation. In fact, it has been proved in [MK11] that multirotor dynamics is differential flat. Only a subset of the motion outputs (the 3D position and yaw angle, called the flat outputs) are able to be controlled independently.

However, not all applications can be satisfied with only specifications on the flat outputs. For certain tasks, such as perching on a vertical wall [MMK12], flying in a narrow space [PBH20] or interacting with the environment [RCS19], the the non-flat outputs, the roll and pitch angles, are independently specified from the position constraints at some particular time.

These tasks require the independent control of position and attitude on the multirotor, at least at some certain time. There has been some works addressing this issue, either by designing specific control or trajectory planning algorithms, or making mechanical modifications.

1.3 Previous Works

For tasks that require independent control of position and attitude of the multirotor aerial platform, two categories of solutions were proposed in the previous works. One designs control or trajectory planning algorithms under task requirements for the conventional multirotor platforms. The other made mechanical modifications to the platforms.

1.3.1 Solutions with Algorithm Design

This category of solutions design control or trajectory planning algorithms on conventional multirotors, to meet independent requirements of position and attitude. Without the modifications on mechanical structure, these solutions are more general to platforms and quicker to realize in implementation.

It has been indicated in [MK11] that specifying position and attitude independently for the entire flight is impractical. However, if the non-flat outputs are only partially designated, it is possible to search for a feasible trajectory whose position path complies with the constraints of roll and pitch angles. It should be noticed that the algorithms do not solve the underactuation problem and achieve independent control of position and attitude. In reality, they only seek for trajectories in accordance with requirements on both in the underactuated dynamical space of multirotors.

Not many works have successfully addressed the underactuation problem in this direction. One representative work was elaborated in [MMK12]. As fully actuated control is not applicable in principle, this work designed five controllers, each of which only regulated a subset of the outputs. The entire trajectory was initialized by manually design a sequence of these controllers with proper triggers for switching. An updating policy was developed to refine the controller parameters with errors. This algorithm has been demonstrated with a few scenarios with requirements on both position and attitude. However, the trajectory generation process involves intensive human efforts and intelligence, and is difficult to extend to different tasks. The constraints of states and inputs can not be included in the planning process and are only considered in the errors when refining the parameters.

Therefore, an algorithm which can automatically generate trajectory under explicit constraints can greatly reduce the design burden and increase the generality, and is thus desired.

1.3.2 Solutions with Mechanical Modification

This category of solutions make mechanical modifications based on conventional multirotor platforms. The new platforms directly increase the actuation capability of system dynamics in certain sense, thus attenuate the underactuation in the aforementioned tasks.

Generally, the modifications are made either by installing add-on mechanisms, or designing new structures. The first type usually focuses on one specific task, such as perching on a surface in [KMR15], [PJW18], [DM19] and [HLS19]. Additional forces are provided by the interaction of add-on mechanisms with the environment. While simple and robust for one specific scenario, these modifications are not flexible for different tasks.

The second type attempts to design mechanical structures with fully actuated dynamics. The conventional multirotors can only provide total thrust force at one certain direction without changing the attitude. These modifications use different approaches to change the direction of the total thrust force. One group of platforms deploy propeller-motor actuators at diverse orientations, as shown in [JVC18], [RRB15], [RMP17] and [PLA18]. The total thrust force is the join force with components in different angles, so its direction can be changed by the manipulation of component magnitudes. These modifications maintains the simple mechanical structure of multirotors, but generally have larger size due to the additional propeller-motor actuators required for full actuation. Orientations of these actuators also remains a challenging problem in research, and usually need to be changed for different tasks for better efficiency.

The other group of structural modifications change the orientations of the propeller-motor actuators actively, thus the direction of thrust forces, during the flight, as shown in [RBG14], [KVE18], [GT18] and [RBF16]. Additional actuators and mechanisms are required for this online thrust vectoring. While able to achieve full actuation with less propeller-motor actuators and flexible to different tasks compared with the previous group of platforms, this

configuration has much more complicated mechanical structure.

Therefore, a fully actuated multirotor platform with mechanical simplicity and flexibility regarding various scenarios is desired.

1.4 Dissertation Organization

This dissertation addresses the underactuation problem in both directions mentioned in the previous section. The goal is to find multirotor solutions for tasks where both position and attitude are specified.

A solution with algorithm design is proposed in Chapter 2. The overall trajectory planning process of conventional quadcopters under position and attitude requirements is formulated as an optimization problem, with the inclusion of explicit state or input constraints. It is then converted into a series of convex problems for efficient and deterministic calculation. The algorithm is applied in a numerical example, where the quadcopter is required to perch at a specific location on the vertical wall. The planned trajectory demonstrates that it meets all the constraints in position and attitude.

A solution with mechanical modification is proposed in Chapter 3. It creates a mechanical design procedure, using commercial quadcopters and passive hinges to construct multirotor platforms with the ability of active thrust vectoring during flight. A prototype with four tiltable thrusting actuators is constructed for demonstration. Its full actuation is dynamics is derived.

Chapter 4 designs a control architecture to realize independent control of position and attitude for the proposed platform. The ability is demonstrated in different flight cases with simulation and experiments.

Chapter 5 analyzes the achievable attitude range of the proposed platform with saturation consideration on quadcopter thrust forces. The overactuation in dynamics renders a redundancy of 2 DoF in space to be explored. Therefore, a controller, which searches for smaller thrust forces to lift and actuate the platform, is proposed. Simulation and experi-

ments verify that the achievable attitude range is enlarged by this controller compared with Chapter 4.

The works of this dissertation are concluded in Chapter 6.

CHAPTER 2

A Convexified Trajectory Planning Algorithm for Quadcopter with Position and Attitude Requirements

2.1 Introduction

Among multirotor aerial platforms, quadcopter is one of the most popular configurations, as it is able to achieve smaller size compared with hexacopter ([BAB17][LPW17]) and octocopter ([BSF18][WHF17]) aerial platforms.

The trajectory planning techniques of quadcopter platforms have been heavily investigated in previous works. The underactuation of quadcopter dynamics makes it impractical to plan for trajectories for all six motion outputs. Generally, planning algorithms mainly focus on the position trajectories, which are constrained with both spacial and dynamical constraints. One typical planning process consists of two steps. At first, a geometric position path is generated under spatial constraints, and parameterized with time. The time scale is then updated to ensure dynamical feasibility. Examples can be found in [CYW07] and [HWT08].

It was proved in [MK11] that the quadcopter dynamics is differential flat. Its flat outputs are the 3D position and the yaw angle. Therefore, the quadcopter dynamics can be rewritten in the flat outputs and their higher order derivatives. Another set of algorithms make use of this property, and formulate an optimization problem under flat dynamics. Spatial and dynamical constraints are converted into the flat space. The trajectories of four flat outputs can be planned in these algorithms, such as [HD11], [NPL20] and [TK20].

However, if the non-flat outputs, the roll and pitch angles, also need to be specified while

planning for the flat outputs, the aforementioned algorithms do not work. One representative application under this category is to drive a quadcopter to perch at a specific location on a vertical wall. Obviously, for this case, the terminal point of the position trajectory is fixed at that required location, and the roll/pitch angle at this point is also determined as the quadcopter has to perch with attitude parallel to the surface of the wall. An algorithm was proposed in [MMK12] regarding the vertical wall perching task. Given the underactuation in quadcopter dynamics, five controllers were defined, each of which only controlled either the position or the attitude of the quadcopter. The entire perching trajectory was manually composed by a sequence of these controllers with proper triggers defined as events or time instants to switch from one control phase to the next. The process is simulated, and the controller parameters were updated based on the regulation errors. This algorithm demonstrated the success in vertical wall perching. However, the initialization of trajectory needs to be manually designed, and there are no general rules for this process. The state and input constraints can not be explicitly included in the planning process.

This chapter proposes a novel trajectory planning algorithm for the conventional quadcopter platform, so that requirements on both position and attitude can be specified. The algorithm is in an optimization form. Techniques are applied to rewrite it as a series of convex problems, and thus can be efficiently solved with existing convex solvers such as CVX package ([GB14][GB08]). Constraints on states and inputs are explicitly included in the optimizations. The algorithm is demonstrated by generating a trajectory, under which the quadcopter perches t a certain location on a vertical wall.

The rest of this chapter is organized as following. Section 2.1 reviews the conventional quadcopter platform and its dynamics. Section 2.3 describes the perching task and shows how it is formulated into optimization format. Section 2.4 addresses the non-convexity in the formulation, and proposes techniques to convert it into a series of convex problems. The algorithm is is elaborated in details in 2.5 to plan for a trajectory for the perching problem. Section 2.6 concludes this chapter.

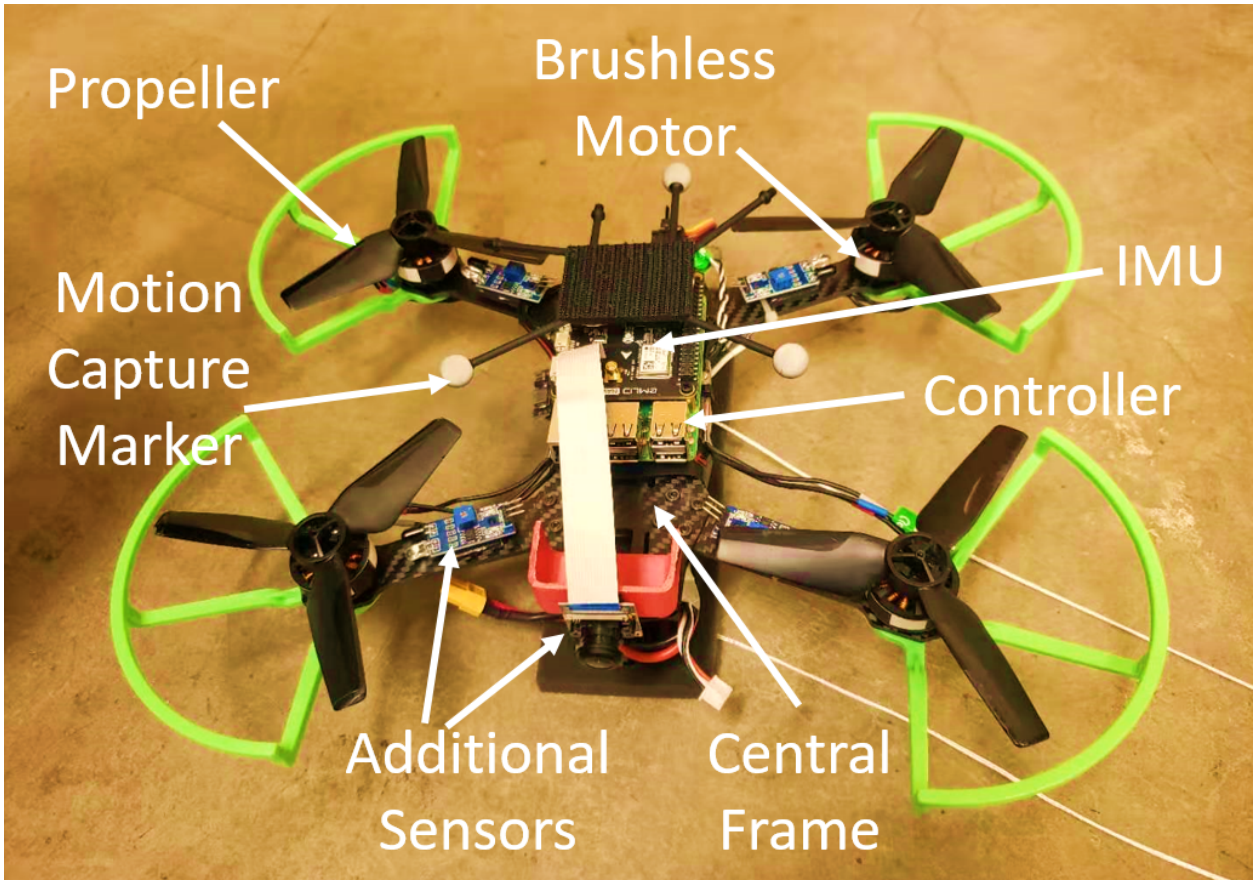


Figure 2.1: A conventional quadcopter platform built in UCLA MacLab.

2.2 Quadcopter

2.2.1 Platform

Figure 2.1 shows the picture of a conventional quadcopter platform. This quadcopter was built in UCLA MacLab. The quadcopter is mainly composed of the central frame, the propeller-motor actuators and the electronic system which generally includes the control board, sensors, motor drivers and battery. The central frame is the picture is made of carbon fiber. Brushless DC motors are utilized to reduce friction and heating at high spinning speed, and are driven with electronic speed controllers (ESCs) for commutation and speed regulation. The inertial measurement unit (IMU) is the major sensor onboard that measures the accelerations and angular velocities of the quadcopter during flight. The data are processed in the controller and usually fused with external positioning systems such as motion

capture systems for indoor applications and global positioning system (GPS) module with barometer for outdoor uses. Additional sensors can also be deployed for specific usages. For example, the picture shows a camera for visual measurement and four laser sensors to measure the spinning speed of the motors. The quadcopter is generally arranged in a symmetric manner for both geometry and mass distribution.

2.2.2 Actuators

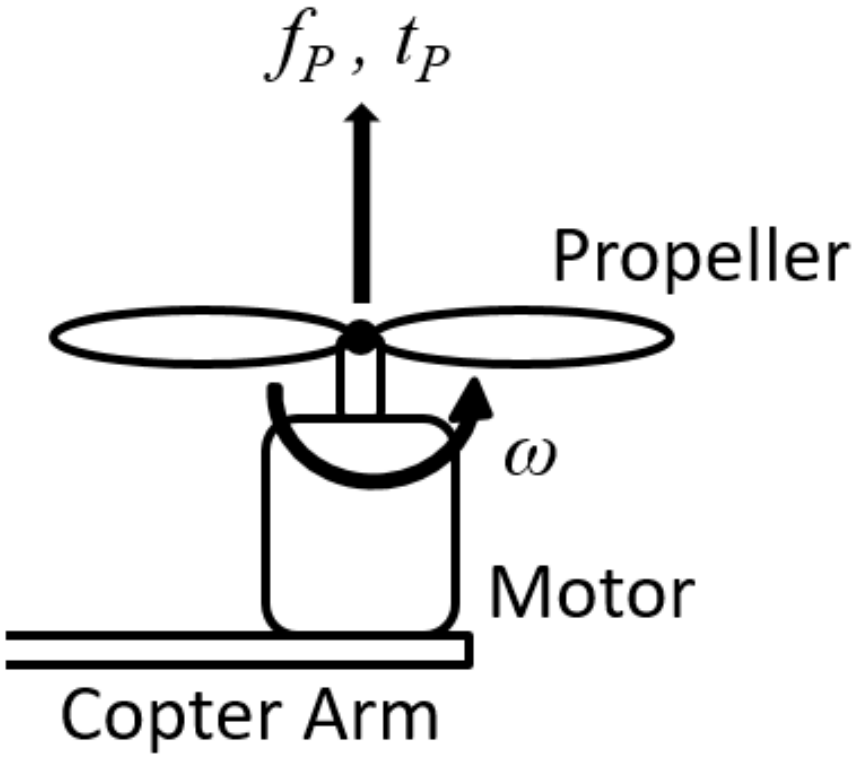


Figure 2.2: Sketch of a typical propeller motor actuator set.

The external thrust forces and torques to drive the quadcopter is generated with propeller-motor actuators, the sketch of which is shown in Figure 2.2. The spinning propeller provides a thrust force f_P and a drag torque t_P, both along the normal direction of the spinning plane. When the propeller pitch angle is fixed, their magnitudes are directly determined by

the spinning speed ω as

$$\begin{aligned} f_P &= K_T \omega^2, \\ t_P &= K_\tau \omega^2. \end{aligned} \tag{2.1}$$

Therefore, when the desired thrust force f_P^d or the desired drag torque t_P^d is known, the desired spinning speed of the motor ω^d can be calculated as

$$\omega^d = \sqrt{f_P^d / K_T} = \sqrt{t_P^d / K_\tau}. \tag{2.2}$$

Brushless DC motors are widely applied, with an example shown in Figure 2.1, because they have lower friction, smaller heating effect and wearings, and thus longer lifetime for high-speed and long-term operations in contrast with the brushed DC motors. As mentioned previously, a ESC module is utilized to deal with the commutation and also establish the control loop for motor speed regulation. A Pulse-Width-Modulated (PWM) servo command with specified frequency is sent to the ESC as the reference spinning speed. This servo command is usually constrained within a small portion of the PWM duty cycle range, and represents a certain reference signal through a one-to-one mapping, shown as

$$\omega^d = f(D). \tag{2.3}$$

Here D refers to the PWM servo command. $f(\cdot)$ is the mapping function from PWM servo command to spinning speed reference, which is monotonically increasing, and usually identified experimentally. Due to one-to-one mapping, the inverse function of $f(\cdot)$ exists as

$$D = f^{-1}(\omega^d). \tag{2.4}$$

Therefore, when the desired thrust force f_P^d is known, the required PWM servo command D can be calculated reversely as

$$D = f^{-1}(\sqrt{f_P^d / K_T}). \tag{2.5}$$

The inner dynamics of motors are largely determined by its rotation inertia and the inductance in circuit. As both are relative small and often neglected in quadcopter applications, it is generally assumed that the motor speed can be regulated without dynamics.

Therefore, the spinning speed of the motor ω can be directly calculated as

$$\omega = \omega^d = f(D) = f(f^{-1}(\sqrt{f_P^d/K_T})). \tag{2.6}$$

Finally, by (2.1), the thrust force f_P is determined by the desired thrust force f_P^d as

$$f_P = K_T f^2(f^{-1}(\sqrt{f_P^d/K_T})). \tag{2.7}$$

The same property applies for the propeller drag torque. Therefore, it can be reasonably assumed from (2.7) that the thrust force and drag torque can be directly controlled without inner dynamics.

It should be noticed that although brushless motors are widely applied, brushed motors are also used in scenarios where small size or low cost are required. However, the assumption of direct thrust/drag torque control applies in both cases.

2.2.3 Body Frame and Rotation

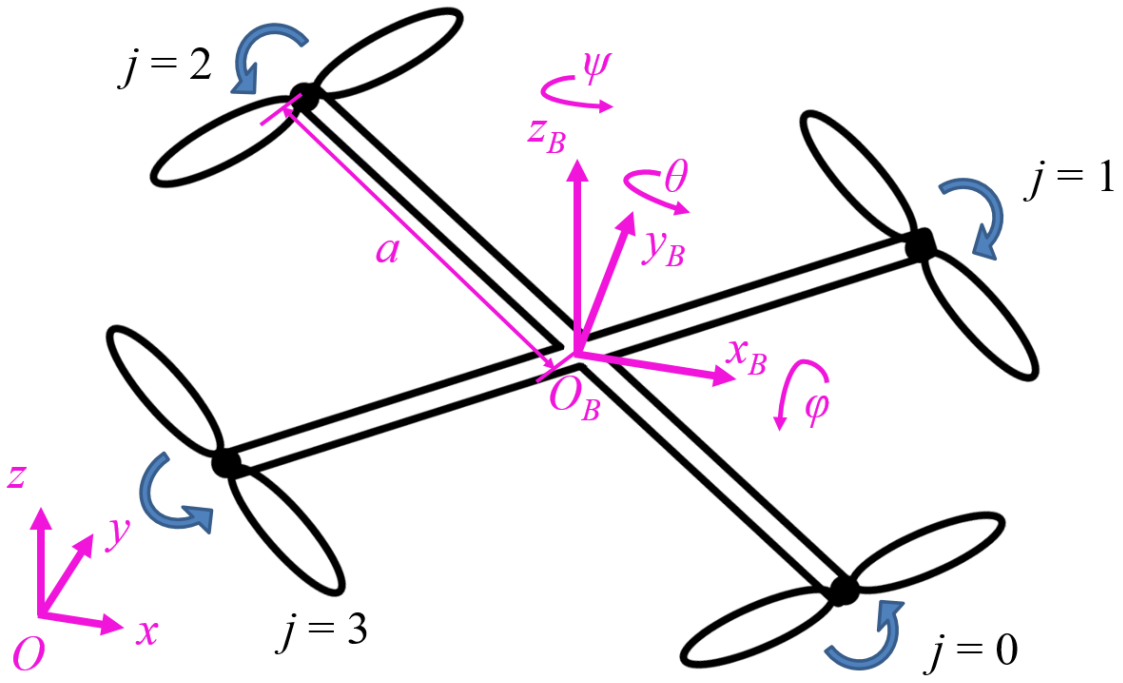


Figure 2.3: Sketch of a conventional quadcopter with coordination system.

The quadcopter body frame is defined on the "X" configuration for flight, as

$$\mathcal{F}^B : \{O_B; \boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}. \tag{2.8}$$

The frame is visualized in Figure 2.3. The four propeller-motor actuators are ordered

$$j = 0, 1, 2, 3 \tag{2.9}$$

respectively. As each propeller-motor actuator generates a thrust force and a drag torque simultaneously while spinning, the four actuators collectively provide four independent inputs for the quadcopter platform as

$$\begin{bmatrix} T \\ M^x \\ M^y \\ M^z \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ -\frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} & -\frac{aK_T}{\sqrt{2}} \\ -\frac{aK_T}{\sqrt{2}} & -\frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} \\ K_\tau & -K_\tau & K_\tau & -K_\tau \end{bmatrix} \begin{bmatrix} \omega_0^2 \\ \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \end{bmatrix}. \tag{2.10}$$

Here the total thrust force T is always along the $\boldsymbol{z}_B$ direction. The torques M^x, M^y and M^z are along $\boldsymbol{x}_B$, $\boldsymbol{y}_B$ and $\boldsymbol{z}_B$ directions respectively.

The body frame is related with the global frame

$$\mathcal{F}^W : \{O; \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\} \tag{2.11}$$

by a rotation, which can be represented with the Euler angle set

$$\boldsymbol{\eta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T, \tag{2.12}$$

representing roll, pitch and yaw angles respectively, as shown in Figure 2.3.

Defining elementary rotation matrices

$$\boldsymbol{R}_x = \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{R}_y = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}, \ \boldsymbol{R}_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\psi & -s\psi \\ 0 & s\psi & c\psi \end{bmatrix}, \tag{2.13}$$

the rotation matrix from body frame to global frame can be calculated as

$${}^W\boldsymbol{R}_B = \boldsymbol{R}_z\boldsymbol{R}_y\boldsymbol{R}_x = \begin{bmatrix} c\theta c\psi & -c\theta s\psi & s\theta \\ c\phi s\psi + s\phi s\theta c\psi & c\phi c\psi - s\phi s\theta s\psi & -s\phi c\theta \\ s\phi s\psi - c\phi s\theta c\psi & s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix}. \tag{2.14}$$

Here

$$s\cdot = \sin\cdot,\ c\cdot = \cos\cdot \tag{2.15}$$

for notation simplicity.

The three axes of the body frame $\mathcal{F}^B$ can be represented by vectors in the world frame as

$${}^W\boldsymbol{R}_B = \begin{bmatrix} \boldsymbol{x}_B & \boldsymbol{y}_B & \boldsymbol{z}_B \end{bmatrix}. \tag{2.16}$$

2.2.4 Equation of Motion

The quadcopter equation of motion is constructed from Newton-Euler method. Define the position of quadcopter center of mass (CoM) as

$$\boldsymbol{\xi} = \begin{bmatrix} x & y & z \end{bmatrix}^T, \tag{2.17}$$

the quadcopter total mass as m and the gravitational acceleration as g, then the quadcopter translational dynamics can be expressed as

$$\ddot{\boldsymbol{\xi}} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m}{}^W\boldsymbol{R}_B \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{T}{m}\boldsymbol{z}_B. \tag{2.18}$$

The rotational dynamic equation is constructed in the body frame $\mathcal{F}^B$. Defining the angular velocity vector in the body frame as

$$\boldsymbol{\nu} = \begin{bmatrix} p & q & r \end{bmatrix}^T \tag{2.19}$$

and the constant inertia matrix of the quadcopter as $\boldsymbol{I}$, the rotation dynamics can be described as

$$\dot{\boldsymbol{\nu}} = \boldsymbol{I}^{-1}(-\boldsymbol{\nu} \times (I\boldsymbol{\nu}) + \boldsymbol{M}), \tag{2.20}$$

where

$$\boldsymbol{M} = \begin{bmatrix} M^x & M^y & M^z \end{bmatrix}^T. \tag{2.21}$$

The angular velocity vector in the body frame $\boldsymbol{\nu}$ can be further related with the derivative of the Euler angle set $\boldsymbol{\eta}$ by transformations, details of which are shown in [Ald95].

2.2.5 Underactuation of Quadcopter Dynamics

Equation (2.10) shows that the conventional quadcopter platform has four independent inputs for motion control. However, there are six motion outputs in 3D space, the position vector $\boldsymbol{\xi}$ and the Euler angle set $\boldsymbol{\eta}$. So the quadcopter is underactuated, as the number of inputs is smaller than the number of outputs. Therefore, the outputs are not fully controllable in 3D space.

Furthermore, from the observation of equation (2.18), it can be concluded that the direction of total thrust force in the translational dynamics is merely determined by the direction of $\boldsymbol{z}_b$. As the vector

$$\boldsymbol{z}_b = \begin{bmatrix} s\theta & -s\phi c\theta & c\phi c\theta \end{bmatrix}^T \tag{2.22}$$

is a function of the roll angle ϕ and pitch angle θ, these two Euler angles are coupled with the translational dynamics, and thus can not be controlled independently without influencing the quadcopter position.

In fact, as mentioned previously, quadcopter has the property of differential flatness, under which only the four flat outputs

$$\boldsymbol{y}_f = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T \tag{2.23}$$

are able to be controlled independently.

2.3 Problem Formulation

This section aims to formulate the vertical wall perching problem as an optimization. The quadcopter dynamics, input and state constraints, terminal conditions and the fly zone are all rewritten as constraints. By solving this optimization, a trajectory that meets all these requirements can be obtained, under the condition of feasibility. The constraints can be separated into dynamical ones and spatial ones, and are elaborated respectively.

2.3.1 Dynamical Constraints

All feasible trajectories must satisfy the quadcopter dynamics in position and attitude. However, as the total dynamics is underactuated and the translational equation are coupled with roll and pitch angles, a two-step approach is proposed, each step of which is fully actuated, or has equal number of inputs and outputs in dynamics.

2.3.1.1 Two-Step Approach for Fully Actuated Dynamics

It can be observed from equation (2.20) that the rotational dynamics itself is fully actuated, and does not have coupling with the quadcopter position, despite underactuation of the whole system. Therefore, feedback linearization can be applied on the quadcopter attitude control to achieve perfect tracking in principle.

The rotational dynamics has three inputs $\boldsymbol{M} \in \mathbb{R}^3$ and three outputs $\boldsymbol{\nu} \in \mathbb{R}^3$. Assign

$$\boldsymbol{M} = \boldsymbol{\nu} \times (I\boldsymbol{\nu}) + I\boldsymbol{M}^v, \tag{2.24}$$

where $\boldsymbol{M}^v$ refers to the virtual input torque vector, then the rotational dynamics can be rewritten as

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}^v. \tag{2.25}$$

Given any desired trajectory of angular velocity $\boldsymbol{\nu}^d$, perfect tracking can be conducted by setting

$$\boldsymbol{M}^v = \dot{\boldsymbol{\nu}}^d. \tag{2.26}$$

Therefore, any trajectory of $\boldsymbol{\nu}$ always has its corresponding trajectory of $\boldsymbol{M}$, thus is always feasible without the consideration of input saturation.

It has been shown that the translational dynamics (2.18) is driven by the total thrust force T, the roll angle ϕ and the pitch angle θ. If these two Euler angles are regarded as inputs, the translational dynamics becomes fully actuated.

Therefore, a two-step approach is designed. In the first step, the position trajectory is planned with inputs T, ϕ and θ. The resulting trajectories of roll and pitch angles are then

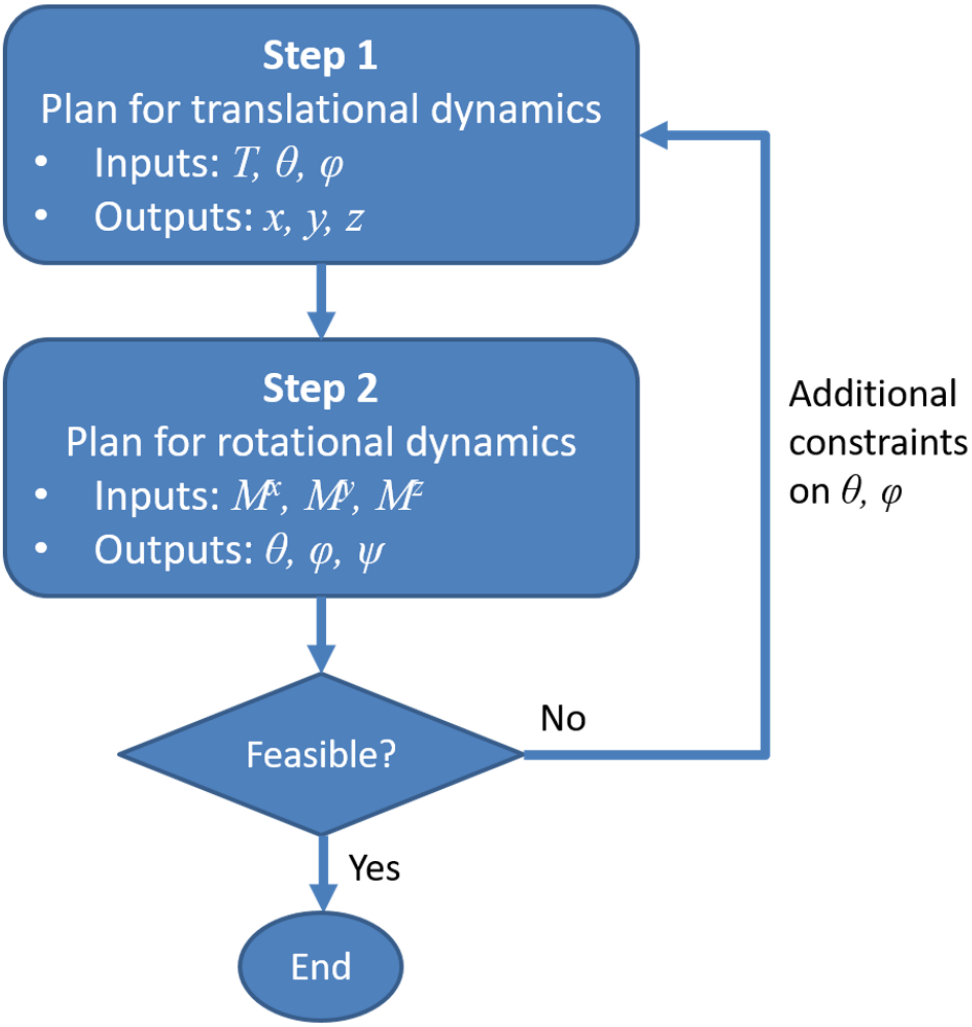


Figure 2.4: The two-step approach for dynamical constraints. It breaks down the quadcopter dynamics into two parts, both of which have full actuation in dynamics, as the number of inputs is equal to the number of outputs.

substituted into the rotational dynamics to test the feasibility of input torque saturation. If the constraints are violated, additional constraints on ϕ and θ are added and the translational planning is redone. If not, the result successfully generates a feasible trajectory for the task. The dynamics for both steps is fully actuated. This process is demonstrated in Figure 2.4.

2.3.1.2 Saturation

The saturation constraints are considered in two aspects. By observation of (2.10), it can be concluded that the all input limitations are essentially determined by the spinning capability of the propellers, as

$$\omega_s^2 \cdot \mathbf{1} \leq \begin{bmatrix} \omega_0^2 \\ \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ -\frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} & -\frac{aK_T}{\sqrt{2}} \\ -\frac{aK_T}{\sqrt{2}} & -\frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} \\ K_\tau & -K_\tau & K_\tau & -K_\tau \end{bmatrix}^{-1} \begin{bmatrix} T \\ M^x \\ M^y \\ M^z \end{bmatrix} \leq \omega_m^2 \cdot \mathbf{1}, \tag{2.27}$$

where ω_s and ω_m refers to the minimum and maximum spinning speeds of the propeller respectively. ω_s can be simply set zero, or tuned as a larger number to improve flight performance.

The trajectories of virtual inputs roll angle ϕ and pitch angle θ in the translational dynamics are implicitly constrained by the limitations of corresponding torque inputs in the rotational dynamics. In order to make these trajectories feasible, the angular velocities and accelerations of these two angles are constrained as an alternative, as

$$\begin{aligned} |\dot{\phi}| &\leq \dot{\phi}_m, \\ |\dot{\theta}| &\leq \dot{\theta}_m, \\ |\ddot{\phi}| &\leq \ddot{\phi}_m, \\ |\ddot{\theta}| &\leq \ddot{\theta}_m, \end{aligned} \tag{2.28}$$

where $\dot{\phi}_m$, $\dot{\theta}_m$, $\ddot{\phi}_m$ and $\ddot{\theta}_m$ are selected maximum values for these variables.

2.3.1.3 Discretization

To make the optimization a finite dimensional problem, the desired trajectory is discretized with respect to time. So is the dynamics. Assume the total time duration of the trajectory is t_f, in which we select $N-1$ time instants of equal distance, then the sampling time can be calculated as

$$t_s = \frac{t_f}{N}. \tag{2.29}$$

Define time instant

$$k \in [0,\ N] \cap \mathbb{Z}, \tag{2.30}$$

then

$$\boldsymbol{x}_d(k) = \boldsymbol{x}_c(k \cdot t_s), \tag{2.31}$$

where $\boldsymbol{x}_d$ and $\boldsymbol{x}_c$ refer to the state variables in discrete time and continuous time respectively.

The dynamics is discretized with Euler's method as

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) + t_s \cdot \dot{\boldsymbol{x}}(k) \tag{2.32}$$

for all state variables.

