Title
An Over-Actuated Multi-Rotor Aerial Platform and Iterative Learning Control Applications

Author
Yu, Pengkang

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

An Over-Actuated Multi-Rotor Aerial Platform and Iterative Learning Control Applications

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Pengkang Yu

2022

ABSTRACT OF THE DISSERTATION

An Over-Actuated Multi-Rotor Aerial Platform and Iterative Learning Control Applications

by

Pengkang Yu
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2022
Professor Tsu-Chin Tsao, Chair

Fully-actuated multi-rotor aerial platforms are receiving increasing research interests for the capability of six degree-of-freedom (DOF) motions such as hovering at non-horizontal attitude angles. Existing fully-actuated aerial vehicles have demonstrated such capability for a limited range of angles and limited thrust efficiencies. This thesis presents an over-actuated aerial platform that achieves maneuvering at arbitrary attitudes with uniformly high thrust efficiency over its achievable configuration space. A novel vectoring thrust force actuator by mounting a regular quad copter on a passive mechanical gimbal mechanism is proposed. The UAV platform achieves full six DOF motion with redundancies from four of these vectoring thrust actuators. We present the hierarchical controller that generates the high level virtual wrench command allocated to each gimbal actuator and the low-level actuator control to track the commanded wrench. And we demonstrate the UAV platform's 6 DOF maneuvers by both simulations and real-world experiments on a prototype we built.

Aerodynamic effects largely affect the performance of aerial vehicles, especially on over-actuated vehicles that could be subject to different airflow configurations. For those aerodynamic effects that are difficult to model but could appear repeatedly, iterative learning control (ILC) has great potential to improve the system performance. This thesis presents the applications of both model-based and data-driven ILC algorithms on the over-actuated

aerial platform and shows great improvements against the aerodynamics effects. A formulation is demonstrated to convert the closed-loop dynamics of the over-actuated aerial platform to linear model with six independent control channels. Model-based and data-driven ILC are applied on one or more control channels, and by simulations and real-world experiments, the ILC algorithms are shown to have great improvement and fast convergence rate against a variety of aerodynamic effects.

The dissertation of Pengkang Yu is approved.

Ankur M. Mehta

Tetsuya Iwasaki

Jacob Rosen

Tsu-Chin Tsao, Committee Chair

University of California, Los Angeles

2022

To my parents,

For their endless love,

support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

My deepest gratitude to Professor Tsu-Chin Tsao, my advisor, who is always supportive and patient. He has great sense in both defining high-level research directions and tackling technical problems. Although most of our conversations are about research, his way of thinking and his value has inspired me a lot outside the research field, guiding me on how to model the real-world challenges and build model-based control on them.

I appreciate my committee, Professor Ankur Mehta, Professor Tetsuya Iwasaki and Professor Jacob Rosen, for their guidance and advice. Special thanks to professor Mehta for providing me with the equipments for the UAV experiments and his advice on the future directions of my research.

I would like to thank all the Maclab members whom I have worked with. Dr. Lecheng Ruan, who has led me into the lab 5 years ago and has mentored me with all his patience. Dr. Yao Su, who has worked with me on various projects and we have great teamwork all the time. Dr. Yu-Hsiu Lee, who has led me to the MR robot project and ILC, and has shared great memories with me as my roommate. And thanks to all other members in Maclab, there are too many names to list.

Finally, thank you all my friends for bringing me good memories. Thanks to my family, my parents for their unconditional support throughout my PhD journey. My appreciation to all of you who have been supporting me during my PhD.

VITA

2016	B.Eng., Mechanical Engineering, Hong Kong University of Science and Technology, Hong Kong, China
2018	M.S., Mechanical Engineering, University of California, Los Angeles (UCLA), Los Angeles, CA, USA
2018–present	Graduate Student Researcher/Teaching Assistant, Mechanical Engineering, University of California, Los Angeles (UCLA), Los Angeles, CA, USA

PUBLICATIONS

Pengkang Yu†, Yao Su†, Matthew J. Gerber, Lecheng Ruan, and Tsu-Chin Tsao. "An over-actuated multi-rotor aerial vehicle with unconstrained attitude angles and high thrust efficiencies." *IEEE Robotics and Automation Letters*, vol. 6, no. 4, 2021

Yao Su†, Pengkang Yu†, Matthew J. Gerber, Lecheng Ruan, and Tsu-Chin Tsao. "Nullspace-based control allocation of overactuated uav platforms." *IEEE Robotics and Automation Letters*, vol. 6, no. 4, 2021

Yao Su†, Pengkang Yu†, Lecheng Ruan†, Chen-Huan Pi, Matthew Gerber, and Tsu-Chin Tsao. "A Fast and Efficient Attitude Control Algorithm of a Tilt-Rotor Aerial Platform Using Inputs Redundancies." *IEEE Robotics and Automation Letters*, 2021

Chen-Huan Pi†, Lecheng Ruan†, Pengkang Yu, Yao Su, Stone Cheng, and Tsu-Chin Tsao. "A simple six degree-of-freedom aerial vehicle built on quadcopters." *IEEE Conference on Control Technology and Applications*, 2021, pp. 329-334. († equal contribution)

CHAPTER 1

Introduction

1.1 Conventional Multirotor UAVs and Their Limitations

Recent development of multirotor unmanned aerial vehicles (UAVs) has largely stimulated related researches and applications with the new promising aerial platform. UAVs have been implemented on various application scenarios, such as monitoring crops [BAB+15], monitoring soil erosion condition [dOMPR12], searching and rescuing [GMG+08], detecting air quality index [YZB+17], traking forest fire [CKBM06]. However, conventional multirotors have all propellers fixed in the same direction and are under-actuated, which results in the coupled position and attitude control and limits the maneuver capabilities of UAV. On the other hand, there are increasing numbers of applications in research or industry that require a UAV to enact six independent forces and torques, such as physical interactions with the environment and inspections in different orientations [RLO18].

1.2 Introduction to Fully-Actuated UAVs

The emergence of the needs of generating six independent forces and torques in the air triggers the development of fully-actuated UAVs, which generally have higher actively controlled degrees of freedom (DOF) and can hover in the air with different attitudes. Fully-actuated UAVs come with various configurations and demonstrate different advantages.

The recent research on fully-actuated UAVs can be sorted in three main categories. For the first type, regular multirotors with at least six propellers are modified by fixing each propeller to a non-parallel direction to provide a wrench of six DOF [RRBF15, RMP+17,

PLA$^+$18, RCS19a]. This strategy comes with simple mechanical design, but the maximum inclination angle of the platform is limited because a large amount of thrust force is needed to balance the body weight. Furthermore, cancellation of thrust vectors internally are inherent to the design, which limits the energy efficiency and battery life.

The second type of UAVs adds actuated mechanisms to create additional DOFs. In one class of design, a regular quadcopter is connected to the main frame via two perpendicular rotational actuators to create exact six-DOF motion for the main frame [LLK$^+$20, NPC19]. The maximum inclination angle is still limited by the driving mechanism. In another class of design, individual propeller axes are articulated with respect to the main frame in one [RBG14, KVE$^+$18, BTKS18] or two rotational angles [GT18, SA13] to create redundant actuation configurations for six-DOF motions. Similar to the aforementioned designs, the articulating actuator-mechanism for each propeller reduces payload with added weight. Such UAVs can have high thrust efficiency if it has two angle adjustment like a gimbal, but suffer from low thrust efficiency if it only has one angle adjustment like a hinge. Furthermore, in the control system design, the actuator torques must be included as internal torques in complex multibody dynamics or as disturbances.

The third type of fully-actuated UAVs connects regular multirotor units to the main frame via passive joints and create the thrust vectors by steering the multirotor thrusts. This class has the advantage that multirotors are used for generating thrust as well as controlling the thrust angles. Therefore, the reduction of payload is minimal without the added active joint actuators. Cables [FMKK11, SST$^+$20] or spherical universal joints [NPPL18] have been used as the passive joints, where the angular range is still constrained by mechanical interference. Passive hinges of unlimited continuous rotations have been used [Rua20, PRY$^+$21] in a unique UAV, which connect the main frame to each of four quadcopters that control both the thrusts and hinge angles. However, the real-world hardware attitudes were limited by the maximum thrust magnitude andlow thrust efficiencies at non-horizontal poses. Continuous and unconstrained vectoring thrust actuator with conventional quadcopter and passive gimbals is proposed [YSG$^+$21], and the platform achieved any arbitrary attitude with uniformly

high thrust efficiency.

1.3 Controller Design of Fully-Actuated UAVs

Controller design of fully-actuated UAVs introduces additional challenges because fully-actuated UAVs generally have more actively controlled components than conventional UAVs, and may involve multibody dynamics which is more difficult to model.

Despite the mechanical complexity of fully-actuated UAVs, a majority of work consider the UAV as a rigid body and can generate any 6 DOF wrench [IGGL20]. The controller comes in a hierarchical or cascaded format, where the high-level calculates the whole-body wrench with feedback controllers, and the following part allocate the control input to each actuator based on the configuration of the UAV. For the type of fully-actuated UAV with propellers fixed to non-parallel directions, the UAV body can still be seen as a rigid body similar to conventional UAVs, and the controller design can come with more different configurations in addition to the basic structure [DSJ19, RCS19b]. [DSJ19] developed a different controller by formulating a constrained optimization problem to exploit the input redundancy. For UAV with conventional quadcopters with passive joints, optimization-based framework was proposed to exploit the over-actuation of the platform [Su21, SYG+21], and compensation control loop was proposed to use the input redundancy to improve the tracking performance [SRY+21].

1.4 Main Contributions

The main contributions of this thesis can be summarized below.

First, an over-actuated UAV platform with four novel vectoring thrust actuators is proposed. The actuator is realized by a quadcopter inside a gimbal with two passive rotational joints, which can freely rotate without constraint. Steering of the passive gimbal is achieved by the torque generated by the quadcopter. The actuator has three major advantages over

those of other systems: a) capable of generating thrust in any direction, with unlimited range of angulation; b) no internal torque exists between rotational joints that might impact system performance; c) system prototyping is mechanically simple because the regular quadcopter integrates thrust generation and steering. No wiring from the central frame is required because each quadcopter uses self-contained power, sensing, and communication.

Second, a prototype of the UAV platform is constructed and evaluated. Through simulation and experiment, the platform is shown to be capable of tracking 6-DOF position and attitude trajectories, and can achieve any arbitrary attitude such as flying in a stable vertical or upside-down orientation. Also, the platform exhibits high thrust efficiency in different body attitudes. To the best of the authors' knowledge, this is the first over-actuated UAV platform that achieves any arbitrary attitude in real-world experiment while maintains uniformly high thrust efficiency.

Third, the proposed UAV platform is demonstrated to have six independent control channels. The formulation of the closed-loop dynamics enables add-on feedforward and/or feedback controllers to be implemented on any of the control channels.

Fourth, both model-based and data-driven iterative learning control algorithms are implemented on the UAV platform to improve the performance of the system under various aerodynamic effects in real-world experiments. The data-driven iterative learning control shows the potential of constructing an accurate model of the aerodynamic effects from the experimental data, without prior knowledge of the model.

The thesis is orgainized as follows. Chapter 2 introduces the mechanical design of the over-actuated UAV platform. Chapter 3 analyzes the dynamics and hierarchical controller of the platform. Chapter 4 conducts several analysis on the mechanical and control properties of the platform. Chapter 5 shows the simulation and experiment results. Chapter 6 demonstrates the model-based iterative learning control applications. Chapter 7 demonstrates the data-driven iterative learning control applications on the platform. And Chapter 8 gives the conclusions.

CHAPTER 2

Design of the Over-actuated UAV Platform

2.1 Mechanical Design Considerations

The essential configuration of a fully-actuated UAV is that the platform must be able to exert thrust vectors in all x, y and z directions; and can generate torques in roll, pitch and yaw directions by differentiating the magnitude of thrusts. For UAV platform that can achieve any arbitrary attitude in the air, the thrust generation capability must be identical in all directions. Therefore, the idea in this thesis is to use multiple modular thrust generation units on the UAV platform, and each unit can generate thrust in any arbitrary direction. The requirements of such thrust generation module are summarized below.

1. The module can generate thrust in any arbitrary direction, and the maximum thrust is identical in different directions;

2. The module integrates all the thrust generation and manipulation functionalities, and the total number of the modules on the UAV platform can be changed easily;

3. The module is light-weighted.

A conventional quad-rotor multicopter is chosen as the thrust generation module because it satisfies all the requirements. The quadcopter can generate torque by increasing the thrusts of some propellers and decreasing the thrusts of other propellers. By differentiating the thrusts of different propeller pairs, the quadcopter can generate torques in all roll, pitch and yaw directions and therefore rotate its body in any direction. With proper design of connection mechanisms with the central frame such as rotational joints, the quadcopter can

reach and generate thrust in any arbitrary orientation. Our lab had previously proposed a thrust generation module using quadcopters and passive hinges in [PRY+21], where the quadcopter can rotate continuously in a 2D plane. The design idea of using light-weighted 3D-printed rotational joints is inherited and expanded to allow for continuous and unlimited rotation in 3D space, which will be introduced in this chapter.

2.2 Design of 2-DOF Gimbal Actuator

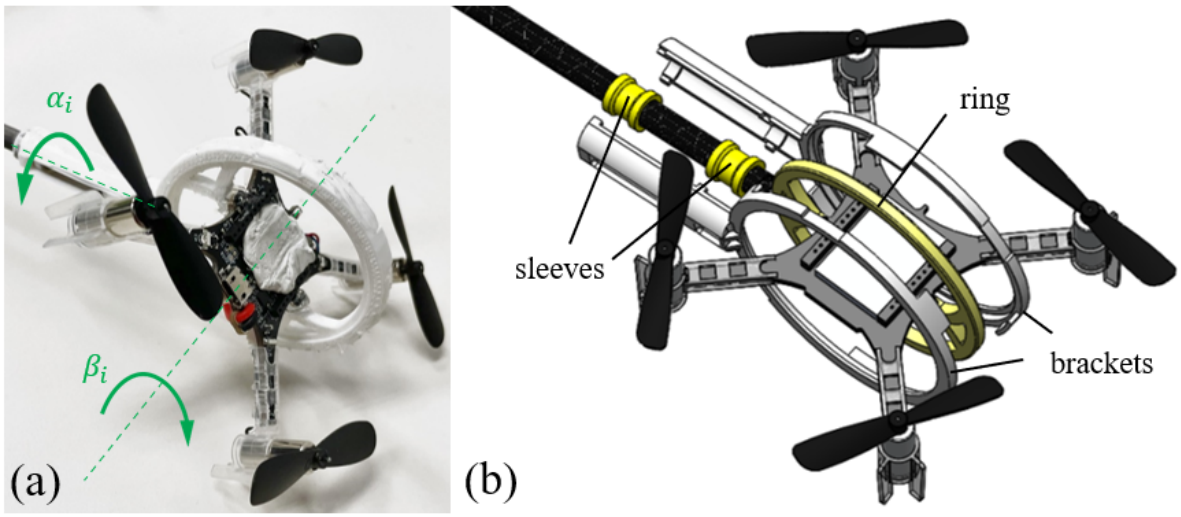


Figure 2.1: The "gimbal actuator". (a) Photograph of a gimbal actuator. Joint angles α_i and β_i rotate perpendicular to each other and intersect at the origin of the geometric center of the quadcopter. (b) Schematic showing the details of the 2-DOF gimbal mechanism.

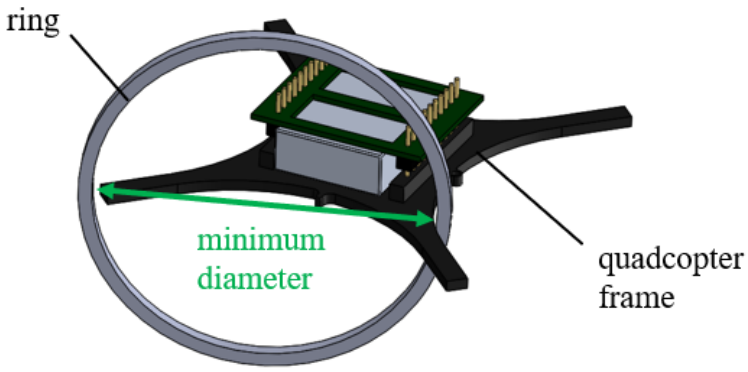


Figure 2.2: The minimum diameter of the ring considering the assembly process.

The first problem to address is the configuration of the connection joints between the

quadcopter and the central frame. We decided to use two perpendicular rotational joints because this is the minimum requirement to enable the thrust vector to point at any arbitrary direction. The thrust vector, denoted as $\boldsymbol{F}$, has 3 DOF if the quadcopter can reach any orientation in the air, such that $\boldsymbol{F} \in \mathbb{R}^3$. Meanwhile, the direction of the thrust vector, denoted as $\hat{\boldsymbol{F}}$, has 2 DOF which is the result of an additional constraint $\|\hat{\boldsymbol{F}}\| = 1$. Therefore, Two perpendicular rotational joints with continuous and unlimited angle will be the effective and efficient choice of connection design.

The thrust generation module consisting of a conventional quadcopter and 3D-printed joints are referred to as a "gimbal actuator" in this thesis. The configuration of the gimbal actuator is shown in Fig. 2.1. The gimbal actuator consists of two passive revolute joints, α_i and β_i, perpendicular to each other and intersect at the origin of the geometric center of the quadcopter. α_i is realized by attaching two sleeves on the shaft which is extended from the central frame. The two brackets wrap the shaft and the gimbal actuator can rotate on the sleeves. The two sleeves are separated to counteract the torque created by the gimbal actuator. β_i is realized by the ring around the quadcopter body. The gimbal mechanism can be seen as a serial link-joint mechanism consisting of 3 links and 2 joints. The first link is the central frame shaft and the sleeves attached on it. The second link is the two brackets. And the third link is the quadcopter and the link attached on it.

The components of the gimbal actuator are all fabricated by 3D printing to reduce manufacturing complexity and reduce the weight. Fused deposition modeling method [KDDCSH+18] is used, and we use PLA plastic as the material. Fast iteration process have been conducted to determine the optimal size and thickness of the components to balance the strength and the weight. The size of the ring is determined by the shape of the quadcopter frame considering the assembly process. As shown in Fig. 2.2, the quadcopter frame is a cross-shaped piece consisting of electronic units and a battery which cannot be separated in the middle. Therefore, the ring must be assembled from the side. The minimum diameter of the ring is the same as the distance between one tip of the frame and the edge on the other side of the tip. To reduce the friction of the joints, the layer of the 3D-printed parts are designed to be

perpendicular to the rotational axes. In addition, lubricant is applied on the joints.

2.3 Configuration of the Over-actuated UAV Platform

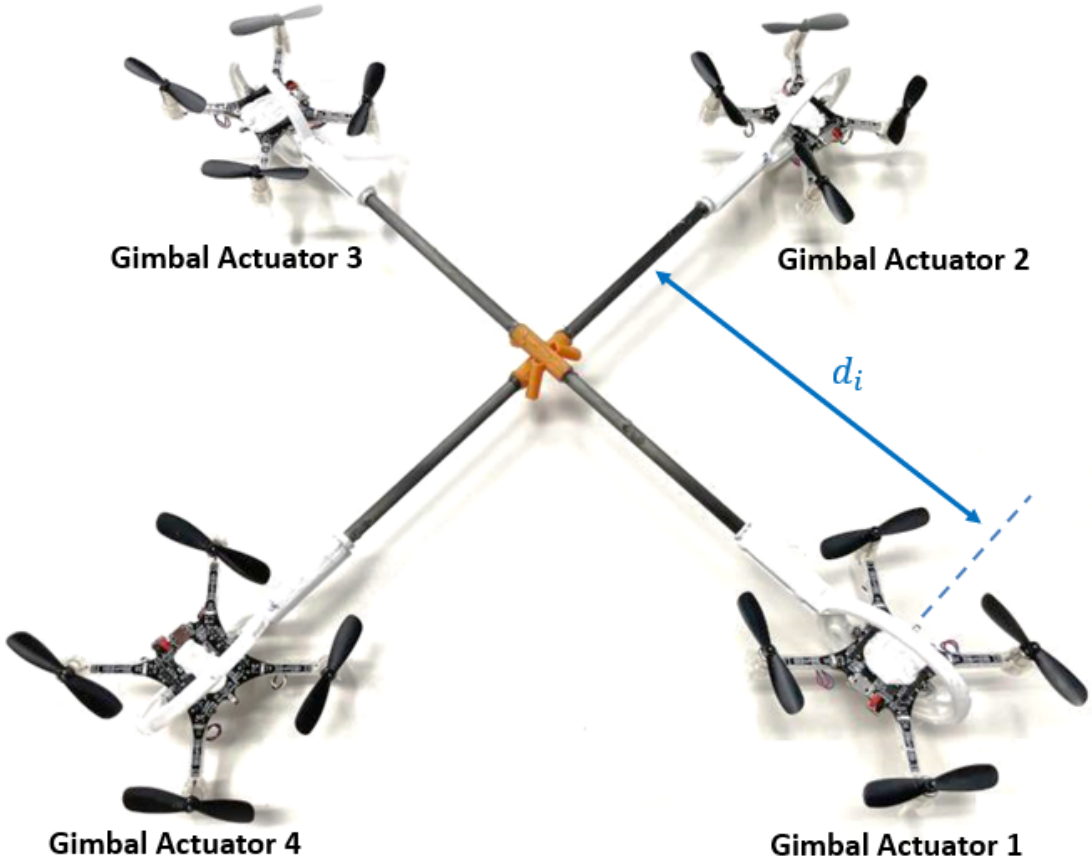


Figure 2.3: Design of the over-actuated UAV platform consisting of 4 gimbal actuators.

From the analysis and validation given by [NPPL18], a minimum of 3 thrust generation modules are needed in order to achieve the full 6-DOF actuation in the air. To make the platform more symmetric and for the ease of the dynamics and control analyses in the following chapters, we use 4 gimbal actuators on our UAV platform. As shown in Fig. 2.3, the central frame of the platform consists of two carbon fiber tubes that are perpendicular and of the same length. The central 3D-printed piece rigidly connects the two tubes. At each end of the carbon fiber tubes attaches a gimbal actuator. The UAV platform is over-actuated and has a total of 12 DOF.

Table 2.1: Physical Parameters of the UAV Platform

Parameter	**Value**
Mass of the central frame	$0.012kg$
Mass of the quadcopter	$0.027kg$
Mass of the gimbal frame	$0.009kg$
Total mass of the UAV platform	$0.156kg$
Inertia of the central frame	$\text{diag}([3e^{-5}\ 3e^{-5}\ 6e^{-5}])kg \cdot m^2$
Inertia of the quadcopter	$\text{diag}([1.2e^{-5}\ 1.2e^{-5}\ 2.2e^{-5}])kg \cdot m^2$
Distance between UAV center and quadcopter	$0.18m$
Distance between adjacent propellers on quadcopter	$0.032m$
Air-resistance torque coefficient k_τ	$0.00597, \tau = k_\tau f_i$
Maximum thrust of motor	$0.147N$

The overall dimension of the platform is a 48cm $\times$ 48cm $\times$ 6cm box. Commercial quadcopter Crazyflie 2.1 from Bitcraze [GSW+17] is used on the gimbal actuator, and BETAFPV 7x16mm DC motors are used on the quadcopter. The parameters of the platform is summarized in Table 2.1. The mass of the components are measured, while the inertia of the components are estimated from the CAD models. Although the selected quadcopter doesn't have higher thrust-to-weight ratio than other over-actuated platforms such as [BBP+19], it is enough to demonstrate all capabilities of our UAV platform. In fact, using the regular quadcopter and 3D printed gimbal frame provides an advantage that different quadcopters can be easily replaced based on the task requirements.

CHAPTER 3

System Dynamics and Controller Design

3.1 System Dynamics

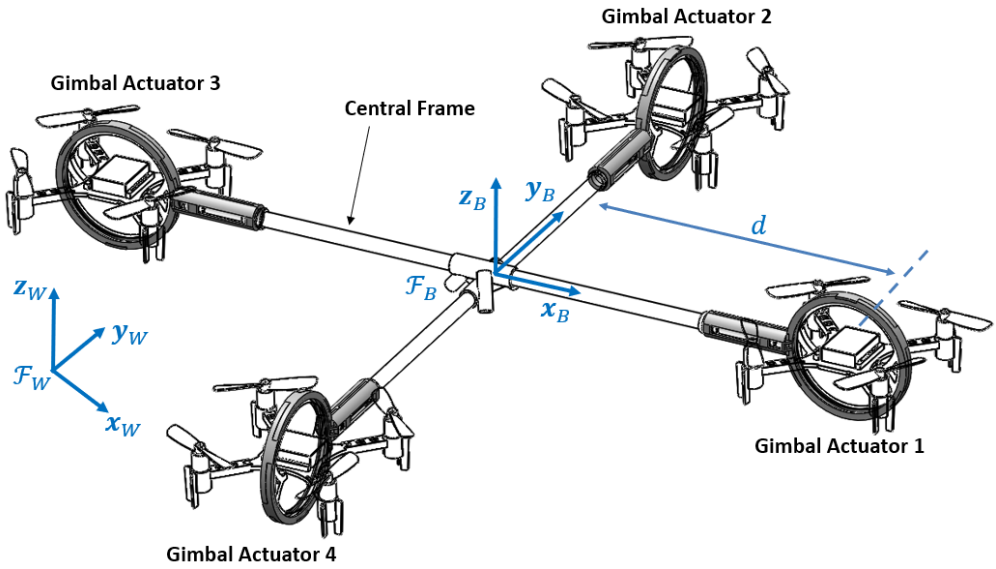


Figure 3.1: Schematics of the proposed over-actuated platform.