The higher order derivatives of state variables, such as those shown in (2.28), can be represented by the discrete time state variables, as

$$\begin{aligned} \dot{\phi}(k) &= (\phi(k+1) - \phi(k))/t_s, \\ \dot{\theta}(k) &= (\theta(k+1) - \theta(k))/t_s, \\ \ddot{\phi}(k) &= (\phi(k+2) - 2\phi(k+1) + \phi(k))/t_s^2, \\ \ddot{\theta}(k) &= (\theta(k+2) - 2\theta(k+1) + \theta(k))/t_s^2. \end{aligned} \tag{2.33}$$

2.3.2 Spatial Constraints

The fly zone can be defined with a set of inequalities on quadcopter positions, as

$$\boldsymbol{C}_P(\boldsymbol{\xi}) \leq 0. \tag{2.34}$$

For example, as the real quadcopter is not a point mass, avoiding quadcopter collating the wall in the $\boldsymbol{xOz}$ plane can be expressed as

$$x(k) + a\cos\theta \leq x_{wall}, \tag{2.35}$$

where a is the quadcopter arm length defined in Figure 2.3.

In addition, the position and attitude of initial and final states can be constrained as

$$\begin{aligned} \boldsymbol{\xi}(0) &= \boldsymbol{\xi}_0, \\ \boldsymbol{\eta}(0) &= \boldsymbol{\eta}_0, \\ \boldsymbol{\xi}(N) &= \boldsymbol{\xi}_f, \\ \boldsymbol{\eta}(N) &= \boldsymbol{\eta}_f. \end{aligned} \tag{2.36}$$

2.3.3 Formulation in $\mathbb{SE}(2)$ Space

To better visualize the trajectory planning process, the task of quadcopter perching on a vertical wall is constrained within the $\boldsymbol{xOz}$ space. The dynamics and constraints are therefore formulated in $\mathbb{SE}(2)$. The formulation is elaborated in this section, and shall be utilized for all later sections in this chapter.

In the $\mathbb{SE}(2)$ space, set

$$\begin{aligned} \phi &= 0, \\ \psi &= 0, \end{aligned} \tag{2.37}$$

at all time. The quadcopter dynamics is rewritten as

$$\begin{aligned} \ddot{x} &= (1/m)T\sin\theta, \\ \ddot{z} &= (1/m)T\cos\theta - g, \\ \ddot{\theta} &= (1/I_{yy})M^y, \end{aligned} \tag{2.38}$$

where I_{yy} is the quadcopter inertia along $\boldsymbol{y}_B$ axis, or the $(2,2)$ element of the quadcopter inertia matrix $\boldsymbol{I}$.

The translational dynamics is then discretized as

$$\begin{bmatrix} x(k+1) \\ \dot{x}(k+1) \\ z(k+1) \\ \dot{z}(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ \dot{x}(k) \\ z(k) \\ \dot{z}(k) \end{bmatrix} + t_s \begin{bmatrix} \dot{x}(k) \\ \ddot{x}(k) \\ \dot{z}(k) \\ \ddot{z}(k) \end{bmatrix} = \begin{bmatrix} 1 & t_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ z(k) \\ \dot{z}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ (t_s/m)T(k)\sin\theta(k) \\ 0 \\ (t_s/m)T(k)\cos\theta(k) - g \end{bmatrix}. \tag{2.39}$$

The rotational dynamics is discretized as

$$\begin{bmatrix}\theta(k+1)\\ \dot{\theta}(k+1)\end{bmatrix} = \begin{bmatrix}\theta(k)\\ \dot{\theta}(k)\end{bmatrix} + t_s \begin{bmatrix}\dot{\theta}(k)\\ \ddot{\theta}(k)\end{bmatrix} = \begin{bmatrix}1 & t_s\\ 0 & 1\end{bmatrix} \begin{bmatrix}\theta(k)\\ \dot{\theta}(k)\end{bmatrix} + \begin{bmatrix}0\\ t_s/I_{yy}\end{bmatrix} M^y(k). \tag{2.40}$$

The constraints for position and attitude states, inputs saturation and terminal states are same with those in $\mathbb{SE}(3)$, and are omitted here for brevity.

Specifically, the terminal constraints on pitch angle for perching in $\mathbb{SE}(2)$ can be formulated as

$$\begin{aligned}\theta(0) &= 0,\\ \theta(N) &= -\pi/2.\end{aligned} \tag{2.41}$$

2.4 Convexification

One remaining challenge for the optimization formulated in the previous section lies on its non-convexity. The non-convex problem is highly based on the initialization of variables, and is usually intensive for computation. Therefore, several techniques are proposed in this section to transform it into a series of convex problems.

2.4.1 Standard Constrained Convex Optimization

A constrained optimization formulation generally consists of the cost function, equality and inequality constraints. If it is convex, the cost function and all inequality constraints must be convex, and all equality constraints must be affine. The general form of a constrained convex optimization is shown as

$$\begin{aligned}&\min f(\boldsymbol{x})\\ s.t.\quad &\\ &\boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{0}\\ &\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} = \boldsymbol{0}\end{aligned} \tag{2.42}$$

where $f(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ are convex functions of $\boldsymbol{x}$, $\boldsymbol{A}$ and $\boldsymbol{b}$ are constant matrices/vectors.

2.4.2 Convexity Analysis

It is obvious from (2.39) that the discrete time translational dynamic equation is nonlinear equality constraint with respect to the input and state variables, thus non-convex as affine constraints are linear. The nonlinearity, or equivalently, the non-convexity has two aspects. Firstly, the equation contains nonlinear functions, the triangular functions, of the pitch angle variable $\theta(k)$. Secondly, these functions of $\theta(k)$ are nonlinearly coupled with the thrust force variable $T(k)$.

The inequality constraints are convex with respect to variables $T(k)$, $M^y(k)$, $x(k)$, $z(k)$ and $\theta(k)$.

2.4.3 Convexification of Dynamical Constraints

Based on previous analysis, the convexification of dynamic constraints (2.39) has two major points. Firstly, to eliminate the nonlinear function of $\theta(k)$, two new input variables are defined as

$$\begin{aligned} \hat{u}_s(k) &= \sin\theta(k), \\ \hat{u}_c(k) &= \cos\theta(k), \end{aligned} \tag{2.43}$$

which are naturally constrained by

$$\begin{aligned} -1 &\leq \hat{u}_s(k) \leq 1, \\ -1 &\leq \hat{u}_c(k) \leq 1. \end{aligned} \tag{2.44}$$

However, it should also be noticed that these two inputs are dependent of each other by the coupling equation

$$\hat{u}_s^2(k) + \hat{u}_c^2(k) = 1. \tag{2.45}$$

Secondly, to deal with the nonlinear multiplication of $T(k)$ with the two new inputs $\hat{u}_s(k)$ and $\hat{u}_c(k)$, in addition to the coupling between $\hat{u}_s(k)$ and $\hat{u}_c(k)$ shown in (2.45), a two-stage process is designed, as shown in Figure 2.5.

In stage 1, a pre-known trajectory for thrust force $T(k)$ from either previous optimization or initialization is substituted into the translational dynamics for $x(k)$ or $z(k)$. The dynamic

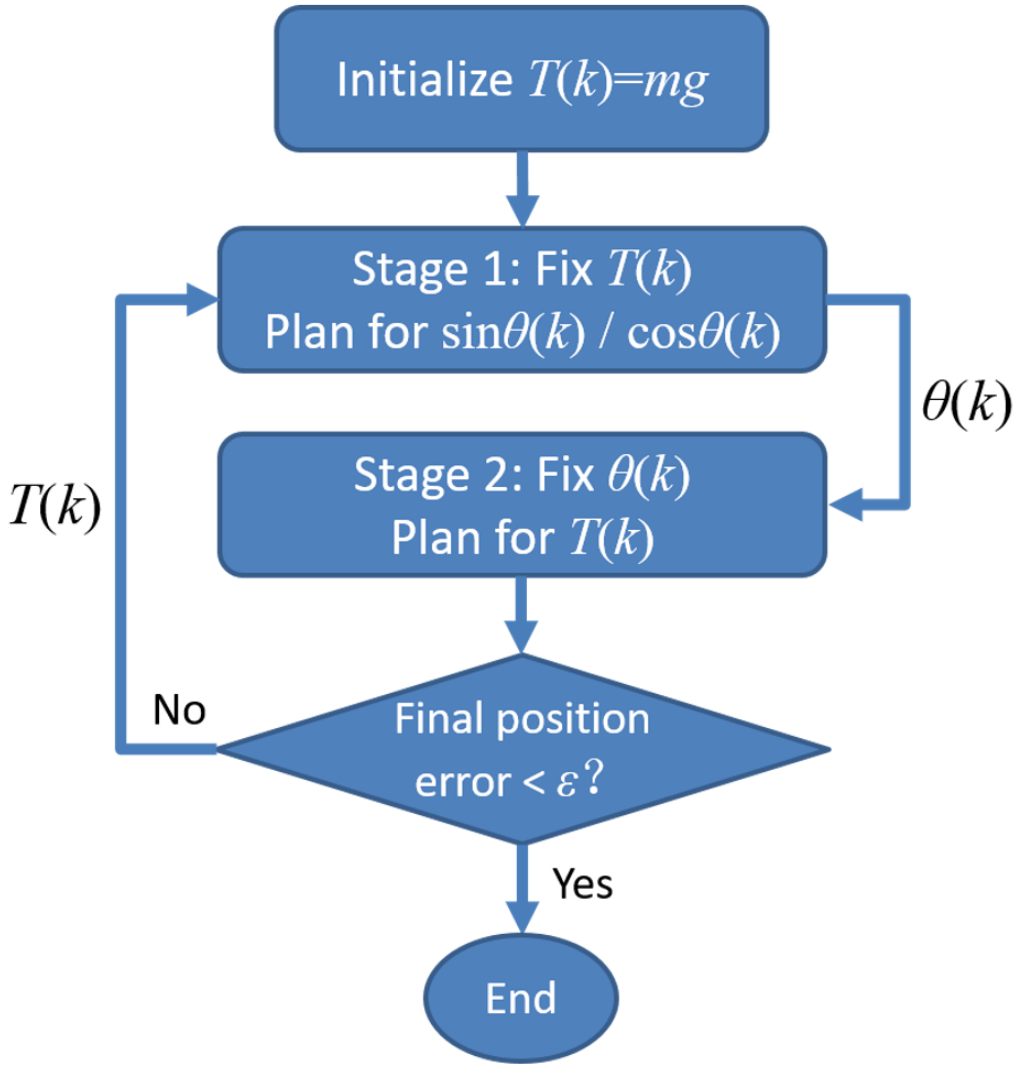


Figure 2.5: The two-stage alternating convexification process to decouple the multiplication of $T(k)$ and functions of $\theta(k)$.

equation thus becomes linear equality constraint with respect to $\hat{u}_s(k)$ or $\hat{u}_c(k)$. For instance, the dynamic equation for $x(k)$ can be written as

$$\begin{bmatrix} x(k+1) \\ \dot{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ (t_s/m)T(k) \end{bmatrix} \hat{u}_s(k), \tag{2.46}$$

which is affine on state and input variables.

Before entering stage 2, a trajectory of $\theta(k)$ has already been available by reverse calculation from either $\hat{u}_s(k)$ or $\hat{u}_c(k)$ trajectories. Therefore, in stage 2, both $\hat{u}_s(k)$ and $\hat{u}_c(k)$ can

be calculated with the trajectory of $\theta(k)$, and substituted into the dynamic equation. The thrust force $T(k)$ is now the solo input. For example, if the $\hat{u}_s(k)$ trajectory was obtained from stage 1, the dynamical constraints can be written as

$$\begin{bmatrix} x(k+1) \\ \dot{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{t_s \hat{u}_s(k)}{m} \end{bmatrix} T(k)$$
$$\begin{bmatrix} z(k+1) \\ \dot{z}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z(k) \\ \dot{z}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{t_s \sqrt{1-\hat{u}_s^2(k)}}{m} \end{bmatrix} T(k) - \begin{bmatrix} 0 \\ g t_s \end{bmatrix}. \tag{2.47}$$

The distance between the trajectory and desired final locations are minimized in stage 2 as the cost function, as

$$\min_{T(k)} d = \sqrt{(x_f - x(N))^2 + (z_f - z(N))^2}. \tag{2.48}$$

Define the error tolerance of perching location as ϵ, then when

$$d > \epsilon, \tag{2.49}$$

the trajectory of $T(k)$ is recorded and substituted into stage one. The whole process is repeated.

When

$$d \leq \epsilon, \tag{2.50}$$

the planned trajectory is considered feasible, and the process is terminated.

2.4.4 Convexification of Constraints on Pitch Angle

Before the convexification of dynamical constraints, all inequality constraints are convex with respect to state and input variables. However, after the change of variables in (2.43), the constraints for higher order derivatives of the Euler angles becomes non-convex.

For example, the angular acceleration of the pitch angle is primarily constrained by

$$|\dot{\theta}(k)| \leq \dot{\theta}_m, \tag{2.51}$$

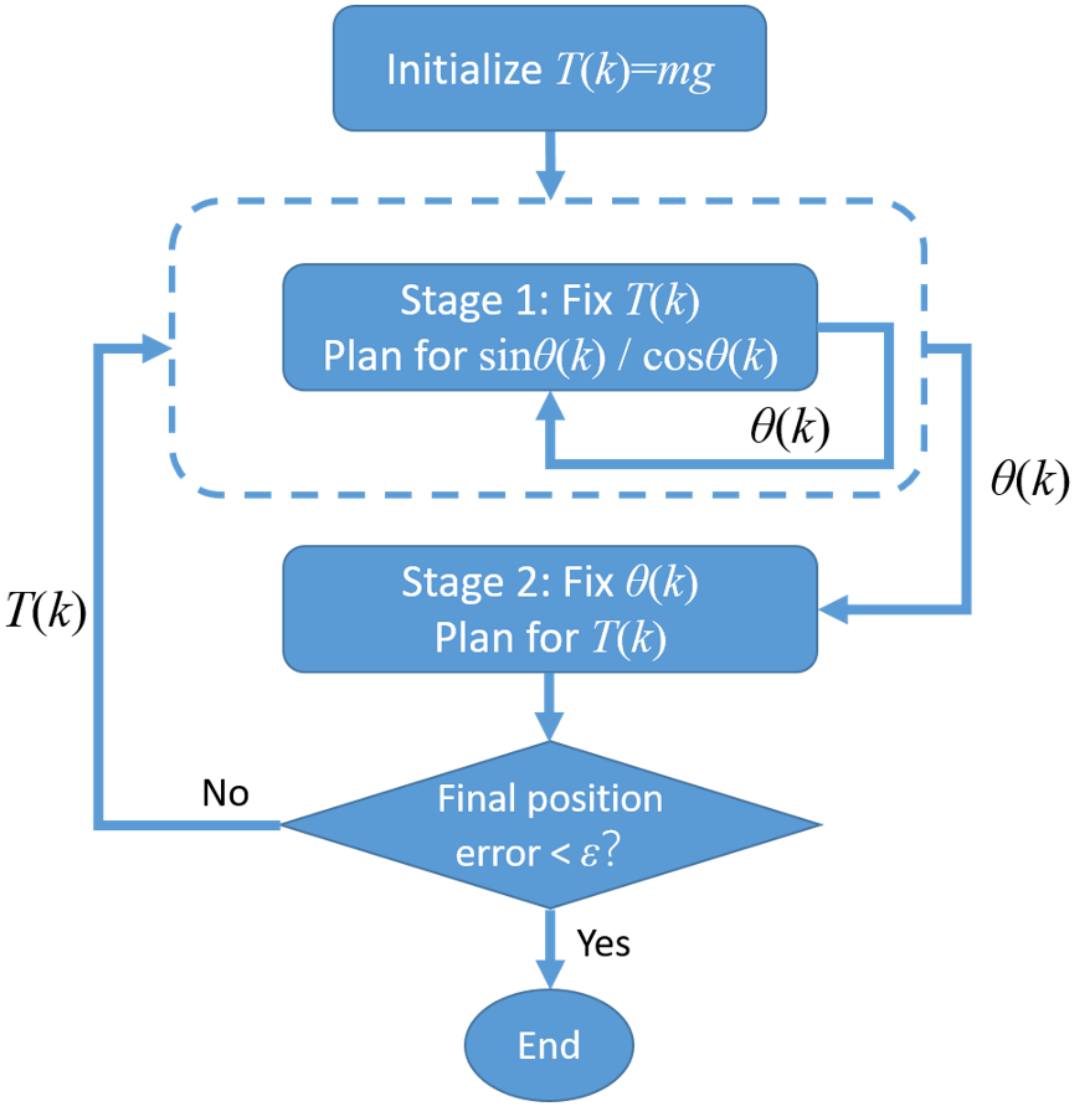


Figure 2.6: The overall convexification process of the trajectory planning optimization. The dashed box refers to the self-iterative process for the convexification of constraints on the angular velocity and acceleration of pitch angle.

which is converted to

$$-t_s \cdot \dot{\theta}_m \leq \theta(k+1) - \theta(k) \leq t_s \cdot \dot{\theta}_m, \tag{2.52}$$

after discretization from (2.33).

However, due to the change of variables in (2.43), the constraint becomes

$$-t_s \cdot \dot{\theta}_m \leq \arcsin \hat{u}_s(k+1) - \arcsin \hat{u}_s(k) \leq t_s \cdot \dot{\theta}_m, \tag{2.53}$$

which is non-convex with respect to the new input variable $\hat{u}_s(k)$.

The convexification is conducted through variations. Make (2.52) as an example. Taking variation of the new input variable $\hat{u}_s(k)$, we obtain

$$\delta\hat{u}_s(k) = \delta(\sin\theta(k)) = \cos\theta(k)\delta\theta(k). \tag{2.54}$$

In discrete time , the variation of $\hat{u}_s(k)$ can be approximated with

$$\begin{aligned} \delta\theta(k) &\approx \theta(k+1) - \theta(k), \\ \delta\hat{u}_s(k) &\approx \hat{u}_s(k+1) - \hat{u}_s(k). \end{aligned} \tag{2.55}$$

Therefore, the inequality constraint (2.52) can be reformulated as

$$-t_s \cdot \dot{\theta}_m \cdot \cos\theta(k) \leq \hat{u}_s(k+1) - \hat{u}_s(k) \leq t_s \cdot \dot{\theta}_m \cdot \cos\theta(k). \tag{2.56}$$

If $\cos\theta(k)$ is known, the constraint (2.56) is convex with respect to $\hat{u}_s(k)$. In practice, this can be realized by self-iteration of stage 1, as shown in Figure 2.6. The trajectory of $\theta(k)$ in the previous iteration is used to calculate for the $\cos\theta(k)$ sequence in the new iteration for the constraint (2.56).

The case for $\hat{u}_c$ is identical to $\hat{u}_s$ and is omitted here for brevity. The angular acceleration of the pitch angle can be refined using the same iterative process with the second-order variation

$$\delta^2\hat{u}_s(k) = \delta^2(\sin\theta(k)) = \cos\theta(k)\delta^2\theta(k) - \sin\theta(k)(\delta\theta(k))^2. \tag{2.57}$$

The term $\sin\theta(k)(\delta\theta(k))^2$ is usually very small and can be ignored for simplicity.

Similarly, the second-order variation can be approximated in discrete time as

$$\begin{aligned} \delta^2\theta(k) &\approx \theta(k+2) - 2\theta(k+1) + \theta(k), \\ \delta^2\hat{u}_s(k) &\approx \hat{u}_s(k+2) - 2\hat{u}_s(k+1) + \hat{u}_s(k). \end{aligned} \tag{2.58}$$

2.5 Numerical Example

2.5.1 Task Description

The goal of this task is to plan generate a trajectory under which the quadcopter perches at a specified location on a vertical wall. Some parameters used in this numerical example are

listed in Table 2.1. Notice that some parameters used in this example are artificially made up, just to demonstrate the effectiveness of the algorithm.

Table 2.1: Parameters used for the numerical example of the vertical wall perching task.

Parameter	Value
m	0.74 kg
a	0.15 m
t_f	0.8 s

The quadcopter start at $(0, 0)$ in the $\boldsymbol{xOz}$ plane, with horizontal initial pitch angle $\theta(0) = 0$. The perching location is selected at $(1, 1)$ and the pitch angle for perching is $\theta(t_f) = -\pi/2$. In order to perch successfully, the quadcopter is required to have a velocity in the $\boldsymbol{x}$ direction at the perching instant as

$$0.05\text{m/s} \leq \dot{x}(t_f) \leq 0.1\text{m/s}. \tag{2.59}$$

The fly zone is defined as

$$\begin{aligned} 0 &\leq x \leq 1, \\ 0 &\leq z \leq 1. \end{aligned} \tag{2.60}$$

The CVX package in Matlab is utilized to solve this problem. Details of CVX package can be found in [GB14] and [GB08].

2.5.2 Initialization

As shown in Figure 2.6, to initialize the planning process, the thrust force $T(k)$ is initialized as

$$T(k) = mg, \tag{2.61}$$

and substituted into the translational dynamic equation for $x(k)$ in discrete time, as

$$\begin{bmatrix} x(k+1) \\ \dot{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ gt_s \end{bmatrix} \hat{u}_s(k). \tag{2.62}$$

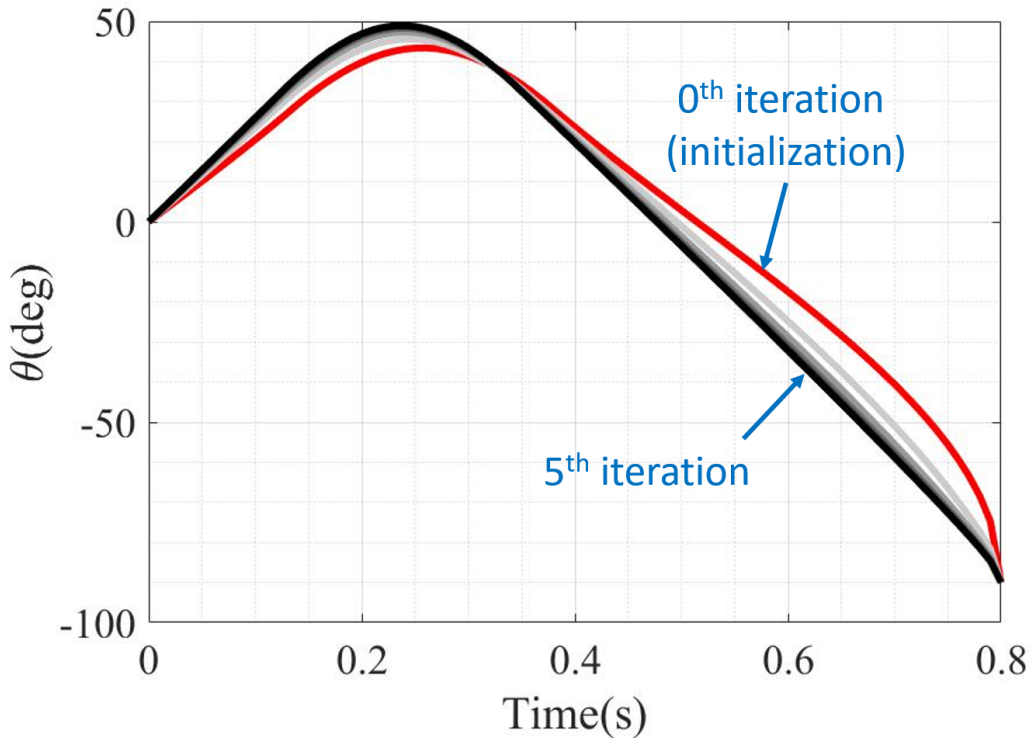


Figure 2.7: The history of $\theta(k)$ trajectories during the self-iterative convexification process. The trajectory from initialization is colored in red, and the trajectory after five iterations is colored in black.

The first and second order derivatives of the input $\hat{u}_s(k)$ are minimized in this optimization, as

$$\begin{aligned} & \min s \\ s.t. & \\ & \Delta\hat{u}_s(k) = \hat{u}_s(k+1) - \hat{u}_s(k) \\ & -s \leq \Delta\hat{u}_s(k) \leq s \\ & \Delta^2\hat{u}_s(k) = \hat{u}_s(k+2) - 2\hat{u}_s(k+1) + \hat{u}_s(k) \\ & -\gamma s \leq \Delta^2\hat{u}_s(k) \leq \gamma s \end{aligned} \tag{2.63}$$

where γ is a tuning parameter.

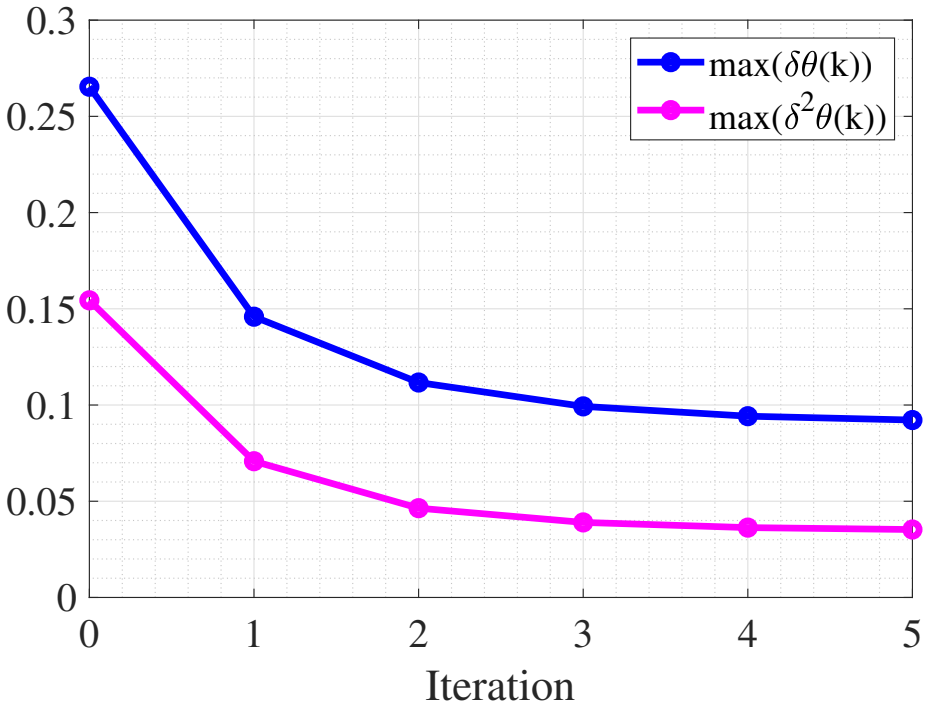


Figure 2.8: The maximum first-order and second-order variations of $\theta(k)$ along the self-iterative convexification process.

2.5.3 Constraints on Angular Velocity and Acceleration of Pitch Angle

The self-iteration process proposed in section 2.4.4 is applied in this section to convexify the constraints on angular velocity and acceleration of the pitch angle. During these iterations, most of the constraints are same as the initialization process, except that (2.63) is replaced with

$$\begin{aligned} &\min s \\ s.t. \quad &\\ &\Delta\hat{u}_s^r(k) = \hat{u}_s^r(k+1) - \hat{u}_s^r(k) \\ &-s \cdot \sqrt{1-\hat{u}_s^{r-1}(k)} \leq \Delta\hat{u}_s^r(k) \leq s \cdot \sqrt{1-\hat{u}_s^{r-1}(k)} \\ &\Delta^2\hat{u}_s^r(k) = \hat{u}_s^r(k+2) - 2\hat{u}_s^r(k+1) + \hat{u}_s^r(k) \\ &-\gamma s \cdot \sqrt{1-\hat{u}_s^{r-1}(k)} \leq \Delta^2\hat{u}_s^r(k) \leq \gamma s \cdot \sqrt{1-\hat{u}_s^{r-1}(k)} \end{aligned} \tag{2.64}$$

where $\hat{u}_s^r(k)$ and $\hat{u}_s^{r-1}(k)$ refer to the trajectories of $\hat{u}_s(k)$ in the r^{th} and $(r-1)^{th}$ iterations respectively.

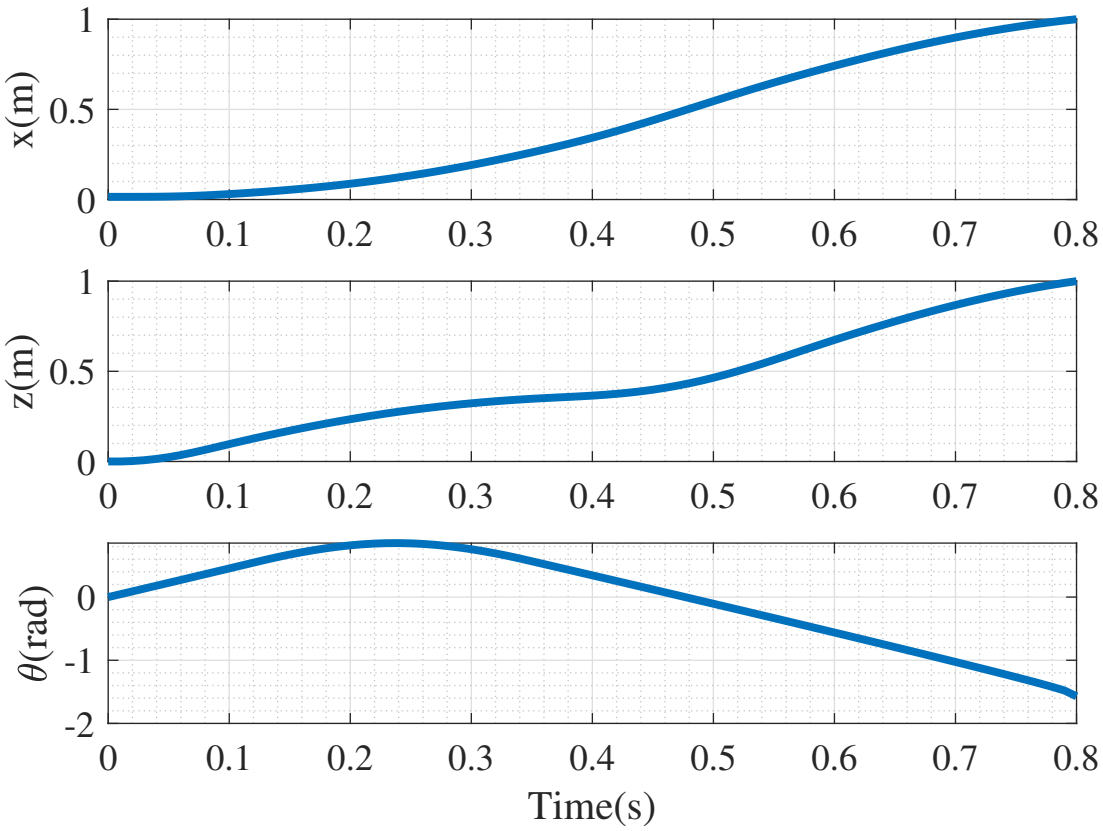


Figure 2.9: The generated trajectories of motion outputs x, z and θ for the task of quadcopter perching at a specified location on a vertical wall in the $\mathbb{SE}(2)$ space.

Specifically for the numerical example, the history of pitch angle trajectory $\theta(k)$ during the self-iterative convexification process is shown in Figure 2.7. The maximum first-order and second-order variations of the generated pitch angle trajectories $\theta(k)$ along these iterations are shown in Figure 2.8, which shows that the maximum values of both variations decrease and gradually converge with more iterations.

2.5.4 Generating Trajectory for $T(k)$

The trajectory of pitch angle $\theta(k)$ is calculated reversely from $\hat{u}_s(k)$ as

$$\theta(k) = \arcsin(\hat{u}_s(k)), \tag{2.65}$$

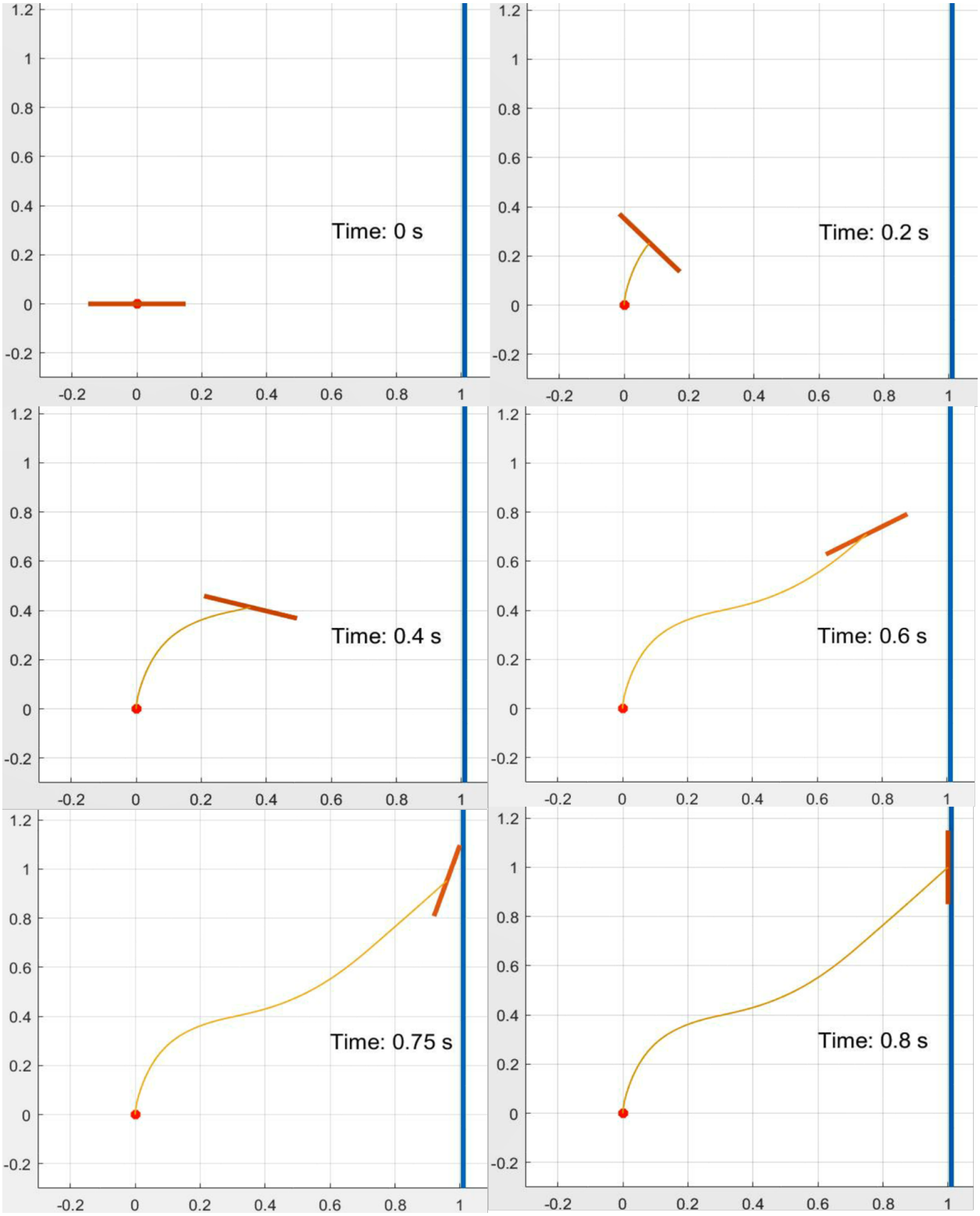


Figure 2.10: Visualization of the planned trajectory.

and conveyed to stage 2, acting as known values in the dynamic equation (2.47). The cost function of stage 2 is

$$\min_{T(k)} d = \sqrt{(x_f - x(N))^2 + (z_f - z(N))^2}, \tag{2.66}$$

where

$$x_f = z_f = 1 \tag{2.67}$$

for this case.

The optimization can be solved, with a perfect optimal solution

$$d^{opt} = 0, \tag{2.68}$$

indicating that the trajectory has met all requirements on position and attitude. The pitch angle trajectory is further substituted into rotational dynamics to check input saturation. Finally, the generated feasible trajectory is shown in Figure 2.9. Its visualization is shown in Figure 2.10.

2.5.5 Discussion

The numerical example shows that the algorithm is able to generate a trajectory with proper requirements on both position and attitude. While the solution is based on convex optimizations, the original problem is highly non-convex and have multiple local minima in the feasible space. Therefore, the trajectory from initialization is significant as all later searches are generally within its neighborhood, as indicated in Figure 2.7. It can be demonstrated by cases that bad initialization can lead to failure in trajectory generation even for this task, of which one feasible solution has already been proposed.

Selection from $\hat{u}_s$ and $\hat{u}_c$ for the optimization in stage 1 of Figure 2.5 is also an open question. For the proposed perching task where the pitch angle is generally constrained within $-\pi/2 \leq \theta \leq \pi/2$, the input $\hat{u}_s$ has more significance, as it can provide bidirectional force along the $\boldsymbol{x}$ axis. On the other hand, any $\hat{u}_c$ within the constraint always provide a force upwards along the $\boldsymbol{z}$ axis, and thus does not influence the position trajectory greatly.

2.6 Conclusion

This chapter proposed algorithm which is able to search for feasible trajectories for conventional quadcopters under requirements of both position and attitude. This is generally a challenging task due to the underactuation of quadcopter dynamics. The algorithm formulated an optimization and converted it into a series of convex problems, thus can be solved efficiently. The algorithm is demonstrated numerically with the task of quadcopter perching at a specified location on a vertical wall.

However, even if a trajectory that specifies roll and pitch angles along with quadcopter position is generated, the tracking still remains a challenging problem due to underactuation. One typical way is to switch between controllers, each of which regulates only a subset of the outputs. However, this control strategy lacks robustness and thus sensitive to model uncertainties or external disturbances. In addition, the algorithm still have heavy computation burdens, and complication for constraints formulation.

Therefore, the implementation of algorithms in this field is challenging in industry. The mechanical modification on multirotor platforms, on the other hand, directly attenuates or even eliminates the underactuation, and shall be elaborated in the next few chapters.

CHAPTER 3

A Novel Fully Actuated Aerial Platform with Tiltable Thrust Actuators

3.1 Introduction

Multirotor, among all popular UAV configurations, shows its advantages on mechanical simplicity and high agility, in addition to the capability of hovering and vertical take-off and landing(VTOL). It is usually driven by multiple propeller-motor sets as thrusting actuators.

However, conventional multirotors have underactuated dynamics, as all thrust forces are aligned in the same direction, usually upwards, to efficiently compensate for gravity, and can vary only with the change of platform poses. This underactuation causes coupling between position and attitude, and thus leads to limitations in applications such as exploration in the constrained space [PBH20] or interacting with the environment [RCS19].

Various mechanical modifications have been proposed regarding this problem, as surveyed in [HGA20]. One class of platforms deployed propeller-motor sets at diverse orientations to collectively generate thrust forces in variable directions, as shown in [RCS19], [JVC18], [RRB15], [RMP17] and [PLA18]. While this configuration maintains mechanical simplicity, selecting the orientations of each propeller-motor set remains open and challenging. It is generally calculated by the optimization customized for one specific case, as shown in [JVC18] and [RRB15], and needs to be adjusted for different tasks. In addition, this configuration needs at least six ([RCS19][JVC18][RRB15][RMP17]) or eight [PLA18] thrusting actuator sets, resulting in larger size, lower agility and greater energy consumption in contrast to the popular configuration with four sets.

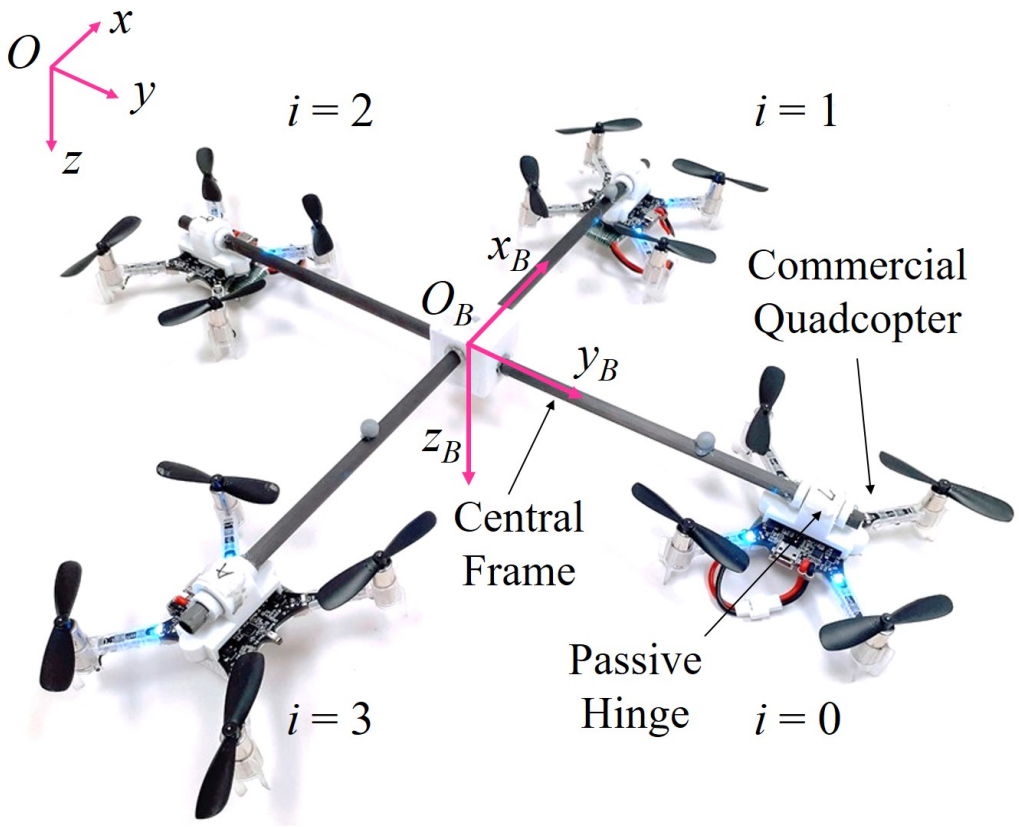


Figure 3.1: Prototype of the proposed platform. Four commercial quadcopters, passively hinged onto the central frame made of 3D-printed parts and carbon fiber tubes, are used as tiltable actuators.

The aforementioned limitations motivate another class of modifications, where the orientations of thrusting actuators are actively controlled during the flight. The concept was firstly proposed and realized by [RBG14], which built a quadrotor platform with tiltable cant angles for each thrusting actuator set. The idea was extended in [GT18], where both cant and dihedral angles became tiltable, to achieve higher thrusting efficiency for counteracting gravity. The omnidirectional flight was achieved by a hexrotor platform with tiltable cant angles, as proposed in [KVE18]. This class of modifications significantly increases the flexibility for different tasks and enlarges the theoretical operational space. However, additional actuators and mechanisms for tilting degrade the advantages of multirotor configurations on mechanical simplicity, and increase the no-load take-off weight. Furthermore, the reaction

torques arising from tilting actuation will be exerted on the platform main body as disturbance. [RBF16] proposed a novel design, where the cant angles of thrusting actuators are driven by one single motor through wire mechanism. The total weight is greatly reduced, but with the price of growing mechanical complexity.

Spinning propellers generate drag torques along with thrust forces. They are utilized collectively in conventional multirotors for yawing control, but have varying directions in the aforementioned modified configurations and are generally ignored and regarded as external disturbance for control and analysis. This problem was circumvented in the configuration of [NPP18], where multiple commercial quadcopters are exploited as thrusting actuators to lift and pose a manipulator in the air. However, it is also addressed by the authors that the operational space is significantly constrained by the limited range of spherical joints used to connect quadcopters with the tool frame.

This chapter presents a novel aerial platform with four tiltable thrusting actuators that can achieve 6 DoF control in space, as shown in Figure 3.1. Each actuator set consists of a commercial quadcopter and a passive hinge to provide actuation on both thrusting and tilting concurrently, as shown in Figure 3.2. The proposed platform circumvents additional actuators and mechanism for tilting, thus maintains mechanical simplicity and easiness for prototyping. It also avoids the introduction of external disturbance from propeller drag and tilting reaction, and has unlimited range for tilting actuation.

The remainder of the chapter is organized as follows. The system dynamics is presented in section 3.2. The actuation capability is analyzed in section 3.3. The simulation and experiment setup is shown in section 3.4. The chapter is concluded in section 3.5.

3.2 Dynamic Model

3.2.1 Preliminaries

Define the world inertia frame $\mathcal{F}_W : \{O; \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ under North-East-Down(NED) convention; the body frame $\mathcal{F}_B : \{O_B; \boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$ attached to the central frame of the platform, as

shown in Figure 3.1; and the quadcopter frames $\mathcal{F}_{Q_i} : \{O_{Q_i}; \boldsymbol{x}_{Q_i}, \boldsymbol{y}_{Q_i}, \boldsymbol{z}_{Q_i}\}$ attached to each quadcopter, as shown in Figure 3.2.

The quadcopters in the platform are ordered

$$i = 0, 1, 2, 3 \tag{3.1}$$

in Figure 3.1. The distance of each quadcopter center to the center of central frame is assumed identical and defined as l. The propellers of each quadcopter are ordered

$$j = 0, 1, 2, 3 \tag{3.2}$$

in Figure 3.2. The distance of each propeller to the quadcopter center is assumed identical and defined as a.

Define the position of the central frame center as

$$\boldsymbol{\xi} = \begin{bmatrix} x & y & z \end{bmatrix}^T. \tag{3.3}$$

The attitude is described by intrinsic rotation. The corresponding Tait–Bryan angles in the roll-pitch-yaw convention are defined as

$$\boldsymbol{\eta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T. \tag{3.4}$$

The angular velocity of the platform in $\mathcal{F}_B$ is defined as

$$\boldsymbol{\nu} = \begin{bmatrix} p & q & r \end{bmatrix}^T. \tag{3.5}$$

The tilting angle of quadcopter i with respect to the central frame is defined as α_i, as shown in Figure 3.2. Define ${}^W\boldsymbol{R}_B \in \mathbb{SO}(3)$ as the rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$ according to the Tait–Bryan angles defined in (3.4), and ${}^B\boldsymbol{R}_{Q_i} \in \mathbb{SO}(3)$ the rotation matrices from $\mathcal{F}_{Q_i}$ to $\mathcal{F}_B$.

For notation simplicity, denote

$$s\alpha_i = \sin\alpha_i \ , \quad c\alpha_i = \cos\alpha_i. \tag{3.6}$$

Key notations are summarized in TABLE 3.1.

Notation	Definition
i	Order of commercial quadcopters in the platform, as in Figure 3.1
j	Order of propellers for each commercial quadcopter, as in Figure 3.2
$\mathcal{F}_W$	World frame $\{O; \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$
$\mathcal{F}_B$	Body frame $\{O_B; \boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$ on the central frame, as in Figure 3.1
$\mathcal{F}_{Q_i}$	Quadcopter frame $\{O_{Q_i}; \boldsymbol{x}_{Q_i}, \boldsymbol{y}_{Q_i}, \boldsymbol{z}_{Q_i}\}$ on quadcopter i, as in Figure 3.2
m	Total mass of the platform
$\boldsymbol{I}$	Inertial matrix of the platform
l	The identical distance of each quadcopter center to the center of central frame
a	The identical distance of each propeller to the quadcopter center
$\boldsymbol{\xi}$	Position of the central frame center, $\boldsymbol{\xi} = \begin{bmatrix} x & y & z \end{bmatrix}^T$
$\boldsymbol{\eta}$	The Euler angle set of platform attitude, $\boldsymbol{\eta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$
$\boldsymbol{\nu}$	The angular velocity of the platform in $\mathcal{F}_B$, $\boldsymbol{\nu} = \begin{bmatrix} p & q & r \end{bmatrix}^T$
α_i	The tilting angle of quadcopter i with respect to the central frame
${}^W\boldsymbol{R}_B$	The rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$
${}^B\boldsymbol{R}_{Q_i}$	The rotation matrix from $\mathcal{F}_{Q_i}$ to $\mathcal{F}_B$
$s[\cdot], c[\cdot]$	Simplified notation of sin[$\cdot$] and cos[$\cdot$] respectively
$\boldsymbol{I}_n$	Identity matrix of dimension n

Table 3.1: Summary of notifications.

3.2.2 Actuators and Tilting Dynamics

General aerial platforms with tiltable rotors require thrusting actuators to generate thrust forces, and tilting actuators to change the direction of thrust vectors. In the proposed platform, however, thrusting and tilting are achieved by the same actuator concurrently and independently.

Propeller j in quadcopter i generates a thrust force f_{ij} and a torque t_{ij} along the normal

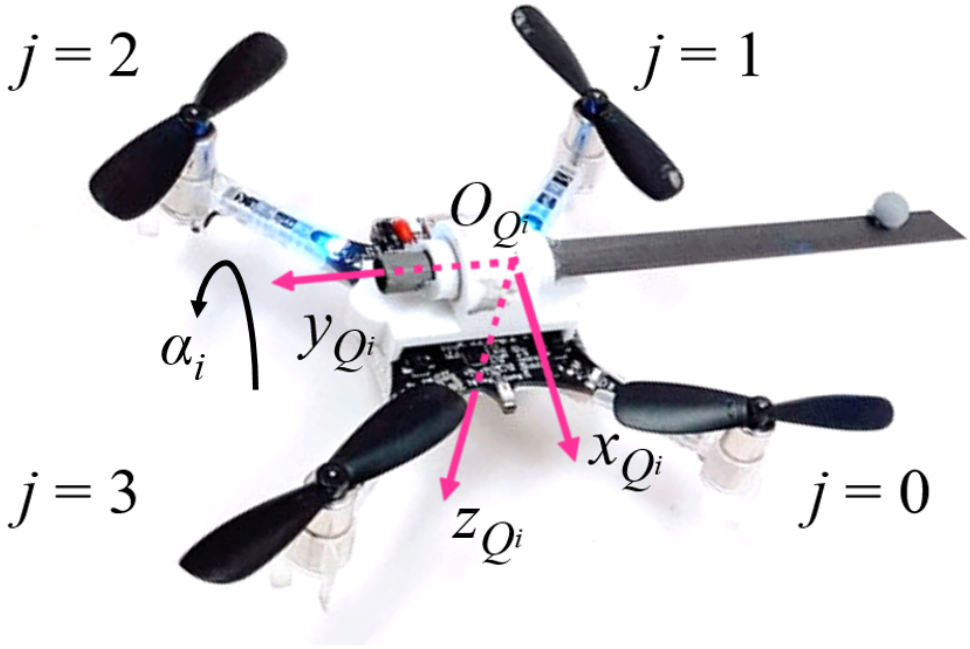


Figure 3.2: The tiltable actuator, consisting of commercial quadcopter and the passive hinge.

direction of its spinning plane

$$\begin{aligned} f_{ij} &= K_T \omega_{ij}^2, \\ t_{ij} &= K_\tau \omega_{ij}^2. \end{aligned} \tag{3.7}$$

Here w_{ij} represents the propeller angular velocity, K_T and K_τ are constants related with propeller specifications and aerodynamics.

Collectively, quadcopter i generates four independent inputs

$$\begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix} = \begin{bmatrix} -K_T & -K_T & -K_T & -K_T \\ -\frac{aK_T}{\sqrt{2}} & -\frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} \\ \frac{aK_T}{\sqrt{2}} & -\frac{aK_T}{\sqrt{2}} & -\frac{aK_T}{\sqrt{2}} & \frac{aK_T}{\sqrt{2}} \\ K_\tau & -K_\tau & K_\tau & -K_\tau \end{bmatrix} \begin{bmatrix} \omega_{i0}^2 \\ \omega_{i1}^2 \\ \omega_{i2}^2 \\ \omega_{i3}^2 \end{bmatrix}. \tag{3.8}$$

Here T_i along $\boldsymbol{z}_{Q_i}$ is the thrust force. M_i^x, M_i^y and M_i^z refer to the external torques in $\mathcal{F}_{Q_i}$.

Therefore, the thrusting actuation is controlled by T_i, and the tilting actuation is controlled by M_i^y, as the quadcopter is passively hinged onto the central frame along $\boldsymbol{y}_{Q_i}$ axis. It can be observed from (3.8) that these two actuations are decoupled.

Furthermore, assuming the quadcopter Center-of-Mass (CoM) on the rotation axis, the

tilting dynamics is

$$\ddot{\alpha}_i = M_i^y - \sin(\frac{\pi}{2}i)\dot{p} - \cos(\frac{\pi}{2}i)\dot{q}, \tag{3.9}$$

where p and q are defined in (3.5).

M_i^x and M_i^z are independent auxiliary inputs, which can be set zero for control simplicity, or exploited to extend control capability, as to be shown in section 3.2.4.

3.2.3 Translational Dynamics

The translational dynamics of the platform is expressed as

$$\ddot{\boldsymbol{\xi}} = \frac{1}{m}{}^W\boldsymbol{R}_B(\sum_{i=0}^{3}{}^B\boldsymbol{R}_{Q_i}\boldsymbol{T}_i) + \boldsymbol{G}, \tag{3.10}$$

where m refers to the total mass of the platform, $\boldsymbol{G}$ the gravitational acceleration in $\mathcal{F}_W$ and

$$\boldsymbol{T}_i = \begin{bmatrix} 0 & 0 & -T_i \end{bmatrix}^T \tag{3.11}$$

the total thrust vector of quadcopter i in $\mathcal{F}_{Q_i}$.

Here

$${}^B\boldsymbol{R}_{Q_i} = \boldsymbol{N}_i\boldsymbol{R}_i, \tag{3.12}$$

where

$$\begin{aligned} \boldsymbol{N}_i &= \begin{bmatrix} \cos(-\frac{\pi}{2}i) & -\sin(-\frac{\pi}{2}i) & 0 \\ \sin(-\frac{\pi}{2}i) & \cos(-\frac{\pi}{2}i) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \boldsymbol{R}_i &= \begin{bmatrix} c\alpha_i & 0 & s\alpha_i \\ 0 & 1 & 0 \\ -s\alpha_i & 0 & c\alpha_i \end{bmatrix}. \end{aligned} \tag{3.13}$$

3.2.4 Rotational Dynamics

Assuming constant inertial matrix $\boldsymbol{I}$, the rotational dynamics of the platform in $\mathcal{F}_B$ is

$$\dot{\boldsymbol{\nu}} = \boldsymbol{I}^{-1}(-\boldsymbol{\nu} \times (\boldsymbol{I}\boldsymbol{\nu}) + \boldsymbol{\tau}), \tag{3.14}$$

where $\boldsymbol{\tau} \in \mathbb{R}^3$ is the total external torque exerted on the platform.

Generally, for aerial platforms with four tiltable thrusting actuators, the external torque consists of three major components

$$\boldsymbol{\tau} = \boldsymbol{\tau}_T + \boldsymbol{\tau}_M + \boldsymbol{\tau}_R. \tag{3.15}$$

Here $\boldsymbol{\tau}_T$ is the torque generated by the four tilting thrust vectors, and is calculated as

$$\boldsymbol{\tau}_T = \begin{bmatrix} -c\alpha_0 l & 0 & c\alpha_2 l & 0 \\ 0 & c\alpha_1 l & 0 & -c\alpha_3 l \\ s\alpha_0 l & s\alpha_1 l & s\alpha_2 l & s\alpha_3 l \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \end{bmatrix}. \tag{3.16}$$

It has been mentioned that each single spinning propeller generates a drag torque when providing thrust force, as indicated in (3.7). The total sum of these torques is denoted as $\boldsymbol{\tau}_M$, and can be calculated by

$$\boldsymbol{\tau}_M = \sum_{i=0}^{3} {}^B\boldsymbol{R}_{Q_i} \boldsymbol{M}_i. \tag{3.17}$$

Here $\boldsymbol{M}_i$ is the drag torque of each thrusting actuator in $\mathcal{F}_{Q_i}$. For previous works where propeller-motors are used for thrusting, such as [RBG14] and [KVE18], $\boldsymbol{M}_i$ is coupled with T_i, and is usually regarded as disturbance in the controller design.

However, in the proposed platform,

$$\boldsymbol{M}_i = \begin{bmatrix} M_i^x & 0 & M_i^z \end{bmatrix}^T \tag{3.18}$$

according to Equation (3.8). Here the zero element is related to the passive hinge along $\boldsymbol{y}_{Q_i}$. M_i^x and M_i^z, as mentioned earlier, are independent inputs. Therefore, $\boldsymbol{\tau}_M$ in the proposed platform is decoupled from $\boldsymbol{\tau}_T$ and can be determined freely.

Finally, $\boldsymbol{\tau}_R$ refers to the reaction torque when tilting the thrusting actuators, which, according to Newton's Third Law, occurs every time $\ddot{\alpha}_i \neq 0$ for the aforementioned tiltable platforms where tilting actuators are fixed on the central frame. However, in the proposed platform, tilting actuation is merely an interaction of the quadcopter and the air due to the passive hinge connection. Therefore,

$$\boldsymbol{\tau}_R = 0 \tag{3.19}$$

holds at all time.

3.3 Actuation Analysis

This section is to show that the unique dynamics of our proposed system is able to provide an analytical solution set of inputs α_i, T_i and $\boldsymbol{M}_i$ to collectively generate any desired accelerations $\ddot{\boldsymbol{\xi}}^d$ and $\dot{\boldsymbol{\nu}}^d$ under arbitrary attitude $\boldsymbol{\eta}$ without the consideration of input saturation. Notice that such property does not hold for conventional quadrotor platforms due to the coupling of position and attitude, and is proved to hold only for simplified models, where propeller drag and tilting reaction torques are ignored, on tiltable actuator platforms such as [RBG14] and [KVE18].

Substitute

$$\begin{aligned} \ddot{\boldsymbol{\xi}} &= \ddot{\boldsymbol{\xi}}^d, \\ \dot{\boldsymbol{\nu}} &= \dot{\boldsymbol{\nu}}^d, \end{aligned} \tag{3.20}$$

into (3.10) and (3.14), then the translational dynamics becomes

$$\ddot{\boldsymbol{\xi}}^d = \frac{1}{m} {}^W\boldsymbol{R}_B \left(\sum_{i=0}^{3} {}^B\boldsymbol{R}_{Q_i} \boldsymbol{T}_i\right) + \boldsymbol{G}, \tag{3.21}$$

and the rotational dynamics is rewritten as

$$\dot{\boldsymbol{\nu}}^d = \boldsymbol{I}^{-1}(-\boldsymbol{\nu} \times (\boldsymbol{I}\boldsymbol{\nu}) + \boldsymbol{\tau}_T + \boldsymbol{\tau}_M). \tag{3.22}$$

The unique dynamics of our proposed configuration decouples $\boldsymbol{\tau}_T$ and $\boldsymbol{\tau}_M$, as demonstrated in (3.16) and (3.18). Therefore, set

$$\boldsymbol{\tau}_M = \boldsymbol{0} \tag{3.23}$$

and define

$$\begin{aligned} \boldsymbol{\zeta} &= \begin{bmatrix} \zeta_x & \zeta_y & \zeta_z \end{bmatrix}^T = m({}^W\boldsymbol{R}_B)^T(\boldsymbol{G} - \ddot{\boldsymbol{\xi}}^d), \\ \boldsymbol{\mu} &= \begin{bmatrix} \mu_x & \mu_y & \mu_z \end{bmatrix}^T = \boldsymbol{I}\dot{\boldsymbol{\nu}}^d + \boldsymbol{\nu} \times (\boldsymbol{I}\boldsymbol{\nu}), \end{aligned} \tag{3.24}$$

then (3.21) and (3.22) can be reformulated in the matrix form as

$$\boldsymbol{W}\boldsymbol{\mathcal{T}} = \boldsymbol{p}, \tag{3.25}$$

where

$$\begin{aligned}
\boldsymbol{W} &= \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -l & 0 & l & 0 \\ 0 & 0 & 0 & 0 & 0 & l & 0 & -l \\ l & l & l & l & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\boldsymbol{\mathcal{T}} &= \begin{bmatrix} s\alpha_0 T_0 & \dots & s\alpha_3 T_3 & c\alpha_0 T_0 & \dots & c\alpha_3 T_3 \end{bmatrix}^T, \\
\boldsymbol{p} &= \begin{bmatrix} \zeta_x & \zeta_y & \zeta_z & \mu_x & \mu_y & \mu_z \end{bmatrix}^T.
\end{aligned} \tag{3.26}$$

Notice that $\boldsymbol{W}$ and $\boldsymbol{p}$ are given, and $\boldsymbol{\mathcal{T}}$ is to be solved. Obviously,

$$\text{rank}(\boldsymbol{W}) = 6. \tag{3.27}$$

So there always exist a matrix $\boldsymbol{W}^\dagger \in \mathbb{R}^{8\times 6}$ such that

$$\boldsymbol{W}\boldsymbol{W}^\dagger = \boldsymbol{I}_6. \tag{3.28}$$

Here $\boldsymbol{I}_6$ refers to the identity matrix of dimension 6.

Therefore, one particular solution of $\boldsymbol{\mathcal{T}}$ for (3.25) is

$$\boldsymbol{\mathcal{T}}_P = \boldsymbol{W}^\dagger \boldsymbol{p}, \tag{3.29}$$

and the null space of is $\boldsymbol{\mathcal{T}}$ explored as

$$\mathcal{N}(\boldsymbol{W}) = \{\boldsymbol{x} \in \mathbb{R}^8 : \ \boldsymbol{x} = (\boldsymbol{I}_8 - \boldsymbol{W}^\dagger \boldsymbol{W})\boldsymbol{\sigma}, \forall \boldsymbol{\sigma} \in \mathbb{R}^8\}. \tag{3.30}$$

Notice that $\mathcal{N}(\boldsymbol{W})$ is supposed to have 2 free variables. By proper calculation, (3.30) can be reduced to

$$\mathcal{N}(\boldsymbol{W}) = \{\boldsymbol{x} \in \mathbb{R}^8 : \ \boldsymbol{x} = \sigma_1 \boldsymbol{v}_1 + \sigma_2 \boldsymbol{v}_2, \forall \sigma_1, \sigma_2 \in \mathbb{R}\}, \tag{3.31}$$

where

$$\begin{aligned}
\boldsymbol{v}_1 &= \begin{bmatrix} -1 & 1 & -1 & 1 & \boldsymbol{0}_{1\times 4} \end{bmatrix}^T, \\
\boldsymbol{v}_2 &= \begin{bmatrix} \boldsymbol{0}_{1\times 4} & -1 & 1 & -1 & 1 \end{bmatrix}^T.
\end{aligned} \tag{3.32}$$

Therefore, the general solution of (3.25) is

$$\boldsymbol{\mathcal{T}} = \boldsymbol{W}^{\dagger}\boldsymbol{p} + \boldsymbol{x},\ \boldsymbol{x} \in \mathcal{N}(\boldsymbol{W}). \tag{3.33}$$

Equation (3.33) can be explicitly calculated as

$$\begin{aligned}
s\alpha_0 T_0 &= -\frac{\zeta_x}{2} + \frac{\mu_z}{4l} - \sigma_1, \\
s\alpha_1 T_1 &= \frac{\zeta_y}{2} + \frac{\mu_z}{4l} + \sigma_1, \\
s\alpha_2 T_2 &= \frac{\zeta_x}{2} + \frac{\mu_z}{4l} - \sigma_1, \\
s\alpha_3 T_3 &= -\frac{\zeta_y}{2} + \frac{\mu_z}{4l} + \sigma_1, \\
c\alpha_0 T_0 &= -\frac{\zeta_z}{4} - \frac{\mu_x}{2l} - \sigma_2, \\
c\alpha_1 T_1 &= -\frac{\zeta_z}{4} + \frac{\mu_y}{2l} + \sigma_2, \\
c\alpha_2 T_2 &= -\frac{\zeta_z}{4} + \frac{\mu_x}{2l} - \sigma_2, \\
c\alpha_3 T_3 &= -\frac{\zeta_z}{4} - \frac{\mu_y}{2l} + \sigma_2.
\end{aligned} \tag{3.34}$$

As the thrust forces $T_i \geq 0$, it can be calculated by

$$T_i = \sqrt{(s\alpha_i T_i)^2 + (c\alpha_i T_i)^2}. \tag{3.35}$$

The tilting angles α_i can be uniquely determined by

$$\alpha_i = \text{atan2}(s\alpha_i T_i, c\alpha_i T_i) \tag{3.36}$$

for

$$\alpha_i \in (-\pi, \pi],\ T_i \neq 0. \tag{3.37}$$

When $T_i = 0$, any tilting angle α_i satisfies the corresponding solution in equation (3.34).

In addition, equation (3.23) has a trivial solution for $\boldsymbol{M}_i$ as

$$\boldsymbol{M}_i = \boldsymbol{0}. \tag{3.38}$$

Therefore, T_i, α_i and $\boldsymbol{M}_i$ can be calculated by (3.35), (3.36) and (3.38) respectively with the results in (3.34), indicating that a set of inputs always exists to generate any specific

translational and rotational accelerations of the platform at arbitrary attitude $\boldsymbol{\eta}$ without the consideration of input saturation. In fact, the result proposed in this section shows that the goal can be achieved with only T_i and α, suggesting that $\boldsymbol{M}_i$ can be utilized as auxiliary inputs to further improve control performance and robustness.

3.4 Simulation and Experimental Environments

3.4.1 Prototype

The prototype in Fig. 3.1 is built in UCLA MacLab for experiments. The central frame is made of carbon fiber tubes. Passive hinges and central frame connector are 3D-printed. Crazyflie 2.1 commercial quadcopters are selected as actuators. Each quadcopter has an onboard STM32 processor and IMU unit, in addition to a radio module for wireless communication. The maximum take-off weight of each quadcopter is 60 g. Other required parameters of Crazyflie 2.1 has been identified in [For15]. Critical parameters of the prototype are measured and listed in TABLE 3.2.

Table 3.2: Physical Parameters of the Prototype in Fig. 3.1

Parameter	Value
m	0.16 kg
l	0.14 m
a	4.60 cm
I_{xx}	1.46×10^{-3} kg$\cdot$ m^2
I_{yy}	1.46×10^{-3} kg$\cdot$ m^2
I_{zz}	2.77×10^{-3} kg$\cdot$ m^2

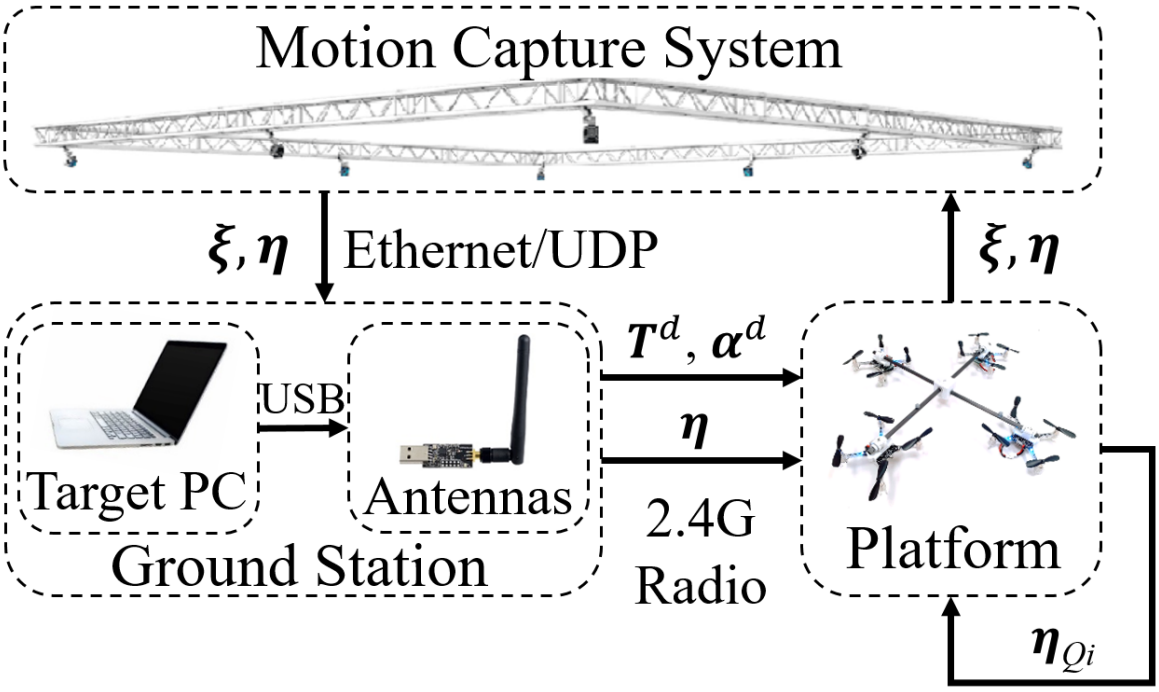


Figure 3.3: Experimental realization of control system. The motion capture system measures the attitude and position of the platform, the position & attitude controllers and the mapper are running at the ground station on the target PC, and the servos are running on each quadcopters.