The symbols and parameters being used in this thesis are summarized in Table 3.1. The world coordinate frame is denoted as $\mathcal{F}_W$, where axes are defined under East-North-Up (ENU) convention. The platform frame $\mathcal{F}_B$ is attached to the geometric center of the over-actuated UAV platform, and is fixed with the central frame. The layout of the platform is shown in Fig. 3.1. To assist derivation of the controller equations, additional platform frames $\mathcal{F}_{Bi}$ are defined by rotating $\mathcal{F}_B$ along $+\boldsymbol{z}$ axis for $\pi(i-1)/2$ rad and placed at the geometric center of each gimbal as shown in Fig. 3.2, and is also fixed with the central frame. Gimbal actuator frames $\mathcal{F}_i$ are attached to the geometric center of the ith quadcopter.

Table 3.1: List of Symbols

Symbol	Description
i	The order of each gimbal actuator. $i \in \{1, \ldots, 4\}$
$\mathcal{F}_W$	World frame $\{O; \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ under East-North-Up (ENU) convention.
$\mathcal{F}_B$	Body frame attached the center of the UAV platform.
$\mathcal{F}_{Bi}$	Body frame attached to gimbal location of the UAV platform
$\mathcal{F}_i$	Body frame attached to the center of the ith quadcopter.
m_0, m_i, m	Mass of the central frame, the ith quadcopter, and the whole platform.
$\boldsymbol{J}_0$, $\boldsymbol{J}_i$, $\boldsymbol{J}$	$\mathbb{R}^{3\times3}$ inertia of the central frame, ith gimbal actuator, and the whole platform
$\boldsymbol{d}_i$	Distance vector from the origin of $\mathcal{F}_B$ to $\mathcal{F}_{Bi}$
f_i	Thrust magnitude for the ith gimbal actuator
α_i	The first rotational joint angle for the ith gimbal actuator
β_i	The second rotational joint angle for the ith gimbal actuator
$\boldsymbol{p}$, $\boldsymbol{\theta}$	Position and orientation in $\mathbb{R}^3$ of the platform.
$\boldsymbol{v}$	Linear velocity in $\mathbb{R}^3$ of the central frame
$\boldsymbol{\omega}$	Angular velocity in $\mathbb{R}^3$ of the central frame
${}^{X}[\cdot]$	Physical term expressed in $\mathcal{F}_X$ other than $\mathcal{F}_W$.
${}^{X}_{Y}R$	Rotation matrix in SO(3) from $\mathcal{F}_X$ to $\mathcal{F}_Y$.
$s[\cdot]$, $c[\cdot]$	Simplified notation of $sin[\cdot]$ and $cos[\cdot]$ respectively.
$I(n)$	Identity matrix of dimension n

3.1.1 Dynamics of the Gimbal Actuator

Each gimbal actuator can generate a thrust along its $+\boldsymbol{z}_i$ axis and enact a torque about each of its three principal axes by differentiating the rotational speed of its propellers. The inertia of the 3D-printed gimbal frame is neglected and the Newton–Euler dynamics for ith

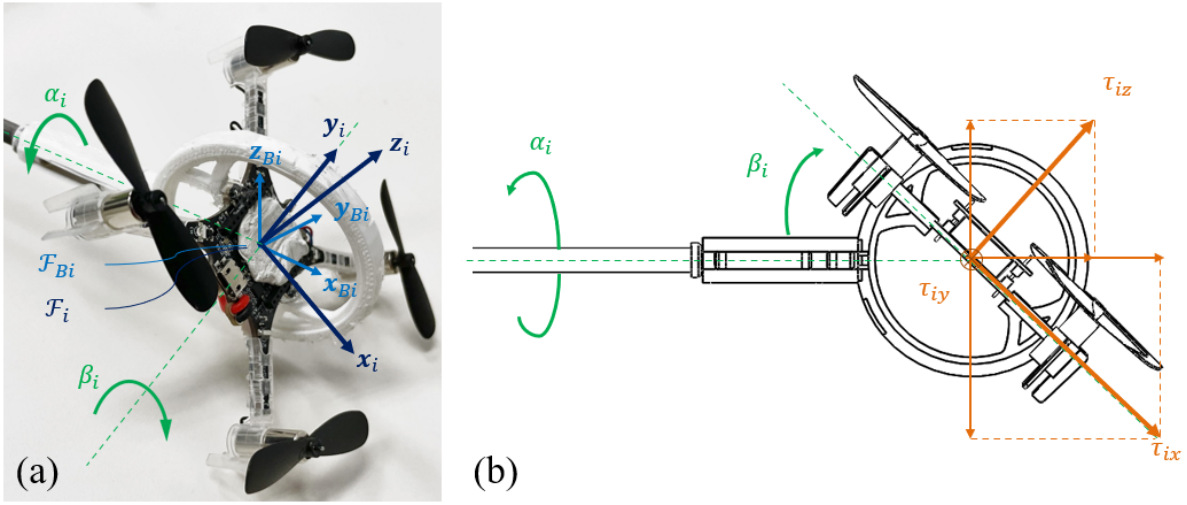


Figure 3.2: The symbols related to the gimbal actuator. (a) $\mathcal{F}_{Bi}$ is at the geometric center of each gimbal and fixed with the central frame; $\mathcal{F}_i$ is attached to the geometric center of the ith quadcopter. (b) 3-DOF torque generated by the quadcopter on gimbal.

gimbal actuator can be written as

$$m_i\, {}^i\dot{\boldsymbol{v}}_i = {}^W_iR^{\mathsf{T}}\, m_i g\hat{\boldsymbol{z}} - {}^B_iR^{\mathsf{T}}\, {}^B\boldsymbol{N}_i + f_i\hat{\boldsymbol{z}} \tag{3.1}$$

$$J_i\, {}^i\dot{\boldsymbol{\omega}}_i + {}^i\boldsymbol{\omega}_i \times J_i\, {}^i\boldsymbol{\omega}_i = \boldsymbol{\tau}_i - {}^B_iR^{\mathsf{T}}\, {}^B\boldsymbol{T}_i + \boldsymbol{\delta}_i \times {}^W_iR^{\mathsf{T}}\, m_i g\hat{\boldsymbol{z}} \tag{3.2}$$

where m_i and J_i are mass and inertia of the gimbal actuator. ${}^i\dot{\boldsymbol{v}}_i$ and ${}^i\boldsymbol{\omega}_i$ are the linear and angular velocities in $\mathcal{F}_i$. f_i represents the magnitude of thrust generated by the gimbal actuator, $\hat{z} = [0\ 0\ 1]^{\mathsf{T}}$. $\boldsymbol{\tau}_i$ is the torque generated by the gimbal actuator. ${}^B\boldsymbol{N}_i$ and ${}^B\boldsymbol{T}_i$ are the force and torque on the central frame exerted by the gimbal actuator. $\boldsymbol{\delta}_i$ is the position vector from the geometric center to the center of mass (COM) of the gimbal actuator.

Because of the "ring" design shown in Section 2.2, the COM of the gimbal actuator is located coincident with both α_i and β_i. Therefore, the torque due to the imbalance of gravity $\delta_i \times {}^W_iR^{\mathsf{T}} m_i g\hat{\boldsymbol{z}}$ is set to zero in the following derivation.

${}^i\dot{\boldsymbol{\omega}}_i$ and ${}^i\boldsymbol{\omega}_i$ can be expressed by α_i and β_i following standard kinematic equations in

[Cra09]:

$$ {}^{i}\dot{\boldsymbol{\omega}}_i = \begin{bmatrix} c\beta_i\ddot{\alpha}_i - s\beta_i\dot{\alpha}_i\dot{\beta}_i \\ \ddot{\beta}_i \\ s\beta_i\ddot{\alpha}_i + c\beta_i\dot{\alpha}_i\dot{\beta}_i \end{bmatrix}, {}^{i}\boldsymbol{\omega}_i = \begin{bmatrix} c\beta_i\dot{\alpha}_i \\ \dot{\beta}_i \\ s\beta_i\dot{\alpha}_i \end{bmatrix} \tag{3.3} $$

where $s[\cdot]$, $c[\cdot]$ are simplified expressions of $\sin[\cdot]$ and $\cos[\cdot]$.

The rotational dynamics Eq. (3.2) can be written as

$$ \begin{aligned} \boldsymbol{M}_i \begin{bmatrix} \ddot{\alpha}_i & \ddot{\beta}_i \end{bmatrix}^{\mathsf{T}} + \boldsymbol{C}_i &= \boldsymbol{\tau}_i - {}^{B}_{i}R^{\mathsf{T}}\, {}^{B}\boldsymbol{T}_i \\ \boldsymbol{M}_i = \begin{bmatrix} J_{ix}c\beta_i & 0 \\ 0 & J_{iy} \\ J_{iz}s\beta_i & 0 \end{bmatrix}, \boldsymbol{C}_i &= \begin{bmatrix} (J_{iz} - J_{ix} - J_{iy})s\beta_i\dot{\alpha}_i\dot{\beta}_i \\ (J_{ix} - J_{iz})s\beta_i c\beta_i\dot{\alpha}_i^2 \\ J_{iz}c\beta_i\dot{\alpha}_i\dot{\beta}_i \end{bmatrix} \end{aligned} \tag{3.4} $$

where J_{ix}, J_{iy} and J_{iz} are diagonal elements of J_i, $s[\cdot]$, $c[\cdot]$ are simplified expressions of $\sin[\cdot]$ and $\cos[\cdot]$. Off-diagonal elements of J_i are neglected due to the symmetry of the gimbal actuator.

3.1.2 Dynamics of the UAV Platform

The central frame is subject to gravity as well as forces and torques transmitted by the gimbal frames from each gimbal actuator. The dynamics of the central frame expressed in $\mathcal{F}_B$ can be written as

$$ m_0\, {}^{B}\dot{\boldsymbol{v}} = {}^{W}_{B}R^{\mathsf{T}} m_0 g\hat{\boldsymbol{z}} + \sum_{i=1}^{4} {}^{B}\boldsymbol{N}_i \tag{3.5} $$

$$ J_0\, {}^{B}\dot{\boldsymbol{\omega}} + {}^{B}\boldsymbol{\omega} \times J_0\, {}^{B}\boldsymbol{\omega} = \sum_{i=1}^{4} (\boldsymbol{d}_i \times {}^{B}\boldsymbol{N}_i + {}^{B}\boldsymbol{T}_i) + {}^{B}\boldsymbol{\tau}_g \tag{3.6} $$

where m_0 and J_0 are mass and inertia of the central frame. $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are linear and angular velocity of platform. ${}^{B}\boldsymbol{\tau}_g$ is the gravity torque due to the displacement of its COM from the

geometric center [GT18]. $\boldsymbol{d}_i$ is the distance vector from the center of $\mathcal{F}_B$ to each $\mathcal{F}_{Bi}$, and

$$\boldsymbol{d}_1 = \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{d}_2 = \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix}, \boldsymbol{d}_3 = \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{d}_4 = \begin{bmatrix} 0 \\ -d \\ 0 \end{bmatrix}. \tag{3.7}$$

An assumption is made in this thesis that the desired velocity of the platform is slow. The second-order term involving the square of the small angular velocity ${}^B\boldsymbol{\omega} \times J_0 {}^B\boldsymbol{\omega}$ can be neglected for the ease of the derivation, and the dynamics can be derived in a compact format as follows.

From Eq. (3.1), we have

$$\begin{aligned} {}^B\boldsymbol{N}_i &= -{}^B_iR m_i {}^i\dot{\boldsymbol{v}}_i + {}^B_iR {}^W_iR^{\mathsf{T}} m_i g \hat{\boldsymbol{z}} + {}^B_iR f_i \hat{\boldsymbol{z}} \\ &= -m_i {}^B\dot{\boldsymbol{v}}_i + {}^W_BR^{\mathsf{T}} m_i g \hat{\boldsymbol{z}} + {}^B_iR f_i \hat{\boldsymbol{z}} \end{aligned} \tag{3.8}$$

where ${}^B\dot{\boldsymbol{v}}_i$ is velocity of the gimbal actuator in $\mathcal{F}_B$.

From [Cra09], under the low-velocity assumption, we have

$${}^B\dot{\boldsymbol{v}}_i = {}^B\dot{\boldsymbol{v}} + {}^B\dot{\boldsymbol{\omega}} \times \boldsymbol{d}_i \tag{3.9}$$

First we apply Eqs. (3.8) and (3.9) to Eq. (3.5) and the translational dynamics of the whole platform can be expressed as

$$m\, {}^B\dot{\boldsymbol{v}} = {}^W_BR^{\mathsf{T}} m g \hat{\boldsymbol{z}} + \sum_{i=1}^{4} {}^B_iR f_i \hat{\boldsymbol{z}} \tag{3.10}$$

where $m = m_0 + \sum_{n=1}^{4} m_i$. Because of the symmetry of the platform, $\sum_{i=1}^{4} m_i {}^B\dot{\boldsymbol{\omega}} \times \boldsymbol{d}_i = 0$ and is already neglected.

Next we apply Eqs. (3.8) and (3.9) to Eq. (3.6) and the rotational dynamics can be

expressed as

$$J_0\,{}^B\dot{\boldsymbol{\omega}} = \sum_{i=1}^{4}(\boldsymbol{d}_i \times (-m_i\,{}^B\dot{\boldsymbol{\omega}} \times \boldsymbol{d}_i + {}^B_iR f_i\hat{\boldsymbol{z}}) + {}^B\boldsymbol{T}_i) + {}^B\boldsymbol{\tau}_g \tag{3.11}$$

Similarly, because of the symmetry of $\boldsymbol{d}_i$,

$$\sum_{i=1}^{4}\boldsymbol{d}_i \times (-m_i\,{}^B\dot{\boldsymbol{v}} + {}^W_BR^{\mathsf{T}} m_i g\hat{\boldsymbol{z}}) = 0 \tag{3.12}$$

Meanwhile, applying Eq. (3.7) we have

$$\sum_{i=1}^{4}\boldsymbol{d}_i \times (m_i\,{}^B\dot{\boldsymbol{\omega}} \times \boldsymbol{d}_i) = d^2\begin{bmatrix} m_2+m_4 & 0 & 0 \\ 0 & m_1+m_3 & 0 \\ 0 & 0 & \sum_{n=1}^{4} m_i \end{bmatrix}{}^B\dot{\boldsymbol{\omega}} \tag{3.13}$$

And Eq. (3.11) can be expressed as

$$J\,{}^B\dot{\boldsymbol{\omega}} = \sum_{i=1}^{4}(\boldsymbol{d}_i \times {}^B_iR f_i\hat{\boldsymbol{z}} + {}^B\boldsymbol{T}_i) + {}^B\boldsymbol{\tau}_g \tag{3.14}$$

where

$$J = J_0 + d^2\begin{bmatrix} m_2+m_4 & 0 & 0 \\ 0 & m_1+m_3 & 0 \\ 0 & 0 & \sum_{n=1}^{4} m_i \end{bmatrix} \tag{3.15}$$

Because the main purpose of the gimbal is to change the direction of thrust, direct torque transmission between gimbal actuator and central frame is unnecessary. In the controller design Section 3.2, ${}^B\boldsymbol{T}_i$ is set to be zero when calculating the desired thrust and torque commands.

Dynamics of the whole platform Eqs. (3.10) and (3.14) has the same format as a rigid body under slow motion with equivalent mass m and inertia J, subject to gravity and force by gimbal actuators. In most over-actuated UAV platforms whose joints are actuated by servo motors [KVE+18, RBG14, GT18], only by neglecting the internal torque between joints can

a platform be treated as a rigid body. For our platform, thanks to the passive joint design, the dynamics is naturally closer to a rigid-body model, which makes model-based controller design on our platform more reasonable.

The second source of disturbance is that the propellers on a fully-actuated UAV is subject to air-resistance torque. At horizontal hovering mode the air-resistance torques are cancelled by propeller pairs that have opposite rotational directions, but when the vehicle body is inclined, the propellers are pointing at different directions, where the air-resistance torques cannot be compensated by themselves. Proper modeling of the magnitude of the air-resistance torque is necessary to derive a compensation algorithm; otherwise, the air-resistance torque is treated as disturbance that will affect the performance of the vehicle. On the proposed over-actuated platform with gimbal actuators, the air-resistance torques are always cancelled by different propellers within each single quadcopter.

3.2 Hierarchical Controller Design

Compared with conventional under-actuated multi-rotor copters, the proposed UAV platform is over-actuated and has control authorities in all 6 DOF in the air. Therefore, a hierarchical controller capable of tracking an arbitrary trajectory in position and attitude is developed for the platform, as shown in . In Section 3.1, the platform can be treated as a rigid body and is actuated by the thrust provided by each of the four gimbal actuators. A high-level controller based on the rigid body dynamics is developed, where the control inputs are the virtual 6-DOF wrench (force and torque) for the whole body. The virtual wrench is then allocated to the gimbal actuators by a wrench mapper, such that the vector sum of all gimbal actuators are equivalent to the high-level virtual wrench at all times. From desired thrust vector, the desired magnitude of thrust and two joint angles can be calculated for each gimbal actuator, and is tracked by an onboard low-level controller.

Although the hierarchical control scheme that determines a virtual wrench and allocates to each unit is also used by various over-actuated UAV platforms [KVE+18,GT18], the detailed

configurations of the high-level controller and the low-level controller are specially designed to guarantee the best performance of the proposed over-actuated UAV platform. Also, the stability of the UAV platform is only determined by the high-level controller, if it has slower closed-loop dynamics than low-level controller [ABN80]. Therefore, the closed-loop behavior of the system is also carefully modeled by the controller design that will be introduced in this chapter.

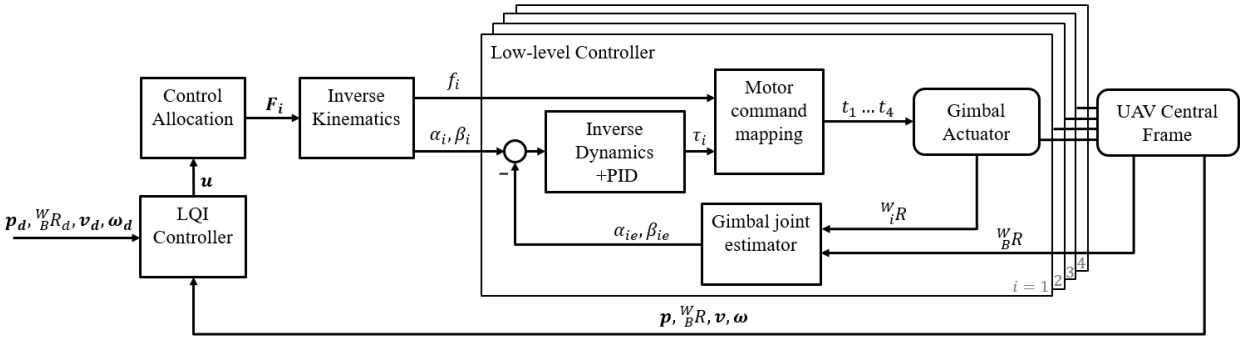


Figure 3.3: Hierarchical control diagram of the over-actuated aerial platform. The high-level LQI controller gives input $\boldsymbol{u}$ as the virtual wrench that regulates the position and attitude of the whole platform. The virtual wrench is then allocated as the desired magnitude of thrust f_i and joint angles α_i, β_i that are tracked by individual gimbal actuator with onboard low-level controller.

3.2.1 High-Level Control of Platform

A model-based linear-quadratic-integral (LQI) control scheme [YW72] is chosen as the high-level controller of the platform out of two reasons. First, the accurate dynamics model of the system is available without making an assumption of neglecting the internal torque at the joints, so the linear-quadratic approach gives an optimal control sequence based on the designed performance index. Second, to account for the modeling error and ensure the tracking performance over a period of time, integral action is necessary.

We first define an equivalent body wrench jointly generated by four gimbal actuators as

${}^B\boldsymbol{u}$ in $\mathcal{F}_B$, and as $\boldsymbol{u}$ in $\mathcal{F}_W$:

$$ {}^B\boldsymbol{u} = \begin{bmatrix} \sum_{n=1}^{4} {}^B_i R f_i \hat{\boldsymbol{z}} \\ \sum_{n=1}^{4} (\boldsymbol{d}_i \times {}^B_i R f_i \hat{\boldsymbol{z}}) \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} {}^W_B R & 0 \\ 0 & {}^W_B R \end{bmatrix} {}^B\boldsymbol{u} \tag{3.16} $$

Then the platform dynamics Eqs. (3.10) and (3.14) in $\mathcal{F}_W$ is written as

$$ \begin{bmatrix} m\dot{\boldsymbol{v}} \\ \boldsymbol{J}\dot{\boldsymbol{\omega}} \end{bmatrix} = \boldsymbol{u} + \begin{bmatrix} mg\hat{\boldsymbol{z}} \\ {}^W_B R \, {}^B\boldsymbol{\tau}_g \end{bmatrix} \tag{3.17} $$

Combining with $\dot{\boldsymbol{p}} = \boldsymbol{v}$ and $\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}$, where $\boldsymbol{p}$ is position and $\boldsymbol{\theta}$ is attitude expressed as Euler angles, the continuous-time platform dynamics can be written in state-space equation as

$$ \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{G} \tag{3.18} $$

where

$$ \boldsymbol{A} = \begin{bmatrix} 0 & 0 & \boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m}\boldsymbol{I}_3 & 0 \\ 0 & \boldsymbol{J}^{-1} \end{bmatrix}, \; \boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\theta} \\ \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}, \; \boldsymbol{G} = \begin{bmatrix} 0 \\ 0 \\ g\hat{\boldsymbol{z}} \\ \boldsymbol{J}^{-1} \, {}^W_B R \, {}^B\boldsymbol{\tau}_g \end{bmatrix} $$

On the other hand, the tracking error in 6-DOF position and attitude defines the performance of the controller, thus the system states must be combined with the reference trajectory to evaluate the performance and calculate the optimal inputs. The states of the tracking error are defined as

$$ \boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{e}_p \\ \boldsymbol{e}_\theta \\ \boldsymbol{e}_v \\ \boldsymbol{e}_\omega \end{bmatrix} = \begin{bmatrix} \boldsymbol{p} - \boldsymbol{p}_d \\ \frac{1}{2}[{}^W_B R^\intercal \, {}^W_B R_d - {}^W_B R_d^\intercal \, {}^W_B R]^\vee \\ \boldsymbol{v} - \boldsymbol{v}_d \\ \boldsymbol{\omega} - \boldsymbol{\omega}_d \end{bmatrix} \tag{3.19} $$

where the subscript d indicates desired values. The operator $[\cdot]^\vee$ is the isomorphism from Lie algebra $\mathfrak{so}(3)$ to $\mathbb{R}^3$ [CSM11]. The reason to use the Lie algebra isomorphism is to make the error states contain the non-linearity of attitude, such that Eq. (3.18) is in linear form.

The LQI control scheme includes integration of position and attitude error as an internal model to improve trajectory tracking performance. The states of the internal model are defined as

$$\dot{\boldsymbol{x}}_m = \begin{bmatrix} \boldsymbol{e}_p \\ \boldsymbol{e}_\theta \end{bmatrix} = \boldsymbol{C}_m \boldsymbol{x}_e \tag{3.20}$$

where

$$\boldsymbol{C}_m = \begin{bmatrix} \boldsymbol{I}_3 & 0 & 0 & 0 \\ 0 & \boldsymbol{I}_3 & 0 & 0 \end{bmatrix}$$

The closed-loop platform dynamics is augmented with the internal model and the augmented model are used to design the LQI controller gains. The augmented system is written as

$$\dot{\boldsymbol{x}}_{aug} = \boldsymbol{A}_{aug} \boldsymbol{x}_{aug} + \boldsymbol{B}_{aug} \widetilde{\boldsymbol{u}} \tag{3.21}$$

$$\begin{bmatrix} \dot{\boldsymbol{x}}_e \\ \dot{\boldsymbol{x}}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{C}_m & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_e \\ \boldsymbol{x}_m \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix} \boldsymbol{u}_{aug} \tag{3.22}$$

where $\boldsymbol{u}_{aug} = \boldsymbol{u} + \boldsymbol{B}^\dagger \boldsymbol{G}$, and $\boldsymbol{B}^\dagger = (\boldsymbol{B}^\mathsf{T}\boldsymbol{B})^{-1}\boldsymbol{B}^\mathsf{T}$ is the Moore–Penrose inverse of $\boldsymbol{B}$.