3.4.2 Control System Realization

The control system is demonstrated in Figure 3.3. The central controller is running as a Python script in the Ubuntu 16.04 operating system on the target PC. The control commands $\boldsymbol{T}^d$ and $\boldsymbol{\alpha}^d$ are sent to each quadcopter wirelessly via the Crazy Radio PA antennas through 2.4G radio. The OptiTrack motion capture system measures the position $\boldsymbol{\xi}$ and attitude $\boldsymbol{\eta}$ of the platform and sends them to the target PC through Ethernet/UDP. The attitude of each quadcopter $\boldsymbol{\eta}_{Q_i}$ is measured and filtered onboard, and exploited to calculate the tilting angle α_i by comparing with the attitude of the platform $\boldsymbol{\eta}$. The motion capture system is tested to have a maximum stable updating rate of 160 Hz, and the servo controllers are running at 500 Hz. The loop time of central controller follows a stochastic distribution as a result

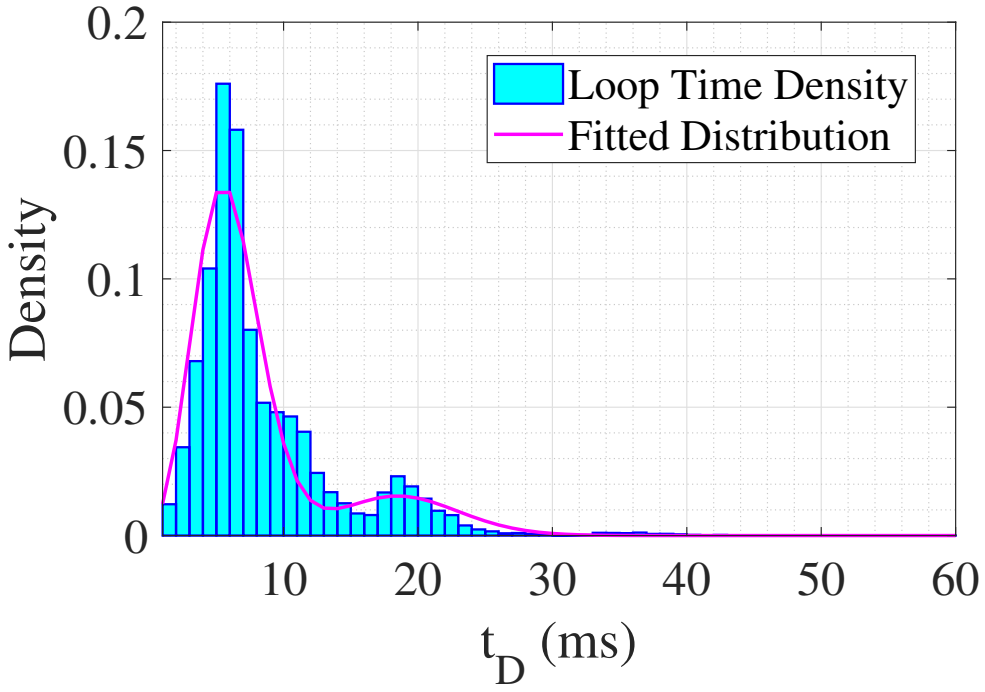


Figure 3.4: The histogram of central controller loop time and the fitted stochastic distribution.

of CPU task management logic and wireless communication, and will be analyzed in section 3.4.3.

3.4.3 Characterization of Central Controller Loop Time

As stated in section 3.4.2, the loop time of the central controller, denoted as t_D with unit millisecond, follows a stochastic distribution due to CPU logic and wireless communication. The goal of this section is to characterize such a distribution for simulation.

A total of $37,287$ data points of t_D are gathered from multiple experiments for different tasks at different time to address generality. The normalized histogram, denoted as $h(t_D)$, is calculated by statistical analysis, shown as the cyan bar in Figure 3.4.

Wireless communication has failures. If data transmission fails in the first attempt, the sender will request a second try. This explains the two peaks in the histogram, where the corresponding loop times 6 ms and 19 ms refers to the nominal loop time for one-time and

two-time transmissions respectively.

The central controller is running on the Ubuntu operating system. Due to its task distribution logic, the completing time is stochastic and generally follows the Poisson Distribution

$$f^p(t_D;\lambda) = \frac{\lambda_D^t e^{-\lambda}}{t_D!},\ t_D \in \mathbb{Z}_{\geq 0}. \tag{3.39}$$

Here λ refers to the peak of density, or the nominal loop time.

Therefore, the loop time t_D is characterized as a weighted Poisson Distribution

$$f^D(t_D) = w \cdot f^p(t_D;6) + (1-w) \cdot f^p(t_D;19). \tag{3.40}$$

Here w refers to the success rate of first-time transmission, and is calculated by the optimization

$$w_{opt} = \arg\min_w ||\boldsymbol{f}^D - \boldsymbol{h}||_\infty = 0.8318, \tag{3.41}$$

where $\boldsymbol{f}^D$ and $\boldsymbol{h}$ refers to the batch vectors of $f^D(t_D)$ and $h(t_D)$ respectively.

The fitted distribution is shown as the magenta curve in Figure 3.4.

3.5 Conclusion

This chapter presented a novel fully actuated aerial multirotor platform, to realize independent control of position and attitude. Such tasks can not be achieved by conventional quadcopters due to underactuation. Commercial quadcopters and passive hinges were exploited as tiltable thrusting actuators. The platform has simple and modulized mechanical structure, and attenuates disturbance torques from thrust vectoring operations. The uniqueness of system dynamics were illustrated, and the capability of actuation under different attitudes were verified theoretically. Simulation and experiment environments were set up for control verification in later works.

CHAPTER 4

Controller Design and Implementation for Independent Position and Attitude Tracking

4.1 Introduction

The previous chapter presented a novel aerial platform using commercial quadcopters and passive hinges as actuators to provide tiltable thrust forces. Actuation analysis showed its capability for 6 DoF control in space. The goal of this chapter is to design a controller for the proposed multirotor aerial platform to achieve independent control of position and attitude. The proposed controller is demonstrated with simulation and experiments in different flight cases.

The rest of this chapter is organized as follows. Section 4.2 proposes a hierarchical computationally effective algorithm to realize 6 DoF control. The servo dynamics is identified in section 4.3 for realistic simulation. Section 4.4 shows the simulation and experimental results to demonstrate the independent control of position and attitude. The chapter is concluded in 4.5.

4.2 Control

4.2.1 Overview

A hierarchical control architecture is designed, as shown in Figure 4.1. The outer loop takes in the 6 DoF errors in the 3D space and calculates the desired total external force $\boldsymbol{F}^d$ and torque $\boldsymbol{\tau}^d$ on the platform, as stated in section 4.2.2. The force and torque are then allocated

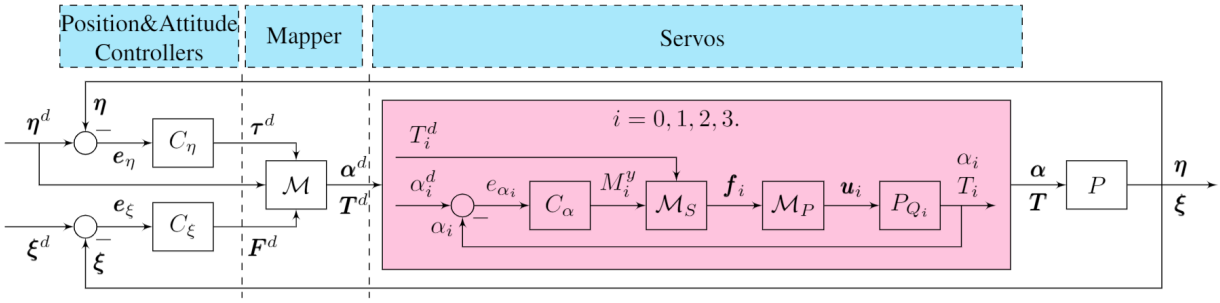


Figure 4.1: Controller architecture.

onto each quadcopter as desired thrusting force T_i^d and tilting angle α_i^d by the mapper $\mathcal{M}$ in section 4.2.3. A servo controller is running on each quadcopter to follow the trajectories of thrust force and tilting angle, as shown in section 4.2.4.

For controller simplicity, set

$$\boldsymbol{M}_i = 0, \tag{4.1}$$

then

$$\boldsymbol{\tau} = \boldsymbol{\tau}_T. \tag{4.2}$$

Notice that block P in Figure 4.1 refers to the dynamics of the proposed platform.

4.2.2 Position and Attitude Controllers

The controllers C_ξ and C_η in the outer loop calculated the desired total external force $\boldsymbol{F}^d$ and torque $\boldsymbol{\tau}^d$ on the platform from 6 DoF errors of rigid body in the 3D space, while compensating the gravitational effect, as

$$\begin{aligned} \boldsymbol{F}^d &= K_\xi^p \boldsymbol{e}_\xi + K_\xi^i \int \boldsymbol{e}_\xi \mathrm{d}t + K_\xi^d \dot{\boldsymbol{e}}_\xi - \boldsymbol{G}, \\ \boldsymbol{\tau}^d &= K_\eta^p \boldsymbol{e}_\eta + K_\eta^i \int \boldsymbol{e}_\eta \mathrm{d}t + K_\eta^d \boldsymbol{e}_\nu, \end{aligned} \tag{4.3}$$

Here $K_\xi^p, K_\xi^i, K_\xi^d, K_\eta^p, K_\eta^i, K_\eta^d$ are controller tuning parameters. The errors are defined in [LLM10] as

$$\begin{aligned}
\boldsymbol{e}_\xi &= \boldsymbol{\xi}^d - \boldsymbol{\xi}, \\
\dot{\boldsymbol{e}}_\xi &= \dot{\boldsymbol{\xi}}^d - \dot{\boldsymbol{\xi}}, \\
\boldsymbol{e}_\eta &= \frac{1}{2}[\boldsymbol{R}(\boldsymbol{\eta})^T \boldsymbol{R}(\boldsymbol{\eta}^d) - \boldsymbol{R}(\boldsymbol{\eta}^d)^T \boldsymbol{R}(\boldsymbol{\eta})]_\vee, \\
\boldsymbol{e}_\nu &= \boldsymbol{R}(\boldsymbol{\eta})^T \boldsymbol{R}(\boldsymbol{\eta}^d)\boldsymbol{\nu}^d - \boldsymbol{\nu}.
\end{aligned} \tag{4.4}$$

Here operator $[\cdot]_\vee$ defines the *vee* mapping from $\mathbb{SO}(3)$ to $\mathbb{R}^3$.

4.2.3 Mapper

The mapper allocates the desired total external force $\boldsymbol{F}^d$ and torque $\boldsymbol{\tau}^d$ onto each quadcopter as desired tilting angles α_i^d and thrust forces T_i^d.

It can be observed from (3.10) and (3.16) that α_i and T_i are nonlinearly coupled in dynamics by multiplication. Therefore, a two-stage mapping scheme is proposed, where the desired tilting angles α_i^d are determined in the first stage, and used as determined values in the second stage to calculate the desired thrust forces T_i^d.

Notice that the mapper has six inputs and eight outputs, so additional constraints are required. It has been derived in [NK14] that the platform roll/pitch angles ϕ, θ and actuator tilting angles α_i have the relationship

$$\begin{aligned}
-\alpha_0 &= \alpha_2 &&= 2\theta, \\
-\alpha_1 &= \alpha_3 &&= 2\phi
\end{aligned} \tag{4.5}$$

at any equilibrium point during hovering. Adjusting this relationship to include the influence of desired forces in the $\boldsymbol{x}_B O_B \boldsymbol{y}_B$ plane gives the first stage of mapping as

$$\begin{bmatrix} \alpha_0^d \\ \alpha_1^d \\ \alpha_2^d \\ \alpha_3^d \end{bmatrix} = \begin{bmatrix} 0 & -2 & -K_1^s & 0 \\ -2 & 0 & 0 & K_1^s \\ 0 & 2 & K_1^s & 0 \\ 2 & 0 & 0 & -K_1^s \end{bmatrix} \begin{bmatrix} \phi^d \\ \theta^d \\ \arctan K_2^s F_{Bx}^d \\ \arctan K_2^s F_{By}^d \end{bmatrix}. \tag{4.6}$$

Here ϕ^d and θ^d are the desired roll and pitch angles of the platform respectively. K_1^s and K_2^s are controller tuning parameters. F_{Bx}^d and F_{By}^d are the desired total external forces along $\boldsymbol{x}_B$ and $\boldsymbol{y}_B$, and are calculated by

$$\begin{bmatrix} F_{Bx}^d & F_{By}^d & F_{Bz}^d \end{bmatrix}^T = ({}^W\boldsymbol{R}_B)^T \boldsymbol{F}^d. \tag{4.7}$$

The inverse tangent function is a selected bounded monotonic function to shape the influence curve of F_{Bx}^d and F_{By}^d, and is subject to change with specific requirements.

The second stage of mapping is based on inverse dynamics

$$\begin{bmatrix} T_0^d \\ T_1^d \\ T_2^d \\ T_3^d \end{bmatrix} = \begin{bmatrix} -c\alpha_0^d l & 0 & c\alpha_2^d l & 0 \\ 0 & c\alpha_1^d l & 0 & -c\alpha_3^d l \\ s\alpha_0^d l & s\alpha_1^d l & s\alpha_2^d l & s\alpha_3^d l \\ -c\alpha_0^d & -c\alpha_1^d & -c\alpha_2^d & -c\alpha_3^d \end{bmatrix}^\dagger \begin{bmatrix} & \\ & \boldsymbol{\tau}^d \\ & \\ F_{Bz}^d & \end{bmatrix}. \tag{4.8}$$

Here $[\cdot]^\dagger$ refers to the operator of Moore-Penrose pseudo inverse.

The proposed mapper is computationally effective, and smoothens trajectories for desired tilting angles α_i.

4.2.4 Servos

Servos are the four identical control loops running on quadcopters to track the desired tilting angles α_i^d and thrust forces T_i^d, as shown in Figure 4.1 colored in magenta.

C_α is the tracking controller for tilting angle α_i, which calculated the required torque along the direction of passive hinge, denoted as M_i^y. $\mathcal{M}_S$ is the mapper that allocates inputs T_i^d and M_i^y onto thrust forces of each propeller $\boldsymbol{f}_i$ as

$$\boldsymbol{f}_i = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{\sqrt{2}}{2a} & -\frac{\sqrt{2}}{2a} & -\frac{\sqrt{2}}{2a} & \frac{\sqrt{2}}{2a} \end{bmatrix}^T \begin{bmatrix} T_i^d \\ M_i^y \end{bmatrix}. \tag{4.9}$$

$\mathcal{M}_P$ is the reverse mapping curve from propeller thrust forces $\boldsymbol{f}_i$ to motor servo commands $\boldsymbol{u}_i$. P_{Q_i} refers to the tilting dynamics of quadcopter i.

4.3 Servo Identification for Simulation Setup

4.3.1 Motivation

The quadcopter with passive hinge is modeled as a double integrator on a non-inertial frame in (3.9). However, the realistic model has more complication such as hinge friction or pendulum effect from CoM offset. Therefore, the closed-loop servo dynamics is experimentally identified in order to create realistic simulation.

4.3.2 Methodology

The discrete-time transfer function of the closed-loop servo dynamics is identified with Hankel matrix analysis.

Assuming the Finite-Impulse-Response (FIR) model of the servo dynamics is defined as

$$H(z^{-1}) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + ..., \tag{4.10}$$

the impulse response of servo can therefore be denoted as

$$h(k) = h_k, \tag{4.11}$$

where k refers to the time stamp of sampling.

Fix the central frame to the ground and send a 0.25 rad step reference signal to one quadcopter. Running the test multiple times and averaging the results gives the step response sequence of the servo, denoted here as $s(k)$. For discrete-time systems, the impulse response can be calculated from step response by

$$h(k) = s(k) - s(k-1). \tag{4.12}$$

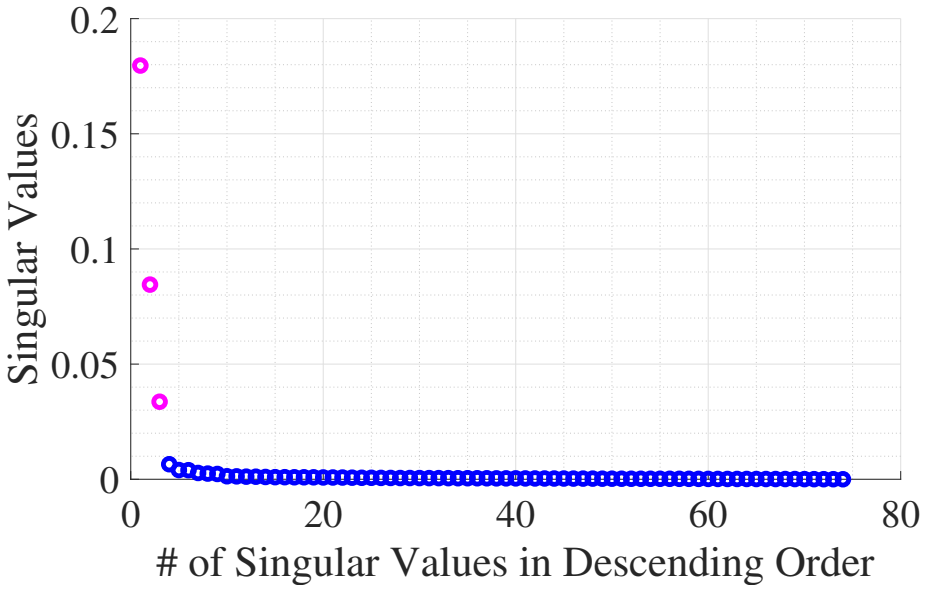


Figure 4.2: Singular values of $\boldsymbol{H}_n$.

Then the Hankel matrix can be constructed as

$$\boldsymbol{H}_n = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots & h_n \\ h_2 & h_3 & h_4 & \cdots & h_{n+1} \\ h_3 & h_4 & h_5 & \cdots & h_{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n+1} & h_{n+2} & \cdots & h_{2n-1} \end{bmatrix}. \tag{4.13}$$

For Single-Input-Single-Output(SISO) system with state-space representation

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}u(k) \\ y(k) &= \boldsymbol{C}\boldsymbol{x}(k) + Du(k) \end{aligned}, \tag{4.14}$$

its impulse response can be rewritten as

$$\begin{aligned} h_0 &= D \\ h_k &= \boldsymbol{C}\boldsymbol{A}^{k-1}\boldsymbol{B}, \quad k = 1, 2, 3... \end{aligned}. \tag{4.15}$$

Therefore, the Hankel matrix can be decomposed as

$$\boldsymbol{H}_n = \boldsymbol{G}_n^O \boldsymbol{G}_n^C, \tag{4.16}$$

where

$$\boldsymbol{G}_n^O = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{n-1} \end{bmatrix}, \ \boldsymbol{G}_n^C = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \tag{4.17}$$

refer to the observability and controllability matrices respectively.

The singular values of $\boldsymbol{H}_n$ are shown in Figure 4.2. It is clear from the figure that the first 3 singular values are dominating the others. Therefore, the Hankel matrix can be approximated with

$$\boldsymbol{H}_n = \begin{bmatrix} \boldsymbol{U}_3 & \boldsymbol{U}_r \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_r \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_3^T \\ \boldsymbol{V}_r^T \end{bmatrix} \approx \boldsymbol{U}_3 \boldsymbol{\Sigma}_3 \boldsymbol{V}_3^T, \tag{4.18}$$

where $\boldsymbol{\Sigma}_3$ refers to the diagonal matrix with the largest 3 singular values, and $\boldsymbol{U}_3$, $\boldsymbol{V}_3$ the corresponding singular vectors.

Therefore, for representation with balanced realization, the observability and controllability matrices can be calculated as

$$\begin{aligned} \boldsymbol{G}_n^O &= \boldsymbol{U}_3 \boldsymbol{\Sigma}_3^{\frac{1}{2}} \\ \boldsymbol{G}_n^C &= \boldsymbol{\Sigma}_3^{\frac{1}{2}} \boldsymbol{V}_3^T \end{aligned}. \tag{4.19}$$

Then

$$\begin{aligned} \boldsymbol{B} &= \mathrm{col}(\boldsymbol{G}_n^C, 1) \\ \boldsymbol{C} &= \mathrm{row}(\boldsymbol{G}_n^O, 1) \end{aligned}. \tag{4.20}$$

Further define

$$\boldsymbol{H}_{n+1} = \begin{bmatrix} h_2 & h_3 & h_4 & \cdots & h_{n+1} \\ h_3 & h_4 & h_5 & \cdots & h_{n+2} \\ h_4 & h_5 & h_6 & \cdots & h_{n+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n+1} & h_{n+2} & h_{n+3} & \cdots & h_{2n} \end{bmatrix}, \tag{4.21}$$

then

$$\boldsymbol{A} = (\boldsymbol{G}_n^O)^\dagger \boldsymbol{H}_{n+1} (\boldsymbol{G}_n^C)^\dagger, \tag{4.22}$$

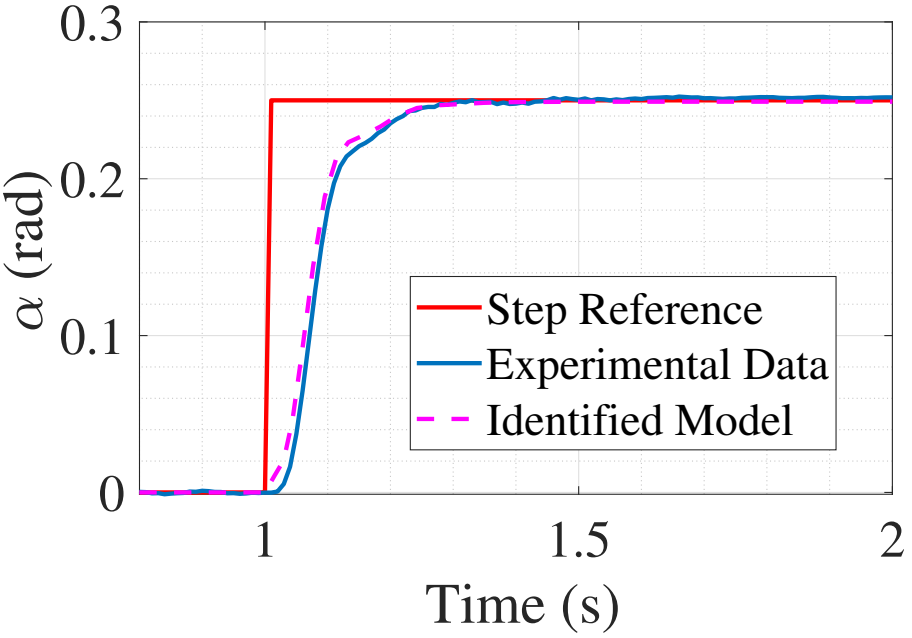


Figure 4.3: Comparison of identified servo dynamics with experimental data under a 0.25 rad step input.

where $[\cdot]^{\dagger}$ refers to the Moore-Penrose Pseudo Inverse of the matrix.

Combining (4.22), (4.20) and (4.15) gives a 3^{rd} transfer function to approximate the closed-loop servo dynamics. A comparison of the identified model with experimental data is shown in Figure 4.3.

4.4 Simulation and Experiments

This section demonstrates different flight cases with the proposed prototype in Figure 3.1. The results verify the capability of the proposed platform for independent control of position and attitude. Simulation and experiments are compared for each case. Every scenario is tested multiple times, and the results are repeatable.

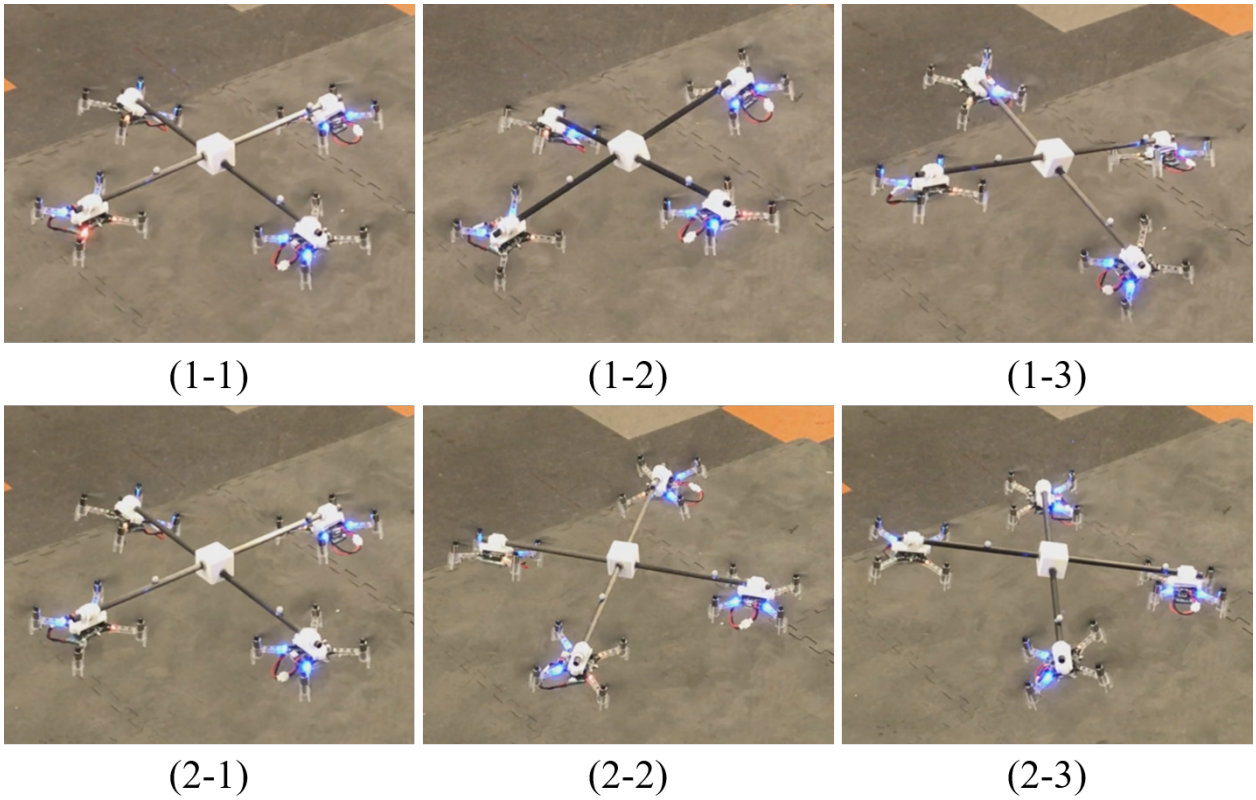


Figure 4.4: Snapshots of attitude tracking while hovering. *Case 1*: Hover at (1-1) $\phi = \theta = 0$; (1-2) $\phi = -\theta = 0.2$ rad; (1-3) $\phi = -\theta = -0.2$ rad. *Case 2*: Hover at (2-1) $\theta = \psi = 0$; (2-2) $\theta = 0.2$ rad, $\psi = -0.6$ rad; (2-3) $\theta = -0.2$ rad, $\psi = 0.6$ rad.

4.4.1 Maximum Hovering Attitude

Section 3.3 has shown full DoF control of the proposed platform without the consideration of physical constraints. However, for the realistic platform, the set of achievable attitudes is limited by the input saturation. This section investigates the maximum achievable attitude of hovering, which is used in latter cases to guide the design of reference trajectories.

The simulation and experiment are shown in Figure 4.6. The pitch angle is tracking a consecutive stair reference signal, while the roll, yaw and platform position are regulated at 0. The stair reference initiates with 0.1 rad and increments 0.5 rad/stair. The stair is maintained at each level for 5 s so that the platform can reach steady state. Consistent results are demonstrated in simulation and experiment, where the control fails after reaching

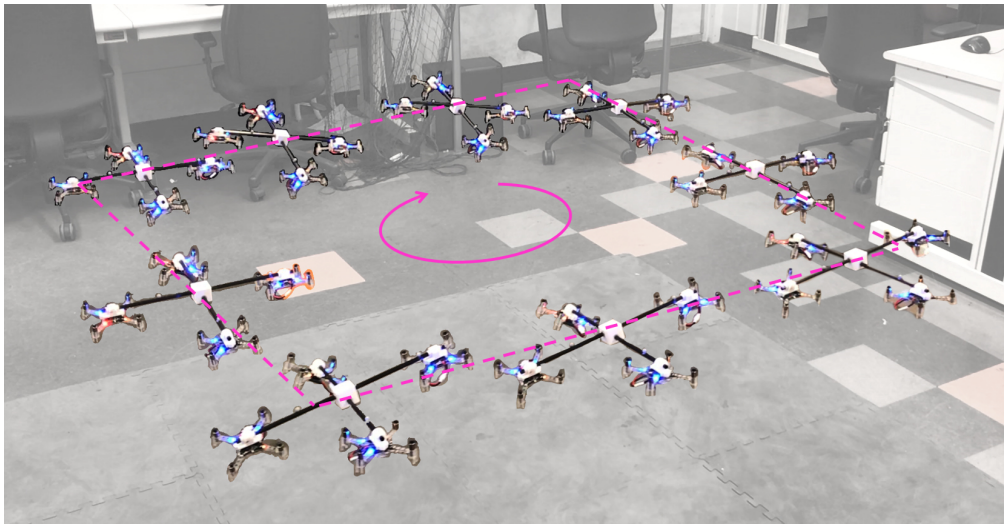
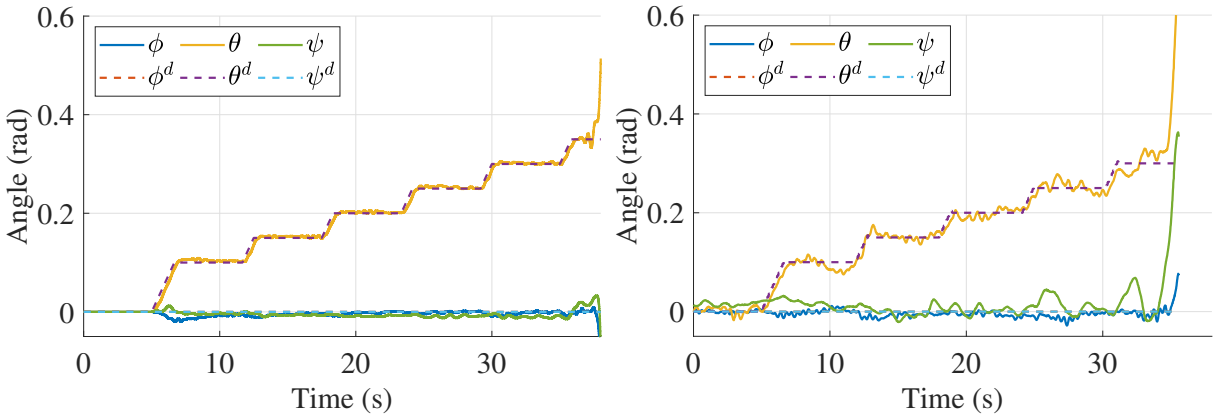


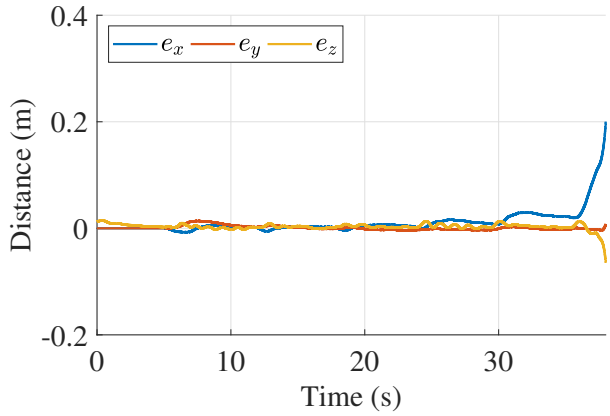
Figure 4.5: Overlay of snapshots for roll and pitch tracking while following a 1 m square trajectory in position.

the step of $\theta = 0.3$ rad. At the failure, the desired thrust forces T_1^d and T_3^d exceed the input saturation. As the platform is symmetric in $\boldsymbol{x}_B$ and $\boldsymbol{y}_B$, identical maximum achievable angle can be expected for the roll angle. Therefore, in the rest of this paper, all attitude trajectories for roll and pitch angles are constrained within this range. The results also verifies that the simulation model captures features of the real platform.

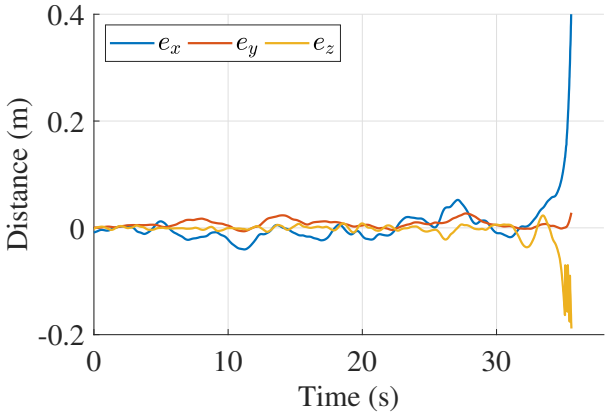


(a) S: Attitude.

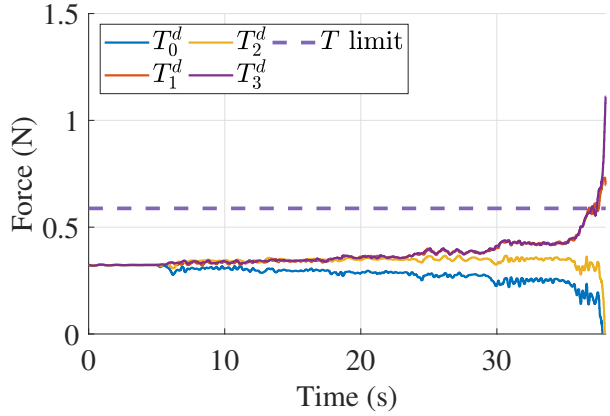
(b) E: Attitude.



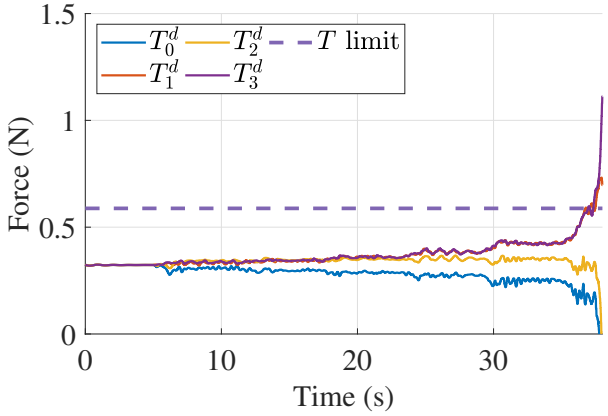
(c) S: Position errors.



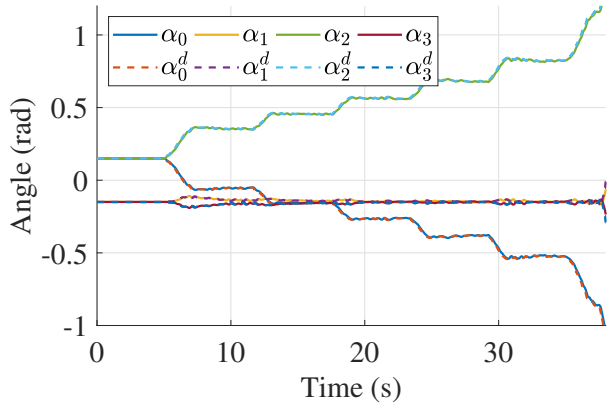
(d) E: Position errors.



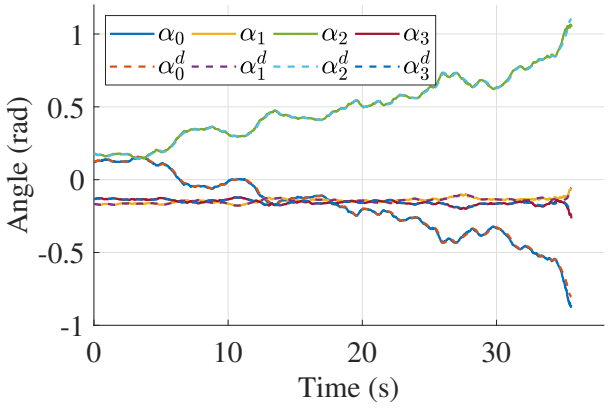
(e) S: Desired thrust forces.



(f) E: Desired thrust forces.

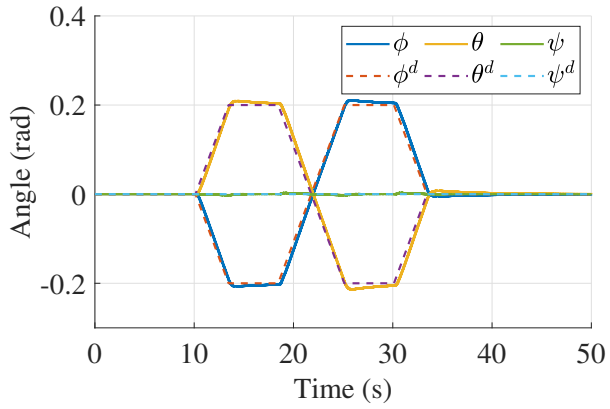


(g) S: Tilting angles.

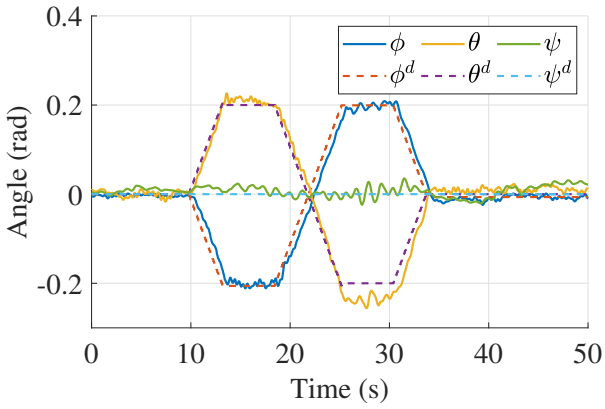


(h) E: Tilting angles.

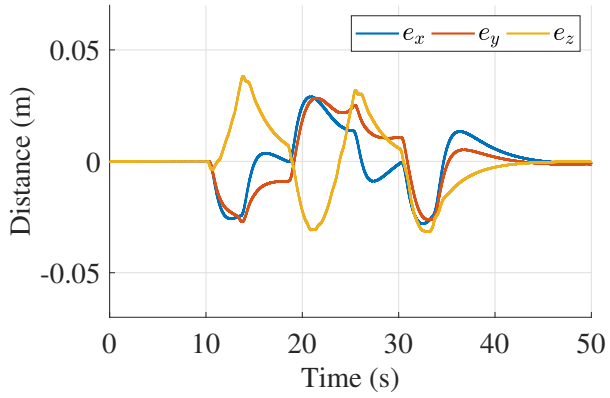
Figure 4.6: Investigation of maximum hovering attitude with existing prototype in Fig. 3.1. The thrust force limit is shown in (c) and (d) by the purple dashed line. (S for simulation, E for experiment)



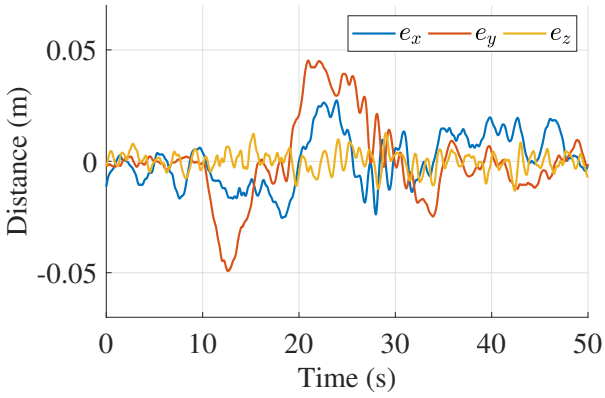
(a) S: Attitude.



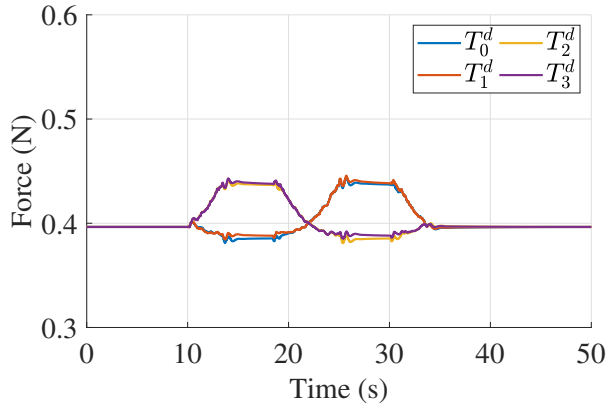
(b) E: Attitude.



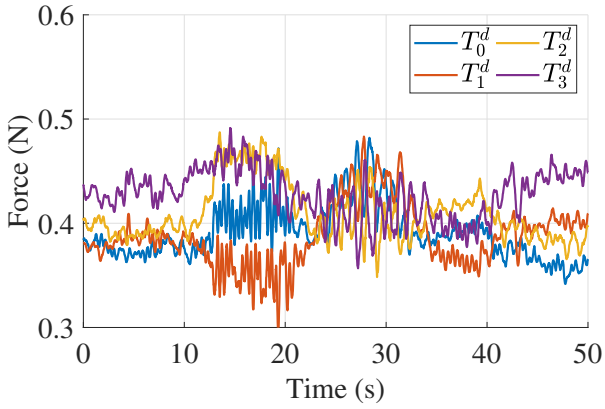
(c) S: Position errors.



(d) E: Position errors.



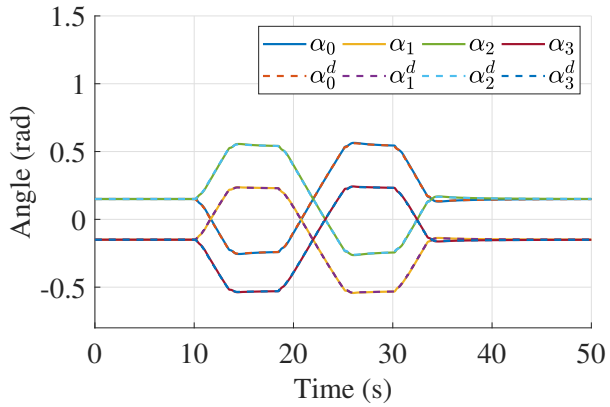
(e) S: Desired thrust forces.



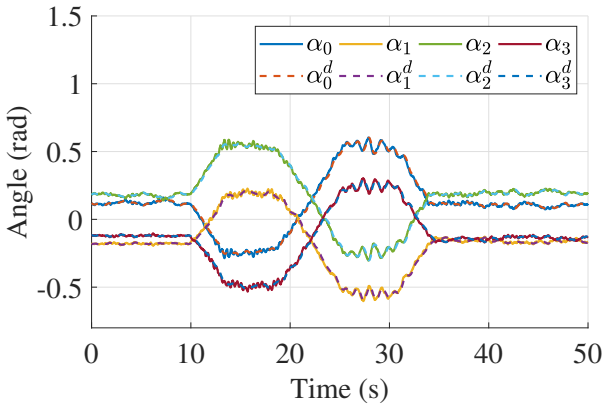
(f) E: Desired thrust forces.

4.4.2 Independent Control of Position and Attitude

Three cases are investigated in this section to demonstrate the decoupling of position and attitude control. The first two cases tracks combined trajectories of Tait-Bryan angles while keep hovering at a specific point. Snapshots of these two experiments are shown in Figure

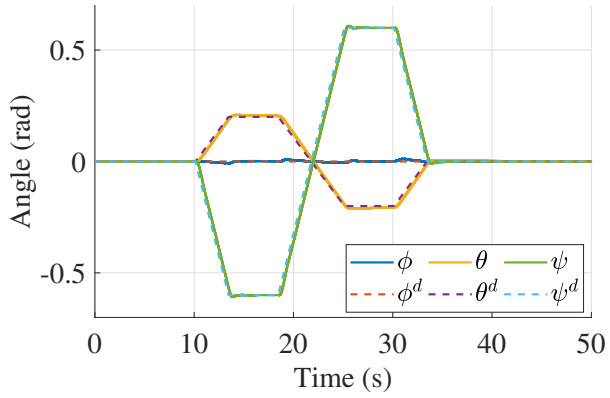


(g) S: Tilting angles.

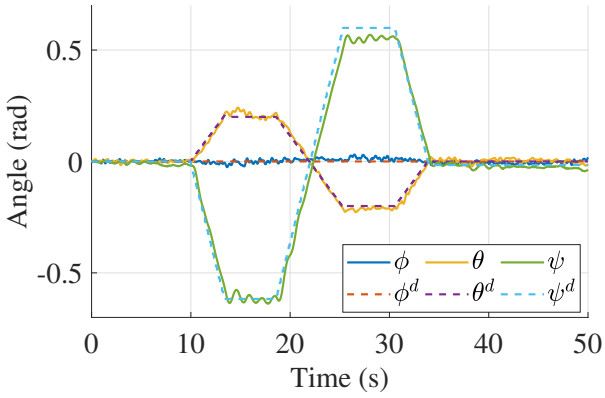


(h) E: Tilting angles.

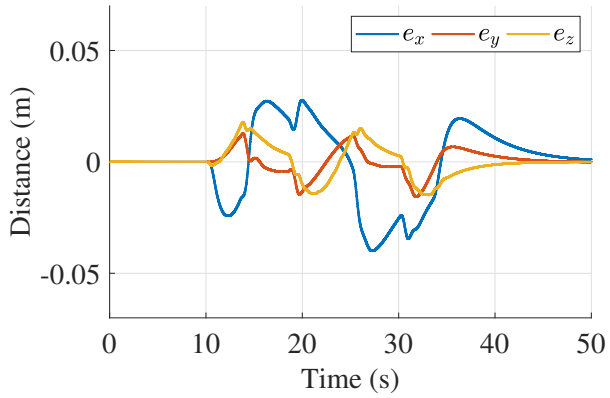
Figure 4.7: Case 1: Tracking roll and pitch angles while hovering. (S for simulation, E for experiment)



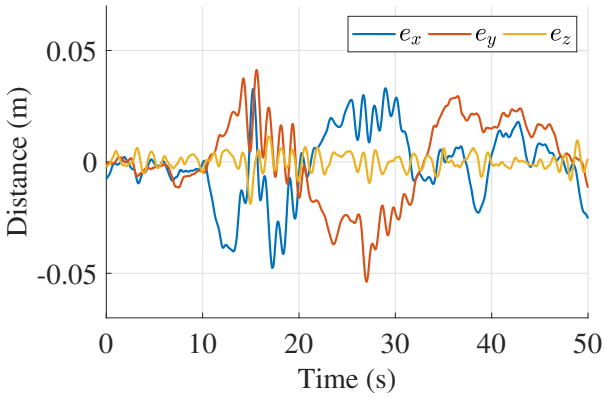
(a) S: Attitude.



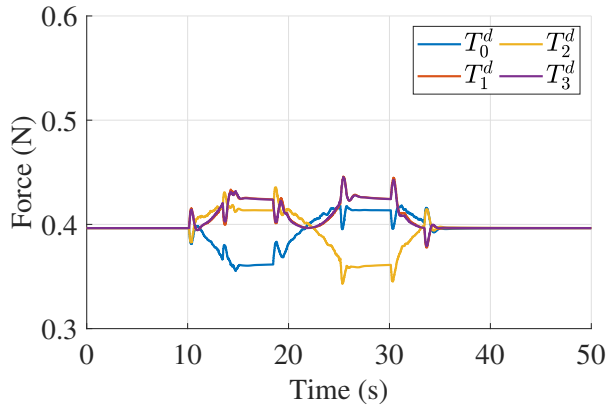
(b) E: Attitude.



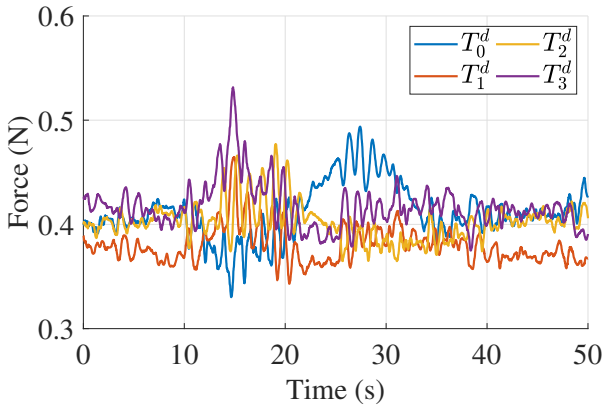
(c) S: Position errors.



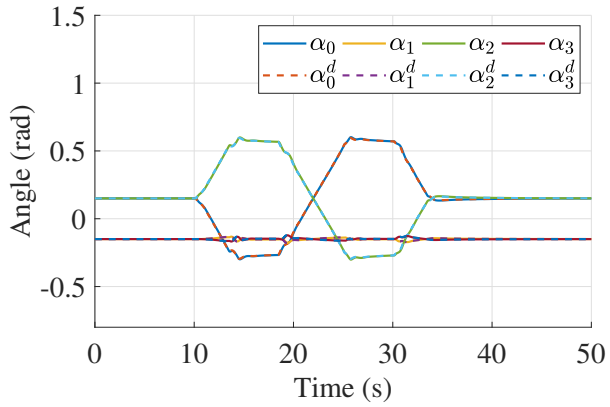
(d) E: Position errors.



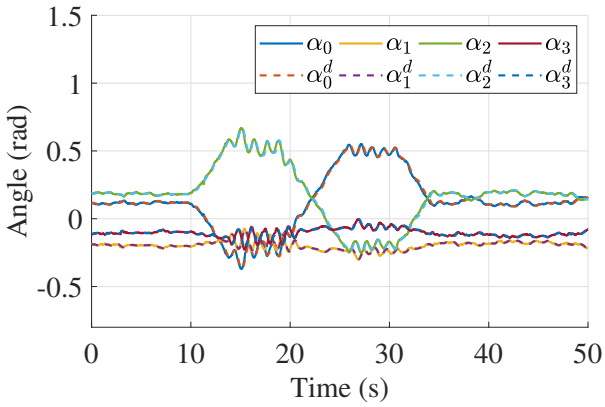
(e) S: Desired thrust forces.



(f) E: Desired thrust forces.

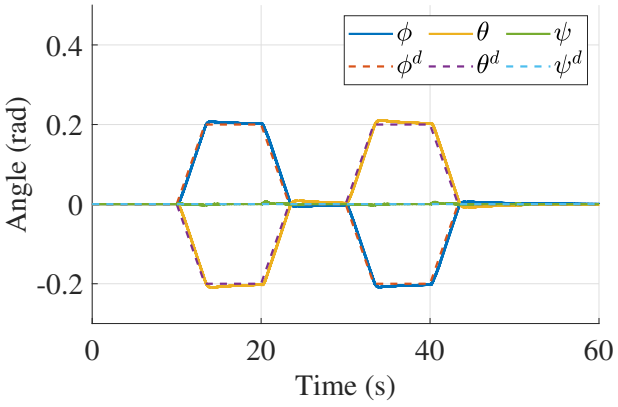


(g) S: Tilting angles.

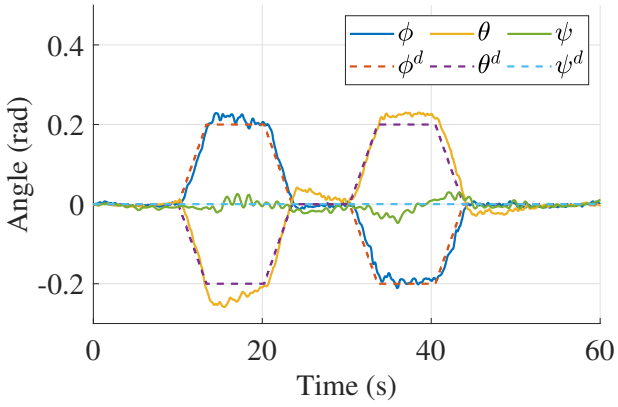


(h) E: Tilting angles.

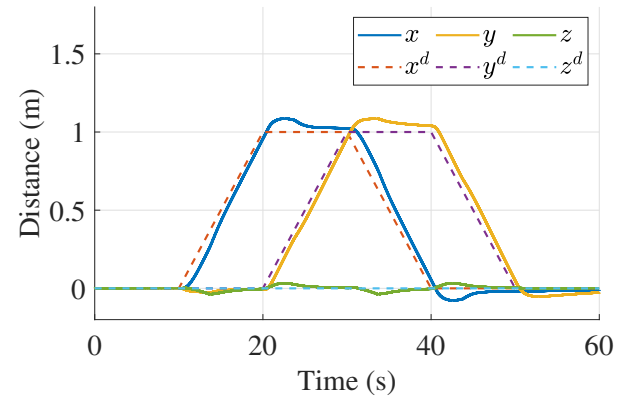
Figure 4.8: Case 2: Tracking pitch and yaw angles while hovering. (S for simulation, E for experiment)



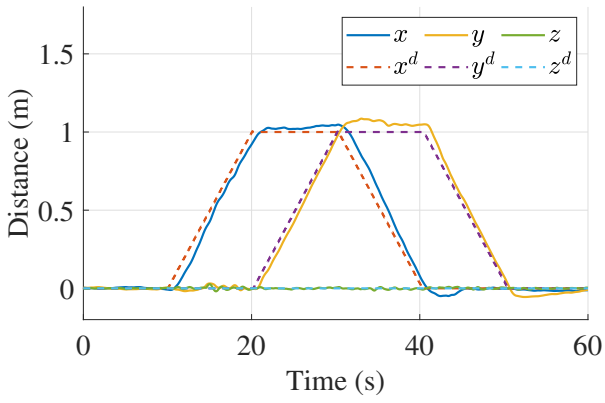
(a) S: Attitude.



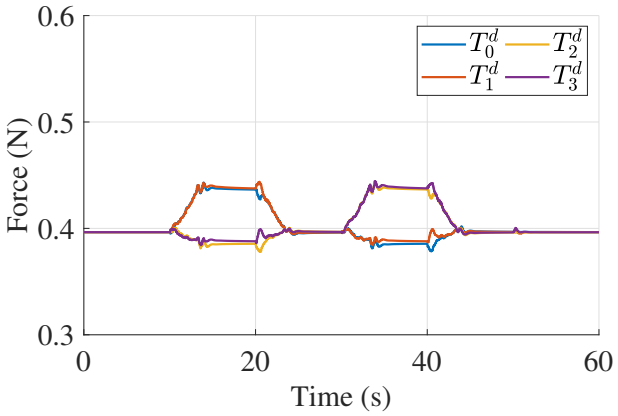
(b) E: Attitude.



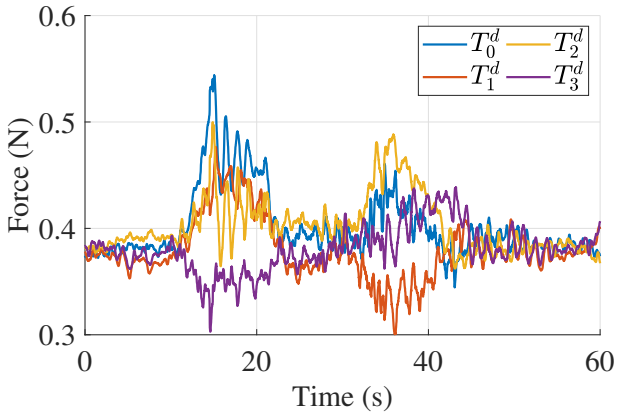
(c) S: Position.



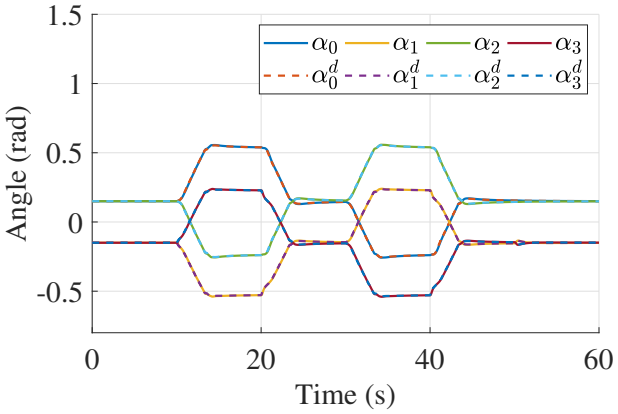
(d) E: Position.



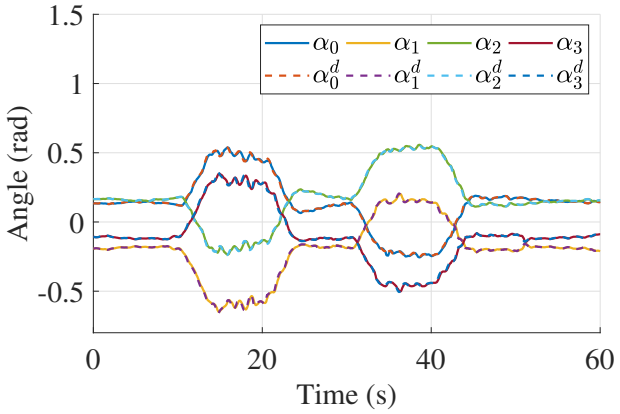
(e) S: Desired thrust forces.



(f) E: Desired thrust forces.



(g) S: Tilting angles.



(h) E: Tilting angles.

Figure 4.9: Case 3: Tracking roll and pitch angles while tracking a square trajectory in position simultaneously. (S for simulation, E for experiment)

4.4. In last case, the platform is tracking moving trajectories for position and attitude respectively. Overlay of snapshots for this experiment is shown in Figure 4.5. The reference signals

Table 4.1: Summary of Rooted-Mean-Square Errors in Simulation (S) and Experiments (E).

	e_ϕ (rad)	e_θ (rad)	e_ψ (rad)	e_x (m)	e_y (m)	e_z (m)
S1	0.008	0.009	0.001	0.012	0.013	0.014
E1	0.017	0.018	0.013	0.012	0.018	0.004
S2	0.003	0.009	0.019	0.017	0.005	0.007
E2	0.009	0.012	0.033	0.016	0.019	0.004
S3	0.008	0.008	0.001	0.048	0.042	0.014
E3	0.011	0.020	0.013	0.052	0.041	0.005

for all cases are designed to be in a slow manner in order to test the system stabilization at different attitudes, as indicated in [RBG14].

Case 1: Tracking roll and pitch angles while hovering. The trajectories of roll and pitch angles are bounded by ± 0.2 rad, and cover both directions. The position of the platform is regulated at a fixed location at all time. Simulation and experimental results are shown in Figure 4.7.

Case 2: Tracking pitch and yaw angles while hovering. The trajectories of pitch and yaw angles are bounded by ± 0.2 rad and ± 0.6 rad respectively, and cover both directions. The position of the platform is still regulated at a specific location. Simulation and experimental results are shown in Figure 4.8.

Case 3: Tracking roll and pitch angles while tracking a square trajectory in position simultaneously. The trajectories of roll and pitch angles are bounded by ± 0.2 rad, and cover both directions. The position trajectory is a 1 m square in space with constant height. Simulation and experimental results are shown in Figure 4.9.

4.4.3 Discussion

The rooted-mean-square (RMS) errors of the 3 aforementioned cases are summarized in TABLE 4.1. It can be observed that the error for simulation and experiment of the same case is of the same level, indicating that the simulation model maintains the main features of the prototype. Comparing the errors of case 1 and case 3 shows that switching the position control from hovering to following a changing trajectory does not influence the tracking error of attitude, which verifies the decoupling of position and attitude control. However, it can be noticed that the position RMS error increases significantly from case 1 to case 3. That is because the mapper in the proposed controller, though computationally effective and avoiding abrupt changes in α_i under smooth reference signals for attitude with inverse tangent function as the shaping function, sacrifices the dynamic property of position tracking.

Comparing with case 1 and case 2 for the desired thrust forces, it can be noticed that ripples occur every time there is a non-differentiable point in the attitude reference. However, the ripples in case 2 is more significant. This can be explained by Equation (4.8). As the tilting angles α_i is relatively small, $c\alpha_i$ is larger than $s\alpha_i$, so a change of required torque in $\boldsymbol{z}_B$ requires larger changes in the thrust forces than $\boldsymbol{x}_B$ or $\boldsymbol{y}_B$.

Another interesting finding is that e_z in experiments are smaller than simulation for all 3 cases. This is expected as the simulation model does not include damping from the air. As the experiments are conducted in a constrained attitude space, the projection of the prototype on the $\boldsymbol{z}$ axis is much larger than that in $\boldsymbol{x}$ or $\boldsymbol{y}$ axis, resulting in larger damping and thus smaller RMS error in the $\boldsymbol{z}$ direction.

4.5 Conclusion

This chapter presented a hierarchical control architecture for the proposed fully actuated multirotor aerial platform, to realize independent control of position and attitude with low computation burden and smoothed reference trajectories for actuator tilting angles α_i. Experimental environment were constructed with motion capture system, ground station and

wireless communication. Closed-loop servo dynamics were identified to improve simulation accuracy. The controller were demonstrated with different flight cases in simulation and experiments.

However, the platform only exhibited a limited range of achievable attitudes due to the saturation of thrust force. As the proposed platform has overactuation in dynamics, the nullspace of inputs allocation can be explored so that the desired external force is generated with the least possible thrust forces. This shall be proposed in the next chapter, to improve the maximum achievable attitude under the same saturation constraints.

CHAPTER 5

Controller Design and Implementation for Increased Range of Achievable Attitudes under Input Saturation

5.1 Introduction

Previous chapters proposed a novel multirotor aerial platform with commercial quadcopters and passive hinges as tiltable thrusting actuator, and a controller to realize independent control of position and attitude. However, the controller in the last chapter generates smooth trajectories of desired actuator tilting angles α_i^d, but degrades the control performance. It is shown in Figure 4.6 that the platform hovering fails when the roll/pitch angle goes beyond 0.35 rad, due to the saturation of thrust forces. The goal of this chapter is to design a new controller which can increase the range of achievable attitude of hovering under the same conditions for input saturation.

Various controllers have been addressed on the tiltable actuator multirotors, such as [JMG19], [SKR17], [KNK17], and [NK14]. However, only a few successfully validated their designs through experiments, of which most applied a hierarchical structure, similar to Figure 4.1. The outer loop generally includes position and attitude controllers in free 3D space, as shown in section 4.2.2. The inner loop is running on tiltable actuators, to generate thrust forces and tilting angles based on desired values. A mapper is used to allocate the desired external force and torque to the desired inputs of the platform, and is usually the key difference of these methods. However, the critical point of the mapper is to decouple tilting angle inputs and thrust force inputs from multiplication. [KVE18] did this by change of variables, where 8 new inputs, each defined as a multiplication of thrust force and triangular function of tilting angle, were defined. A least square mapping is then used to calculate

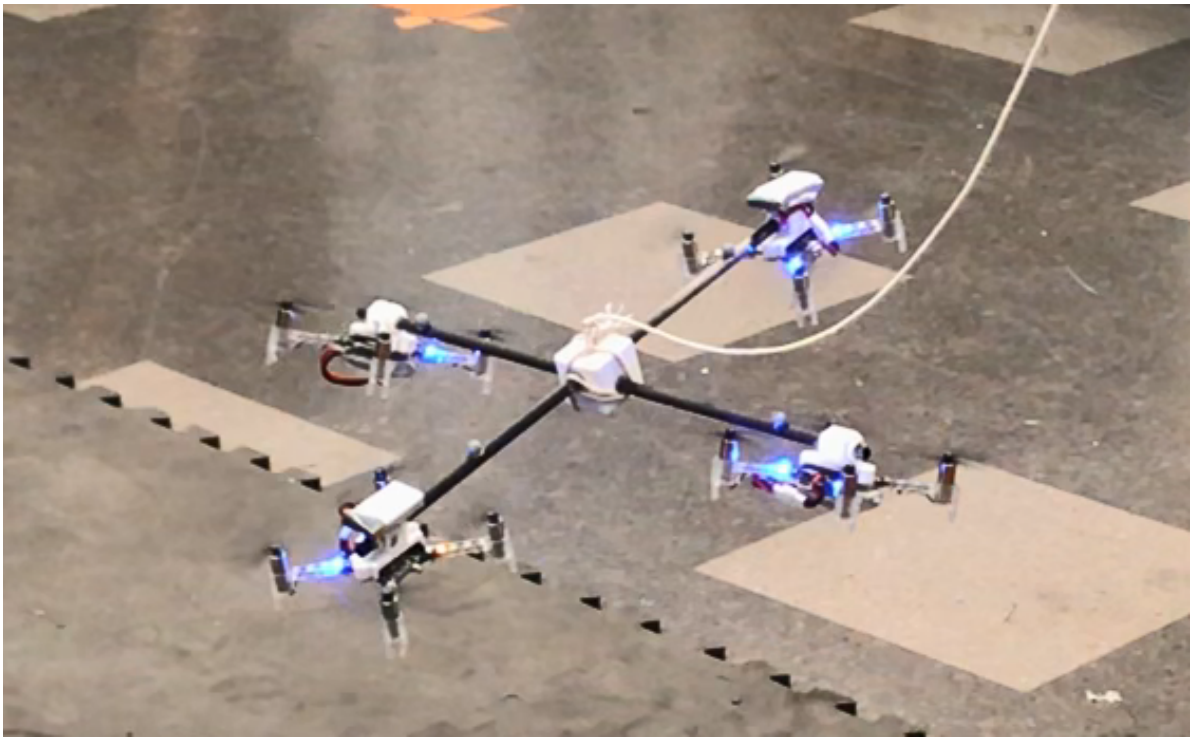


Figure 5.1: The tiltable actuator aerial platform in the air.

these inputs, and the original inputs are recovered through the reverse nonlinear change of variables. This mapping is straightforward and computationally effective, but is hard to take into consideration the input saturation, as the physical representations of new inputs are not clear. On the other hand, [RBG14] proposed another mapping scheme, which decoupled the inputs by taking derivatives of the original dynamics. The new system is feedback linearizable, with increments of thrust forces and tilting angles as the new inputs. Compared with [KVE18], this scheme makes use of the overactuation of dynamics by formulating an online optimization to investigate the null space in mapping, and is thus able to explicitly constrain the desired thrust forces by saturation. However, the model predictive methodology greatly increases the computational burden, and also requires the model to be precise.

This chapter proposes a novel controller, of which the mapping scheme is extended from [KVE18] by exploring the nullspace of input allocation, in order to search for the least possible thrust forces required to hover at a specified attitude. The range of achievable attitudes for hovering is therefore maximized.

The rest of this chapter is organized as following. Section 5.2 describes the least square mapping scheme on which the the proposed controller is based. Section 5.3 formulates the deployment of thrust forces as a convex problem to search for the least possible thrust forces. The analytical solution can be obtained for this optimization. The proposed controller is compared with the controller in [KVE18] in simulation and experiments for hovering with non-horizontal attitudes. The conclusion is in section 5.5.

5.2 Least Square Mapping with Change of Variables

The mapping scheme applied in this section is analogous to the one proposed in [KVE18], but adapted to fit the quadrotor configuration. By defining a set of new input variables, the system addresses decoupled full actuation, and is feedback linearizable for controller design. A nonlinear mapping is then adopted to resume the original system inputs.

5.2.1 Methodology

5.2.1.1 Change of Variables

In order to reformulate the dynamics for change of variables, two assumptions are made.

- The gyroscopic effect of rotational dynamics is neglectable.
- $\boldsymbol{\tau}_M = \mathbf{0}$ at all time.

The simplified dynamics under these assumptions is expressed in the matrix form as

$$\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}{}^{W}\boldsymbol{R}_B & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T} + \begin{bmatrix} \boldsymbol{G} \\ \mathbf{0} \end{bmatrix}, \tag{5.1}$$

where

$$\begin{aligned}
\boldsymbol{J}_\xi &= \begin{bmatrix} -s\alpha_0 & 0 & s\alpha_2 & 0 \\ 0 & s\alpha_1 & 0 & -s\alpha_3 \\ c\alpha_0 & c\alpha_1 & c\alpha_2 & c\alpha_3 \end{bmatrix}, \\
\boldsymbol{J}_\eta &= \begin{bmatrix} -c\alpha_0 l & 0 & c\alpha_2 l & 0 \\ 0 & c\alpha_1 l & 0 & -c\alpha_3 l \\ s\alpha_0 l & s\alpha_1 l & s\alpha_2 l & s\alpha_3 l \end{bmatrix}, \\
\boldsymbol{T} &= \begin{bmatrix} T_0 & T_1 & T_2 & T_3 \end{bmatrix}^T.
\end{aligned} \tag{5.2}$$

Define

$$\boldsymbol{F} = \begin{bmatrix} F_{s0} & \dots & F_{s3} & F_{c0} & \dots & F_{c3} \end{bmatrix}^T, \tag{5.3}$$

where

$$F_{si} = s\alpha_i T_i,\ F_{ci} = c\alpha_i T_i, \tag{5.4}$$

then the dynamics in (5.1) is rewritten as

$$\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}{}^W\boldsymbol{R}_B & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}^{-1} \end{bmatrix} \boldsymbol{W}\boldsymbol{F} + \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix}, \tag{5.5}$$

where $\boldsymbol{W}$ is a constant matrix

$$\boldsymbol{W} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & -l & 0 & l & 0 \\ 0 & 0 & 0 & 0 & 0 & l & 0 & -l \\ l & l & l & l & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.6}$$

with full row rank, thus having a Moore-Penrose pseudo inverse $\boldsymbol{W}^\dagger$ such that

$$\boldsymbol{W}\boldsymbol{W}^\dagger = \boldsymbol{I}_6. \tag{5.7}$$

Obviously, the dynamics in (5.5) is overactuated, as indicated by the null space of $\boldsymbol{W}$.