The cost function of the augmented system is

$$J(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{u}}) = \int_0^\infty (\widetilde{\boldsymbol{x}}^\mathsf{T} Q \widetilde{\boldsymbol{x}} + \widetilde{\boldsymbol{u}}^\mathsf{T} R \widetilde{\boldsymbol{u}}) dt \tag{3.23}$$

where Q, R are positive definite matrices that are designed to balance the system bandwidth and the control efforts.

The optimal solution to Eq. (3.23) is obtained by solving the algebraic Riccati equation

$$\boldsymbol{A}^\mathsf{T} P + P\boldsymbol{A} - PMP + Q = 0 \tag{3.24}$$

where P is a self-adjoint matrix to be solved, and $M = \boldsymbol{B}R^{-1}\boldsymbol{B}^{\mathsf{T}}$. Then the optimal input that minimizes the cost is

$$\boldsymbol{u}_{aug} = -K\boldsymbol{x}_{aug} = -R^{-1}\boldsymbol{B}^{\mathsf{T}}P\boldsymbol{x}_{aug} \tag{3.25}$$

And the actual input $\boldsymbol{u}$ in $\mathcal{F}_W$ is

$$\boldsymbol{u} = -K\boldsymbol{x_{aug}} - \boldsymbol{B}^{\dagger}\boldsymbol{G} \tag{3.26}$$

Because the proposed over-actuated platform can enact 6 independent forces and torques, there is no need to restrict the behavior of the intersections of different control channels. Therefore, the matrices Q, R are designed to be diagonal. Solving Eqs. (3.23) and (3.24), the format of the optimal gain matrix can be obtained as $K = [K_1\ K_2\ K_3]$, where K_1, K_2, K_3 are diagonal matrices in $\mathbb{R}^{6\times 6}$.

3.2.2 Control Allocation

The body wrench ${}^{B}\boldsymbol{u}$ is then allocated to be the desired thrusts and joint angles of the gimbal actuators. Because the gimbal actuators have full directional authority, the thrust vectors are first obtained from ${}^{B}\boldsymbol{u}$ and then used to determine the joint angles, as shown in Fig. 3.4.

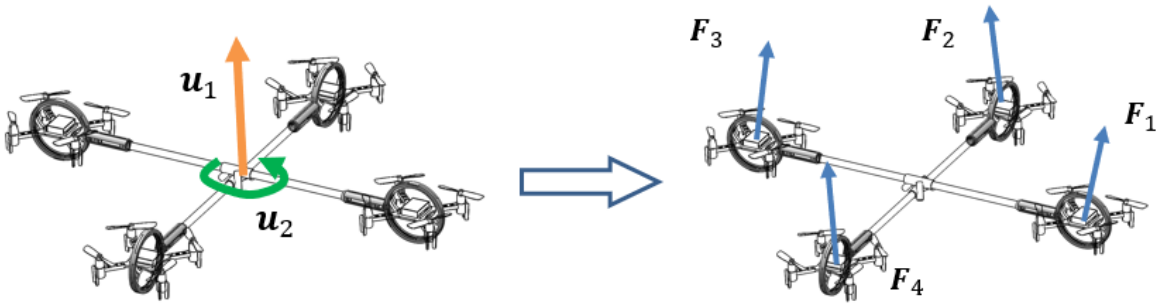


Figure 3.4: Mapping from the virtual wrench to thrust vector of each gimbal actuator.

We define $\boldsymbol{F}_i = [F_{ix}\ F_{iy}\ F_{iz}]^{\mathsf{T}}$ as the desired thrust vector of gimbal actuator i expressed

in $\mathcal{F}_{Bi}$, and $\boldsymbol{F} = [\boldsymbol{F_1}\ \boldsymbol{F_2}\ \boldsymbol{F_3}\ \boldsymbol{F_4}]^\mathsf{T}$. With this definition we can rewrite Eq. (3.16) as

$$^{B}\boldsymbol{u} = W\boldsymbol{F} \tag{3.27}$$

where

$$W = \begin{bmatrix} {}^{B}_{B1}R & ... & {}^{B}_{B4}R \\ (\boldsymbol{d}_1)^\wedge\, {}^{B}_{B1}R & ... & (\boldsymbol{d}_4)^\wedge\, {}^{B}_{B4}R \end{bmatrix} \tag{3.28}$$

and $[\cdot]^\wedge$ is the isomorphic operator from $\mathbb{R}^3$ to $\mathfrak{so}(3)$. With Eq. (3.7) we have

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & d & 0 & 0 & 0 & 0 & 0 & d \\ 0 & 0 & -d & 0 & 0 & 0 & 0 & 0 & d & 0 & 0 & 0 \\ 0 & d & 0 & 0 & d & 0 & 0 & d & 0 & 0 & d & 0 \end{bmatrix} \tag{3.29}$$

As shown in Eq. (3.29), $W \in \mathbb{R}^{6\times 12}$ is a constant matrix, and $\text{rank}(W) = 6$. It is then feasible to obtain a least-norm solution for $\boldsymbol{F}$ such that

$$\boldsymbol{F} = W^{\dagger}\, {}^{B}\boldsymbol{u} \tag{3.30}$$

The least-norm solution guarantees a minimum control effort provided jointly by the four thrust vectors. Also, the pseudo-inverse of W is constant as well, which can be calculated offline to save the onboard computational power.

The direction cosine matrix from $\mathcal{F}_{Bi}$ to $\mathcal{F}_i$ can be obtained by subsequent rotation of α_i and β_i:

$${}^{Bi}_{i}R = \begin{bmatrix} c\beta_i & 0 & s\beta_i \\ s\alpha_i s\beta_i & c\alpha_i & -s\alpha_i c\beta_i \\ -c\alpha_i s\beta_i & s\alpha_i & c\alpha_i c\beta_i \end{bmatrix} \tag{3.31}$$

And each thrust vector $\boldsymbol{F}_i$ can be written as

$$\boldsymbol{F}_i = {}^{Bi}_{i}R f_i \hat{\boldsymbol{z}} = f_i \begin{bmatrix} s\beta_i \\ -s\alpha_i c\beta_i \\ c\alpha_i c\beta_i \end{bmatrix} \tag{3.32}$$

Thus we have

$$\begin{aligned} f_i &= \|\boldsymbol{F}_i\| \\ \alpha_i &= \text{atan2}(-F_{iy}, F_{iz}) \\ \beta_i &= \text{asin}(F_{ix}/f_i) \end{aligned} \tag{3.33}$$

where $\text{atan2}(\cdot,\cdot)$ is the quadrant-corrected arc-tangent function with two inputs.

3.2.3 Low-Level Control of Gimbal Actuator

The onboard microprocessor on each gimbal actuator is responsible for tracking the desired magnitude of thrust and joint angles in the low-level control loop. The desired thrust f_i directly feeds into gimbal actuator motor command, while the desired joint angles α_i and β_i are controlled by the torque generated by the gimbal actuator. Because direct measurements of the joint angles by encoders are unavailable on the passive gimbal, an estimation by the onboard IMU is developed.

The attitude of the overall platform ${}^{W}_{B}R$ is sent to each gimbal actuator. The onboard IMU provides attitude measurements of the gimbal actuator. The reference frame of the IMU is calibrated at startup to be aligned with $\mathcal{F}_W$, so that the attitude of each gimbal actuator ${}^{W}_{i}R$ is available. Thus we have the measured rotation matrix

$${}^{Bi}_{i}R = ({}^{W}_{B}R\, {}^{B}_{Bi}R)^{\mathsf{T}}\, {}^{W}_{i}R \tag{3.34}$$

From Eq. (3.31), an estimate of α_i and β_i can be obtained by

$$\begin{aligned} \alpha_{ie} &= \text{atan2}({}^{Bi}_{i}R_{32}, {}^{Bi}_{i}R_{22}) \\ \beta_{ie} &= \text{atan2}({}^{Bi}_{i}R_{13}, {}^{Bi}_{i}R_{11}) \end{aligned} \tag{3.35}$$

where ${}^{Bi}_{i}R_{xy}$ indicates the matrix element of the xth row and yth column from Eq. (3.34).

In Eq. (3.4), ${}^{B}\boldsymbol{T}_i$ is always perpendicular to α_i and β_i because of the gimbal geometry. Also, since $\boldsymbol{M}_i$ is in full column rank despite the value of β_i, the 3-DOF $\boldsymbol{\tau}_i$ can always generate desired angular accelerations $\ddot{\alpha}_i$ and $\ddot{\beta}_i$ while making ${}^{B}\boldsymbol{T}_i$ to be zero. Therefore, 3.4 can be written as

$$\boldsymbol{\tau}_i = \boldsymbol{M}_i \begin{bmatrix} \ddot{\alpha}_i & \ddot{\beta}_i \end{bmatrix}^{\mathsf{T}} + \boldsymbol{C}_i \tag{3.36}$$

The joint angles are controlled by separate PID controllers based on the error dynamics as

$$\begin{aligned} \ddot{\alpha}_i &= k_{D\alpha}\dot{e}_\alpha + k_{P\alpha}e_\alpha + k_{I\alpha}\int e_\alpha dt \\ \ddot{\beta}_i &= k_{D\beta}\dot{e}_\beta + k_{P\beta}e_\beta + k_{I\beta}\int e_\beta dt \end{aligned} \tag{3.37}$$

where $k_{[.]\alpha}$ and $k_{[.]\beta}$ are constant PID gains, $e_\alpha = \alpha_{ie} - \alpha_i$ and $e_\beta = \beta_{ie} - \beta_i$.

Then the thrust of each propeller can be uniquely determined by the linear mapping:

$$\begin{bmatrix} f_{id} \\ \tau_{ix} \\ \tau_{iy} \\ \tau_{iz} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ a & -a & -a & a \\ -a & -a & a & a \\ -c_\tau & c_\tau & -c_\tau & c_\tau \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} \tag{3.38}$$

where $t_{[.]}$ is the desired thrust of each individual propeller, a is the distance from propeller to quadcopter axis $\boldsymbol{x}$ or $\boldsymbol{y}$, and c_τ is the ratio of propeller air resistance torque and thrust. The value of c_τ is provided by the vendor of the regular quadcopter. The propeller thrusts are then converted to PWM signals to drive the motors.

CHAPTER 4

Mechanical System Analysis

4.1 Thrust-Generation Capacity of the Gimbal Actuator

The torque used to control the gimbal angles is jointly generated by four propellers, and the maximum torques that actuate α_i and β_i are related to each other. To verify the thrust generation capability and angulation capability using the conventional quadcopter, the relation of maximum $\ddot{\alpha}_i$ and $\ddot{\beta}_i$ are calculated in this section. Commercial quadcopter Crazyflie 2.1 from Bitcraze [GSW+17] is used on the gimbal actuator, and BETAFPV 7x16mm DC motors are used on the quadcopter. The parameters of the quadcopter are obtained from the manufacturer and are summarized in Table 2.1. Given these parameters, the maximum angular velocities can be calculated from Eqs. (3.36) and (3.38). The evaluation is given under the condition that a gimbal actuator provides thrust equal to 1/4 of the platform gravity (therefore each propeller provides 1/16 of total gravity). In the calculations, the joint velocities are neglected because the desired trajectory of the platform is assumed to be slow.

The result in Fig. 4.1 shows that when $\ddot{\alpha}_i = 0$, $\ddot{\beta}_i$ has a maximum value of 340 rad/s^2. Meanwhile, when $\ddot{\beta}_i = 0$, the maximum value of $\ddot{\alpha}_i$ depends on β_i. The gimbal actuator generates a smallest maximum $\ddot{\alpha}_i$ of 64 rad/s^2 when $\beta_i = \pi/2$ (or $-\pi/2$ due to symmetry). This is because α_i is actuated by τ_{iz} when $\beta_i = \pi/2$. In addition, τ_{iz} is generated by the air-resistance torque which has a smaller torque-to-thrust ratio than τ_{ix} or τ_{iy}. However, the capability of the gimbal actuator when $\beta_i = \pi/2$ is sufficient for the experimental evaluation. For more aggressive maneuvering, more suitable quadcopters can be easily incorporated into the system thanks to the advantages of the design.

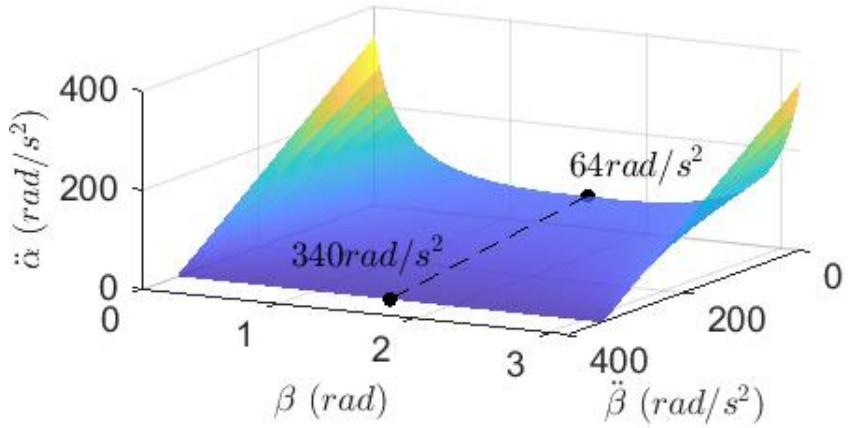


Figure 4.1: Relation of maximum $\ddot{\alpha}_i$ and $\ddot{\beta}_i$.

4.2 Handling of Singularity

Ideally the hierarchical controller proposed in Section 3.2 is effective to regulate the over-actuated platform at any arbitrary attitude, because the system dynamics can be seen as a rigid body and presents the same property at different attitude as analyzed in Section 3.1. In Chapter 5, the simulation and experiment results will show that the controller is robust against noises and unmodelled dynamics. However, when the platform is rotated around its roll or pitch direction for around 90° , the whole platform will be unstable under this control scheme. This is because the individual thrust actuators will be subject to singularity caused by the use of hierarchical controller [BTKS18]. The cause of the singularity and the singularity handling techniques are analyzed below.

4.2.1 Singularity Configuration of the Platform

As shown in Fig. 4.2, when the platform is at vertical flight configuration, the gimbal actuators directly above and underneath the central frame are in kinematic singularity. The singularity on gimbal actuators are caused by the use of the hierarchical controller, and is revealed by solving for α_i and β_i from the desired thrust. Taking derivatives of the normalized

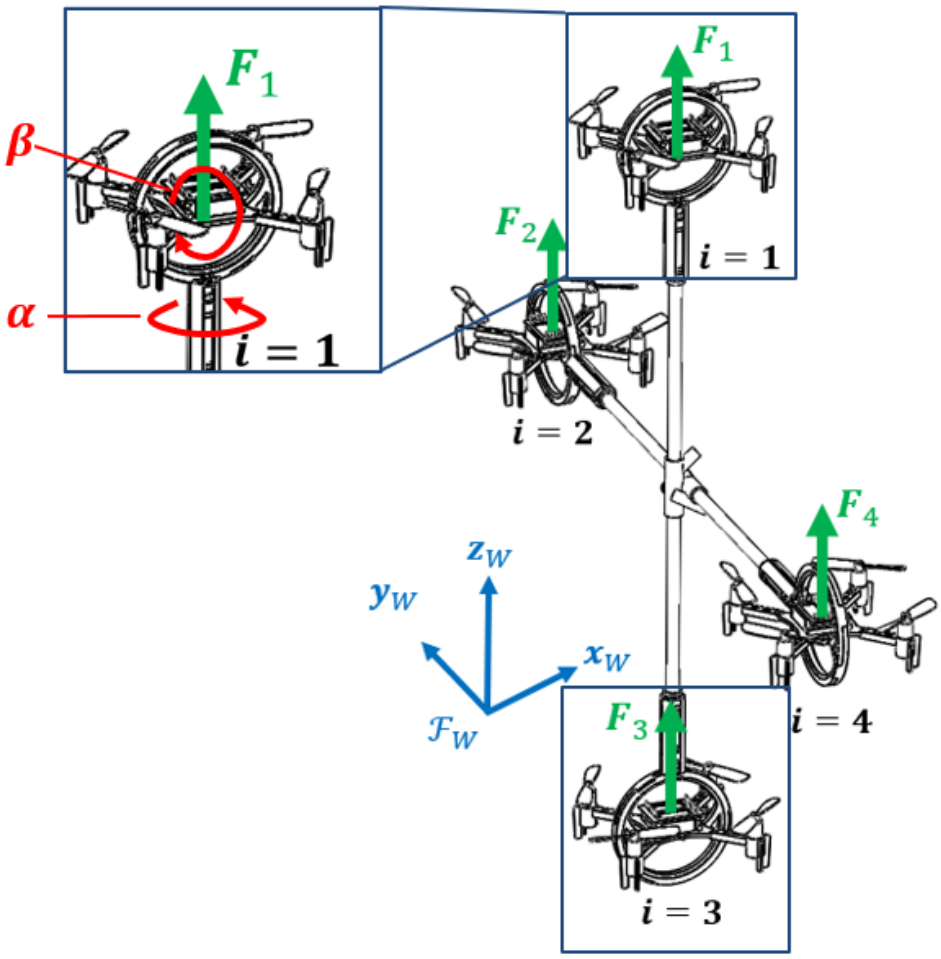


Figure 4.2: One of the singularity poses when pitch $= -90°$. Gimbal actuator 1 and 3 directly above and underneath the central frame are in singularity at this configuration.

thrust $\hat{\boldsymbol{F}}_i$ in Eq. (3.32), the rate of change of joint angles $\dot{\alpha}_i$ and $\dot{\beta}_i$ can be written as

$$\dot{\hat{\boldsymbol{F}}}_i = J(\alpha_i, \beta_i) \begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \end{bmatrix} = \begin{bmatrix} 0 & c\beta_i \\ -c\alpha_i c\beta_i & s\alpha_i s\beta_i \\ -s\alpha_i c\beta_i & -c\alpha_i s\beta_i \end{bmatrix} \begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \end{bmatrix} \tag{4.1}$$

where $J(\alpha_i, \beta_i)$ is the Jacobian matrix [Cra09]. When $\beta_i = \pm\frac{\pi}{2}$, $J(\alpha_i, \beta_i)$ will lose rank and $\dot{\alpha}_i$ will be close to infinity; in result, the control will attempt to track infinitely large joint speeds, which is not a viable input command for the system.

Note that the dramatic raise of $\dot{\alpha}_i$ with small perturbation does not only happens at 90°

roll or pitch angle strictly. Instead, when the over-actuated platform rotates near that angle range, the $\dot{\alpha}_i$ with small perturbation will continuously increase following a trend of the inverse function as observed from Eq. (4.1).

Another view of the singularity condition of the gimbal actuator is demonstrated by the manipulability ellipsoid, which describe the capability for a gimbal actuator to change its orientation in an arbitrary direction, and can be written as

$$\dot{\hat{\boldsymbol{F}}}_i^T \dot{\hat{\boldsymbol{F}}}_i = 1 \tag{4.2}$$

where the change of vector $\dot{\hat{\boldsymbol{F}}}_i$ is confined on a unit sphere. Combined with Eq. (4.1) it can be written as

$$\begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \end{bmatrix}^T J(\alpha_i, \beta_i)^T J(\alpha_i, \beta_i) \begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \end{bmatrix} = 1 \tag{4.3}$$

The manipulability ellipsoid is $J(\alpha_i, \beta_i)^T J(\alpha_i, \beta_i)$, which is used to describe the manipulability of joint α_i and β_i. The closer the principle axes of the ellipsoid, the stronger manipulability the system would have. From Eq. (4.1), the manipulability ellipsoid of the rotor is

$$J(\alpha_i, \beta_i)^T J(\alpha_i, \beta_i) = \begin{bmatrix} \cos^2 \beta_i & 0 \\ 0 & 1 \end{bmatrix} \tag{4.4}$$

When β_i is closed to $\pm\frac{\pi}{2}$, the manipulability ellipsoid is flat which is ill-conditioned. The matrix reduces its rank to 1, which means the rotor is losing one DOF in certain direction. This kinematic singularity results in the failure of the control.

When β_i is closed to $\pm\frac{\pi}{2}$, a small perturbation of $\dot{\hat{\boldsymbol{F}}}_i$ could results in very large angular velocity $\dot{\alpha}_i$. A sudden maneuver of the α joint with the inertia will create large unmodelled angular momentum to the quadcopter, and the low-level controller will fail to track the reference. This may explain the failure of the controller near the vertical flight condition.

4.2.2 Singularity Prevention by Assigning Values of Joint Angles

The singularity can be handled by manually assigning the values of the joint angles. The singularity occurs when a thrust direction is aligned with its α_i axis, so there are at most two gimbals in singularity given that the desired trajectory of the platform is slow. Fig. 4.2 illustrates an example: the top and bottom gimbal actuators are in singularity when the system is hovering in a vertical configuration. In such configuration, α_1 and α_3 are the only two joint angles that have dramatically large angular velocity. Given that the aerial platform is over-actuated with 12 DOF in total, the values of the two angles can be manually overwritten and the platform is still over-actuated.

In Eq. (3.32), we set $\alpha_i = 0$ for the two gimbal actuators in singularity. The equation becomes

$$\boldsymbol{F}_i = \begin{bmatrix} s\beta_i \\ 0 \\ c\beta_i \end{bmatrix} \tag{4.5}$$

Then $\boldsymbol{F}_i$ has two unknowns, and $F_{iy} = 0$. Because $\boldsymbol{F}_i$ is determined by allocation from ${}^B\boldsymbol{u}$ in Eq. (3.30) and W is a constant matrix, at most two columns in W (that correspond to F_{iy}) are removed, and W maintains full rank. When the pitch angle of the platform is 90° , the allocation matrix becomes

$$W' = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & d & 0 & 0 & 0 & 0 & d \\ 0 & -d & 0 & 0 & 0 & 0 & d & 0 & 0 & 0 \\ 0 & 0 & 0 & d & 0 & 0 & 0 & 0 & d & 0 \end{bmatrix} \tag{4.6}$$

And the allocation in Eq. (3.27) becomes

$${}^B\boldsymbol{u} = W'\boldsymbol{F}' \tag{4.7}$$

In a more general case, when we need to assign the α_i angle in singularity with values other

than 0, the allocation matrix can still be modified depending on the value of the assigned α_1 and/or α_3. Apply Eq. (3.32) to Eq. (3.27) with α_1 and/or α_3 extracted, the equation becomes

$$
{}^{B}\boldsymbol{u} = W\,\mathrm{diag}(1, -s\alpha_1, c\alpha_1, 1, 1, 1, 1, -s\alpha_3, c\alpha_3, 1, 1, 1)
\begin{bmatrix}
f_1 s\beta_1 \\ f_1 c\beta_1 \\ f_1 c\beta_1 \\ f_2 s\beta_2 \\ -f_2 s\alpha_2 c\beta_2 \\ f_2 c\alpha_2 c\beta_2 \\ f_3 s\beta_3 \\ f_3 c\beta_3 \\ f_3 c\beta_3 \\ f_4 s\beta_4 \\ -f_4 s\alpha_4 c\beta_4 \\ f_4 c\alpha_4 c\beta_4
\end{bmatrix} \tag{4.8}
$$

Where the values of α_1 and α_3 are constant and are determined by the task requirements. By combining the terms $f_1 c\beta_1$ and $f_3 c\beta_3$, the equation can be written as

$$
{}^{B}\boldsymbol{u} = W'\boldsymbol{F}' = W
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -s\alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & c\alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -s\alpha_3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & c\alpha_3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
f_1 s\beta_1 \\ 2f_1 c\beta_1 \\ f_2 s\beta_2 \\ -f_2 s\alpha_2 c\beta_2 \\ f_2 c\alpha_2 c\beta_2 \\ f_3 s\beta_3 \\ 2f_3 c\beta_3 \\ f_4 s\beta_4 \\ -f_4 s\alpha_4 c\beta_4 \\ f_4 c\alpha_4 c\beta_4
\end{bmatrix} \tag{4.9}
$$

where W' is still a constant allocation matrix.