5.2.1.2 Feedback Linearization

A feedback linearized controller is designed for the dynamics in (5.5). Specifying the input vector as

$$\boldsymbol{F} = \boldsymbol{W}^{\dagger} \begin{bmatrix} m({}^{W}\boldsymbol{R}_{B})^{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{u}_{\xi} \\ \boldsymbol{u}_{\nu} \end{bmatrix} - \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix} \right), \tag{5.8}$$

the dynamics is reduced to

$$\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_{\xi} \\ \boldsymbol{u}_{\nu} \end{bmatrix}, \tag{5.9}$$

where $\boldsymbol{u}_{\xi}$ and $\boldsymbol{u}_{\nu}$ are virtual inputs for position and attitude dynamics respectively.

Two second-order tracking controllers are designed on the virtual inputs as

$$\begin{aligned} \boldsymbol{u}_{\xi} &= \ddot{\boldsymbol{\xi}}^{d} + K_{\xi 1}\dot{\boldsymbol{e}}_{\xi} + K_{\xi 2}\boldsymbol{e}_{\xi} + K_{\xi 3}\int \boldsymbol{e}_{\xi} dt, \\ \boldsymbol{u}_{\nu} &= \ddot{\boldsymbol{\nu}}^{d} + K_{\nu 1}\dot{\boldsymbol{e}}_{\nu} + K_{\nu 2}\boldsymbol{e}_{\eta} + K_{\nu 3}\int \boldsymbol{e}_{\eta} dt, \end{aligned} \tag{5.10}$$

where the errors are defined in [LLM10], and explicitly shown in (4.4).

Substituting (5.10) to (5.8) gives the new inputs $\boldsymbol{F}$, from which the original inputs can be resumed by

$$T_i = \sqrt{F_{si}^2 + F_{ci}^2}, \tag{5.11}$$

and

$$\alpha_i = \text{atan2}(F_{si}, F_{ci}) \tag{5.12}$$

for

$$\alpha_i \in [-\pi, \pi).$$

5.2.2 Simulation and Experiments

5.2.2.1 Realistic Simulation Setup

The realistic simulation is established under Matlab Simulink environment. The SimScape toolbox is utilized to establish the dynamic model of the platform in a physical manner, where the connections and relative constraints between physical components are described by the

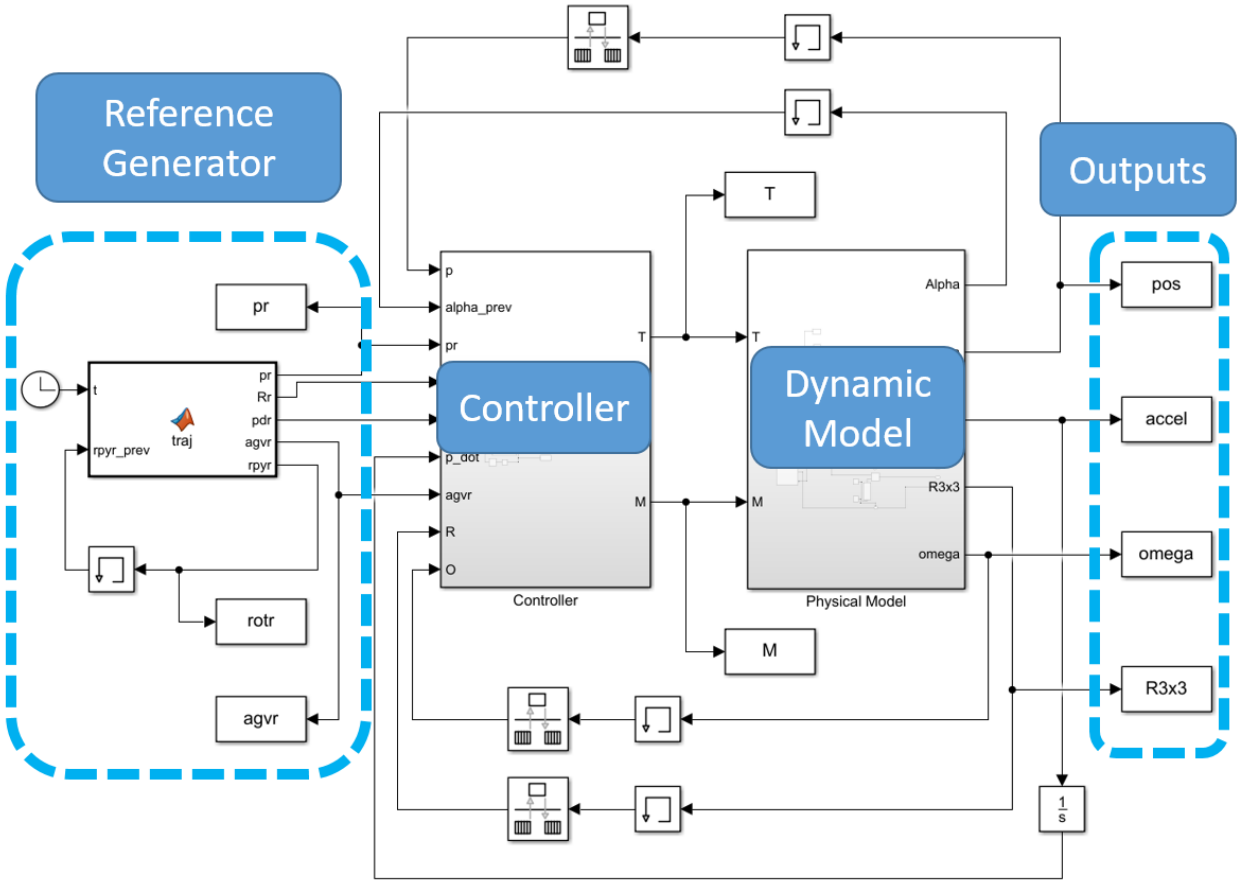


Figure 5.2: The overall simulation diagram, with dynamic model of the proposed platform, controller and generator for 6 DoF reference signals.

connections of SimScape blocks. The SimScape model has direct interface with Simulink diagrams.

The overall block diagram of simulation is shown in Figure 5.2. The diagram mainly consists of 3 parts. The reference generator provides desired trajectories for 6 DoF position and attitude. The controller takes in the reference signals and the measurements of 6 DoF motion outputs of the platform, and calculates the thrust force and tilting torque command for each quadcopter. The dynamic model is constructed by SimScape toolbox.

The simulation diagram of the controller is shown in Figure 5.3. It follows the structure described in Figure 4.1. The outer loop controllers takes in the desired reference trajectories

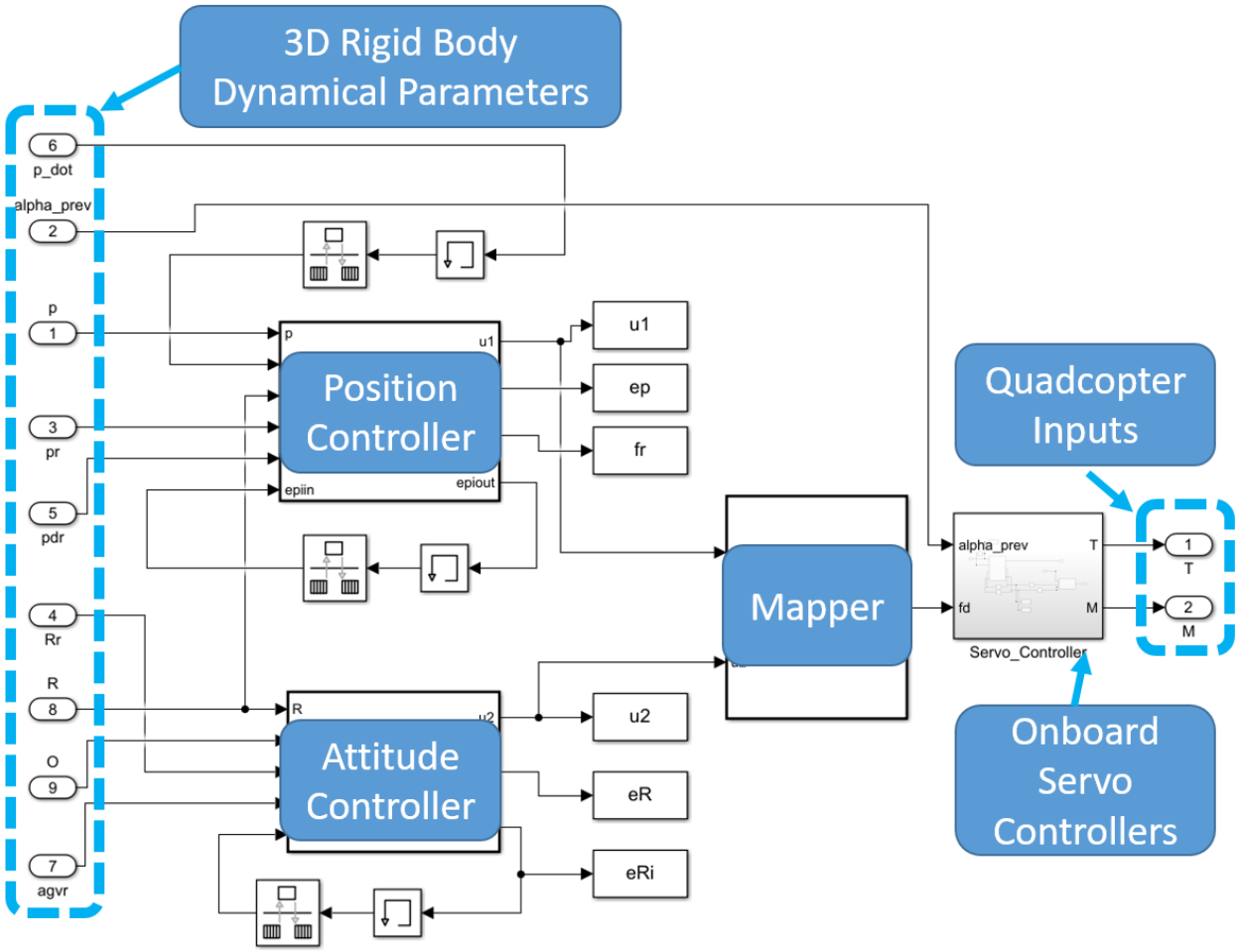


Figure 5.3: The simulation diagram of the controller, which consists of the outer loop tracking controllers for position and attitude, the mapper that allocates the desired external torque and force onto each quadcopter as thrust vectors, and the onboard servo controllers on quadcopters to calculate the desired total thrust forces and torques along the hinge directions.

for position and attitude, in addition to the 6 DoF measurements of the platform, and calculates the desired total external force and torque on the platform. The mapper then allocates the force and torque onto each quadcopters as thrust vectors. The servo controllers calculate the thrust forces and tilting torques required to generate such thrust vectors. Notice that the communication delay characterized in section 3.4.3 is included in the servo controller diagrams.

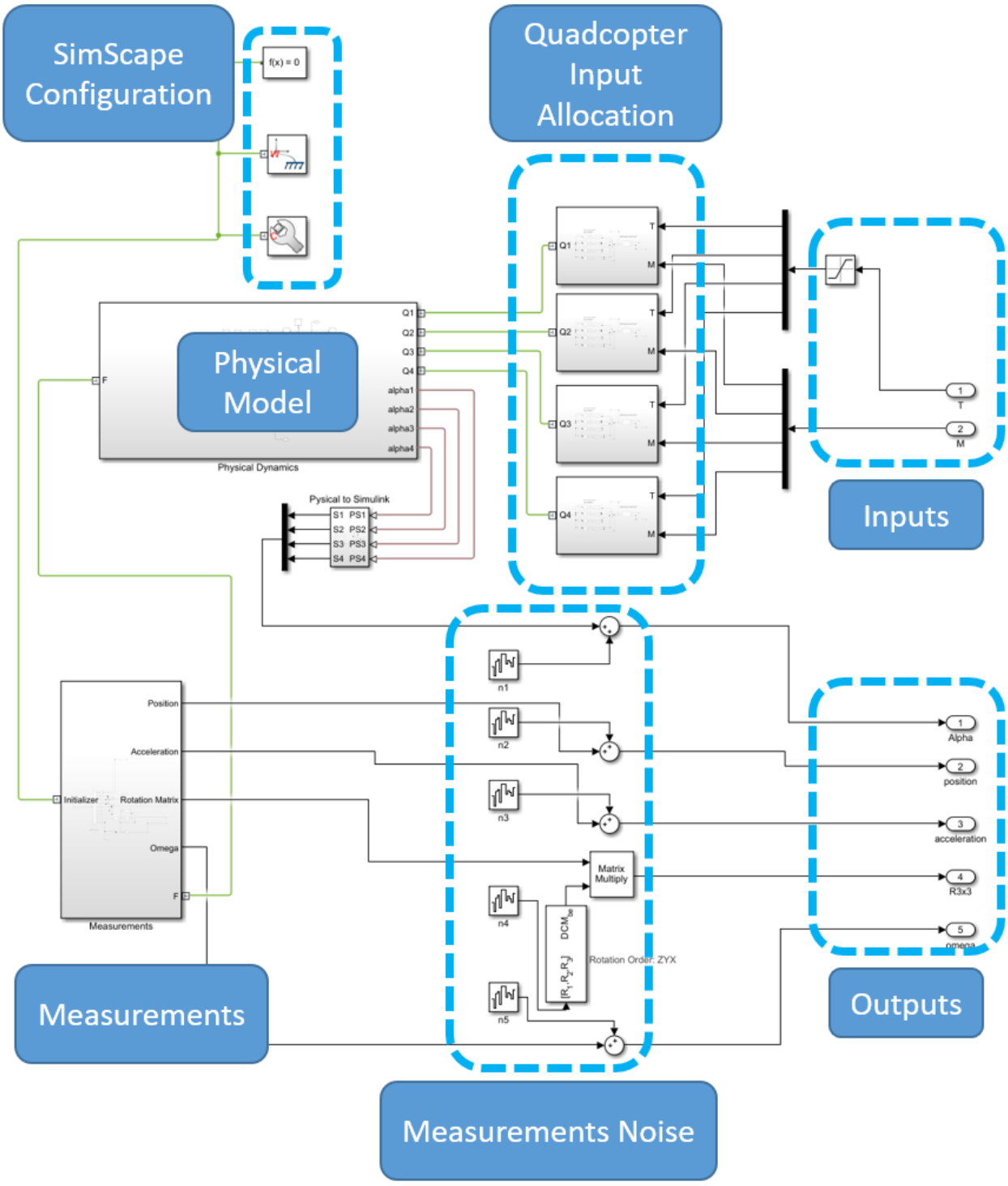


Figure 5.4: The simulation diagram of the dynamic model, which mainly consists of the physical model, measurements and quadcopter input allocation.

Figure 5.5: The CAD model of the proposed aerial platform, which is utilized to construct physical model with Simulink SimScape toolbox.

The simulation diagram of the dynamic model is shown in Figure 5.4. The major part is the physical model, which is constructed with SimScape toolbox and express the platform dynamics in a more realistic method. The SimScape configuration defines the gravity, ground and the world frame that is attached to it. It also selects the solver for the physical model. The measurements of platform 6 DoF information are taken, and proper noises are artificially added before the measurements are sent to the controller. The quadcopter input allocation part distribute the required thrust force and torque of each quadcopter to its propeller speeds.

The physical model of the proposed aerial platform is based on it CAD model, as shown in Figure 5.5. The block diagram is shown in Figure 5.6. Clearly, the model can be divided into 3 parts, the central frame, the passive hinges and the quadcopters. In this approach, the platform is analyzed with multi-body dynamics, instead of the conventional mathematical model, which describes the system as a rigid body and 4 tilting thrust forces with inner dynamics.

The quadcopter input allocation block is detailed in Figure 5.7. The desired thrust force and tilting torque are allocated onto the thrust forces of the 4 propeller-motor actuators

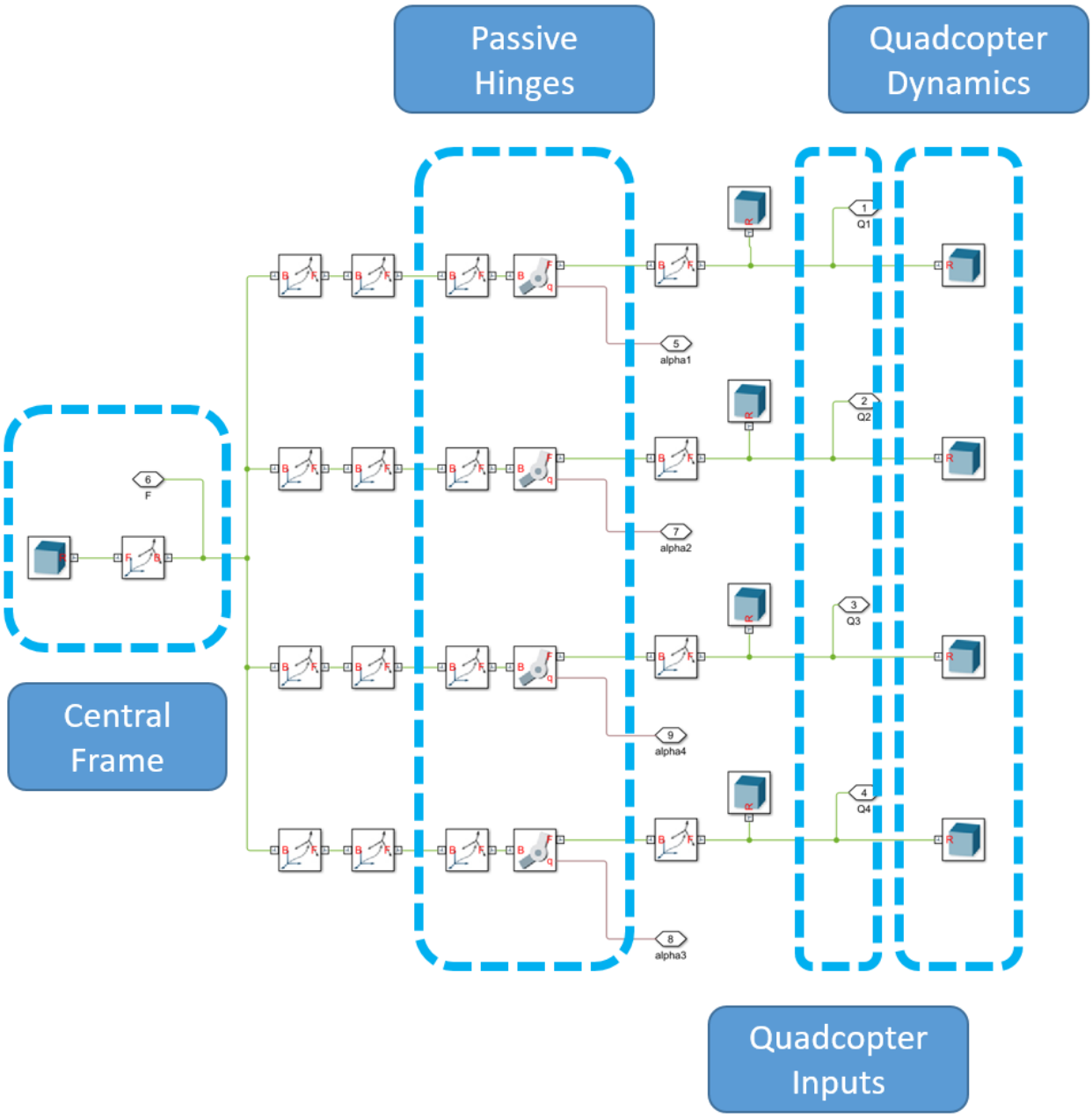


Figure 5.6: The simulation diagram of the physical model established with SimScape toolbox, which contains the models for central frame, passive hinges and quadcopters.

of each quadcopter. The spinning speeds are then calculated based on the thrust forces, and converted to servo commands to the motor drivers. The saturation is directly put on the spinning speed, instead of total thrust forces provided by the quadcopter, to make the simulation realistic. The Simulink-to-physical converter block is the communication interface

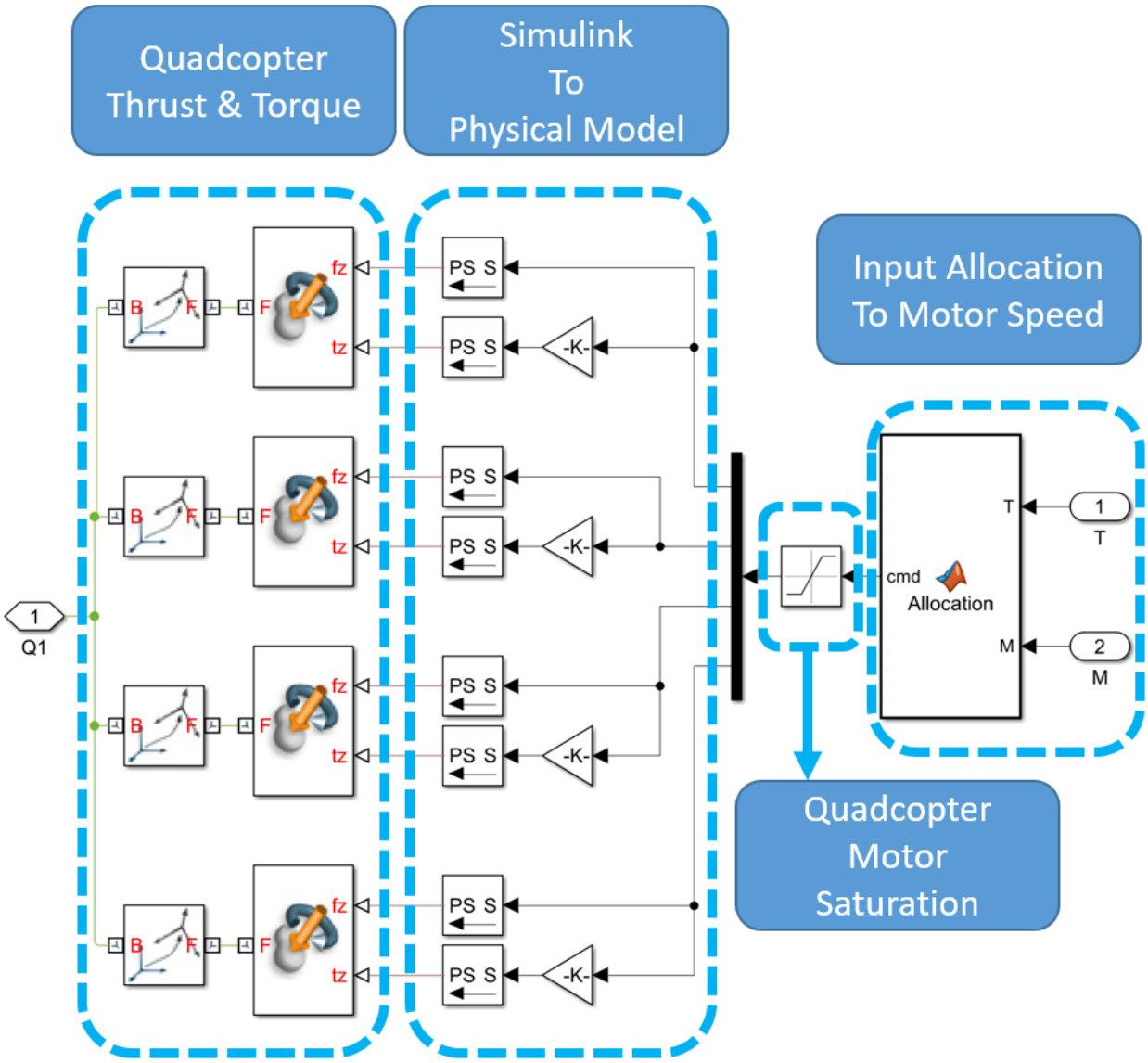


Figure 5.7: The simulation diagram of the quadcopter input allocation, which takes in the desired thrust force and torque of each quadcopter, and calculates the corresponding spinning speed of each propeller-motor actuator. The signals are then converted into physical quantities and exerted on the physical model.

that converts between physical quantities and mathematical signals.

5.2.2.2 Ramp Test for Maximum Achievable Pitch Angle

In this chapter, the pitch angle is used to indicate the attitude for simplicity, but all results can be directly transferred to roll angle. The maximum achievable pitch angle is acquired in simulation and experiments through a ramp test during the flight, where the platform hovers at a fixed position, and its pitch angle tracks a ramp reference. The references of roll and yaw angles are set zero at all time. The 6 DoF trajectories designed for the ramp test is written as

$$\begin{bmatrix} \boldsymbol{\xi}^d \\ \boldsymbol{\eta}^d \end{bmatrix} = \begin{bmatrix} x^d(t) \\ y^d(t) \\ z^d(t) \\ \phi^d(t) \\ \theta^d(t) \\ \psi^d(t) \end{bmatrix} = \begin{bmatrix} 0 & \text{m} \\ 0 & \text{m} \\ 0 & \text{m} \\ 0 & \text{rad} \\ \beta t & \text{rad} \\ 0 & \text{rad} \end{bmatrix}, \tag{5.13}$$

where β is the selected ramp slope.

5.2.2.3 Defining Achievable Pitch Angle for the Ramp Test

During the ramp test, a mathematical criterion is required to determine whether a pitch angle is achievable or not. The largest achievable pitch angle is used to compare the capability of different controllers.

Firstly, the summed error for platform position and attitude at time t are defined respectively as

$$\begin{aligned} e_\xi(t) &= \sqrt{e_x^2(t) + e_y^2(t) + e_z^2(t)} \\ e_\eta(t) &= \sqrt{e_\phi^2(t) + e_\theta^2(t) + e_\psi^2(t)} \end{aligned}. \tag{5.14}$$

The controller is considered failed at time t when the tracking errors exceed the threshold in attitude

$$t \in \mathcal{S}_F^\eta = \{t_0 : e_\eta(t_0) \geq \epsilon_\eta, e_\eta(t) \geq \epsilon_\eta, \forall t \geq t_0\}, \tag{5.15}$$

or in position

$$t \in \mathcal{S}_F^\xi = \{t_0 : e_\xi(t_0) \geq \epsilon_\xi, e_\xi(t) \geq \epsilon_\xi, \forall t \geq t_0\}, \tag{5.16}$$

where ϵ_η and ϵ_ξ are the selected thresholds of errors in attitude and position respectively.

The achievable pitch angles are defined as those from the $\theta^d(t)$ trajectory when the controller does not fail. The set of achievable pitch angles are described as

$$\mathcal{S}_A^\theta = \{\theta : \theta = \theta^d(t), t \in [0, t_f] - \mathcal{S}_F^\xi - \mathcal{S}_F^\eta\}. \tag{5.17}$$

The maximum achievable pitch angle is therefore obtained by

$$\theta_m = \max \theta, \ \ \theta \in \mathcal{S}_A^\theta. \tag{5.18}$$

5.2.2.4 Simulation

The simulation is conducted under Matlab Simulink/Simscape environment, as elaborated in section 5.2.2.1. Apart from the platform dynamics, the simulation also includes input saturation, control sampling and zero-order-hold, communication delay and propeller motor dynamics.

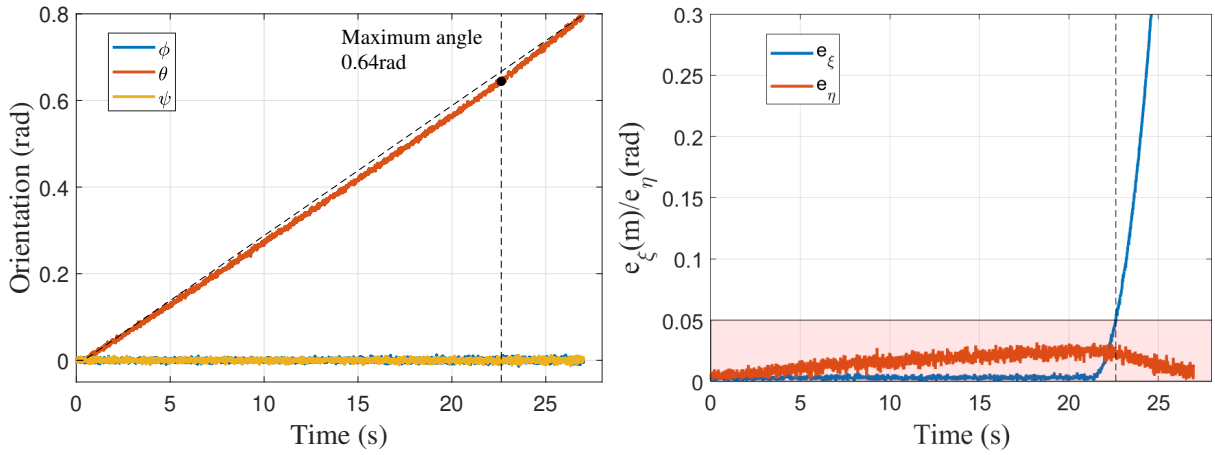


Figure 5.8: Simulation of attitude tracking results of the least square mapping scheme with change of variables. The summed error signals for position and attitude are shown on the right figure. The maximum achievable pitch angle is obtained when the first tracking failure happens.

The ramp test is conducted in simulation with the proposed controller in this section. The attitude tracking result and the corresponding error signals are shown in Figure 5.8.

The error tolerances are selected to be

$$\begin{aligned} \epsilon_\eta &= 0.05 \text{ rad} \\ \epsilon_\xi &= 0.05 \text{ m} \end{aligned}, \tag{5.19}$$

by which the maximum achievable pitch angle is found to be

$$\theta_m = 0.64 \text{ rad}. \tag{5.20}$$

It can be observed that after reaching θ_m, though the tracking of pitch angle remains within error tolerance, the position drifts quickly, so the tracking controller fails.

5.2.2.5 Experiments

The corresponding experiments are conducted at UCLA MacLab, on the prototype proposed in Chapter 3. The attitude tracking results and corresponding error signals are shown in Figure 5.9. Snapshots of the experiment are shown in Figure 5.10.

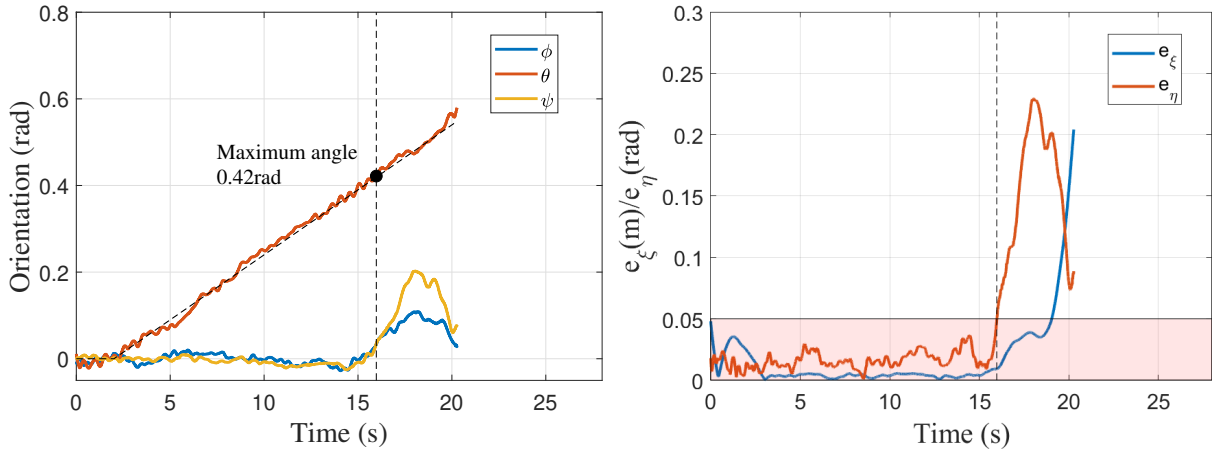


Figure 5.9: Experimental attitude tracking results of the least square mapping scheme with change of variables. The summed error signals for position and attitude are shown on the right figure. The maximum achievable pitch angle is obtained when the first tracking failure happens.