As shown in these equations where the allocation matrix remains full-rank after the modifications, the singularity-handling method does not affect the over-actuated configuration of the platform. The maximum wrench of the platform is evaluated when hovering at 90° pitch angle. At hovering, the gimbal actuators provide thrusts to balance the gravity. The magnitude of the nominal thrust can be calculated based on the parameters provided in

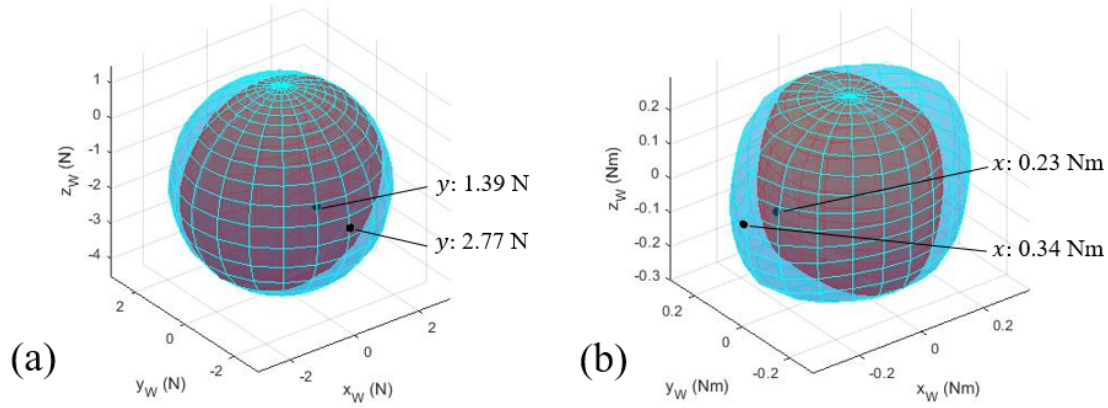


Figure 4.3: (a) Force envelope and (b) torque envelope of the platform. The red/blue regions are the envelope with/without singularity handling (setting α_1 and α_3 to zero). Selected data points indicated the edges of the regions.

Table 2.1. The extra thrust capabilities are used for trajectory tracking and disturbance rejection. The maximum force and torque envelopes (Fig. 4.3(a) and Fig. 4.3(b)) represent the range of the extra wrench ${}^{B}\boldsymbol{u}$. The maximum force is obtained by setting the torque to zero, and vice versa. The blue region in Fig. 4.3 is the maximum force and torque envelope without singularity handling, and the red region is the envelope with singularity handling. The singularity-handling method reduces the maximum force in the $\boldsymbol{y}$ direction and the torque in the $\boldsymbol{x}$ direction, but maintains the same wrench-generation capabilities in the other directions. Although the singularity-handling method reduces the robustness, the whole platform is still controllable in all six DOF.

Another advantage of this singularity handling method is that the thrust efficiency is maintained, because the least-norm result of the thrust vectors are obtained. As will be demonstrated in Chapter 5, the thrust vectors of all gimbal actuators will still be pointing upwards without canceling each other in lateral dirctions at vertical flights.

4.2.3 Singularity Handling by Avoiding the Singularity Zone

The other potential solution to the kinematic singularity of the gimbal actuator is realized by keeping the desired thrust vector outside the singularity zone. In other words, avoiding

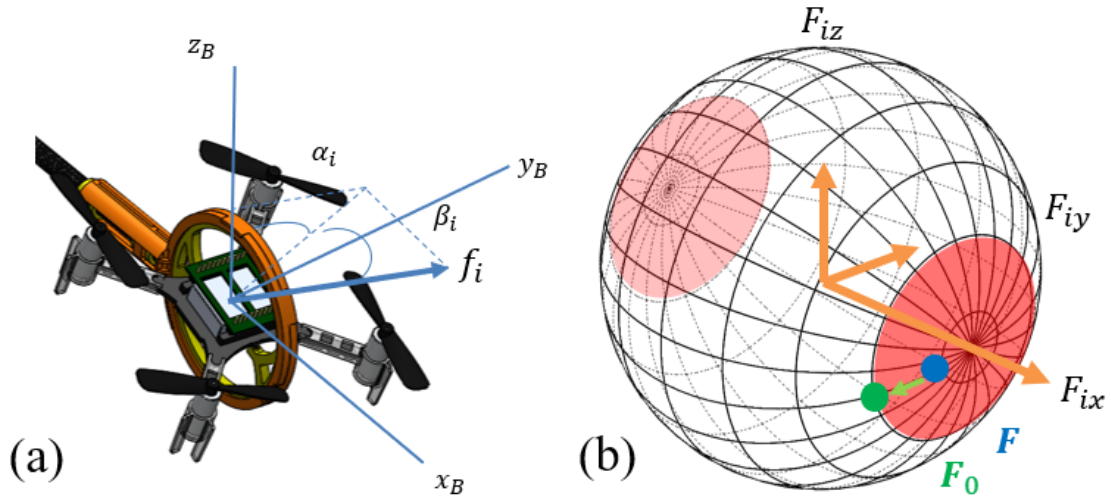


Figure 4.4: A gimbal actuator in singularity zone. (a) The direction of thrust. (b) The red zone defined on a sphere indicates the direction of thrust vector which will cause singularity.

β_i to get close to $\pm 90°$ as shown in Fig. 4.4, where the red zone on the unit sphere is defined as the singularity zone. Other than manually setting the values of singular joint angles, singularity avoidance can also be done by active avoidance for each rotor. To explore the possibility of establishing a more general solution, a singularity avoidance solution based on the nullspace of the allocation matrix and damped least-squares optimization for every gimbal actuator is proposed in this section.

The idea of this singularity avoidance technique is based on exploiting the over-actuation of the whole platform. From Eq. (3.27), It's possible to get a variety of other solutions of $\boldsymbol{F}$ without affecting the original $\boldsymbol{u}$, because the allocation matrix has more columns than its rows. For every other solution, it would be the least norm solution plus an additional vector which is in the null space of W. From the perspective of each gimbal actuator, the thrust vector has changed by an additional term. If one or more gimbal actuators are originally in singularity configuration, the additional term drives the thrust vector away. Then the singularity of the platform is avoided with the additional terms in $\boldsymbol{F}$.

A naive solution is to add a constant vector to $\boldsymbol{F}$ which is calculated offline to be in null(W). However, to ensure a smooth transition, the additional term should be larger when some gimbal actuator is more closed to singularity. What's more, as $W \in \mathbb{R}^{6\times 12}$, the

nullspace of W has a rank of 6. It's difficult to determine a unique solution that could be applied to all experimental scenarios. Therefore the singularity avoidance technique is proposed below.

The aforementioned additional term, denoted as $\boldsymbol{F}_{null} \in null(W)$, can be exploited. The new solution to $\boldsymbol{F}$ becomes $\boldsymbol{F_n} = \boldsymbol{F}_{least-norm} + \boldsymbol{F}_{null}$, where $\boldsymbol{F}_{least-norm}$ is the original least-norm solution. In the literature [Lie77] it is given that

$$\boldsymbol{F_n} = W^{\dagger}\boldsymbol{u} + (I - W^{\dagger}W)k_a \nabla H(\boldsymbol{F}) \tag{4.10}$$

where $(I - W^{\dagger}W)$ is to project an arbitrary vector on the nullspace of W, k_a is a constant gain, and $\nabla H(\boldsymbol{F})$ is the gradient of an objective function of $\boldsymbol{F}$. In this case, the gradient is designed to be

$$\nabla H(\boldsymbol{F}) = \boldsymbol{F_0} - W^{\dagger}\boldsymbol{u} \tag{4.11}$$

where $\boldsymbol{F_0}$ is the expected thrust vector of each gimbal actuator. $\boldsymbol{F_0}$ is defined as a correction to the a priori thrust vector $\boldsymbol{F}_{least-norm}$, or simply $\boldsymbol{F}$. If the least-norm solution falls within a user-defined range of vicinity of the singularity configuration, it will be pulled out of the singularity region. The singularity region is defined as $\sqrt{\frac{F_{iy}^2 + F_{iz}^2}{F_{ix}^2}} < \epsilon$, as shown in Figure Fig. 4.4(b). Thus, $\boldsymbol{F_0}$ is the closest vector that is outside the singularity region.

The idea behind Eqs. (4.10) and (4.11) is that detection of singularity of a single gimbal actuator will drive the change on all rotors, so that the joint effect of the change of all thrust vectors falls in the null space of W. We look into a single gimbal actuator at first, and find the objective thrust vector $\boldsymbol{F_0}$ for singularity avoidance on this gimbal actuator. Then the gradient of this objective thrust vector, in other words how much the thrust vector needs to change, drives the change of thrust vector of other rotors, such that the joint effect of the changes falls in the nullspace of W. The gain k_a is designed to guarantee a smooth transition and avoid a sudden impact to the system.

The next step is to find a proper $\boldsymbol{F_0}$ for the gimbal actuator with singular $\boldsymbol{F}_{least-norm}$. An

optimization problem is formulated to calculate $\boldsymbol{F_0}$ with minimized $\begin{bmatrix}\dot{\alpha}_i \\ \dot{\beta}_i\end{bmatrix}$.

From Eq. (4.1), a least-squares solution is obvious to get a feasible $\begin{bmatrix}\dot{\alpha}_i \\ \dot{\beta}_i\end{bmatrix}$. In order to minimize joint velocity itself, a damped least-squares problem as proposed in [Wam86] is formulated. The objective function is

$$\text{minimize } k_b^2 \left\| \begin{bmatrix}\dot{\alpha}_i \\ \dot{\beta}_i\end{bmatrix} \right\|^2 + \left\| J \begin{bmatrix}\dot{\alpha}_i \\ \dot{\beta}_i\end{bmatrix} - \dot{\hat{\boldsymbol{F}}}_{least-norm} \right\|^2 \tag{4.12}$$

where k_b is a tunable gain. The solution to this problem is

$$\begin{bmatrix}\dot{\alpha}_i \\ \dot{\beta}_i\end{bmatrix} = (J^T J + k_b^2 I)^{-1} J^T \dot{\hat{\boldsymbol{q}}}_{least-norm} \tag{4.13}$$

It is obvious that $(J^T J + k_b^2 I)$ is a diagonal matrix in $\mathbb{R}^{2\times 2}$, and its inverse doesn't consume much computational power in implementation.

The rate of change of $\boldsymbol{F_0}$ is

$$\dot{\hat{\boldsymbol{F}}}_{\boldsymbol{0}} = J \begin{bmatrix}\dot{\alpha}_i \\ \dot{\beta}_i\end{bmatrix} = J(J^T J + k_b^2 I)^{-1} J^T \dot{\hat{\boldsymbol{F}}}_{least-norm} \tag{4.14}$$

And finally $\boldsymbol{F_0}$ at timestamp k is

$$\boldsymbol{F_0}(k) = \boldsymbol{F_0}(k-1) + ||\boldsymbol{F}_{least-norm}|| \cdot \dot{\hat{\boldsymbol{F}}}_{\boldsymbol{0}} \cdot T \tag{4.15}$$

where T is the sampling time.

This solution calculates the value of $\boldsymbol{F_0}$ which is closed to $\boldsymbol{F}_{least-norm}$ enough and ensures minimal rate of change of α_i and β_i with a tunable gain k_b.

This singularity avoidance technique could ideally enables the over-actuated platform to

hover at vertical flight. However, this solution requires each gimbal actuator to provide additional thrust that counteracts with each other and reduce the overall thrust efficiency. This is a conflict with the design goal. In real-world implementation, the internal thrust cancellation caused by this singularity avoidance method largely increases the power requirement to the system. Therefore, this method is not used and the method that maintains high thrust efficiency introduced in Section 4.2.2 will be demonstrated in experiment. The other work that applies an optimization on the over-actuated aerial platform that exploits the nullspace of the allocation while maintains high thrust efficiency is demonstrated in [SYG+21].

CHAPTER 5

Simulation and Experiment Validation

5.1 Simulation Setup

A high fidelity computer simulation model is needed prior to experimental validation, in order to verify the dynamics model of the system, the full-actuation nature of the UAV platform, the hierarchical controller design and discover unexpected problems of the dynamics and control. In this thesis, the simulation model is built in Simulink in Matlab. The simulations included several physical and hardware properties that were not considered in the model used for controller design in Section 3.2, which are listed below.

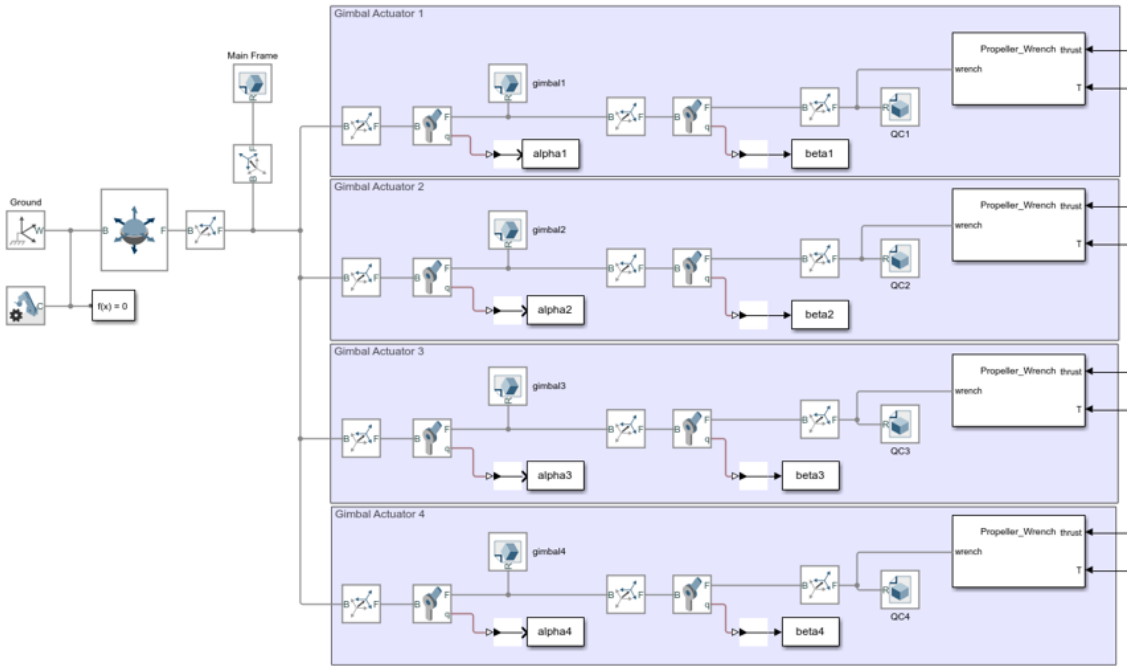


Figure 5.1: Complete multibody dynamics simulation using Simscape.

Simscape Multibody module in Simulink was used to model the complete dynamics of

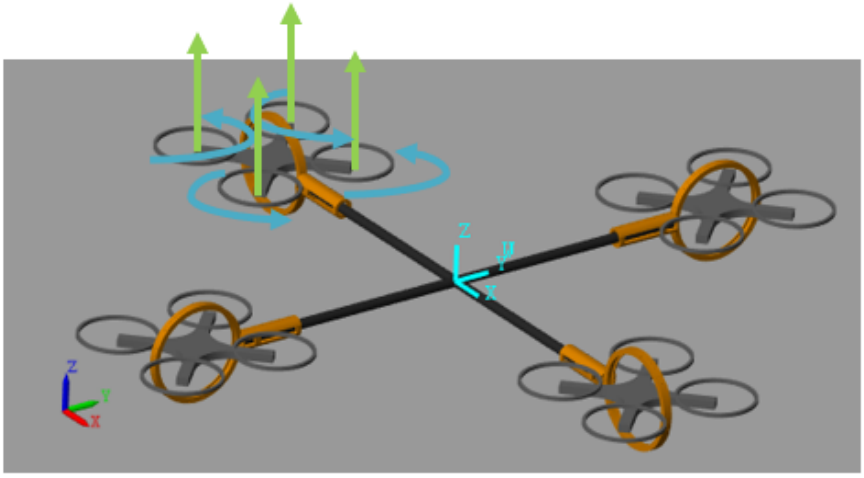


Figure 5.2: Visualization of the over-actuated aerial platform in Simscape.

the platform, as shown in Fig. 5.1. It contains the main frame and four gimbal actuators, and each gimbal actuator has two revolute joints and the propellers on the quadcopter is subject to external forces and torques calculated by the motor command calculation module. Compared with the whole-body dynamics Section 3.1, the Simscape model also included the mass and inertia of the gimbal frames and the quadcopters, connected by passive revolute joint blocks, and therefore the gyroscopic effects caused by the gimbal rotation are included. The saturation of motor speed for each of the 16 propeller was specified and the relationship between the motor speed and the magnitude of thrust was from the product data sheet from the manufacturer. The Simscape module also generate a virtual realization of the over-actuated platform for observation (Fig. 5.2).

The complete hierarchical controller introduced in Section 3.2 is built in Simulink, as shown in Fig. 5.3. Controller frequencies are set to be the same as in the experiment in Section 5.2. The high-level controller, including the position and attitude controllers using LQI, the control allocation and the inverse kinematics, is set to be 100 Hz and is highlighted in green. The low-level controller which is run onboard in real-world implementation, is set to be 500 Hz and is highlighted in red. The Simscape multibody dynamics are run in continuous time, and the continuous-time blocks are shown in black. The Simscape dynamics block is in yellow (multi-rate) because it receives the inputs of the desired propeller thrusts

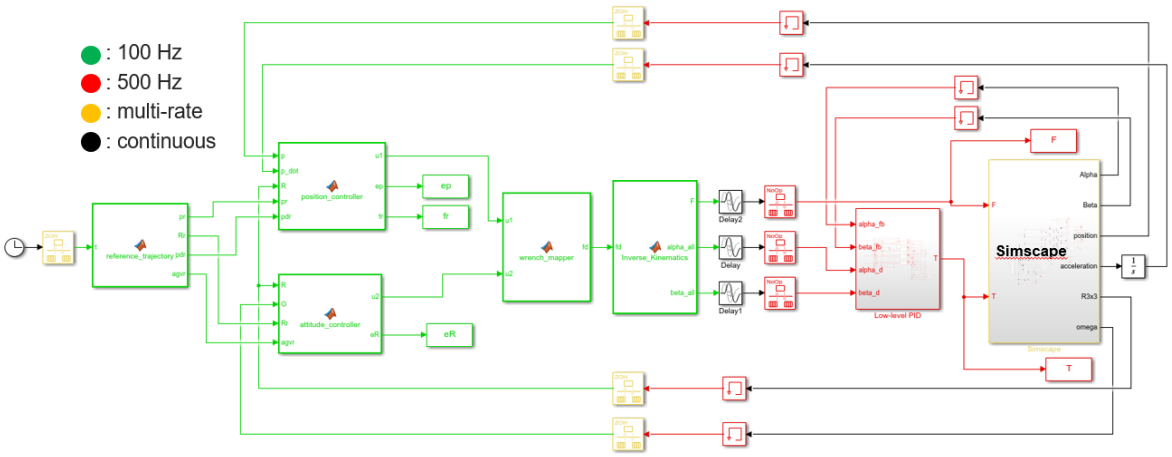


Figure 5.3: Hierarchical controller with multiple sampling rates and delays in Simulink.

and torques from the low-level controller in 500 Hz, and converts them to continuous-time signal. A communication delay of 20 ms is inserted between the high-level control and low-level control, as in the real-world implementation the ground station calculates the high-level inputs and sends them to each onboard processor via wireless communication. Also, sensor noises that are representative to real-world situations are also included.

On the other hand, some real-world physical properties that are difficult to model were simplified or neglected in the simulation. In particular, the aerodynamics of each propeller were simplified to exert direct thrust force and air resistance torque on the body. The air flows generated by one quadcopter that might affect the other quadcopters or the main body are ignored. Finally, friction on the passive gimbal joints was neglected.

5.2 Experiment Setup

Although the proposed over-actuated UAV platform is designed for applications involves both outdoor and indoor environments, the experiments demonstrated in this thesis are accomplished in an indoor environment because of the additional challenges of acquiring position measurements in an outdoor environment.

An Optitrack Prime 13 motion-capture system was used to measure the position and

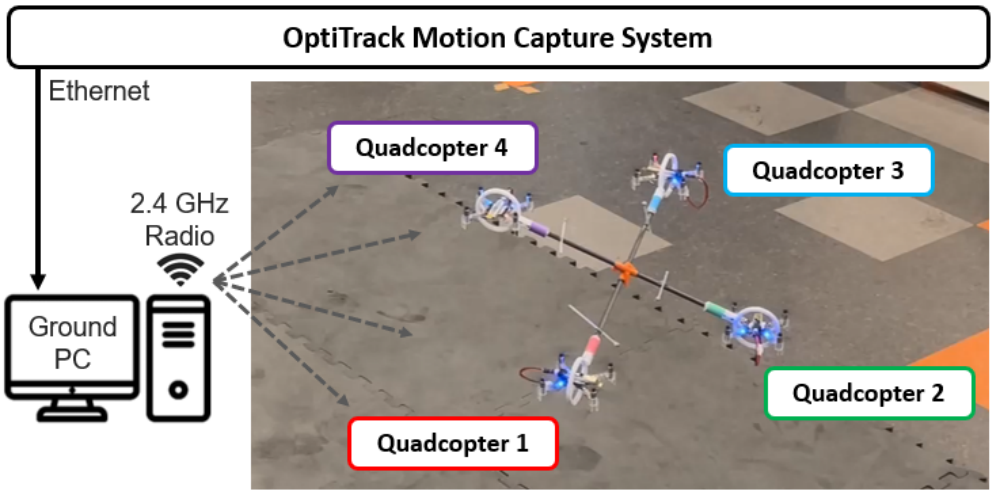


Figure 5.4: Experiment setup of the over-actuated UAV platform in an indoor environment.

attitude of the platform in an indoor environment. The high-level controller was run on a ground-based computer, which communicated with the motion capture system via Ethernet. The main controller calculated the desired commands for each gimbal actuator. These desired commands, along with the platform attitude from the motion-capture system, were sent to each gimbal actuator, which communicated with the ground-based computer via a 2.4 GHz Crazyradio PA [GSW+17]. The low-level controller was run onboard each quadcopter. The low-level controller rate was set higher than that of the high-level controller. The hardware setup is shown in Fig. 5.4.

The software architecture of the platform is shown in Fig. 5.5. The ground PC runs Python programs in Linux environment. To balance the calculation load among the available CPUs, the program is designed to have three processes using the multiprocess module in Python. Data exchange channels are set among the three processes.

The main process is responsible for the data processing. It retrieves the 6 DOF measurment data of the over-actuated platform from the Optitrack system, and initiates the process of the measurment data, such as converting the attitude data between quaternion and rotation matrix formats for the high-level controller. The trajectory can be pre-defined in the hard drive, or can be controlled in real-time by user inputs using the keyboard. The main process also gathers all the data in real time and saves log to the hard drive.

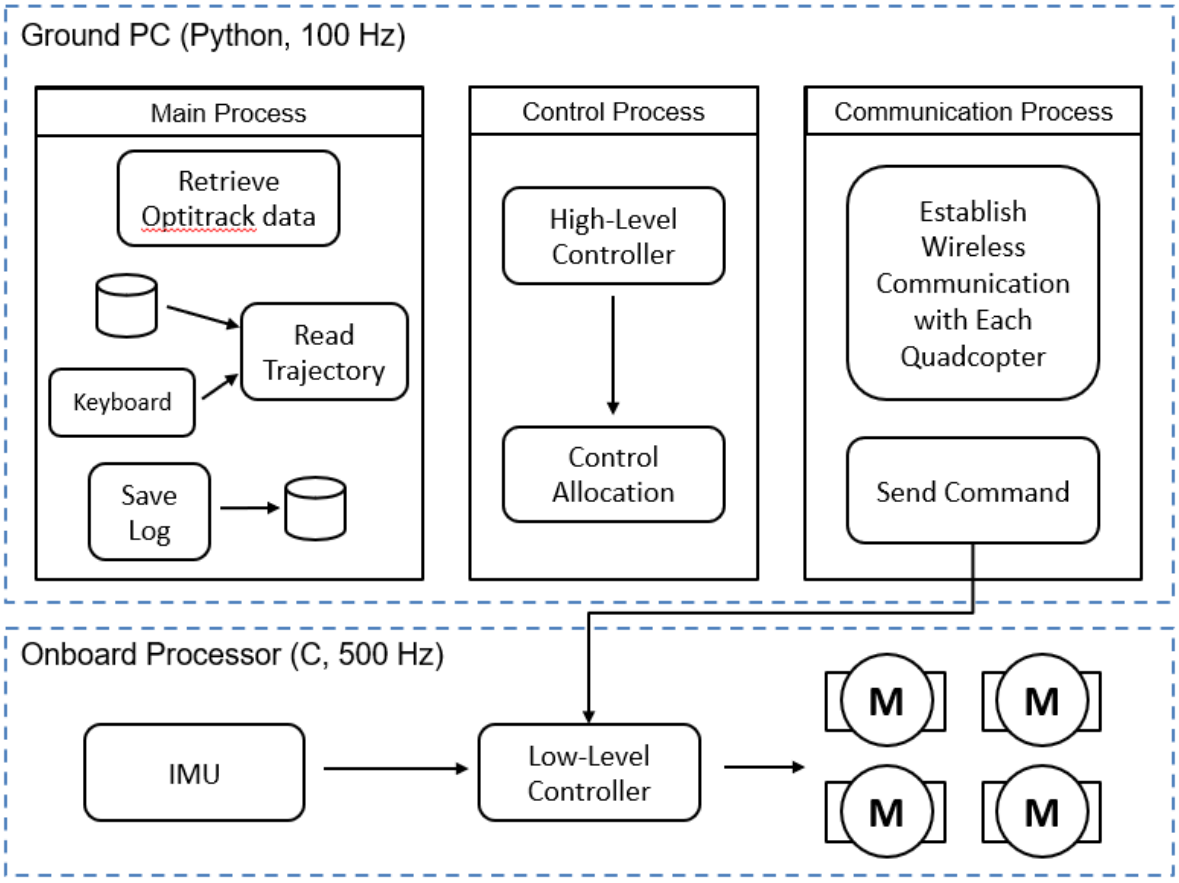


Figure 5.5: Software architecture of the over-actuated UAV platform.

The control process on the ground PC runs the high-level controller, including the LQI position and attitude controllers, the control allocation and the inverse kinematics that calculates the desired joint angles and magnitude of thrust for each gimbal actuator. The high-level controller reads the 6 DOF feedback and the reference trajectory from the main process in real time.

The commands of the desired joint angles and magnitude of thrust are passed to the communication process, which establishes wireless communication with each quadcopter using the library provided by Crazyflie [GSW+17]. The commands are then sent to each quadcopter. The communication process is initiated first, and the first group of data is sent when the main process and the control process are ready. Initially the data being sent to each quadcopter are all 0, including the thrust command, so the platform is in standby mode. When the first group of input commands from the control process is available, the communi-

cation process starts to sent the real-time commands in 100 Hz sampling rate. The attitude of the main frame is also sent to each quadcopter in the same rate.

The onboard processor of the crazyflie runs C programs in 500 Hz sampling rate. The onboard IMU with built-in sensor fusion algorithms calculates the rotation matrix of the quadcopter body. The low-level controller uses the attitude information of both the main frame of the platform and the quadcopter body to estimate the angles of the passive joints, because direct measurement of the passive joint angles is not available on the hardware, as introduced in Section 3.2.3. The estimated joint angles are used as feedback for the low-level controller, and the desired body torque to track the joint angles are calculated. With the desired thrust for the gimbal actuator and the desired body torque, the desired angular speed of each propeller motor is calculated, and the PWM signals are sent to perform the control.

To ensure the stability of the hierarchical controller, the inner-loop controller must be significantly faster than the outer-loop controller [ABN80]. The low-level controller was first designed by closed-loop pole placement in consideration of the maximum actuator capacity as introduced in Section 4.1. The gains are $k_{P\alpha} = k_{P\beta} = 7000, k_{D\alpha} = k_{D\beta} = 300, k_{I\alpha} = k_{I\beta} = 100$. And the step response of the gimbal joint angle is shown in Fig. 5.6.

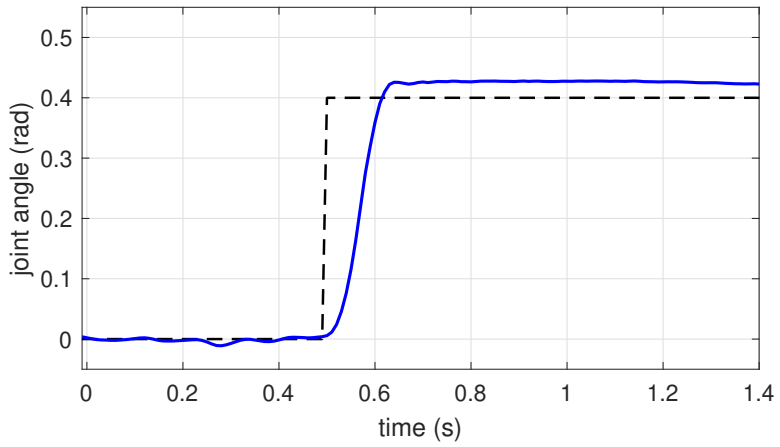


Figure 5.6: Step response of a gimbal joint angle with the low-level controller.

Then the high-level LQI controller was designed accordingly to ensure slower high-level

Table 5.1: Step Response of Hierarchical Controllers

	Rise time (sec)	Overshoot (%)	Settling time (sec)
Low-level	0.005	5	0.1
High-level (Position)	0.4	20	4.5
High-level (Attitude)	0.2	20	1.3

response. The weighting matrices are:

$$Q = \text{diag}(1,1,1,\ 1,1,1,\ 0.1,0.1,0.1,\ 0.01,0.01,0.01,\ 1,1,1,\ 0.1,0.1,0.1)$$

$$R = \text{diag}(5,5,5,\ 700,700,700)$$

The weights of position and attitude control are determined by the maximum linear and angular accelerations of the platform. The step response performance calculated from closed-loop transfer functions is shown in Table 5.1.

5.3 Results and Discussions

5.3.1 6-DOF Step Response

Prior to implementing the actual 6-DOF trajectory on the over-actuated platform, a step-response test is conducted to test the closed-loop behavior of the system. The step signal is given to each of the 6 DOF separately, while the commands of the other directions are 0. Then the response in all 6 DOF are recorded. The magnitude of the step should not be too large, or actuator will hit saturation; it should not be too small, or the step response will be difficult to identify from the disturbances. Therefore, considering the signal-to-noise ratio of the real-world platform, the magnitude of the step is chosen to be 0.1 m in position and 0.3 rad in attitude. For each direction, the step signal is given twice in the opposite directions to observe any difference of the closed-loop behavior in the opposite directions.

The experiment results are shown in Fig. 5.7 and Fig. 5.8. The 6-DOF data is logged in each test. It can be observed that the step response of the system in all 6 DOF agrees with

the expected outcome in Table 5.1. The step in each direction doesn't trigger the reaction in any other direction. Notice that in Fig. 5.7, because the magnitude of the step in x, y and z position is smaller, and position and attitude data are plotted in the same y-axis scale, the perturbations in roll, pitch and yaw directions are visually larger than those in Fig. 5.8, but the perturbations are essentially in the same level. Therefore, a preliminary conclusion can be made that the presented over-actuated UAV platform can enact 6 independent forces and torques in each direction in space, without interfering with the system performance in other directions. This is an important observation of the system, which encourages the idea of treating the system as containing 6 independent control channels and can be utilized to simplify the process of controller design as described in details in Chapter 6 and Chapter 7.

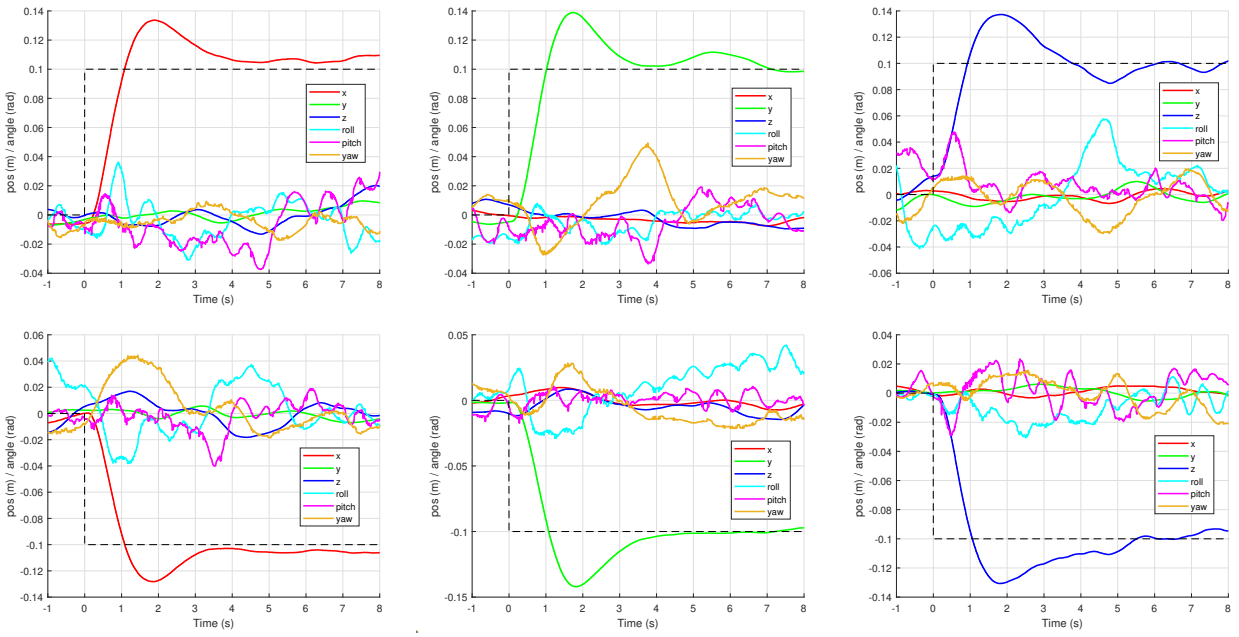


Figure 5.7: Experimental data of step response, with steps exerted in x, y and z directions.

The step test is also applied on the simulation model in Simulink. The magnitude of steps in position and attitude remain the same, and the step response data is compared with the experiment data in each of the 6 DOF. In addition, as discussed in Section 3.2, when the low-level controller is significantly faster than the high-level, the closed-loop system behavior is dominated by the high-level controller. In this work, the high-level controller is a model-based LQI control, and the dynamics can be treated as a rigid body due to the passive joint

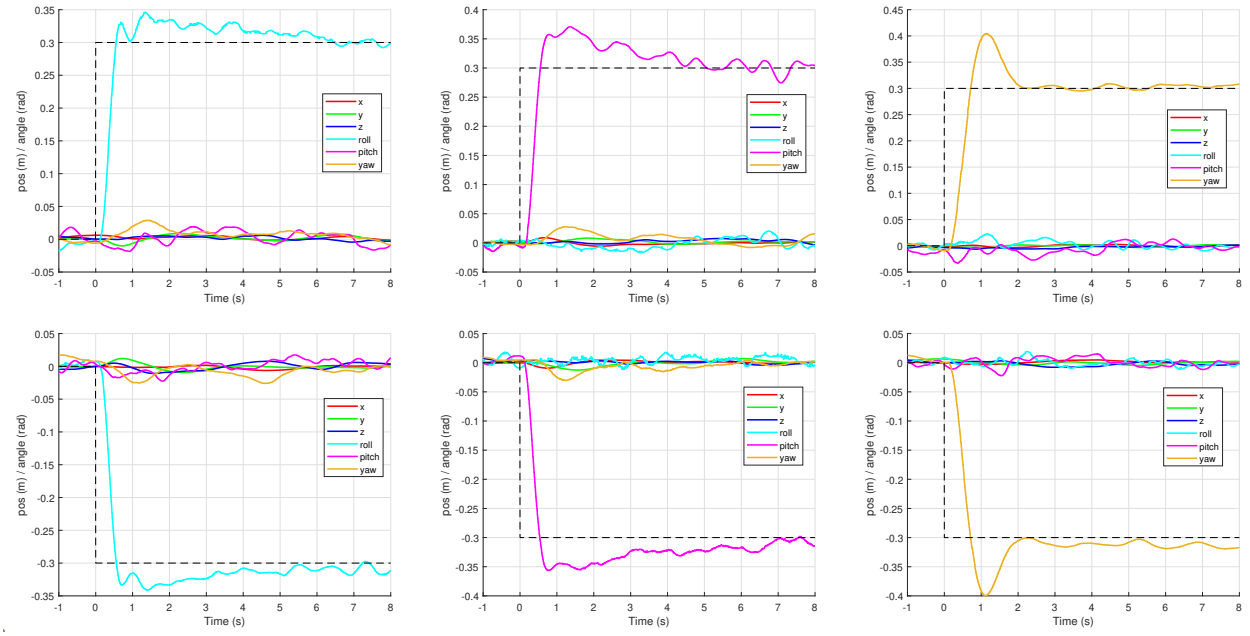
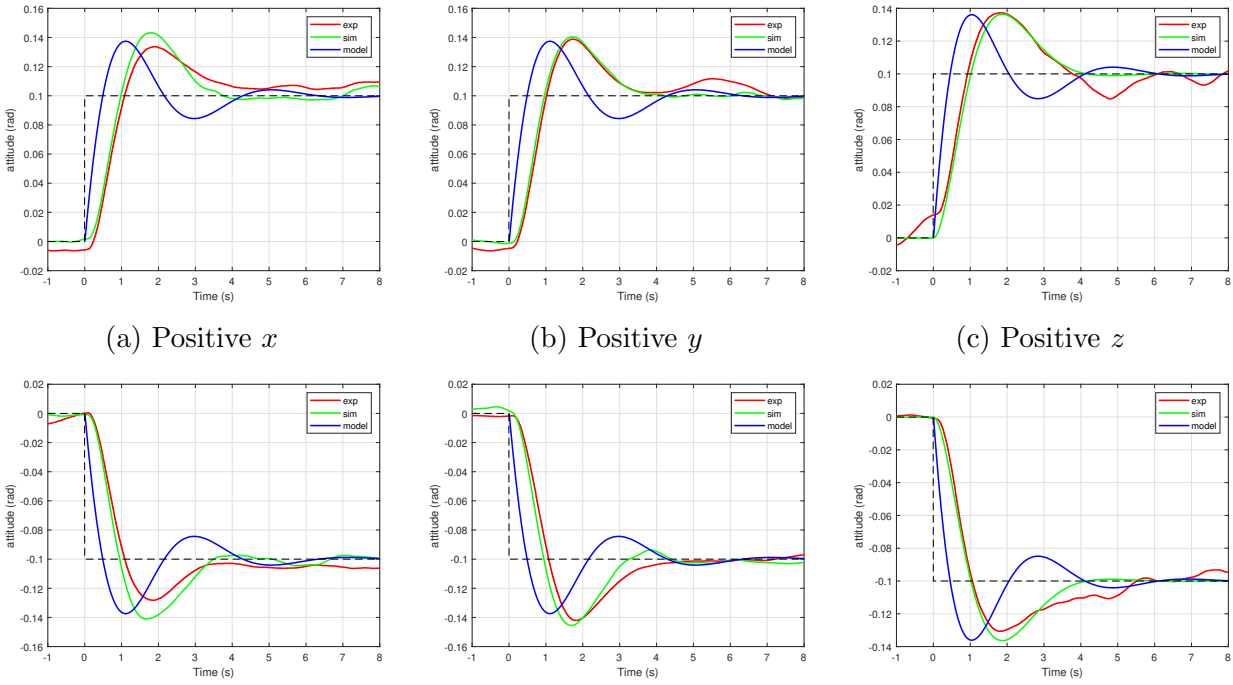


Figure 5.8: Experimental data of step response, with steps exerted in roll, pitch and yaw directions.

and the multibody dynamics is naturally simplified. Then the high-level dynamics model is available and an emulated step response on the high-level dynamics model can be obtained. The step response on the high-level model is also included in this comparison.

The results are shown in Fig. 5.9 and Fig. 5.10. It is obvious that the simulation result is close to the experiment result in all 6 DOF, despite some perturbations in the experiment data due to external disturbances that are common on an aerial vehicle. That means the simulation model has included most of the real-world physical properties of the system, and the modeling is accurate. On the other hand, the emulated step response data from the high-level dynamics model is different from the simulation and experiment results. This shows that the dynamics model we derived cannot reflect the complete physical properties of the system, and the high-level dynamics is not enough to describe the system behavior. Yet, the hierarchical controller, especially the high-level LQI controller that is built upon the high-level dynamics model, successfully stabilized the whole platform. It shows that the derived model can be a referral to describe a partial of the system properties, and the presented over-actuated aerial platform as well as the hierarchical control scheme is robust

to model uncertainty.



(a) Positive x (b) Positive y (c) Positive z

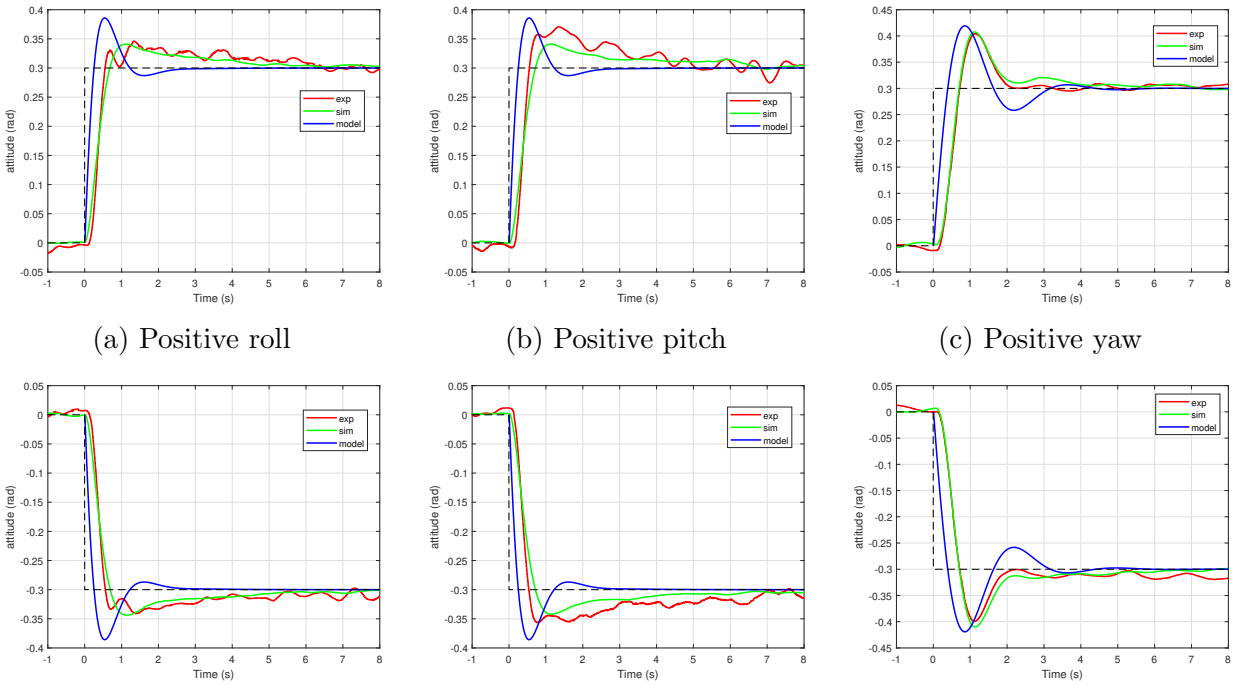
(d) Negative x (e) Negative y (f) Negative z

Figure 5.9: Comparison of the step response in x, y and z directions.

5.3.2 Tracking of 6-DOF Trajectory

The first application scenario assigned to the presented over-actuated UAV platform is to track a trajectory in which the 6-DOF position and attitude change with time independently. The position and attitude trajectories are designed arbitrarily, and different trajectories can be executed easily. In this section, we demonstrate one trajectory. The purpose of designing the independent 6-DOF trajectory is to verify the capability of the over-actuated UAV to enact 6 independent control inputs in every direction. The tracking results from the simulation and the experiment are shown in Fig. 5.11.

Since the high-level controller is slower, the 6-DOF trajectory is designed to be changing slowly to avoid triggering satuation. In both simulation and experiment, the over-actuated UAV platform tracks the reference trajectory well, and the root-mean-square (RMS) errors are shown in Table 5.2. The RMS errors in simulation and experiment are close to each



(a) Positive roll (b) Positive pitch (c) Positive yaw

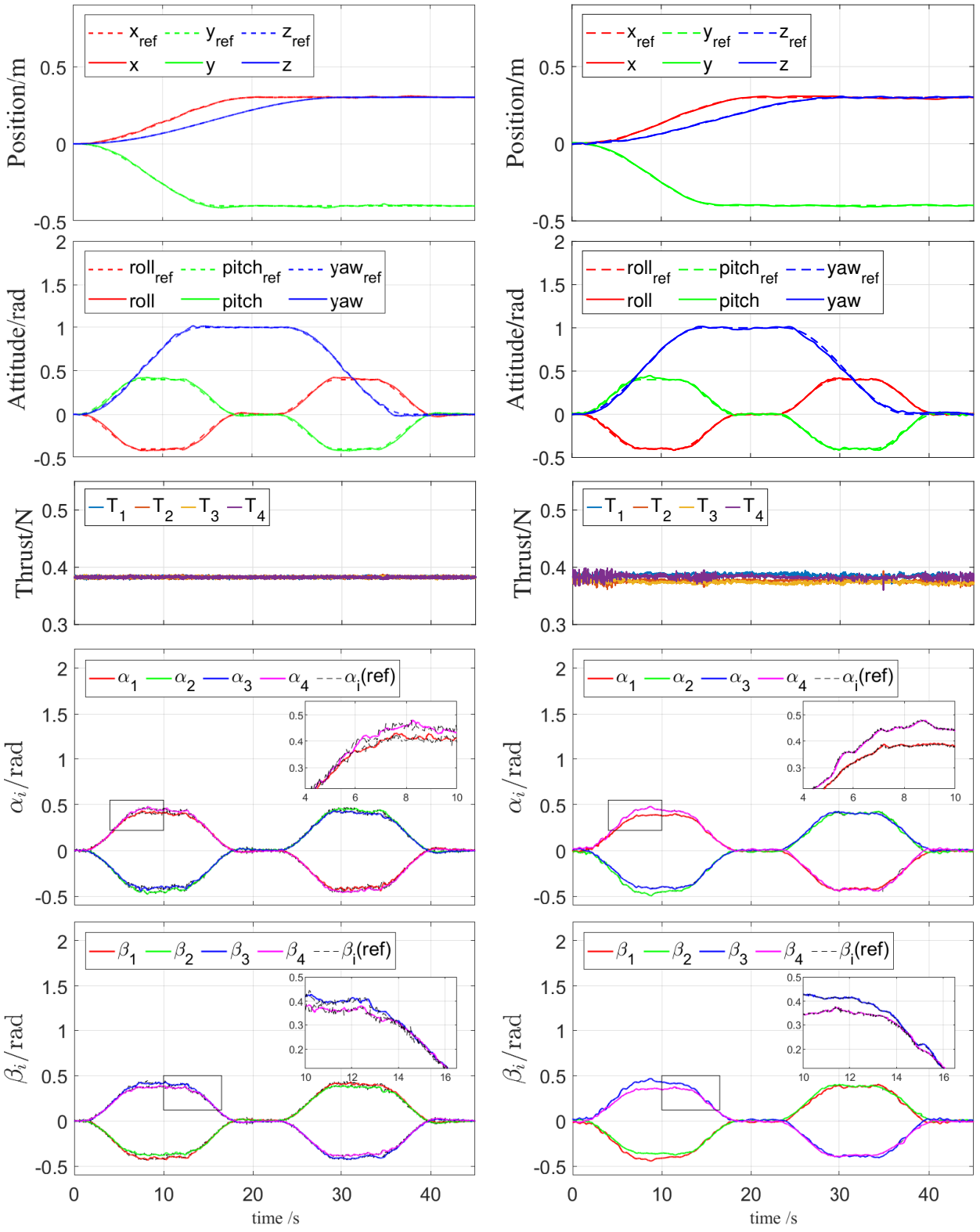
(d) Negative roll (e) Negative pitch (f) Negative yaw

Figure 5.10: Comparison of the step response in roll, pitch and yaw directions.

other except in the z direction. One explanation is that in the simulation the aerodynamic effects of the thrusts are modeled in a much simpler way, i.e. directly exerting support force upwards. In the real-world situation, thrusts generated by the propellers are obviously more complicated, and may include additional disturbances that reduce the tracking accuracy in z direction.

The magnitude of thrust of each gimbal actuator in both simulation and experiment maintain constant and identical of 0.38 N, despite some spikes due to sensor noises in the experiment. Note that from Table 2.1, the weight of the whole platform is $0.156 \times 9.81 = 1.53$ N, and the sum of the thrusts are $0.38 \times 4 = 1.52$ N. The thrust efficiency of the platform executing this 6-DOF trajectory is as high as 99%. In other words, there is almost no internal thrust cancellation when the platform is in different attitude. The thrust efficiency is near that of a regular quadcopter.

The zoom-in view of the α_i and β_i angles in Fig. 5.11 shows the desired joint angle commands are not as smooth as the high-level trajectory, because the gimbal actuators are



(a) Simulation (b) Experiment

Figure 5.11: 6-DOF trajectory tracking performance in (a) Simulation, and (b) Experiment.

controlled to compensate for sensor noises and disturbances. From the tracking results of the joint angle commands, it can be referred that the low-level controller tracks the desired joint angles much faster than the high-level controller.

Table 5.2: RMS Error of Trajectory Tracking

Case	x(mm)	y(mm)	z(mm)	roll(rad)	pitch(rad)	yaw(rad)
6DOF(Sim)	3.1	4.8	0.4	0.015	0.015	0.013
6DOF(Exp)	4.9	3.8	3.2	0.009	0.013	0.017
360°(Sim)	13.6	4.2	3.1	0.003	0.026	0.010
360°(Exp)	5.6	10.7	54.0	0.033	0.055	0.011

5.3.3 Disturbance Rejection Performance

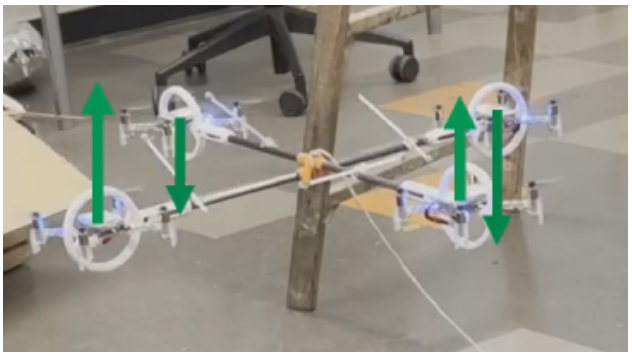


Figure 5.12: Artificial disturbance exerted by overwriting the thrust commands.