Selecting the same set of error tolerances (5.19), the maximum achievable pitch angle is found as

$$\theta_m = 0.42 \text{ rad}. \tag{5.21}$$

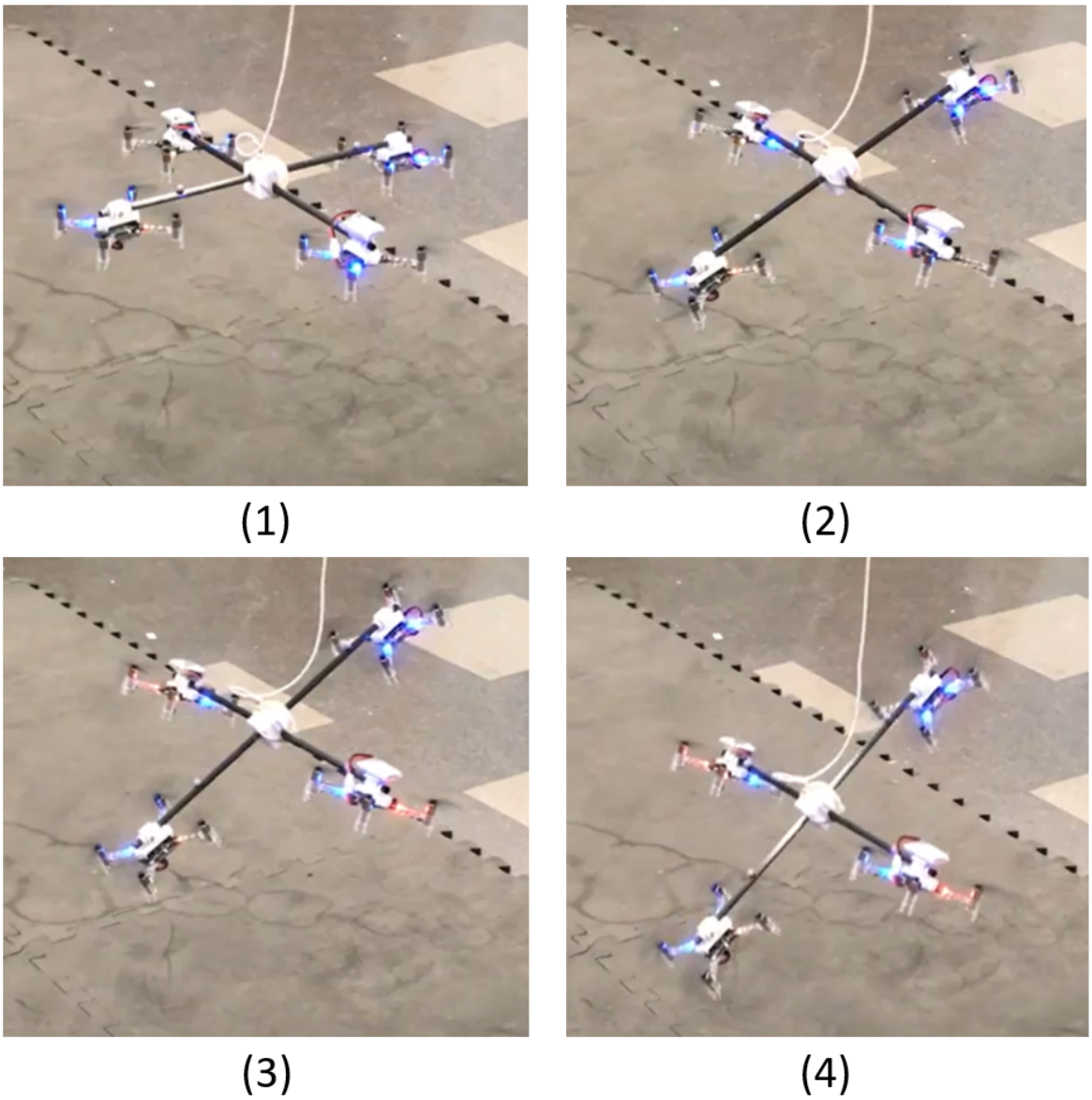


Figure 5.10: Snapshots of ramp test experiment for least square mapping with change of variables. (1) Hover at horizontal attitude. (2) Pitch angle tracks the ramp reference signal. (3) Reach the maximum pitch angle at $\theta_m = 0.42$ rad. (4) Tracking fails after reaching the maximum pitch angle.

5.2.2.6 Results Analysis

Compared with the mapping scheme in Chapter 4, the least square mapping with change of variables gives a maximum pitch angle of 0.42 rad, increased by 20%.

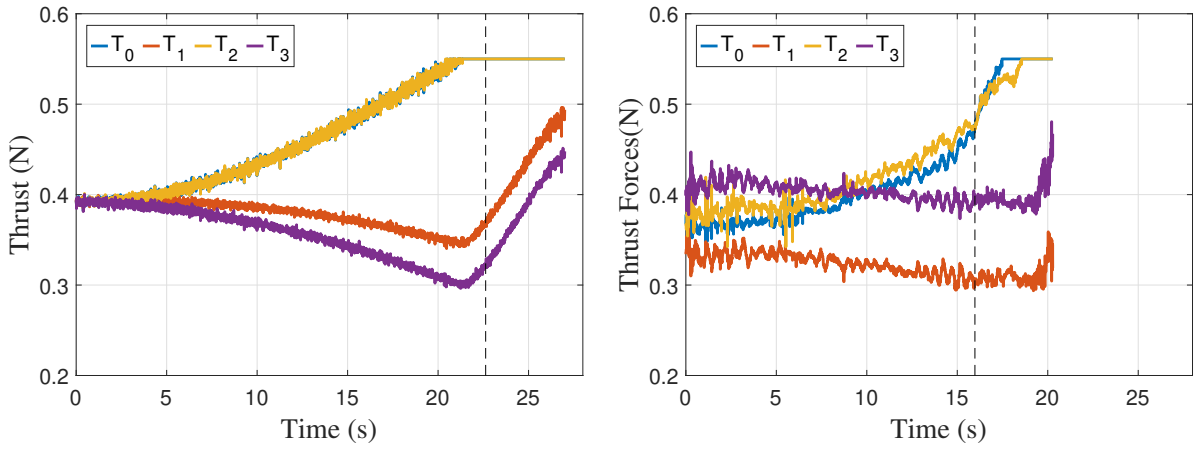


Figure 5.11: The thrust forces for ramp test on simulation(left) and experiment(right).

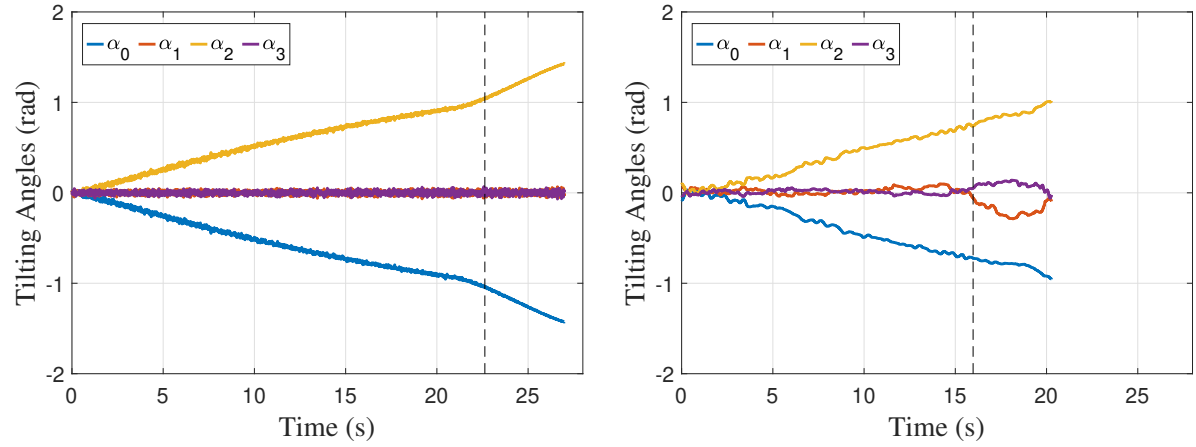


Figure 5.12: The actuator tilting angles for ramp test on simulation(left) and experiment(right).

The original platform inputs are shown in Figure 5.11 and Figure 5.12. It can be observed that the thrust forces diverge quickly with the increase of desired pitch angle, and the lifting force of the platform is largely provided by only two of the thrust forces. Therefore, changing the input mapping scheme to efficiently use all four thrust forces for lifting can enlarge the maximum achievable pitch angle. This is to be elaborated in the following section.

5.3 Null Space Exploration

As has been analyzed in 3.3, the platform is theoretically able to achieve arbitrary pose $\boldsymbol{\eta}$ with proper selection of tilting angles α_i and thrust forces T_i. In reality, though, this capability is constrained by the saturation of thrust forces, so only a smaller range of attitudes are practically achievable, as indicated in 4.4 and 5.2. However, as the system dynamics is overactuated, a mapping scheme with least required thrust forces can always be found in the solution space. Under such mapping, the thrust saturation shall be hit at the largest possible attitude, thus provide the largest range of achievable attitudes. The goal of this section, therefore, is to find this mapping scheme. The analysis and verification are still based on pitch angle.

5.3.1 Gravity Compensation at Hovering

When the platform hovers a specific attitude $\boldsymbol{\eta}$, we have

$$\begin{aligned} \ddot{\boldsymbol{\xi}} &= \mathbf{0} \\ \dot{\boldsymbol{\nu}} &= \mathbf{0} \\ \boldsymbol{\nu} &= \mathbf{0} \end{aligned} \tag{5.22}$$

Under this circumstance, the system inputs collaboratively compensate for its gravity, as

$$\boldsymbol{\zeta} = \begin{bmatrix} \zeta_x & \zeta_y & \zeta_z \end{bmatrix}^T = m({}^W\boldsymbol{R}_B)^T\boldsymbol{G}. \tag{5.23}$$

Recalling section 3.3, the general mapping scheme for hovering condition can be explicitly calculated as

$$\begin{aligned} s\alpha_0 T_0 &= -\frac{\zeta_x}{2} - \sigma_1, \quad c\alpha_0 T_0 = -\frac{\zeta_z}{4} - \sigma_2, \\ s\alpha_1 T_1 &= \frac{\zeta_y}{2} + \sigma_1, \quad c\alpha_1 T_1 = -\frac{\zeta_z}{4} + \sigma_2, \\ s\alpha_2 T_2 &= \frac{\zeta_x}{2} - \sigma_1, \quad c\alpha_2 T_2 = -\frac{\zeta_z}{4} - \sigma_2, \\ s\alpha_3 T_3 &= -\frac{\zeta_y}{2} + \sigma_1, \quad c\alpha_3 T_3 = -\frac{\zeta_z}{4} + \sigma_2, \end{aligned} \tag{5.24}$$

where σ_1 and σ_2 are free variables for the null space of input mapping.

As mentioned previously, the original platform inputs thrust forces T_i and actuator tilting angles α_i can be reversely calculated by the nonlinear functions

$$T_i = \sqrt{(s\alpha_i T_i)^2 + (c\alpha_i T_i)^2}, \tag{5.25}$$

and

$$\alpha_i = \text{atan2}(s\alpha_i T_i, c\alpha_i T_i) \tag{5.26}$$

for

$$\alpha_i \in [-\pi, \pi).$$

When investigating the pitch angle, (5.23) can be specified as

$$\boldsymbol{\zeta} = \begin{bmatrix} \zeta_x & 0 & \zeta_z \end{bmatrix}^T = \begin{bmatrix} mgs\theta & 0 & mgc\theta \end{bmatrix}^T. \tag{5.27}$$

Substituting (5.27) back to (5.24), the thrust forces can be calculated by (5.25) as

$$\begin{aligned} T_0 &= \sqrt{(\frac{\zeta_x}{2} + \sigma_1)^2 + (\frac{\zeta_z}{4} + \sigma_2)^2}, \\ T_1 &= \sqrt{\sigma_1^2 + (\frac{\zeta_z}{4} - \sigma_2)^2}, \\ T_2 &= \sqrt{(-\frac{\zeta_x}{2} + \sigma_1)^2 + (\frac{\zeta_z}{4} + \sigma_2)^2}, \\ T_3 &= \sqrt{\sigma_1^2 + (\frac{\zeta_z}{4} - \sigma_2)^2}. \end{aligned} \tag{5.28}$$

5.3.2 Convex Formulation

When the pitch angle for hovering attitude is determined, the thrust forces (5.28) are merely functions of the null space free variables σ_1 and σ_2.

Therefore, in order to find the mapping scheme with the least possible thrust forces, the free variables σ_1 and σ_2 are searched such that

$$(\sigma_1^{opt}, \sigma_2^{opt}) = \arg \min_{(\sigma_1, \sigma_2)} (\max_i T_i). \tag{5.29}$$

It is obvious that T_i are convex functions of σ_1 and σ_2. Therefore, (5.29) is a unconstrained convex optimization problem and thus should have a unique optimal solution.

Define

$$h(\sigma_1, \sigma_2) = \max_i T_i, \tag{5.30}$$

then it always gives the largest total thrust force among the four quadcopters for a given set of (σ_1, σ_2). As

$$T_1 = T_3 \tag{5.31}$$

at all time in the given scenario, $h(\sigma_1, \sigma_2)$ will always be one of T_0, T_1 and T_2 depending on the selection of point in null space, or explicitly expressed as

$$h(\sigma_1, \sigma_2) = \begin{cases} T_0, & (\sigma_1, \sigma_2) \in \mathcal{S}_0 \\ T_1, & (\sigma_1, \sigma_2) \in \mathcal{S}_1 \\ T_2, & (\sigma_1, \sigma_2) \in \mathcal{S}_2 \end{cases}, \tag{5.32}$$

$\mathcal{S}_0$ is the region in (σ_1, σ_2) space where T_0 is the largest among all thrust forces, and can be calculated as

$$\begin{aligned} T_0 &\geq T_1 \\ T_0 &\geq T_2 \end{aligned}. \tag{5.33}$$

The analytical result is

$$\mathcal{S}_0 = \{(\sigma_1, \sigma_2) : \sigma_1 \geq 0, \sigma_2 \geq -\frac{\zeta_x}{\zeta_z}\sigma_1 - \frac{\zeta_x^2}{4\zeta_z}\}. \tag{5.34}$$

Similarly, $\mathcal{S}_2$ is the region in (σ_1, σ_2) space where T_2 is the largest among all thrust forces, and can be calculated as

$$\begin{aligned} T_2 &\geq T_0 \\ T_2 &\geq T_1 \end{aligned}. \tag{5.35}$$

The analytical result is

$$\mathcal{S}_2 = \{(\sigma_1, \sigma_2) : \sigma_1 \leq 0, \sigma_2 \geq \frac{\zeta_x}{\zeta_z}\sigma_1 - \frac{\zeta_x^2}{4\zeta_z}\}. \tag{5.36}$$

Therefore, $\mathcal{S}_1$, representing the region in (σ_1, σ_2) space where T_1 is the largest among all thrust forces, can be calculated through set operation as

$$\mathcal{S}_1 = \mathbb{R}^2 - \mathcal{S}_0 - \mathcal{S}_2. \tag{5.37}$$

The optimization (5.29) is then rewritten as

$$(\sigma_1^{opt}, \sigma_2^{opt}) = \arg \min_{(\sigma_1,\sigma_2)} \{h_0, h_1, h_2\}, \tag{5.38}$$

where

$$\begin{aligned} h_0 &= \min T_0, \quad (\sigma_1, \sigma_2) \in \mathcal{S}_0 \\ h_1 &= \min T_1, \quad (\sigma_1, \sigma_2) \in \mathcal{S}_1 \\ h_2 &= \min T_2, \quad (\sigma_1, \sigma_2) \in \mathcal{S}_2 \end{aligned} . \tag{5.39}$$

5.3.3 Analytical Optimal Solution

This section calculated the analytical solution for (5.38).

Define points

$$\begin{aligned} C_0 &= (-\frac{\zeta_x}{2}, -\frac{\zeta_z}{4}), \\ C_1 &= (0, \frac{\zeta_z}{4}), \\ C_2 &= (\frac{\zeta_x}{2}, -\frac{\zeta_z}{4}) \end{aligned} \tag{5.40}$$

in the (σ_1, σ_2) space. The by observation of (5.28), the thrust forces can be interpreted in the geometrical sense as

$$\begin{aligned} h_0 &= \min_{P \in \mathcal{S}_0} C_0 P = C_0 P_0, \\ h_1 &= \min_{P \in \mathcal{S}_1} C_0 P = C_1 P_1, \\ h_2 &= \min_{P \in \mathcal{S}_2} C_0 P = C_2 P_2. \end{aligned} \tag{5.41}$$

Visualizations of the optimization are shown in Figure 5.13 and Figure 5.14. The regions $\mathcal{S}_0$, $\mathcal{S}_1$ and $\mathcal{S}_2$ are filled with cyan, green and pink respectively. It is noticed that the pitch angle is constrained within the range

$$\theta \in [0, \frac{\pi}{2}], \tag{5.42}$$

because the platform is symmetric with respect to $\boldsymbol{y}_B$, and the tracking capability of any pitch angle in the range

$$\theta \in [\frac{\pi}{2}, \pi] \tag{5.43}$$

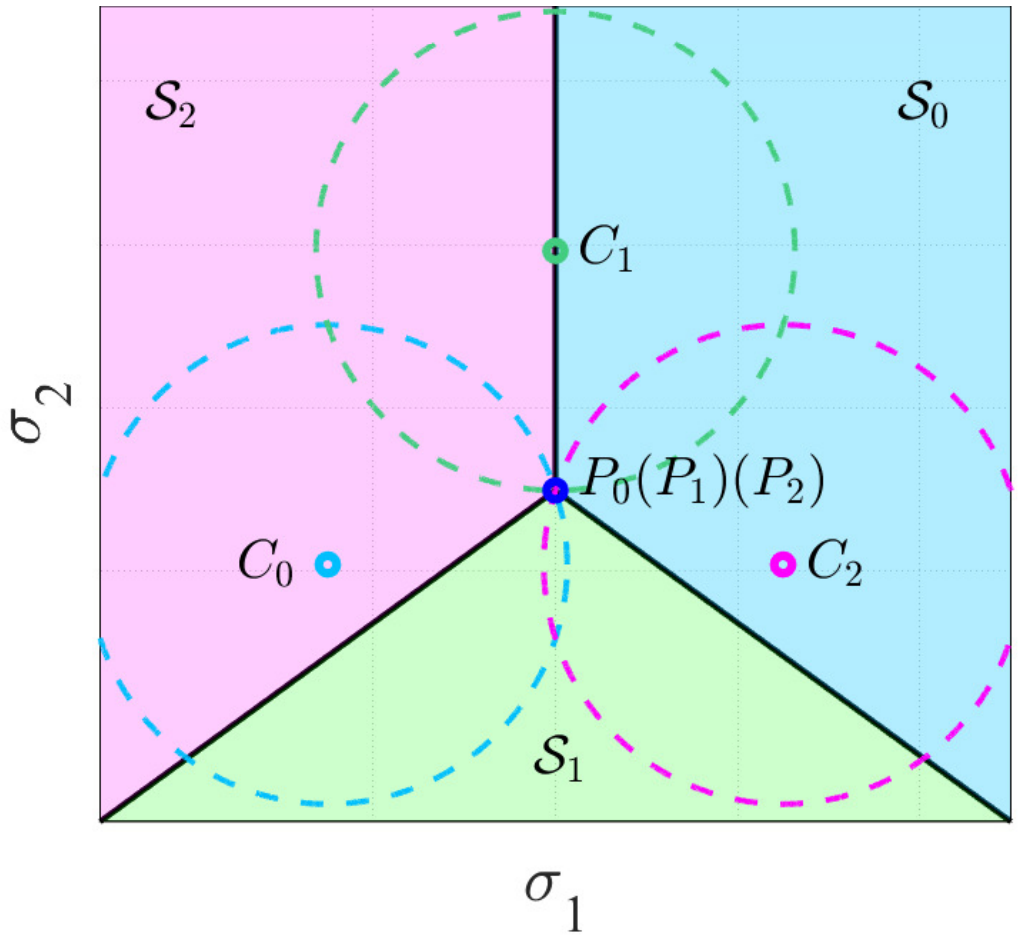


Figure 5.13: Geometrical interpretation of optimization for $0 \leq \theta \leq \frac{\pi}{4}$. The global minimum value is obtained at $P_0/P_1/P_2$.

is equivalent with the tracking of another pitch angle

$$\theta' = \pi - \theta \in [0, \frac{\pi}{2}]. \tag{5.44}$$

It can be seen from Figure 5.13 and Figure 5.14 that the optimal solution is discussed separately in two cases.

When

$$0 \leq \theta \leq \frac{\pi}{4}, \tag{5.45}$$

we have

$$-\frac{\zeta_z}{4} \leq -\frac{\zeta_x^2}{4\zeta_z}. \tag{5.46}$$

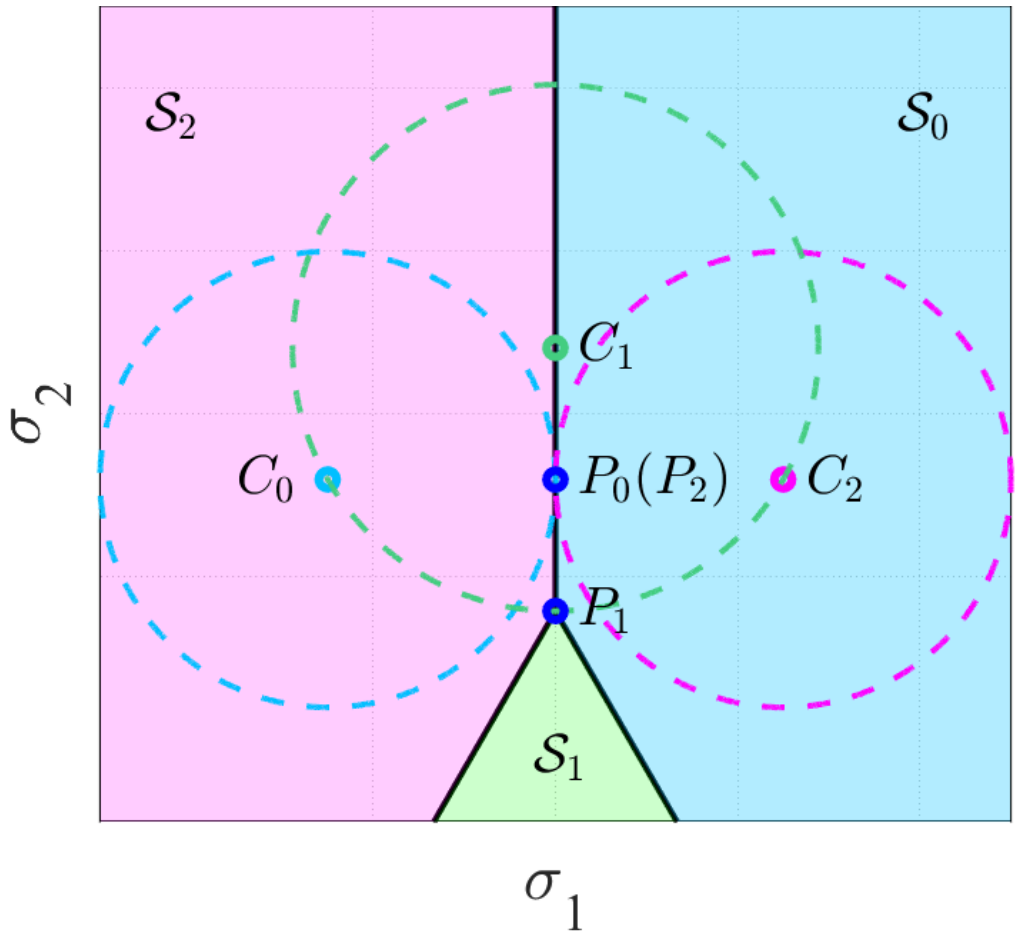


Figure 5.14: Geometrical interpretation of optimization for $\frac{\pi}{4} \leq \theta \leq \frac{\pi}{2}$. The global minimum value is obtained at P_0/P_2.

From Figure 5.13, we can observe that the optima is obtained when

$$h_0 = h_1 = h_2, \tag{5.47}$$

where all total thrust forces of individual quadcopters take the same value. In this case,

$$P_0 = P_1 = P_2 = (0, -\frac{\zeta_x^2}{4\zeta_z}), \tag{5.48}$$

and the corresponding optimal value is calculated as

$$\min_{\sigma_1,\sigma_2} h(\sigma_1, \sigma_2) = \frac{\zeta_x^2 + \zeta_z^2}{4\zeta_z}. \tag{5.49}$$

In the second case, when

$$\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}, \tag{5.50}$$

we have

$$-\frac{\zeta_z}{4} \geq -\frac{\zeta_x^2}{4\zeta_z}. \tag{5.51}$$

In this case, we can observe from 5.14 that

$$h_0 = h_2 \leq h_1. \tag{5.52}$$

The optimal point is

$$P_0 = P_2 = (0, -\frac{\zeta_z}{4}), \tag{5.53}$$

and the corresponding optimal value can be obtained as

$$\min_{\sigma_1,\sigma_2} h(\sigma_1, \sigma_2) = \frac{\zeta_x}{2}. \tag{5.54}$$

Substituting (5.27) into the optimal points gives the analytical solution for (5.38) as

$$(\sigma_1^{opt}, \sigma_2^{opt}) = \begin{cases} (0, -mgs^2\theta/(4c\theta)) & \theta \in [0, \pi/4] \\ (0, mgc\theta/4) & \theta \in (\pi/4, \pi/2] \end{cases}. \tag{5.55}$$

5.3.4 Mapping Scheme Modification

When the reference trajectories for attitude and position is smooth and gradual, the optimal solution for hovering in the previous section can be directly added to the virtual inputs $\boldsymbol{F}$. Therefore, the original inputs under new mapping scheme are

$$T_i = \sqrt{F_{si}^2 + (F_{ci} + (-1)^{i+1}\sigma_2^{opt})^2}, \tag{5.56}$$

and

$$\alpha_i = \text{atan2}(F_{si}, F_{ci} + (-1)^{i+1}\sigma_2^{opt}) \tag{5.57}$$

for

$$\alpha_i \in [-\pi, \pi).$$

Here F_{si} and F_{ci} represent the least square mapping solution in (5.8), and σ_2^{opt} is defined in (5.55).

5.3.5 Simulation and Experiments

5.3.5.1 Simulation

The same ramp test proposed in section 5.2.2.2 is conducted on the new mapping scheme. The criterion defined in 5.2.2.3 is still applied here to judge tracking failure. The results is shown in Figure 5.15.

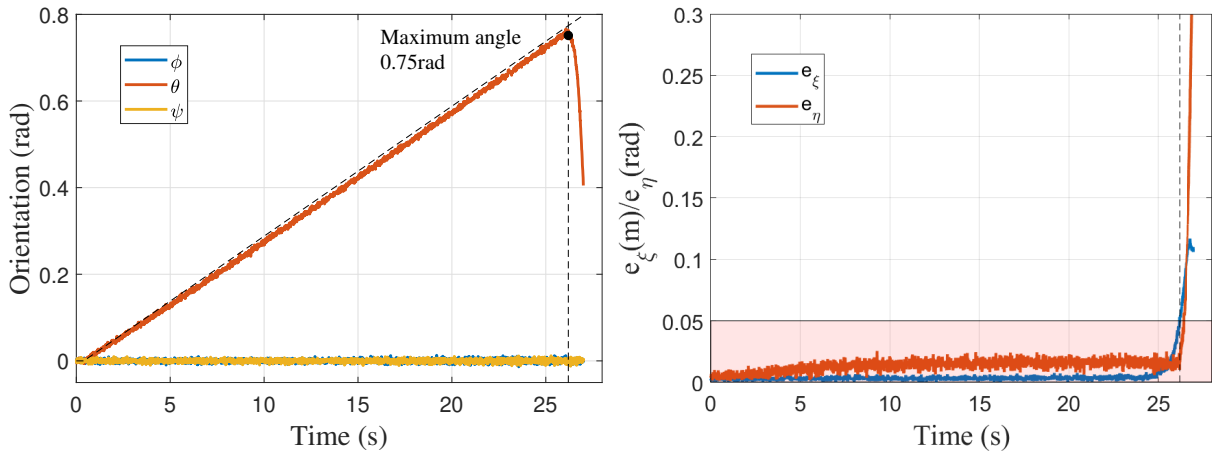


Figure 5.15: Simulation of attitude tracking results of the proposed mapping scheme with null space exploration. The summed error signals for position and attitude are shown on the right figure. The maximum achievable pitch angle is obtained when the first tracking failure happens.

The maximum achievable pitch angle is found as

$$\theta_m = 0.75 \text{ rad}. \tag{5.58}$$

5.3.5.2 Experiments

The corresponding experiments for this proposed controller are conducted at UCLA MacLab, on the prototype proposed in previous chapters. The attitude tracking results and corresponding error signals are shown in Figure 5.16. Snapshots of the experiment are shown in Figure 5.17.

Selecting the same set of error tolerances (5.19), the maximum achievable pitch angle

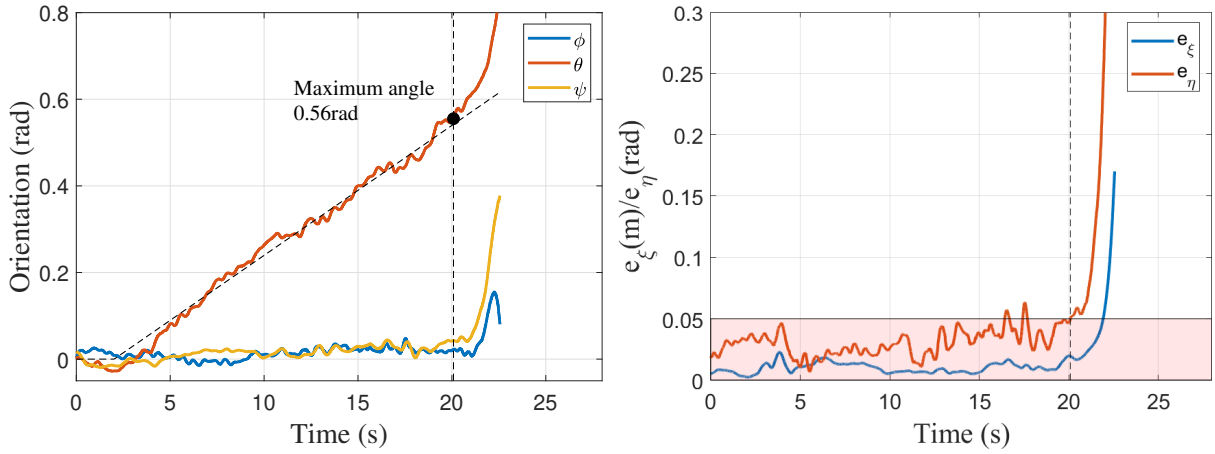


Figure 5.16: Experimental attitude tracking results of the proposed mapping scheme with null space exploration. The summed error signals for position and attitude are shown on the right figure. The maximum achievable pitch angle is obtained when the first tracking failure happens.

can be found as

$$\theta_m = 0.56 \text{ rad}. \tag{5.59}$$

5.3.5.3 Results Analysis

Compared with the mapping scheme in Chapter 4, the maximum achievable pitch angle is increased by 60%. Compared with the least square mapping proposed in the previous section, the maximum achievable pitch angle is increased by 33%

The original platform inputs are shown in Figure 5.18 and Figure 5.19. It is obvious compared with Figure 5.11 that all 4 thrust forces are collectively providing forces for lifting the platform, thus reduces the burden for individual thrusting actuators.

5.3.6 Lifting Capability of the Platform

Apart from making modifications to the mapping scheme, the null space exploration can also be used to size the lifting capability of the platform, or to guide the design process to the proper commercial quadcopters.

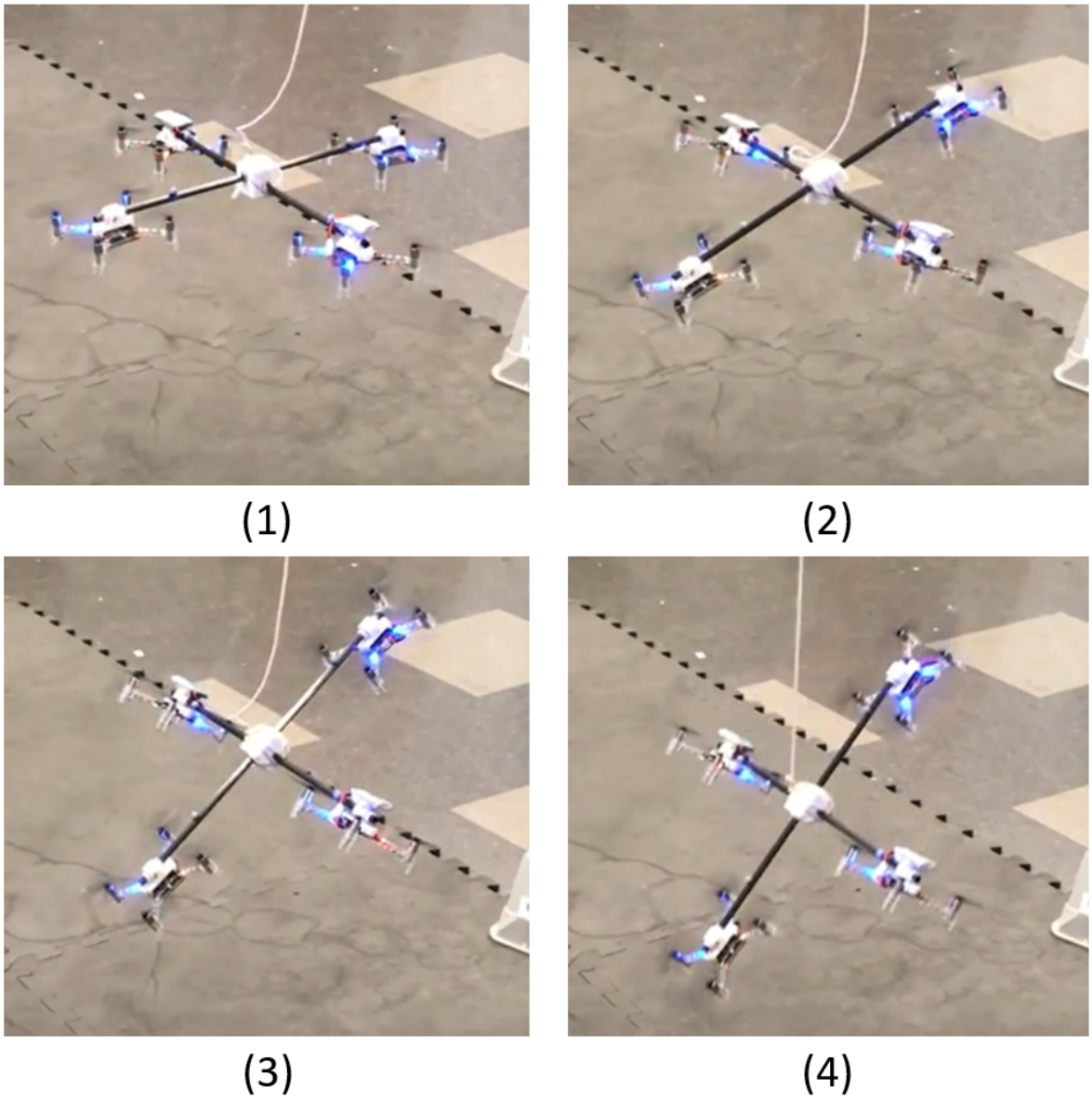


Figure 5.17: Snapshots of ramp test experiment for least square mapping with null space exploration. (1) Hover at horizontal attitude. (2) Pitch angle tracks the ramp reference signal. (3) Reach the maximum pitch angle at $\theta_m = 0.56$ rad. (4) Tracking fails after reaching the maximum pitch angle.