In this case, force and torque disturbances are exerted on the platform while hovering. The disturbance is realized at the software level, where the desired thrust command of each gimbal actuator calculated by the high-level controller is added with a designated value for a period of time, as indicated in Fig. 5.12. The equivalent disturbance consists of a constant force of $[0\ 0\ 0.03]^{\mathsf{T}}$ N and torque of $[0.05\ -0.04\ 0]^{\mathsf{T}}$ Nm from 1 to 1.3 second, and the results of the platform reacting to this disturbance are shown in Fig. 5.13. Because the disturbance is only exerted from 1 to 1.3 second and no disturbance after that, the system is equivalent to having a step response starting at 1.3 second if we ignore the behavior prior to 1.3 second. The settling time of the platform is less than 1.5 second due to the closed-loop dynamics designed by the hierarchical controller. Since the low-level controller is designed to

be significantly faster than the high-level controller, the closed-loop dynamics is dominated by the high-level LQI controller. This can be verified by observing the low-level behavior in this disturbance rejection case. By observing the peaks of the reference joint angle commands and the actual tracking of the joint angles, the settling time of low-level controller is around 0.1 second, which is much faster than the high-level controller and thus it guarantees the stability of the whole system.

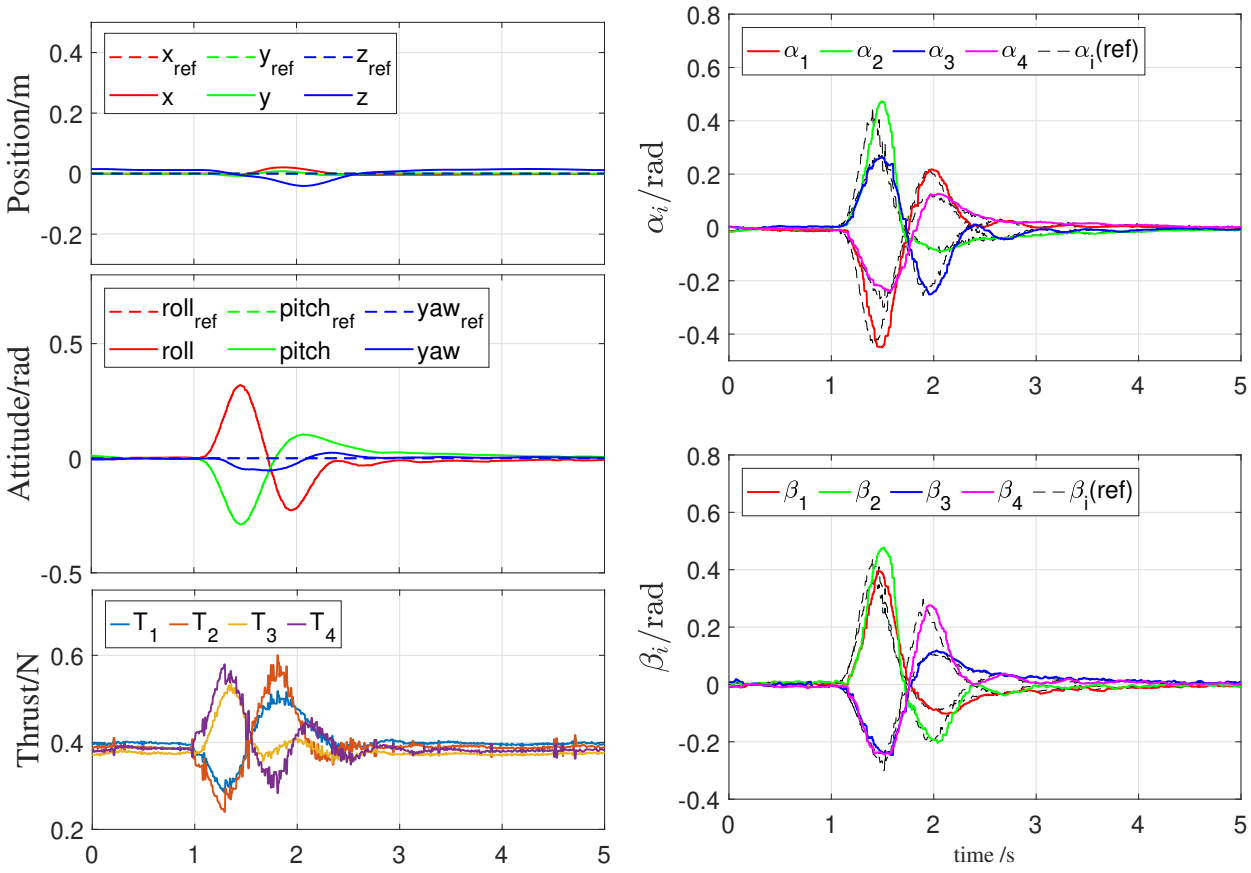


Figure 5.13: Disturbance rejection performance.

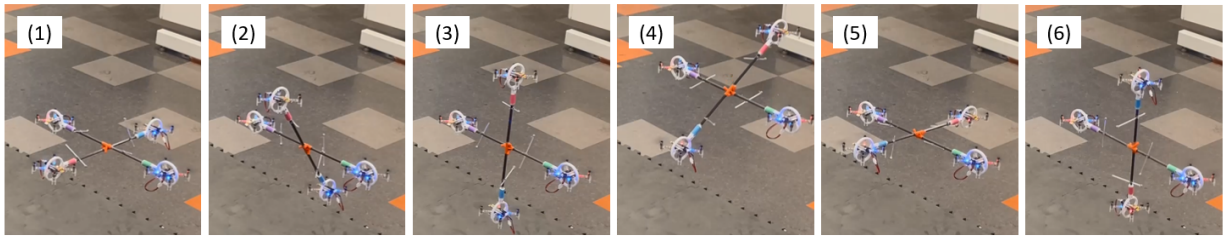
5.3.4 Complete 360° Rotation

This application scenario is to demonstrate the capability of the presented over-actuated UAV platform to reach any arbitrary attitude. The platform was commanded to rotate around its pitch axis for 2π rad, while maintaining the initial states of the other 5 DOF.

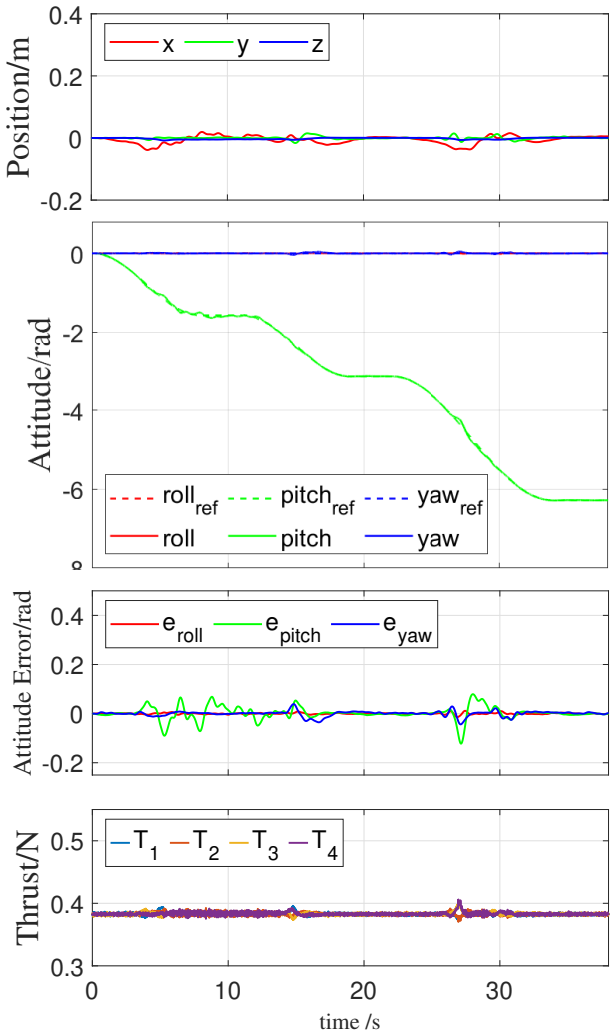
During the rotation, the platform intentionally remained at $\pi/2$ rad and π rad for a few seconds each to demonstrate its capability to hover at these poses. As discussed in Section 4.2, when the pitch angle approaches $\pi/2$ rad or $3\pi/2$ rad, the top and bottom gimbal actuators $i = 1, 3$ are in singularity configuration. In this experiment, the singularity handling solution by manually overwrite the values of joint angles α_i in singularity is implemented.

The simulation and experimental results for the 2π rad rotation case are shown in Fig. 5.14. The results indicate that the platform remains stable throughout the rotation process and tracks the desired angles accurately, while the performance was degraded around the two singularity regions. On possible explanation is the quadcopter downwash effects [MMLK10]: when the bottom gimbal quadcopter is directly below the top gimbal quadcopter, the airflow generated by the top quadcopter reduces the effective thrust of the other one. The reduction of effective thrust explains the drop of z position and the rise of desired thrusts when entering the singularity regions. The z position gradually goes back to normal by the integral action in the LQI controller. The stored information in the integrator then causes the rise of the platform when it exits the singularity regions, and is regulated back to normal again. Notice that this situation does not appear in the simulation, because the aerodynamic effects between different quadcopters were not included.

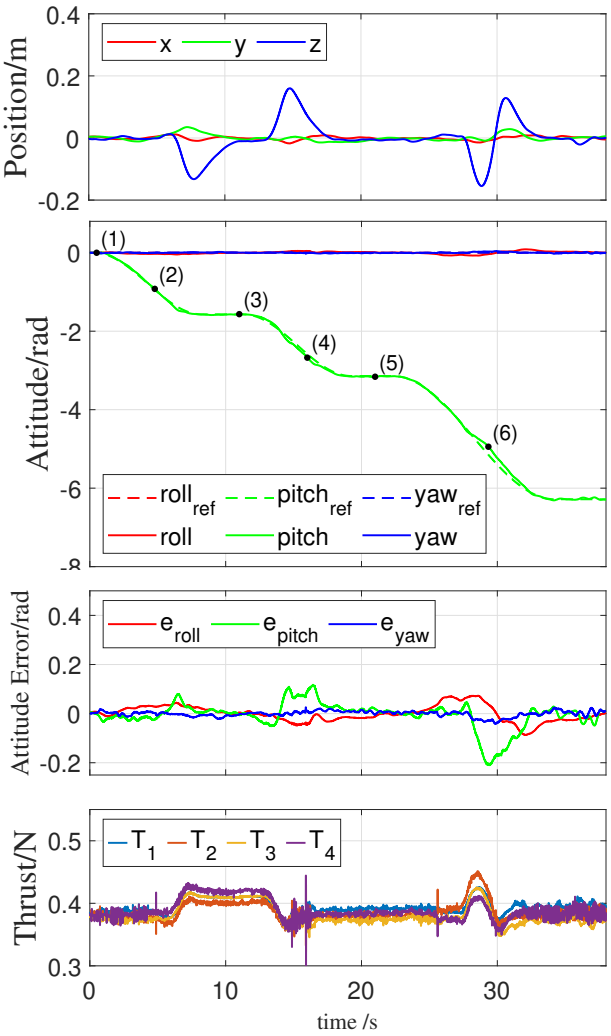
Despite the possible downwash effects that affects the experiment results in z direction, the tracking of the other 5 DOF is accurate, which is also shown by the RMS tracking error in Table 5.2. The simulation and experiment results demonstrate that the over-actuated platform with gimbal actuators can achieve continuous and unrestricted rotation in the air and can reach any arbitrary attitude angle. The platform therefore shows great potential in real-world applications where it needs to perform physical interactions with the surroundings in different attitude. In addition, the thrusts of each gimbal actuator maintain at the same level throughout the rotation process despite the downwash effects. The average thrust efficiency is 0.99% in simulation and 0.98% in experiment.



(a) 360° Rotation Process



(b) Simulation



(c) Experiment

Figure 5.14: 360° rotation around pitch axis. (a) Rotation process of the platform in experiment. Showing the results in (b) Simulation, and (c) Experiment.

CHAPTER 6

Model-Based Iterative Learning Control on the Over-Actuated UAV

6.1 Introduction

Controller design of UAVs have been extensively studied in the last decades with increased popularity of aerial vehicles. A UAV is an unstable system in general, and feedback controllers must actively engage throughout the flight. Precise trajectory tracking of UAVs is important to guarantee the quality of aerial tasks such as monitoring and interaction with the environment. Yet the trajectory tracking is challenging because UAVs are prone to disturbances introduced by wind and the aerodynamic effects of propellers [BMSP09]. In addition, for over-actuated UAVs introduced in previous chapters, the airflow generated by different propellers affects each other since the thrusts are not all parallel. The multi-body dynamics also creates internal forces and torques that are nonlinear and will affect the performance of the whole platform.

The effects of wind and the aerodynamic effects on aerial vehicles have widely been studied in the literature. Estimation models of wind fields, aerodynamic disturbances by ground or obstacles, and propeller aerodynamics are given in multiple works in [WW09, KNC13, SCHO17, YDBJ+17, CBQ21] and the modeling of the aerodynamic effects mostly relies on the quality of sensor measurements. Specifically, tests of the downwash effects when one quadcopter flies under another and other aerodynamic effects introduced by a group of quadcopters are conducted in [MMLK10]. Online estimation of and active rejection of the aerodynamic effects are proposed in several works, such as [LAN11, YSPS15, BCSM16]. In

[BCMC14], the authors use the geometry of the obstacles to estimate the disturbances, which doesn't require accurate pose and position measurements but largely rely on the modeling of the predictor.

Several groups have proposed feedback controllers to improve the performance by utilizing the various onboard sensors of the UAV. In [HZX14], the authors designed an L_1 adaptive controller to estimate the wind effects. In [WWW+18], the authors proposed backstepping control with disturbance observer based on linear extended state observer to compensate for fluid torque on a tilt-rotor quad-rotor underwater vehicle. In [AML20], the authors designed a robust model reference adaptive controller on a tilt-rotor quadcopter to address the unknown inertia of the vehicle. In these works the baseline feedback controllers were all redesigned to meet the performance requirements.

Feedforward compensation is another method that can be implemented as a separate control module without affecting the baseline feedback controllers. Traditional methods of improving the control performance include formulating the model of the aerodynamic effects [YSPA15] and multi-body dynamics [SHV+06], and actively compensating for the nonlinear effects in real time. However, these methods require accurate model of the system, and are sensitive to modeling error. For an over-actuated UAV, the controller must calculate the system states containing complete multi-body dynamics in parallel with the baseline feedback controller, which is challenging for UAVs with limited onboard computational power.

Iterative learning control (ILC), a feedforward controller that is used for repeated tasks, has shown great performance in various applications. The main function of ILC is using the tracking error information from the previous trial to update the reference or input commands of the next trial, and keep updating as the task repeats. Prior to UAVs, ILC has succeeded in various applications such as industrial robot arms, wafer stage motion systems, motors and engine valves [BTA06]. ILC has also been applied to improve the tracking performance of conventional UAVs [BK13, DQR+18, FLW20, ML21]. In [PO13], a PD-type ILC was used to perform the take-off, translation and circular trajectories of a quadrotor UAV. In [PD09], the authors used the dynamic inverse as the learning filter of the ILC and used it to perform

aggressive motions of a quadrotor UAV. In [AZ20], the authors designed an optimization-based ILC that minimized the H-infinity norm of the UAV system.

Among different feedforward ILC filter design approaches, the inversion-based approaches give fast convergence rate and small steady-state error if an accurate model inverse can be obtained [BTA06, TT15]. However, similar to feedforward control, the performance of the ILC is sensitive to modeling error. Besides, if there is a nonminimum-phase zero in the model, in other words a zero in the right-half pane on the s-plane, direct inversion will result in an unstable filter. Therefore, several algorithms were proposed to obtain the system inverse for feedforward controller and ILC [vZO18]. Approximate inverse techniques of single-input-single-output (SISO) systems with nonminimum-phase zeros are proposed, including NPZ-ignore [GTM94], zero-phase-error tracking controller (ZPETC) [Tom87, TT87, TT88], and zero-magnitude-error tracking controller (ZMETC) [RPL09]. Optimization-based methods are proposed, such as norm-optimal ILC based on Riccati equations in linear quadratic control [GN01]. ILC based on $\mathcal{H}_\infty$ [WZWT16] and the extension with preview [HL08] are proposed. However, the inversion of multi-input-multi-output (MIMO) systems are too complicated and are mostly ignored.

In this chapter, a model-based ILC design is proposed for fully-actuated UAVs to improve the trajectory tracking performance for repeated tasks. Simplification process of the over-actuated UAV model is demonstrated. Although fully-actuated UAVs are essentially nonlinear MIMO systems, the independent control authority in each of the six DOF makes the system equivalent to having 6 SISO channels, and each of the SISO channel can be regulated by SISO ILC techniques mentioned above. This ILC filter is implemented on the high-level controller of the over-actuated UAV with gimbal actuators in Section 3.2, where the low-level dynamics are treated as disturbances. The ILC uses ZPETC method to obtain the inverse from the closed-loop system of the rigid-body dynamics with the baseline LQI controller. The ILC design is validated by both simulation and experiments, where the UAV platform is subject to aerodynamic effects caused by the downwash of the propellers and ground. It is demonstrated that with the accurate system model available, the model-

based ILC approach can significantly improve the trajectory tracking performance without redesigning the hierarchical feedback controller.

6.2 Backgrounds of ILC Algorithm

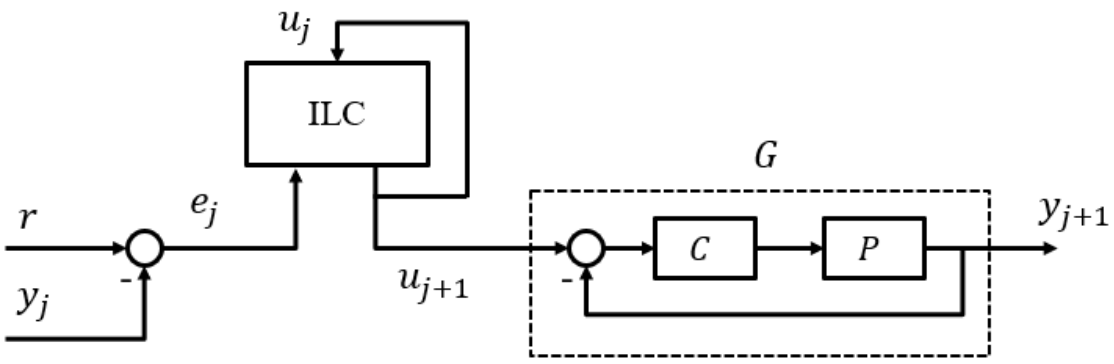


Figure 6.1: Block diagram of a generic ILC.

ILC is applied in situations where a system is asked to track a fixed trajectory in the same environment for multiple times, as shown in Fig. 6.1. The tracking error in the previous execution is used to update the input commands in the next execution and improve the tracking performance. Consider a SISO linear time-invariant (LTI) system G. The system has stable dynamics or is stabilized by a feedback controller C. The system is represented in discrete-time and finite impulse response (FIR) format as $G(z)$, and

$$G(z) = \sum_{k=-N}^{k=N} g(k) z^{-k} \tag{6.1}$$

where $g(k)$ is the impulse response of G, and N is the sampling length.

The system dynamics when tracking a fixed trajectory can be expressed as

$$y(z) = G(z)u(z) + d(z) \tag{6.2}$$

where u is the input signal, y is the output, and d is the disturbance. In ILC implementation, d contains the external disturbance that repeats every iteration.

A widely applied ILC update law [BTA06, Moo12] can be written as

$$u_{j+1}(k) = Q(z)[u_j(k) + L(z)(r(k+1) - y_j(k+1))] \tag{6.3}$$

where r is the desired trajectory of the task, j is the index of the task execution, and k is the time index; $L(z)$ is the learning filter in FIR representation, and $Q(z)$ is called a Q-filter, which is usually a low-pass filter to improve the robustness.

Because the ILC changes the input signal in every execution, the system $G(z)$ must be stable or pre-stabilized by feedback controllers. The ILC update process is conducted offline, and the feedback information of the complete trajectory is available. Therefore, $L(z)$ and $Q(z)$ can be non-causal and include information of the future time indices, which can further improve the performance and is not available in real-time feedback control. Eq. (6.3) is usually written in lifted format as

$$\boldsymbol{u}_{j+1} = \boldsymbol{Q}[\boldsymbol{u}_j + \boldsymbol{L}(\boldsymbol{r} - \boldsymbol{y}_j)] \tag{6.4}$$

where $\boldsymbol{r}$, $\boldsymbol{u}$ and $\boldsymbol{y}$ are the signals in $\mathbb{R}^{n\times 1}$ of the whole execution. $\boldsymbol{Q}$ and $\boldsymbol{L}$ are Toeplitz matrices constructed by shifting the FIR coefficients. Note that by using the lifted format, the ILC can be easily applied on LTV systems where $\boldsymbol{Q}$, $\boldsymbol{L}$ are not necessarily Toeplitz.

Writing the system dynamics Eq. (6.2) in lifted form and combining with the update law Eq. (6.4), the iteration domain dynamics can be expressed as

$$\boldsymbol{u}_{j+1} = \boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{G})\boldsymbol{u}_j + \boldsymbol{Q}\boldsymbol{L}(\boldsymbol{r} - \boldsymbol{d}) \tag{6.5}$$

We define the tracking error as $\boldsymbol{e}_j = \boldsymbol{r} - \boldsymbol{y}_j$. If the system is asymptotically stable, the

convergence error $\boldsymbol{e}_\infty = \lim_{y\to\infty} \boldsymbol{e}_j$, and $\boldsymbol{e}_\infty$ can be expressed as

$$\boldsymbol{e}_\infty = [\boldsymbol{I} - \boldsymbol{G}[\boldsymbol{I} - \boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{LG})]^{-1}\boldsymbol{QL}](\boldsymbol{r} - \boldsymbol{d})\boldsymbol{e}_\infty - \boldsymbol{e}_{j+1} = \boldsymbol{GQ}(\boldsymbol{I} - \boldsymbol{LG})\boldsymbol{G}^{-1}(\boldsymbol{e}_\infty - \boldsymbol{e}_j) \tag{6.6}$$

From [NG02], it is defined that the ILC system is asymptotically stable if and only if $\max|\lambda(\boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{LG}))| < 1$, where λ is the eigenvalue of a matrix. When $(\boldsymbol{I} - \boldsymbol{LG}) = \boldsymbol{0}$, the system is not only asymptotically stable, but also converge to zero error at $j = 1$. In other words, if the learning filter is the inverse of the system, $\boldsymbol{L} = \boldsymbol{G}^{-1}$, the ILC algorithm reaches steady state within one iteration and gives zero tracking error.

In this chapter, model-based approaches to obtain the learning filter as the system inverse will be presented. In Chapter 7, data-driven approaches to obtain the learning filter will be presented. The effectiveness of the ILC design in both chapters will be demonstrated by simulation and real-world experiments.

6.3 Model-Based ILC Algorithm

6.3.1 Closed-Loop System Representation of UAV Platform

In Chapter 3, a hierarchical controller is implemented on the over-actuated UAV platform. Since the low-level controller bandwidth is designed to be significantly faster than the high-level controller, the closed-loop dynamics are dominated by the high-level controller. In this section, the high-level dynamics will be the basis for designing the ILC controller, and the low-level dynamics are treated as disturbances.

From Eqs. (3.18) and (3.26), the closed-loop dynamics of the platform can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + -\boldsymbol{B}K\boldsymbol{x}_{\boldsymbol{aug}} \tag{6.7}$$

As discussed in Section 3.2.1, the optimal gain matrix $K = [K_1\ K_2\ K_3]$, where K_1, K_2, K_3

are diagonal matrices in $\mathbb{R}^{6\times6}$. Also, $\frac{1}{m}\boldsymbol{I}_3$ and $\boldsymbol{J}^{-1}$ as the submatrices in $\boldsymbol{B}$ are also diagonal. Therefore, the closed-loop dynamics can be written as

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\theta} \\ \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ K_{B1} & K_{B2} & K_{B3} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_p \\ \boldsymbol{e}_\theta \\ \boldsymbol{e}_v \\ \boldsymbol{e}_\omega \\ \int \boldsymbol{e}_p \\ \int \boldsymbol{e}_\theta \end{bmatrix} \tag{6.8}$$

where K_{B1}, K_{B2}, K_{B3} are diagonal matrices in $\mathbb{R}^{6\times6}$. It is clear that in each of the 6 DOF, the closed-loop dynamics can be expressed in an equivalent PID control format as

$$\ddot{w} = k_p e_w + k_d e_{\dot{w}} + k_i \int e_w \tag{6.9}$$

where w represents any of x, y, z, roll, pitch or yaw. k_p, k_d, k_i are the corresponding PID gains.

Therefore, the high-level dynamics of the UAV platform can be simplified as containing 6 SISO control channels, and each of the 6 DOF are controlled independently. ILC design approaches based on SISO plants can be used on this system.

6.3.2 Inversion-Based ILC Algorithm

The equivalent closed-loop dynamics in each 6 DOF in Eq. (6.9) can be treated as a double-integrator open-loop plant and a PID controller, which can be expressed in transfer function

$$P(s) = \frac{1}{s^2}, C(s) = k_p + k_d s + \frac{k_i}{s} \tag{6.10}$$

To implement the controller on the physical UAV platform hardware, the system is ex-

pressed in discrete-time with zero-order hold and a sampling rate of $T = 0.01sec$ as

$$\begin{aligned} P(z) &= \frac{z-1}{z} \cdot Z[\frac{P(s)}{s}] = \frac{T^2}{2}\frac{z+1}{(z-1)^2} \\ C(z) &= k_p + \frac{k_d(z-1)}{Tz} + \frac{k_i T(z+1)}{z-1} \\ G(z) &= \frac{C(z)P(z)}{1+C(z)P(z)} \end{aligned} \tag{6.11}$$

Given the optimal state-feedback gain matrix used in the experiment sec, the equivalent high-level PID gains and the discrete-time transfer function in each direction are shown in Table 6.1. Note that nonminimum-phase zeros appear in each direction of the transfer function. The nonminimum-phase zeros can result from getting the discrete-time model out of the continuous-time model, and from the use of the integral control with unit-step delay. With the nonminimum-phase zeros, obtaining the system inverse by direct inversion of the poles and zeros will result in an unstable filter. As introduced in Section 6.1, three popular techniques are available to obtain the stable inverse from a SISO system: NPZ-ignore, ZPETC and ZMETC. NPZ-ignore approach simply ignores any nonminimum-phase zeros in the transfer function, which is the least accurate because a part of the system dynamics is totally ignored. From the tests given in [BPA12], with the presence of nonminimum-phase zeros in the left-half plane ("$1 + z^{-1}$" in this case), ZPETC gives better performance in the high-frequency part of the system. Therefore, the ZPETC approach is used to obtain the system inverse in this thesis.

The closed-loop system with stable and unstable zeros separated can be written as

$$G(z) = \frac{y(k)}{r(k)} = \frac{z^{-d}B_s(z^{-1})B_u(z^{-1})}{A(z^{-1})} \tag{6.12}$$

where $A(z)$ are the poles of the original system, $B_s(z)$ are the stable zeros, and $B_u(z)$ are the unstable zeros. The ZPETC inverse [Tom87] is given as

$$F(z) = \frac{r(k)}{y_d(k+d)} = \frac{A(z^{-1})B_u(z)}{B_s(z^{-1})[B_u(1)]^2} \tag{6.13}$$

Table 6.1: Equivalent PID Gains and Transfer Function in Each Direction

Dir.	k_p	k_i	k_d	**Transfer Function**
x	4.5	2.0	2.7	$G_x(z) = \frac{0.013726z^{-1}(1+z^{-1})(1-1.983z^{-1}+0.9835z^{-2})}{(1-0.985z^{-1})(1-0.01388z^{-1})(1-1.987z^{-1}+0.9876z^{-2})}$
y	4.5	2.0	2.7	$G_y(z) = \frac{0.013726z^{-1}(1+z^{-1})(1-1.983z^{-1}+0.9835z^{-2})}{(1-0.985z^{-1})(1-0.01388z^{-1})(1-1.987z^{-1}+0.9876z^{-2})}$
z	5.0	2.5	3.0	$G_z(z) = \frac{0.015251z^{-1}(1+z^{-1})(1-1.983z^{-1}+0.9835z^{-2})}{(1-0.982z^{-1})(1-0.01547z^{-1})(1-1.987z^{-1}+0.9875z^{-2})}$
roll	17.0	6.0	5.5	$G_{\text{roll}}(z) = \frac{0.028353z^{-1}(1+z^{-1})(1-0.9891z^{-1})(1-0.9806z^{-1})}{(1-0.9906z^{-1})(1-0.02912z^{-1})(1-1.952z^{-1}+0.9533z^{-2})}$
pitch	17.0	6.0	5.5	$G_{\text{pitch}}(z) = \frac{0.028353z^{-1}(1+z^{-1})(1-0.9891z^{-1})(1-0.9806z^{-1})}{(1-0.9906z^{-1})(1-0.02912z^{-1})(1-1.952z^{-1}+0.9533z^{-2})}$
yaw	8	3	3	$G_{\text{yaw}}(z) = \frac{0.015402z^{-1}(1+z^{-1})(z^2-1.974z+0.9739)}{(1-0.99z^{-1})(1-0.01546z^{-1})(1-1.979z^{-1}+0.9797z^{-2})}$

where $B_u(z)$ is obtained by replacing z^{-1} in the polynomial with z. To implement this inverse filter, the desired trajectory $y(k)$ is shifted forward by d steps and feeds into the system to make the filter causal, as shown in Fig. 6.2.

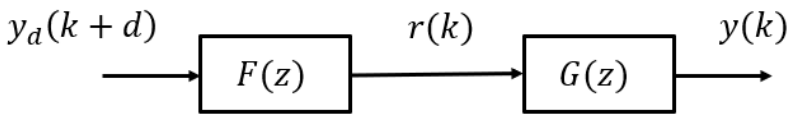


Figure 6.2: Block Diagram of the ZPETC Feedforward Filter

The overall transfer function from the desired output to the actual output is

$$\frac{y(k)}{y_d(k)} = \frac{B_u(z^{-1})B_u(z)}{[B_u(1)]^2} \tag{6.14}$$

In the frequency domain, the overall system has zero phase across all frequencies. In low frequencies, the gain is close to 1. The transfer function of the UAV platform in x direction

is shown below as an example. The ZPETC inverse of $G_x(z)$ is

$$F_x(z) = \frac{18.214(1 - 0.985z^{-1})(1 - 0.01388z^{-1})(1 - 1.987z^{-1} + 0.9876z^{-2})}{z^{-1}(1 - 1.983z^{-1} + 0.9835z^{-2})} \tag{6.15}$$

And the FIR format of $F_x(z)$ is shown in Fig. 6.3. Notice that the FIR is non-causal because of the one-step phase shift in the transfer function.

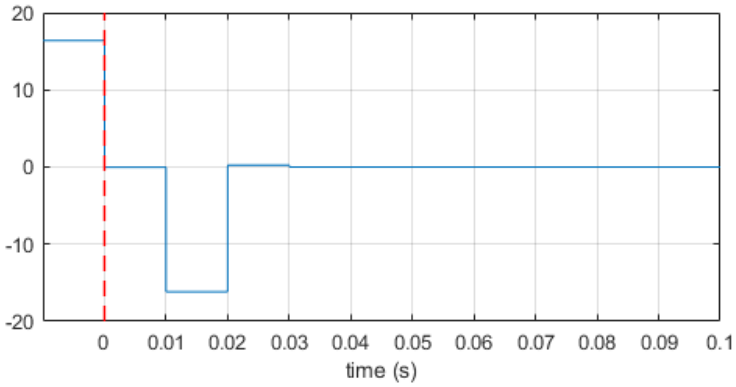


Figure 6.3: FIR representation of the ZPETC inverse.

The Bode plot of the overall system is shown in Fig. 6.4. The ZPETC approach effectively creates the system inverse with zero phase tracking result. The system has unity gain when the frequency is lower than around 20 Hz. For the other 5 DOF of the UAV platform, due to the presence of the nonminimum phase zero at $z = -1$, the overall tracking performance of the inverse filter is similar.

In the next steps, the impulse response of $F(z)$ is taken and is used to build the learning filter in Eq. (6.4) in lifted format as a toeplitz matrix. $\boldsymbol{Q}$ is necessary, as the overall system response has unity gain in low frequencies only. $\boldsymbol{Q}$ is usually designed as a low-pass filter, and the cut-off frequency is closely related to the bandwidth of $\boldsymbol{L}$ to ensure stability of the system. With the ZPETC inverse as the $\boldsymbol{L}$ filter, $\boldsymbol{Q}$ is chosen as a zero-phase low-pass filter and can be written as

$$Q(z) = \cdots + q_{-2}z^2 + q_{-1}z^1 + q_0 + q_1z^{-1} + q_2z^{-2} + \cdots \tag{6.16}$$

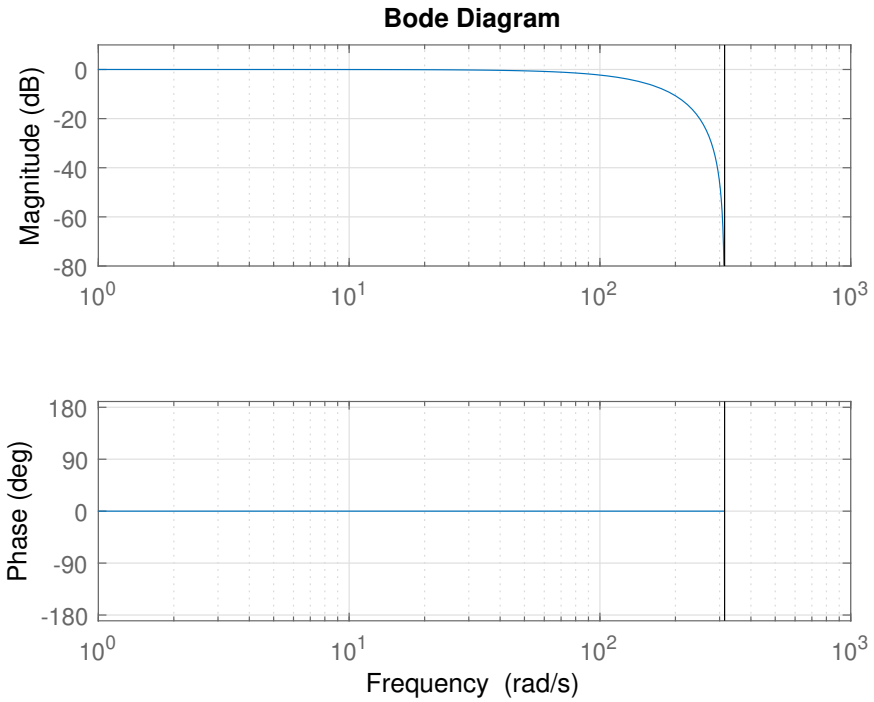


Figure 6.4: Overall frequency response of the ZPETC feedforward filter in x direction.

where

$$q_i = 2\omega_c T \text{sinc}(2\omega_c t) \tag{6.17}$$

and ω_c is the cut-off frequency of the filter. The purpose of the zero-phase low-pass filter is to totally reject the learning of the error in high frequencies. When $\omega_c = 5Hz$, the representation of the Q filter in time and frequency domain is shown in Fig. 6.5. The time domain sequence is used to construct the $\boldsymbol{Q}$ in lifted form.

6.4 An Estimation of Downwash Effects in Simulation

It has been tested that One quadrotor can influence the dynamic behavior of neighboring quadrotors because of the downwash from its rotors [MMLK10]. The experiment cases of our over-actuated UAV platform in Section 5.3.4, where the platform rotates around the pitch angle for 90° show the downwash effects that reduce the tracking performance. To verify the effectiveness of the ILC algorithm on the platform, an estimation model of the downwash

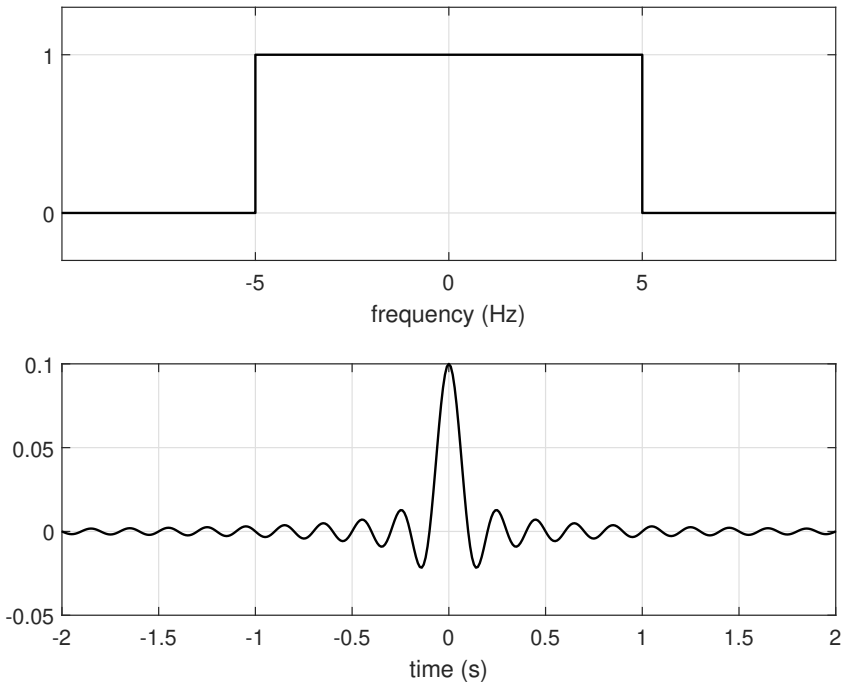


Figure 6.5: Frequency and time domain representation of Q filter when $\omega_c = 5Hz$.

effects is first created in the simulation.

As shown in the experiment, the downwash disturbance is most obvious on z direction in $\mathcal{F}_W$. Therefore, the estimated downwash model is designed to exert a downward force on z direction, and is exerted on the bottom quadcopter as the effect of reduced thrust. The magnitude of the downwash force depends on the distance in the horizontal plane of the top and bottom quadcopters, as shown in Fig. 6.6. The magnitude of the downwash force is 0 when the horizontal distance is large. In other words, when the top and bottom quadcopters are not overlapping, their thrusts do not interfere with each other. When the distance is smaller than a threshold d_{zero}, the force gradually increases to a maximum value F_{max} at d_{full}. The values of these corner cases are estimated from the experiment data, where $d_{zero} = 0.133\,\mathrm{m}$, $d_{full} = 0.036\,\mathrm{m}$, $F_{max} = 0.1\,\mathrm{N}$.

The simulation result of the platform tracking the same complete 360° trajectory is shown in Fig. 6.7. In the position plots, the z position in simulation shows the same drop and rise near the vertical flight region. The thrust plot of the simulation shows the same amount of

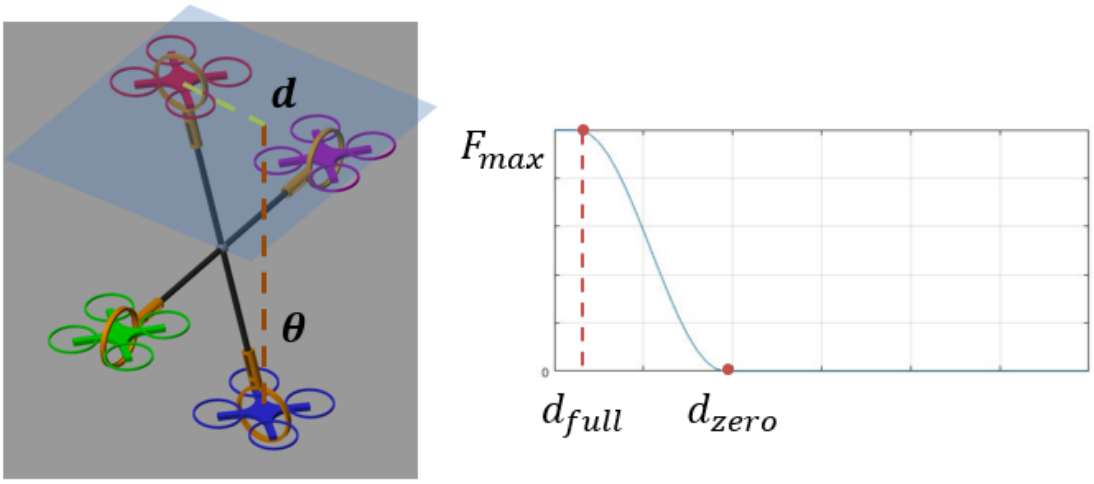


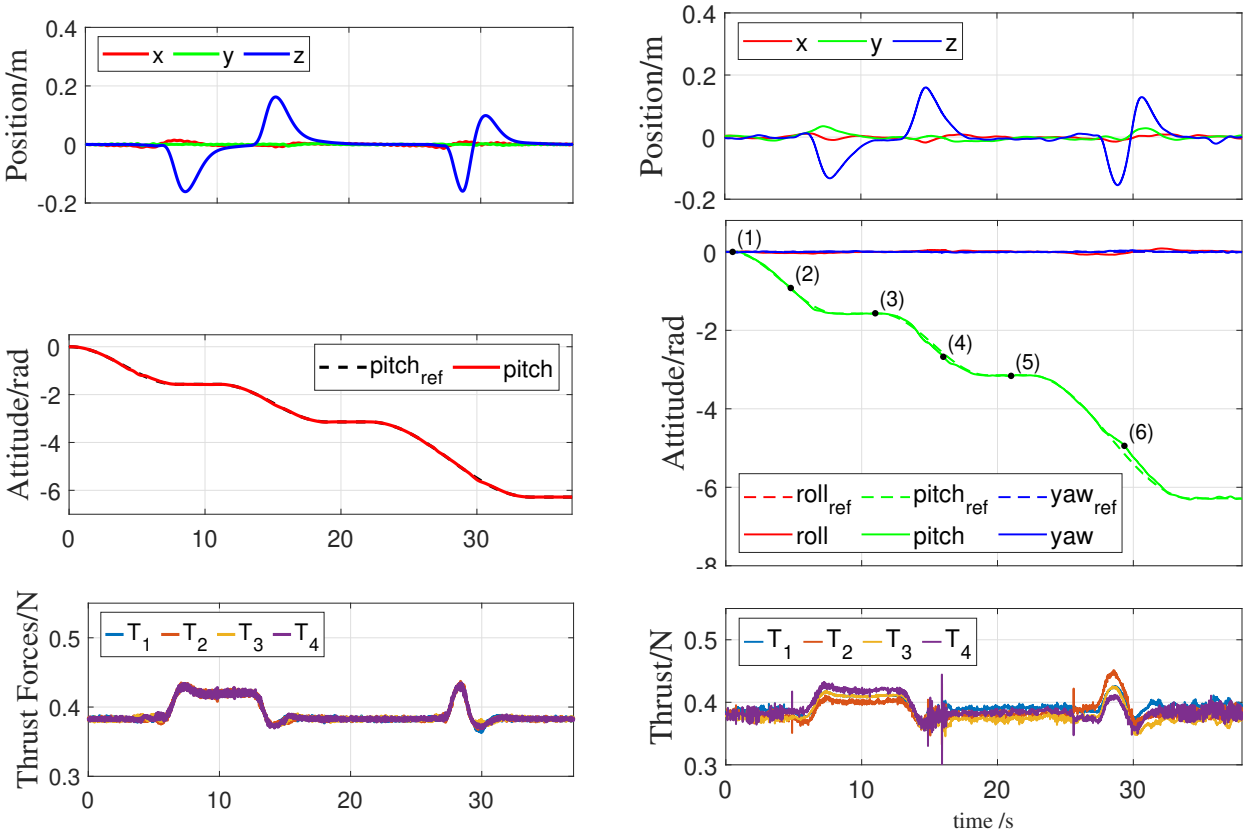
Figure 6.6: Simulation of the downwash effects in simulation. The downwash force is on z direction, and the magnitude depends on the horizontal distance of the top and bottom quadcopters.

increase near the vertical flight region. Therefore, the estimated downwash force model has similar behavior as in the real-world experiment case, and this estimated downwash model can be used to verify the effectiveness of the ILC algorithm in simulation.

6.5 Model-Based ILC on z Direction

6.5.1 Simulation Results

As demonstrated previously in Section 6.3, the closed-loop behavior of the over-actuated UAV platform can be simplified as containing 6 SISO LTI channels in each DOF, and the equivalent closed-loop transfer function of each channel is available. In the first application, the model-based ILC with the learning filter as the zero-phase inverse of the transfer function is implemented on z direction of the platform, in order to address the downwash effects at vertical flights. The ILC changes the input of the desired z position every iteration, while the hierarchical controller remains unchanged throughout the process. Given the frequency response of the ZPETC filter and the shape of the downwash disturbance on the output, the Q filter is designed as a zero-phase ideal low-pass filter with cut-off frequency $\omega_c = 5Hz$. The reference trajectory of the platform pitch angle is shown in Fig. 6.8, where the platform



(a) Simulation (b) Experiment

Figure 6.7: 360° rotation around pitch axis with downwash effect. (a) Simulation, and (b) Experiment.

rotates from from 0 to 180°, and remains at 90° for 3 seconds. At 90° the simulated downwash force is at maximum magnitude. The desired z position is 0 throughout the trajectory.

The simulation result is shown in Fig. 6.9. The first iteration $j = 0$ is the dry run without ILC and the input is equal to the reference. The model-based ILC changes the position input u, and the over-actuated platform creates additional force when attempting to track the input. The additional force, combined with the external downwash force, reduces the error in z direction in the next iterations. Within the first few iterations, the error in z direction is significantly reduced. However, the high-frequency signals on the input u accumulates as more iterations are executed. The RMS error and the maximum absolute error on z direction are shown in Fig. 6.10. Both the RMS and max error quickly reduce to the lowest level within the first 3 iterations. Starting at the 7th iteration, the RMS and max

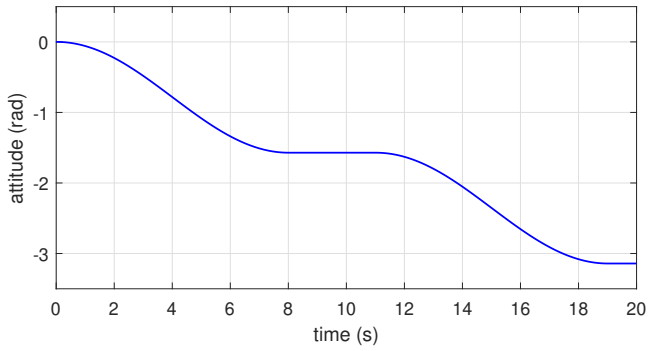


Figure 6.8: Reference trajectory of pitch angle from 0 to 180°, and remains at 90° for 3 seconds.

error raise again and maintain at a higher error level.

Obviously the high-frequency part of the input signal u is excited by the measurement noise and the error from the previous iterations, which may affect the system stability because the learning filter is not close to the system inverse at high-frequencies. In this case the selection of Q filter is essential to guarantee the robustness of the system.

Next, the cut-off frequency of the Q filter is reduced to 1 Hz, and the simulation is conducted again. The result is shown in Fig. 6.11. Compared with the 5 Hz case, the error furthur reduced after 14 iterations. The error is still varying because of the added sensor noises and uncertainties in the simulation. The convergence is not as expected as to reduce to minimum within one iteration because of the learning filter design. The results indicates that the ZPETC inverse based on the high-level transfer function cannot represent the simulated physical model, which is close to the real-world system as shown in Section 5.3. Despite this, experiments have been done to show that the proposed ILC algorithm can reduce the tracking error to some extent, and the experiment results are shown in the following sections.