From (5.49) and (5.54), it can be calculated that

$$T_m(\theta) = \begin{cases} \frac{G}{4c\theta}, & \theta \in [0, \frac{\pi}{4}] \\ \frac{Gs\theta}{2}, & \theta \in (\frac{\pi}{4}, \frac{\pi}{2}] \end{cases}. \tag{5.60}$$

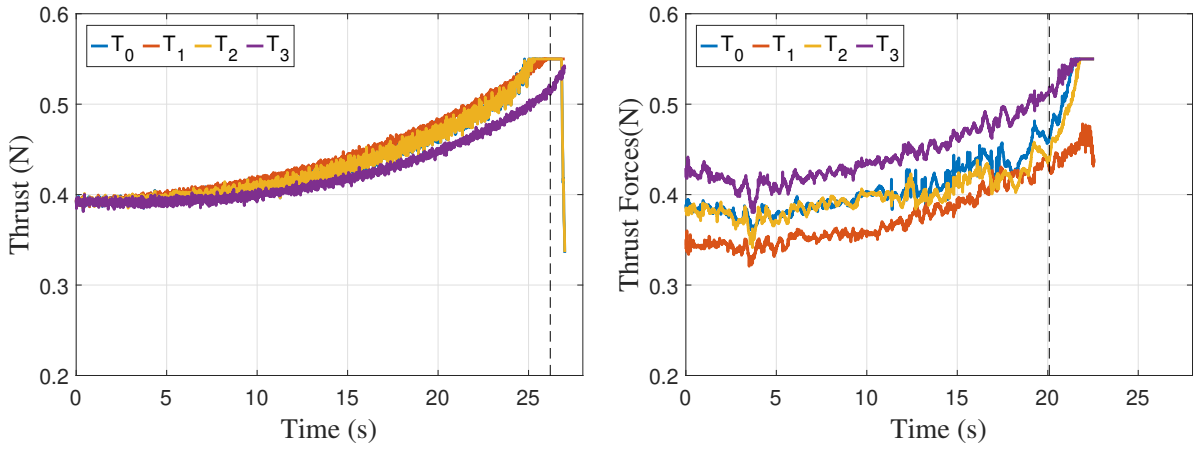


Figure 5.18: The thrust forces for ramp test on simulation(left) and experiment(right).

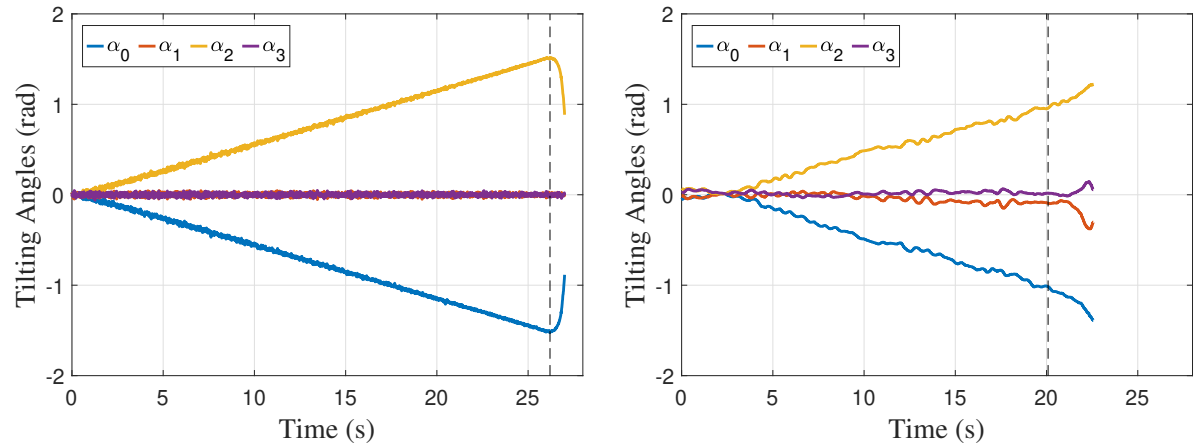


Figure 5.19: The actuator tilting angles for ramp test on simulation(left) and experiment(right).

Define the gravity of each quadcopter as G_Q, and the gravity of the central frame as G_F, then

$$G = G_F + 4G_Q. \tag{5.61}$$

Define the thrust-to-weight ratio of the quadcopter as

$$\gamma_Q = \frac{T_Q}{G_Q}, \tag{5.62}$$

and the scaling factor of the central frame weight with respect to the quadcopter as

$$\rho = \frac{G_F}{G_Q}. \tag{5.63}$$

Obviously,

$$\gamma_Q \geq 1,\ \rho > 0. \tag{5.64}$$

For existing prototype of the proposed configuration, γ_Q and ρ are given, and the maximum pitch angle θ_m is calculated by

$$\theta_m(\gamma_Q, \rho) = \begin{cases} \arccos\left[\text{sat}(\frac{\rho+4}{4\gamma})\right], & \rho \geq 2\sqrt{2}\gamma_Q - 4 \\ \arcsin\left[\text{sat}(\frac{2\gamma_Q}{\rho+4})\right], & \rho < 2\sqrt{2}\gamma_Q - 4 \end{cases}, \tag{5.65}$$

where the saturation function

$$\text{sat}(x) = \begin{cases} x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}. \tag{5.66}$$

5.4 Perching on a Vertical Wall with Lifted Thrust Limit

Although the null space exploration techniques improves the lifting efficiency of thrust forces, it has been indicated from the experiments in this chapter that the existing prototype built in UCLA MacLab does not have sufficient thrust force limit to achieve vertical pose where $\theta = \pi/2$. In fact, it can be calculated from (5.60) that the smallest thrust force limit required to achieve $\pi/2$ in pitch angle is

$$T_m = \frac{G\sin(\pi/2)}{2} = 0.784\text{N}. \tag{5.67}$$

In the realistic simulation constructed in section 5.2.2.1, however, the thrust force limit can be artificially lifted to simulate the task of perching on a vertical wall. According to (5.67), the lifted thrust limit is set as

$$T_m = 0.8\text{N}. \tag{5.68}$$

The perching problem is described in Figure 5.20. The aerial platform is at rest on the ground at location $(0, 0)$, with horizontal attitude $\theta = 0$. The goal is to control the platform to perch on the vertical wall at the location $(1, -1)$, with attitude $\theta = \pi/2$.

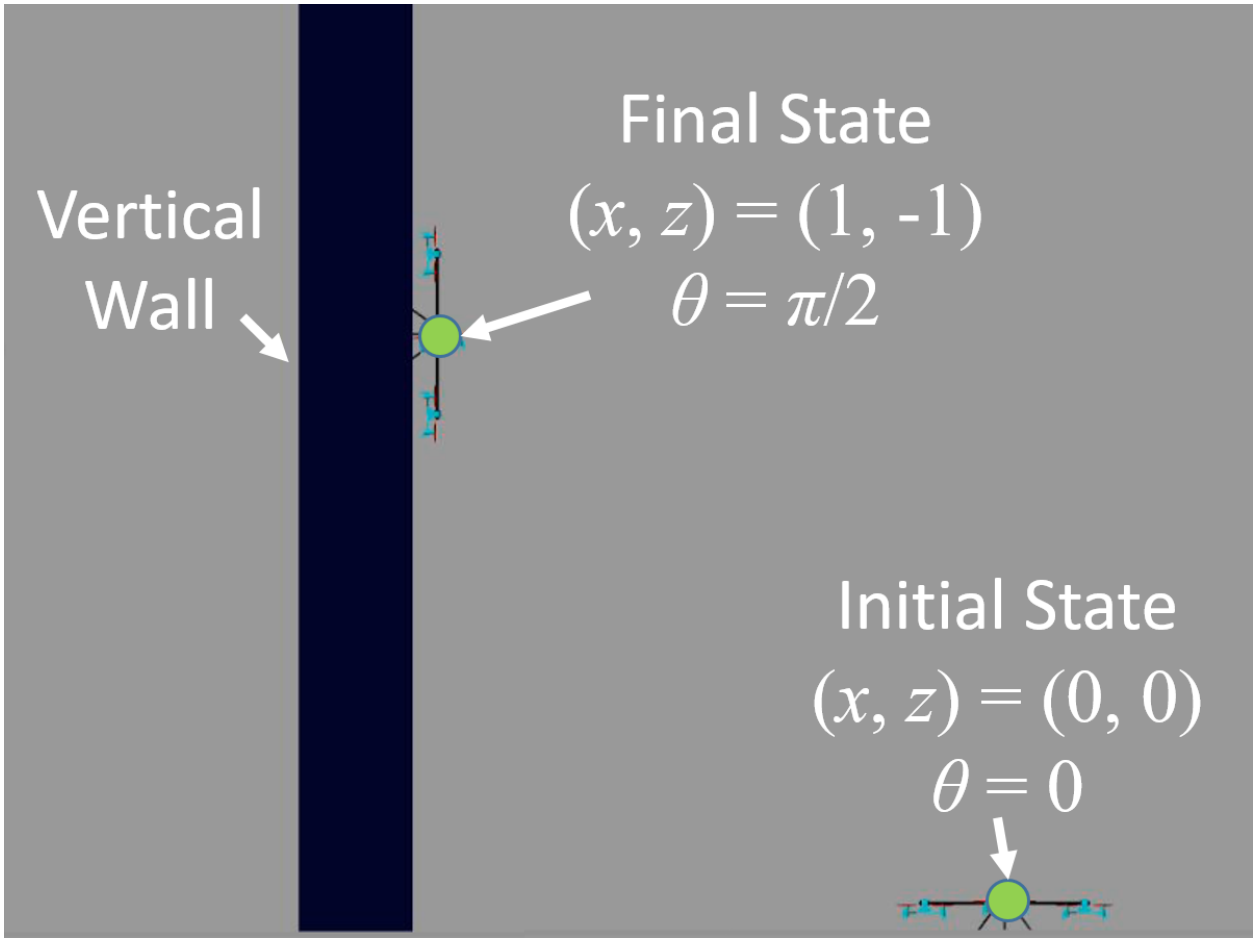


Figure 5.20: The perching task in the Simulink SimScape Environment, where the aerial platform is initialized at the horizontal attitude on the ground, and attempts to perch on a vertical wall at $(x, z) = (1\text{m}, -1\text{m})$. The coordinates refer to the position of platform CoM, which is marked with a green dot. The quadcopters hinged on the platform are colored cyan for better distinction.

The perching is visualized in Figure 5.21. The tracking performance of position and pitch angle are shown in Figure 5.22, It can thus be concluded that the platform is able to complete the task of perching on a vertical wall with sufficient thrust limit.

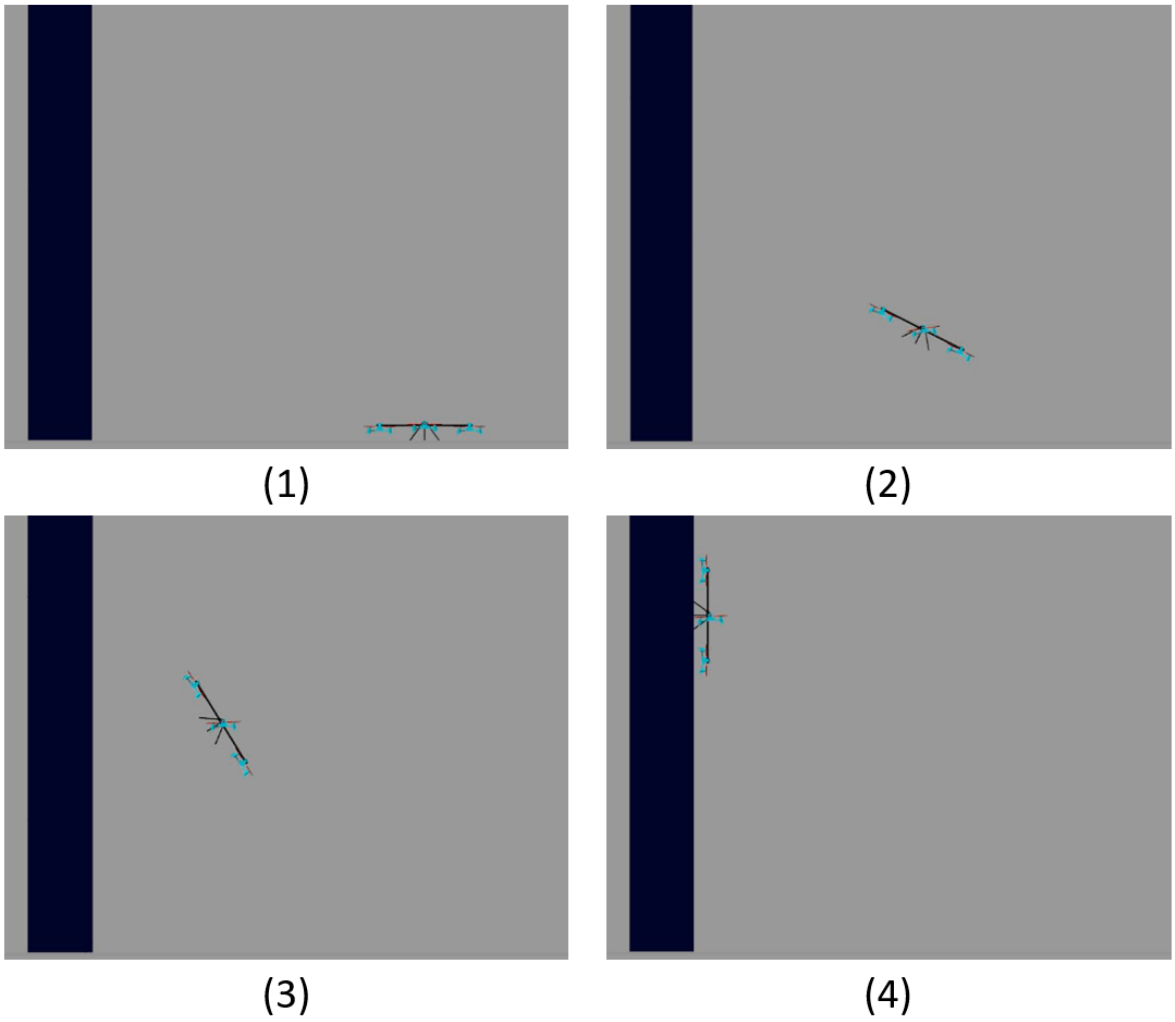


Figure 5.21: The perching results demonstration in the Simulink SimScape Environment. (1) Rest on the ground with horizontal attitude. (2) Take off and approach the wall. (3) Decrease velocity and adjust attitude to prepare for perching. (4) Perch on (1m, 1m) position on the wall.

5.5 Conclusion

This chapter designed a novel controller for the proposed fully actuated multirotor aerial platform with tiltable thrust actuators. The controller is based on the least square solution in [KVE18]. The mapping scheme of explored the null space of input allocation and searched for the least possible thrust forces required to hover at a certain attitude. The controller was

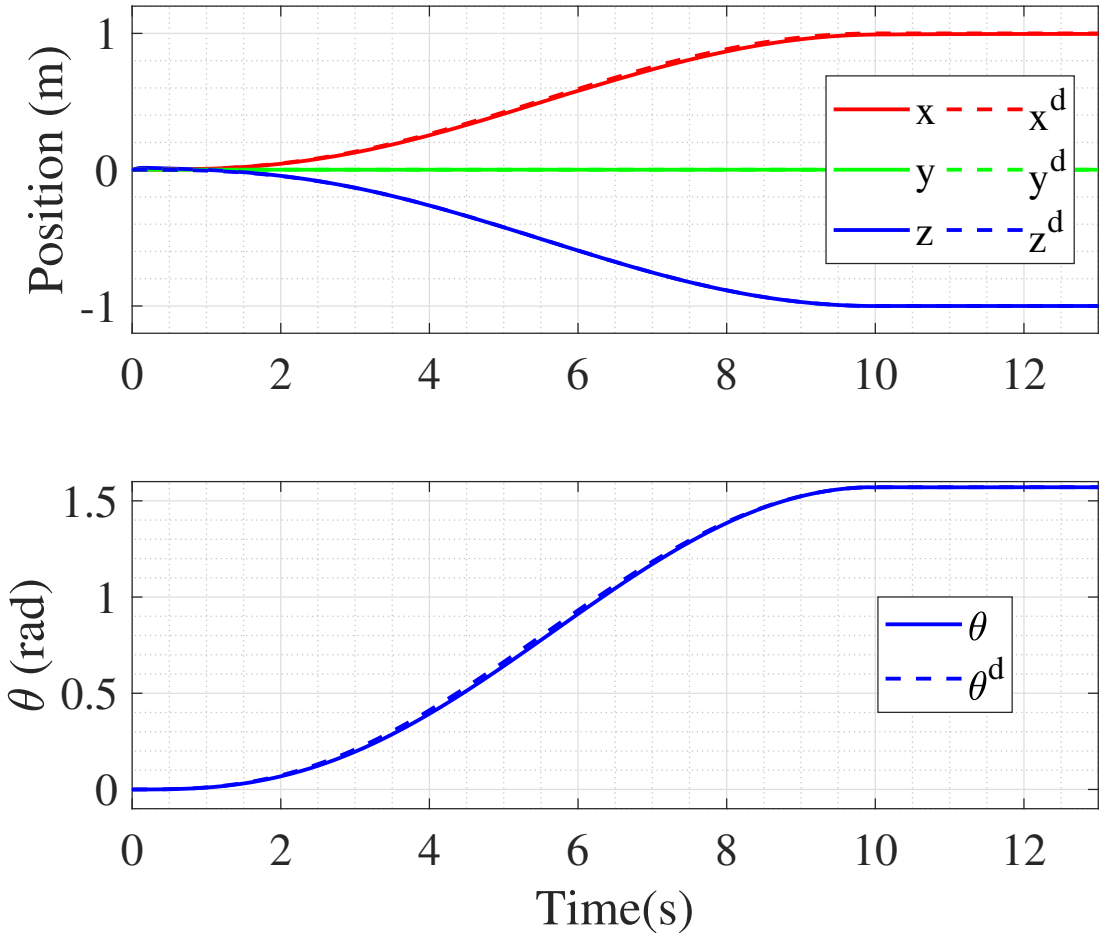


Figure 5.22: Position and pitch angle trajectories of for perching in the Simulink SimScape environment.

formulated as a convex optimization, which can be solved analytically and thus avoids heavy online computations. The controller was tested experimentally. It showed 60% improvement in maximum achievable pitch angle compared with the controller in Chapter 4, and 33% increase compared with the least square solution in [KVE18]. It is also verified from realistic simulation that this controller is able to drive the proposed aerial platform to perch on a vertical wall with sufficient thrust limit.

CHAPTER 6

Conclusion

Multirotor UAV has been widely applied in research and industry, for its advantages in mechanical simplicity, high agility and ability of vertical take-off and landing. One critical challenge for conventional multirotors lies on its underactuation in dynamics, which makes it impossible to independently control position and attitude. Challenge remains for multirotor applications with combined requirements on position and attitude, such as perching on a vertical wall or hovering at a non-horizontal pose. This dissertation addressed the challenge in two aspects.

An algorithm was proposed for the conventional quadcopters to generate trajectories with requirements on both position and attitude. The planning process was formulated as an optimization, and further reduced to a series of convex problems for efficient calculation. Constraints on input and state variables were explicitly included. The planning process of the trajectory of a conventional quadcopter perching at a designated location on the vertical wall was demonstrated for verification. While algorithm solutions can be quickly implemented, it does not really avoid the underactuation of multirotor dynamics, thus can only be successfully applied under very specific conditions where a desired trajectory does exist. They are also generally degraded in robustness, and have further difficulties in tracking.

A novel multirotor aerial platform with fully actuated dynamics was developed. Commercial quadcopters and passive hinges worked as tiltable thrusting actuators. The platform avoids additional actuators and mechanisms for thrust force vectoring, and thus has advantages in mechanical simplicity. It also avoids the disturbance torques introduced from tilting actuation and propeller drag. A controller was designed for independent control of position and attitude, and were demonstrated in different flight cases in simulation and experiments.

Another controller was proposed to search for efficient deployment of thrust forces when the platform hovered at a non-horizontal attitude. Simulation and experiments showed that this controller was able to hover at larger attitude compared with the previous controller under the same thrust force limit.

One common critical process for the trajectory planning algorithm in Chapter 2 and the controllers in Chapter 4 and 5 is the decoupling of two multiplied variables. This is usually one significant step to convexify the dynamic equations. Chapter 2 and 4 proposed two-step approaches, where one variable was determined with additional knowledge, and used in the second step as constant to calculate for the other variable. These approaches are easily implementable, but narrow down the variables within particular subspaces of the original problems. Chapter 5 decoupled the multiplication by the change of variables, which kept the original search space but had difficulties in converting the constraints of original variables to the new ones due to the nonlinear mapping in between.

It is also obvious that the use of thrust forces for lifting against gravity is very inefficient in large attitude for the aerial platform with tiltable thrusting actuators along only cant angles, even with the controller proposed in Chapter 5. This is the natural disadvantage of this particular mechanical configuration, and can be improved by adding another tilting direction on the thrusting actuators. A prototype of such configuration has already been built in UCLA MacLab, where four commercial quadcopters are connected to the platform central frame by passive revolute joints with DoF on both cant and dihedral angles. The thrust forces for this configuration can be deployed very efficiently for lifting against the gravity. The range of achievable attitude is therefore greatly enlarged for the same selection of commercial quadcopters.

References

[Ald95] Thomas S Alderete. "Simulator aero model implementation." *NASA Ames Research Center, Moffett Field, California*, p. 21, 1995.

[AN19] A. Asheralieva and D. Niyato. "Game Theory and Lyapunov Optimization for Cloud-Based Content Delivery Networks With Device-to-Device and UAV-Enabled Caching." *IEEE Transactions on Vehicular Technology*, **68**(10):10094–10110, 2019.

[BAB17] Pavol Božek, Aiman Al Akkad M, Peter Blištan, and Ibrahim Ibrahim N. "Navigation control and stability investigation of a mobile robot based on a hexacopter equipped with an integrated manipulator." *International Journal of Advanced Robotic Systems*, **14**(6):1729881417738103, 2017.

[BSF18] N Belmonte, S Staulo, S Fiorot, C Luetto, P Rizzi, and M Baricco. "Fuel cell powered octocopter for inspection of mobile cranes: Design, cost analysis and environmental impacts." *Applied energy*, **215**:556–565, 2018.

[CKC18] Vishnu S Chipade, Mangal Kothari, Rushikesh R Chaudhari, et al. "Systematic design methodology for development and flight testing of a variable pitch quadrotor biplane VTOL UAV for payload delivery." *Mechatronics*, **55**:94–114, 2018.

[CYW07] Ian D Cowling, Oleg A Yakimenko, James F Whidborne, and Alastair K Cooke. "A prototype of an autonomous controller for a quadrotor UAV." In *2007 European Control Conference (ECC)*, pp. 4001–4008. IEEE, 2007.

[DM19] David J Dunlop and Mark A Minor. "Modelling and Simulation of Perching With a Quadrotor Aerial Robot With Passive Bio-Inspired Legs and Feet." In *Dynamic Systems and Control Conference*, volume 59162, p. V003T21A011. American Society of Mechanical Engineers, 2019.

[For15] Julian Förster. *"System identification of the crazyflie 2.0 nano quadrocopter."*. B.S. thesis, ETH Zurich, 2015.

[GB08] Michael Grant and Stephen Boyd. "Graph implementations for nonsmooth convex programs." In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.

[GB14] Michael Grant and Stephen Boyd. "CVX: Matlab Software for Disciplined Convex Programming, version 2.1." http://cvxr.com/cvx, March 2014.

[GSW18] Jingjing Gu, Tao Su, Qiuhong Wang, Xiaojiang Du, and Mohsen Guizani. "Multiple moving targets surveillance based on a cooperative network for multi-UAV." *IEEE Communications Magazine*, **56**(4):82–89, 2018.

[GT18] Matthew J Gerber and Tsu-Chin Tsao. "Twisting and Tilting Rotors for High-Efficiency, Thrust-Vectored Quadrotors." *Journal of Mechanisms and Robotics*, **10**(6), 2018.

[HD11] Markus Hehn and Raffaello D'Andrea. "Quadrocopter trajectory generation and control." *IFAC proceedings Volumes*, **44**(1):1485–1491, 2011.

[HGA20] Ramy Hashem, Jelmer Goerres, Ronald G Aarts, Johan B C Engelen, and Stefano Stramigioli. "Fully Actuated Multirotor UAVs: A Literature Review." *IEEE Robotics & Automation Magazine*, pp. 0–0, 2020.

[HHM13] M. Hua, T. Hamel, P. Morin, and C. Samson. "Introduction to feedback control of underactuated VTOLvehicles: A review of basic control design ideas and principles." *IEEE Control Systems Magazine*, **33**(1):61–75, 2013.

[HLS19] Kaiyu Hang, Ximin Lyu, Haoran Song, Johannes A Stork, Aaron M Dollar, Danica Kragic, and Fu Zhang. "Perching and resting—A paradigm for UAV maneuvering with modularized landing gears." *Science Robotics*, **4**(28), 2019.

[HSS18] K Harikumar, J Senthilnath, and Suresh Sundaram. "Multi-uav oxyrrhis marina-inspired search and dynamic formation control for forest firefighting." *IEEE Transactions on Automation Science and Engineering*, **16**(2):863–873, 2018.

[HWT08] Gabriel Hoffmann, Steven Waslander, and Claire Tomlin. "Quadrotor helicopter trajectory tracking control." In *AIAA guidance, navigation and control conference and exhibit*, p. 7410, 2008.

[JMG19] Ruihang Ji, Jie Ma, and Shuzhi Sam Ge. "Modeling and Control of a Tilting Quadcopter." *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[JVC18] G. Jiang, R. M. Voyles, and J. J. Choi. "Precision Fully-Actuated UAV for Visual and Physical Inspection of Structures for Nuclear Decommissioning and Search and Rescue." In *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–7, 2018.

[KMR15] Arash Kalantari, Karan Mahajan, Donald Ruffatto, and Matthew Spenko. "Autonomous perching and take-off on vertical walls for a quadrotor micro air vehicle." In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4669–4674. IEEE, 2015.

[KNK17] Rumit Kumar, Alireza Nemati, Manish Kumar, Rajnikant Sharma, Kelly Cohen, and Franck Cazaurang. "Tilting-rotor quadcopter for aggressive flight maneuvers using differential flatness based flight controller." In *Dynamic Systems and Control Conference*, volume 58295, p. V003T39A006. American Society of Mechanical Engineers, 2017.

[KPJ16] Keeryun Kang, JVR Prasad, and Eric Johnson. "Active control of a uav helicopter with a slung load for precision airborne cargo delivery." *Unmanned Systems*, **4**(03):213–226, 2016.

[KPL20] W. Kwon, J. H. Park, M. Lee, J. Her, S. Kim, and J. Seo. "Robust Autonomous Navigation of Unmanned Aerial Vehicles (UAVs) for Warehouses' Inventory Application." *IEEE Robotics and Automation Letters*, **5**(1):243–249, 2020.

[KVE18] Mina Kamel, Sebastian Verling, Omar Elkhatib, Christian Sprecher, Paula Wulkop, Zachary Taylor, Roland Siegwart, and Igor Gilitschenski. "The Voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle." *IEEE Robotics & Automation Magazine*, **25**(4):34–44, 2018.

[LLM10] T. Lee, M. Leok, and N. H. McClamroch. "Geometric tracking control of a quadrotor UAV on SE(3)." In *49th IEEE Conference on Decision and Control (CDC)*, pp. 5420–5425, 2010.

[LPW17] Jeroen AJ Ligthart, Pakorn Poksawat, Liuping Wang, and Henk Nijmeijer. "Experimentally validated model predictive controller for a hexacopter." *IFAC-PapersOnLine*, **50**(1):4076–4081, 2017.

[LST18] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, M. Saska, and V. Kumar. "Localization, Grasping, and Transportation of Magnetic Objects by a Team of MAVs in Challenging Desert-Like Environments." *IEEE Robotics and Automation Letters*, **3**(3):1576–1583, 2018.

[MK11] Daniel Mellinger and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2520–2525. IEEE, 2011.

[MKC12] R. Mahony, V. Kumar, and P. Corke. "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor." *IEEE Robotics Automation Magazine*, **19**(3):20–32, 2012.

[MMK12] Daniel Mellinger, Nathan Michael, and Vijay Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors." *The International Journal of Robotics Research*, **31**(5):664–674, 2012.

[NJL16] Abdullah Al Redwan Newaz, Sungmoon Jeong, Hosun Lee, Hyejeong Ryu, and Nak Young Chong. "Uav-based multiple source localization and contour mapping of radiation fields." *Robotics and Autonomous Systems*, **85**:12–25, 2016.

[NK14] A. Nemati and M. Kumar. "Modeling and control of a single axis tilting quadcopter." In *2014 American Control Conference*, pp. 3077–3082, 2014.

[NKO17] Kazuhiro Nakadai, Makoto Kumon, Hiroshi G Okuno, Kotaro Hoshiba, Mizuho Wakabayashi, Kai Washizaki, Takahiro Ishiki, Daniel Gabriel, Yoshiaki Bando, Takayuki Morito, et al. "Development of microphone-array-embedded UAV for search and rescue task." In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5985–5990. IEEE, 2017.

[NPL20] Ngoc Thinh Nguyen, Ionela Prodan, and Laurent Lefèvre. "Flat trajectory design and tracking with saturation guarantees: a nano-drone application." *International Journal of Control*, **93**(6):1266–1279, 2020.

[NPP18] Hai-Nguyen Nguyen, Sangyul Park, Junyoung Park, and Dongjun Lee. "A novel robotic platform for aerial manipulation using quadrotors as rotating thrust generators." *IEEE Transactions on Robotics*, **34**(2):353–369, 2018.

[OOP16] Samuel Ortega-Farías, Samuel Ortega-Salazar, Tomas Poblete, Ayse Kilic, Richard Allen, Carlos Poblete-Echeverría, Luis Ahumada-Orellana, Mauricio Zuñiga, and Daniel Sepúlveda. "Estimation of energy balance components over a drip-irrigated olive orchard using thermal and multispectral cameras placed on a helicopter-based unmanned aerial vehicle (UAV)." *Remote Sensing*, **8**(8):638, 2016.

[PBH20] M. Petrlík, T. Báča, D. Heřt, M. Vrba, T. Krajník, and M. Saska. "A Robust UAV System for Operations in a Constrained Environment." *IEEE Robotics and Automation Letters*, **5**(2):2169–2176, 2020.

[PJW18] Katie M Popek, Matthew S Johannes, Kevin C Wolfe, Rachel A Hegeman, Jessica M Hatch, Joseph L Moore, Kapil D Katyal, Bryanna Y Yeh, and Robert J Bamberger. "Autonomous grasping robotic aerial system for perching (agrasp)." In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9. IEEE, 2018.

[PLA18] S. Park, J. Lee, J. Ahn, M. Kim, J. Her, G. Yang, and D. Lee. "ODAR: Aerial Manipulation Platform Enabling Omnidirectional Wrench Generation." *IEEE/ASME Transactions on Mechatronics*, **23**(4):1907–1918, 2018.

[PWM17] Pakorn Poksawat, Liuping Wang, and Abdulghani Mohamed. "Gain scheduled attitude control of fixed-wing UAV with automatic controller tuning." *IEEE Transactions on Control Systems Technology*, **26**(4):1192–1203, 2017.

[RBF16] Markus Ryll, Davide Bicego, and Antonio Franchi. "Modeling and control of FAST-Hex: A fully-actuated by synchronized-tilting hexarotor." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1689–1694. IEEE, 2016.

[RBG14] Markus Ryll, Heinrich H Bülthoff, and Paolo Robuffo Giordano. "A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation." *IEEE Transactions on Control Systems Technology*, **23**(2):540–556, 2014.

[RCS19] R. Rashad, F. Califano, and S. Stramigioli. "Port-Hamiltonian Passivity-Based Control on SE(3) of a Fully Actuated UAV for Aerial Physical Interaction Near-Hovering." *IEEE Robotics and Automation Letters*, **4**(4):4378–4385, 2019.

[RMP17] Markus Ryll, Giuseppe Muscio, Francesco Pierri, Elisabetta Cataldi, Gianluca Antonelli, Fabrizio Caccavale, and Antonio Franchi. "6D physical interaction with a fully actuated aerial robot." In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5190–5195. IEEE, 2017.

[RRB15] Sujit Rajappa, Markus Ryll, Heinrich H Bülthoff, and Antonio Franchi. "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers." In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4006–4013. IEEE, 2015.

[RTL18] Vincent Roberge, Mohammed Tarbouchi, and Gilles Labonté. "Fast genetic algorithm path planner for fixed-wing military UAV using GPU." *IEEE Transactions on Aerospace and Electronic Systems*, **54**(5):2105–2117, 2018.

[SKB16] K. Steich, M. Kamel, P. Beardsley, M. K. Obrist, R. Siegwart, and T. Lachat. "Tree cavity inspection using aerial robots." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4856–4862, 2016.

[SKR17] Siddharth Sridhar, Rumit Kumar, Mohammadreza Radmanesh, and Manish Kumar. "Non-linear sliding mode control of a tilting-rotor quadcopter." In *Dynamic Systems and Control Conference*, volume 58271, p. V001T09A007. American Society of Mechanical Engineers, 2017.

[SNT19] S. Sawadsitang, D. Niyato, P. Tan, and P. Wang. "Joint Ground and Aerial Package Delivery Services: A Stochastic Optimization Approach." *IEEE Transactions on Intelligent Transportation Systems*, **20**(6):2241–2254, 2019.

[SSL17] Jean Smith, Jinya Su, Cunjia Liu, and Wen-Hua Chen. "Disturbance observer based control with anti-windup applied to a small fixed wing UAV for disturbance rejection." *Journal of Intelligent & Robotic Systems*, **88**(2-4):329–346, 2017.

[TK20] Ezra Tal and Sertac Karaman. "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness." *IEEE Transactions on Control Systems Technology*, 2020.

[TVM16] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. "Sensor planning for a symbiotic UAV and UGV system for precision agriculture." *IEEE Transactions on Robotics*, **32**(6):1498–1511, 2016.

[WHF17] Konradin Weber, Georg Heweling, Christian Fischer, and Martin Lange. "The use of an octocopter UAV for the determination of air pollutants–a case study of the traffic induced pollution plume around a river bridge in Duesseldorf, Germany." *International Journal of Education and Learning Systems*, **2**, 2017.

[WZT18] Dandan Wang, Qun Zong, Bailing Tian, Fang Wang, and Liqian Dou. "Finite-time fully distributed formation reconfiguration control for UAV helicopters." *International Journal of Robust and Nonlinear Control*, **28**(18):5943–5961, 2018.

[ZZZ17] Cheng Zhan, Yong Zeng, and Rui Zhang. "Energy-efficient data collection in UAV enabled wireless sensor network." *IEEE Wireless Communications Letters*, **7**(3):328–331, 2017.