6.5.2 Experiment Results

The closed-loop behavior of the over-actuated UAV platform can be simplified as containing 6 SISO LTI channels in each DOF, and the equivalent closed-loop transfer function of each channel is available. Same as the simulation, the model-based ILC with the learning

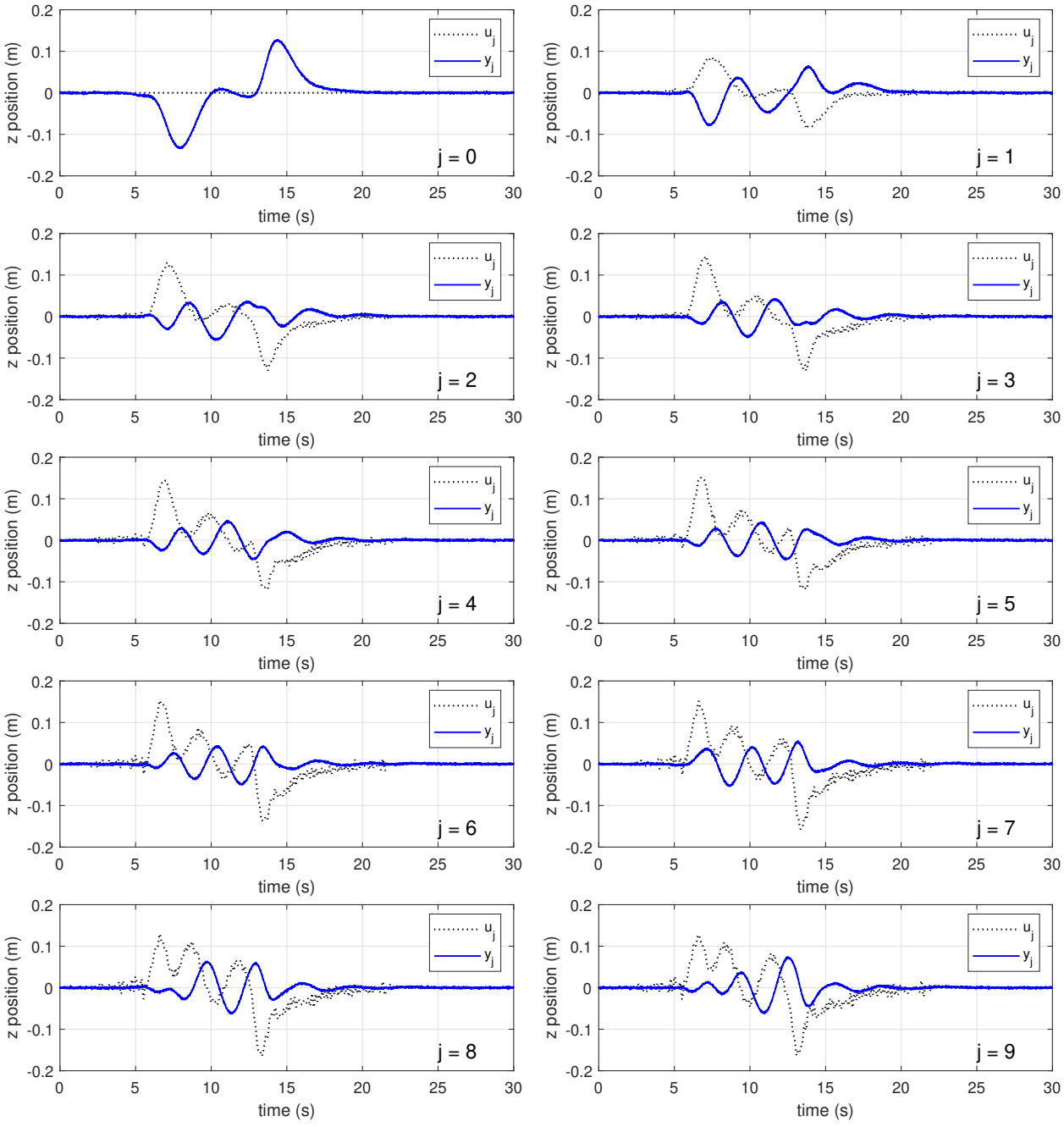


Figure 6.9: Simulation result of model-based ILC on z direction from $j = 0$ to 9.

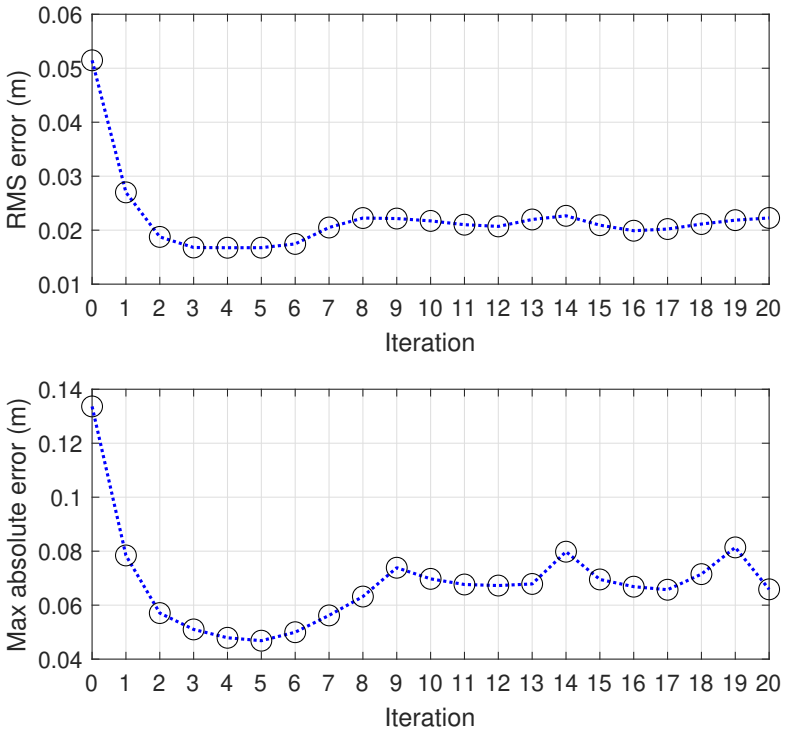


Figure 6.10: Simulation - RMS error and max absolute error of model-based ILC on z direction, Q: 5 Hz.

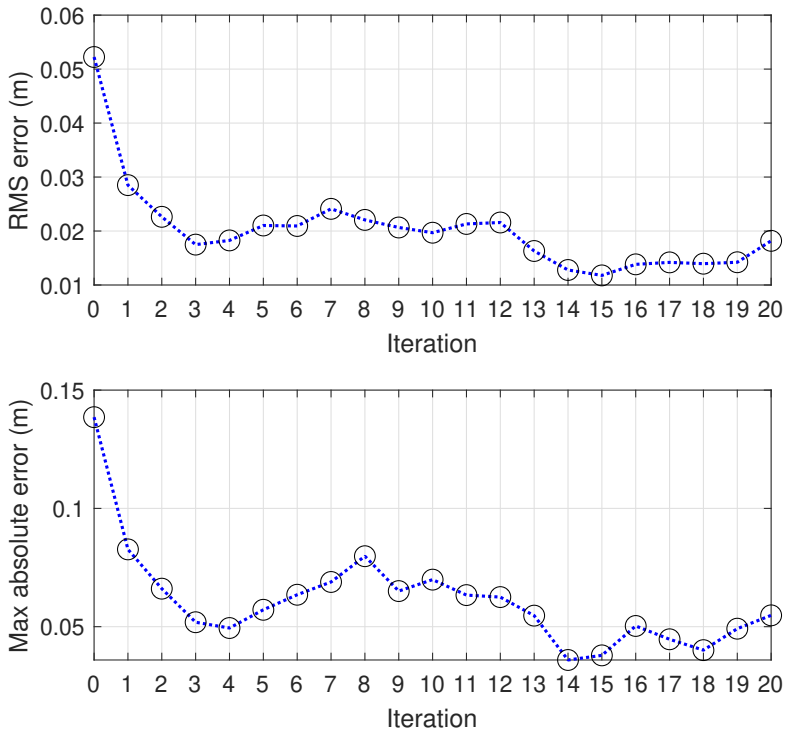


Figure 6.11: Simulation - RMS error and max absolute error of model-based ILC on z direction, Q: 1 Hz.

filter as the zero-phase inverse of the transfer function is implemented on z direction of the platform, in order to address the downwash effects at vertical flights. From the insights shown in the simulation the Q filter should be in lower cut-off frequency to filter out the high-frequency noises from the measurements in previous iterations. In experiment, the measurement of the pose and position of the over-actuated aerial platform is subject to multiple sources of noises, and thus the Q filter is designed as a zero-phase ideal low-pass filter with cut-off frequency $\omega_c = 1Hz$. The reference trajectory of the platform pitch angle is shown in Fig. 6.8, where the platform rotates from from 0 to 180°, and remains at 90° for 3 seconds. The desired z position is 0 throughout the trajectory. In the experiment, the data is collected during each iteration, and the ILC learning process is conducted offline to obtain the input trajectory for the next iteration.

The simulation result is shown in Fig. 6.12. The first iteration $j = 0$ is the dry run without ILC and the input is equal to the reference. The model-based ILC changes the

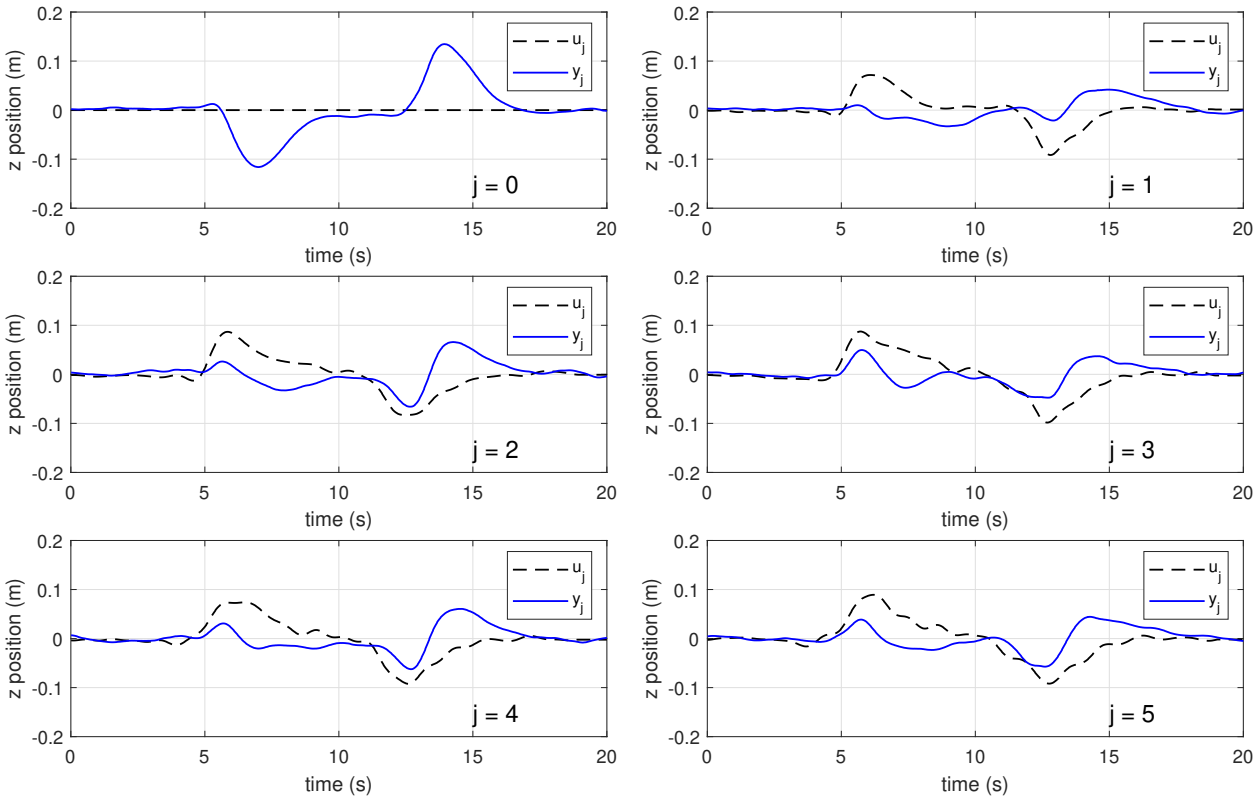


Figure 6.12: Experiment result of model-based ILC on z direction from $j = 0$ to 5.

position input u, and the over-actuated platform creates additional force when attempting to track the input. The additional force, combined with the external downwash force, reduces the error in z direction in the next iterations. The RMS error and the maximum absolute error on z direction are shown in Fig. 6.13. Both the RMS error and max error in z direction is significantly reduced within the first iteration, and the errors maintains at the same level for the future iterations.

Since the setup in experiment is much more complicated than the experiment, only a few iterations are repeated. But the experiment data is sufficient to show the proposed ILC implemented on the z direction of the over-actuated platform is stable. The ILC learned from the previous measurement data and can reduce the error to a lower level under the repeated downwash effects. Moreover, with the learning filter designed as the ZPETC inverse of the high-level transfer function on z direction of the dynamics, the error converges within the first few iterations.

Therefore, a conclusion can be drawn that the linearization of the over-actuated platform is successful, and the ILC is helpful to address the aerodynamic effects on the platform that are normally difficult to model. Meanwhile, the existing issues of the presented model-based ILC approach are two folds. First, the performance of the ILC is highly dependent on the model accuracy. Second, the low-level dynamics of the over-actuated UAV platform is not included in the high-level model, and the effects are nonlinear. It will be difficult to model the low-level dynamics under the same model-based approach. Since the system inverse is obtained by an estimation method, better approximation of the system inverse for the learning filter remains a topic to discuss in the next chapters.

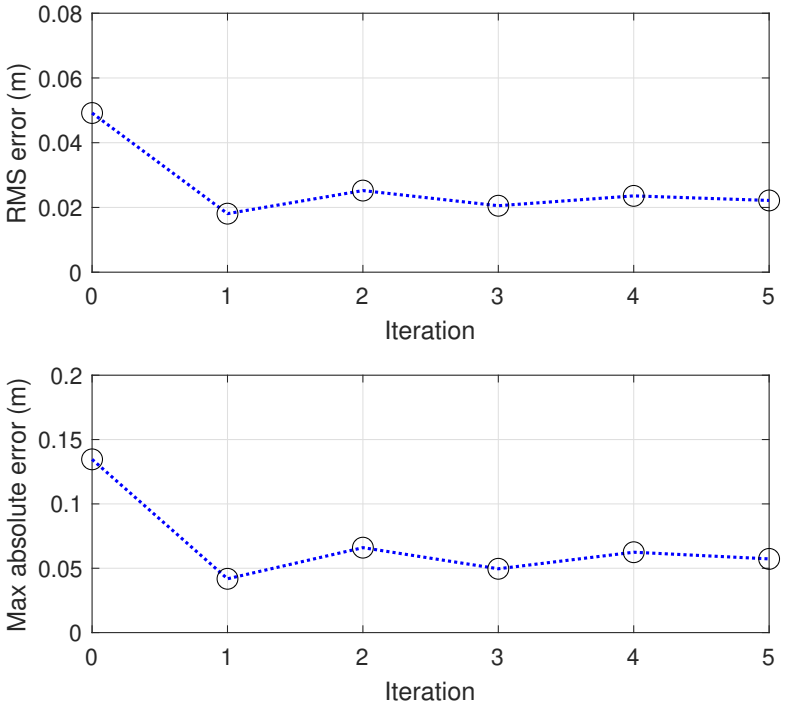


Figure 6.13: Experiment - RMS error and max absolute error of model-based ILC on z direction, Q: 1 Hz.

CHAPTER 7

Data-Driven Iterative Learning Control on the Over-Actuated UAV

7.1 Introduction

The performance of typical feedforward controllers, repetitive control and ILC heavily depends on the quality of the modeling of the system inverse filter. The inversion-based ILC approaches give fast convergence rate and small steady-state error if an accurate model inverse can be obtained [BTA06, TT15]. The accuracy of the system inverse depends on the accuracy of the system model. System identification is a popular data-driven approach by estimating model parameters from output data with designated inputs to the real-word system [ÅE71, FL13]. System identification methods can be applied to models with or without partial knowledge of the system configurations, and can provide results with high accuracy.

On the other hand, the accuracy of the system inverse also depends on the phase condition of the model. If a nonminimum-phase zero is present, direct inversion of the model will result in an unstable filter. Even if an accurate model is obtained by system identification methods, the inversion of the nonminimum-phase zeros still involves estimations to keep the inverse filter stable. Given this limitation, data-based inversion approaches are proposed, including the B-spline filtering with limited preview [DYO18], the modeling-free inversion-based iterative feedforward control [KZ12], and the nonlinear iterative inversion-based control [dRO19]. It is shown that these methods are sensitive to external disturbances and measurement noises because the least-squares approach or point-wise division in the frequency domain are involved.

A data-driven approach to construct an FIR filter of the system inverse is proposed by MacLab members [CRT19]. This method uses a baseline ILC to track an impulse signal with limited bandwidth as the output. The converged input sequence is directly applied as an FIR filter of the system inverse. This approach provides more robustness to output disturbances and measurement noises because of the bandwidth limit of the impulse, but the accuracy of the system inverse is limited by the selection of the baseline ILC filter. This approach is improved by progressively updating the baseline ILC filter using the learned system inverse as proposed in [CT21]. The process can be summarized as follows. First, an impulse signal with its bandwidth limited by a zero-phase filter whose cut-off frequency is chosen based on system bandwidth is selected as the reference. Next, a baseline ILC is constructed to learn the reference impulse. After a few iterations, the input signal is used as an estimation of the system inverse, and is used to update the learning filter in the baseline ILC. Then the process repeats and the system output is getting close to the impulse reference with an accelerated speed, and the converged input becomes an accurate estimation of the system inverse in FIR format. It is demonstrated that the convergence rate of the learning process is significantly increased, and the convergence error is further reduced by the progressive updating process. The robustness to noises in high frequencies is also analyzed.

In this chapter, the data-based feedforward filter construction using ILC [CRT19] with progressive updates [CT21] is demonstrated on the over-actuated UAV. This method is implemented on the high-level controller of the over-actuated UAV with gimbal actuators in Section 3.2, where the low-level dynamics are designed to converge faster than the high-level dynamics and are treated as disturbances. The baseline ILC filter is constructed by data-driven inverse time filtering of the system in each of the 6 DOF [WYZ14], and is progressively updated by the previous learned inputs. The process is demonstrated by both simulation and experiments, where the UAV platform is commanded to learn a bandwidth limited impulse signal separately in each 6 DOF. The converged system inverse filters are used to design an ILC to track trajectories under external disturbances as different aerodynamic effects. The simulation and experiment results demonstrate that the data-driven ILC can significantly

improve the tracking performance under of the UAV platform with fast convergence rates under unknown external disturbances.

7.2 Data-Driven ILC Algorithm

7.2.1 ILC-Based Feedforward Filter Construction

It has been demonstrated in Section 6.2 that the performance of an ILC filter is largely dependent on the quality of the learning filter L. When the learning filter is equivalent to the inverse of the system, the ILC algorithm converges to zero tracking error within one learning iteration. Therefore, to maximize the tracking accuracy and reduce the number of repeated task executions, obtaining an accurate model of the physical inverse is essential in real-world applications. The ILC-based method proposed in [CRT19] is used to construct a feedforward filter of the UAV platform. This method is data-driven, without any prior knowledge of the closed-loop dynamics in each direction of the platform.

The main idea of the ILC-based feedforward filter construction is that, suppose an ILC filter as demonstrated in Fig. 6.1 is already established and converged on a system. The system can track any reference signal within the bandwidth of the system, $y_\infty(z) = r(z)$. When the reference signal is an impulse $\delta(t)$, the converged output satisfies $y_\infty(t) = \delta(t)$, and in the z-transform domain $y_\infty(z) = 1$. Then we have

$$G(z)u_\infty(z) = y_\infty(z) = 1 \tag{7.1}$$

which implies

$$u_\infty(z) = G^{-1}(z) \tag{7.2}$$

The converged input sequence to the system is the FIR of the system inverse in this case, $F(z) = u_\infty(z)$. Ideally, the FIR of the system inverse can be directly applied as a feedforward controller, or be used to construct the learning filter of an ILC.

However, in real-world applications, there are two issues related to this approach that

needs to be solved. First, the ideal impulse signal $\delta(t)$ is noncausal, and the attempt to directly track the impulse signal with an ILC is not applicable. Therefore, the reference signal is processed by a phase shift, making the peak of the impulse appear at $t > 0$ in time domain. After the ILC converges, the system inverse $F(z)$ is constructed by a phase shift of $u_\infty(z)$ in the reverse direction. Second, the $F(z)$ is the exact inverse of $G(z)$ across all frequencies. This will result in high gains at high frequencies, which will trigger control saturation in real-world applications. To limit the high frequency gains, low-pass filter must be applied to the reference signal.

A reference signal $M(z)$ that is the impulse response of a zero-phase low-pass filter with delay is used in the revised learning framework, as shown in Fig. 7.1. The reference signal is

$$M(z) = z^{-N_q}(\frac{z^{-1} + 2 + z}{4})^{N_q} \tag{7.3}$$

where N_q is the order of the filter. The selection of the N_q value depends on the system bandwidth. When N_q is small, the reference signal is closer to an impulse, and the system inverse will be more accurate at high frequencies yet the risk of hitting control saturation will be increased. On the other hand,, when N_q is large, the reference signal is smoother and cannot trigger the high-frequency response of the system, but the control input convoluted with the system inverse will be smoother as well.

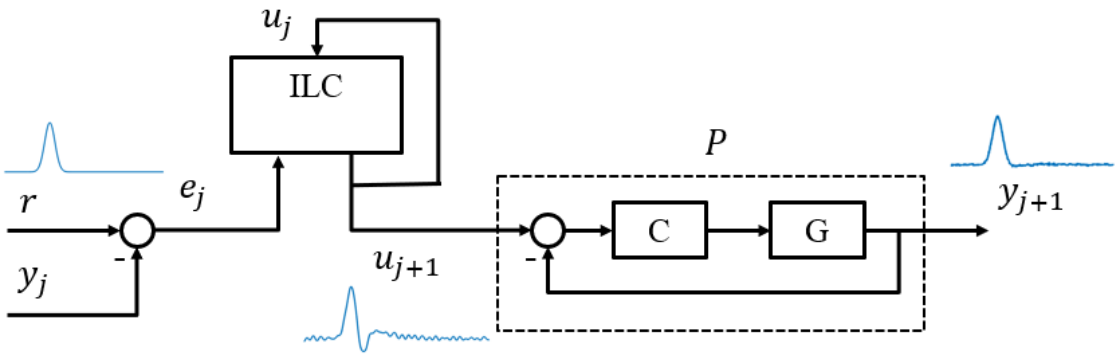


Figure 7.1: Block diagram of the ILC-based feedforward filter construction.

After the ILC converged, the system inverse can be obtained by

$$F(z) = z^{N_q} u_{j+1}(z) \cdot T \tag{7.4}$$

where T is the sampling rate of the system, and $F(z)$ has the same bandwidth as the reference signal $M(z)$.

7.2.2 Initialization by Reverse Time Filtering

To initiate the learning of the inverse filter, a baseline ILC must be established. At this stage it is assumed that the dynamics of the system is unknown. A common solution is to use a PD-type ILC [AKM84]. PD-type ILC doesn't require any prior knowledge of the platform; instead, it relies on manually tuning the learning gains. The learning function fo the PD-type ILC can be written as

$$u_{j+1}(k) = u_j(k) + K_p e_j(k+1) + K_d[e_j(k+1) - e_j(k)] \tag{7.5}$$

However, because of the manual tuning of the learning gains, the performance cannot be guaranteed. Finding an optimal gain without knowing the system dynamics is very time consuming in experiment. To ensure the tracking performance and accelerate the learning process, a different method that can quickly estimate the system dynamics is needed.

Adjoint-type ILC is proposed as a modeling-free ILC approach [SA94], where the output error was used backward in time to create the system adjoint. The performance of the adjoint operator is studied in [KSA02]. The time-reversal technique is first proposed in [KJ74], and it is shown that the time-reversal technique can be used to realize the system adjoint [WYZ14], which can be used as the zero-phase approximation of the system inverse and be used in ILC design. The adjoint filter is obtained by using time reversal of the output signal, and is explained below.

The initial response of the system from an impulse is defined in lifted format as

$$\boldsymbol{y}_1 = \boldsymbol{G}\boldsymbol{r}_\delta \tag{7.6}$$

where $\boldsymbol{r}_\delta$ is the FIR of the smooth impulse $M(z)$. The real-world model of the system $\boldsymbol{G}$ is unknown, but by feeding the input $\boldsymbol{r}_\delta$, $\boldsymbol{y}_1$ is directly available as the output.

We define the time reversal operator as I_R, where the diagonal elements from lower left to upper right are 1, and the other elements are 0. The time reversal operator is written as

$$I_R = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \iddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \tag{7.7}$$

The initial output is reversed in time and input to the system again. The second output is

$$\boldsymbol{y}_2 = \boldsymbol{G}I_R\boldsymbol{y}_1 \tag{7.8}$$

The adjoint filter is the time reversal of the second output

$$\boldsymbol{w} = I_R\boldsymbol{y}_2 \tag{7.9}$$

Then the adjoint filter can be written as

$$\boldsymbol{w} = I_R\boldsymbol{G}I_R\boldsymbol{G}\boldsymbol{r}_\delta = \boldsymbol{G}^*\boldsymbol{G}\boldsymbol{r}_\delta \tag{7.10}$$

For a LTI system in lifted format as introduced in Section 6.2, $\boldsymbol{G}$ is a toeplitz matrix, and it is obvious that $\boldsymbol{G}^* = \boldsymbol{G}^\mathsf{T}$. Therefore, the filter $\boldsymbol{G}^*\boldsymbol{G}$ is a system with positive magnitudes and zero phase accross all frequencies. With sufficiently small gain α, the stability criterion $\max|\lambda(\boldsymbol{I} - \boldsymbol{G}^*\boldsymbol{G})| < 1$ for an ILC on the system $\boldsymbol{G}$ holds. Therefore, a learning filter can be

determined by the process above as

$$\boldsymbol{L} = \alpha \boldsymbol{G}^* \tag{7.11}$$

Since the system dynamics $\boldsymbol{G}$ is unknown, the gain α is manually tuned. Compared with PD-type ILC, if α is sufficiently small and ILC stability is guaranteed, the value of α only affects the convergence rate of ILC. Therefore, finding the value of α that provides both the stability and the convergence rate can be done by setting the initial value to be relatively large, and gradually decreasing the value until the ILC is stabilized.

7.2.3 Progressive Update of the ILC Filter

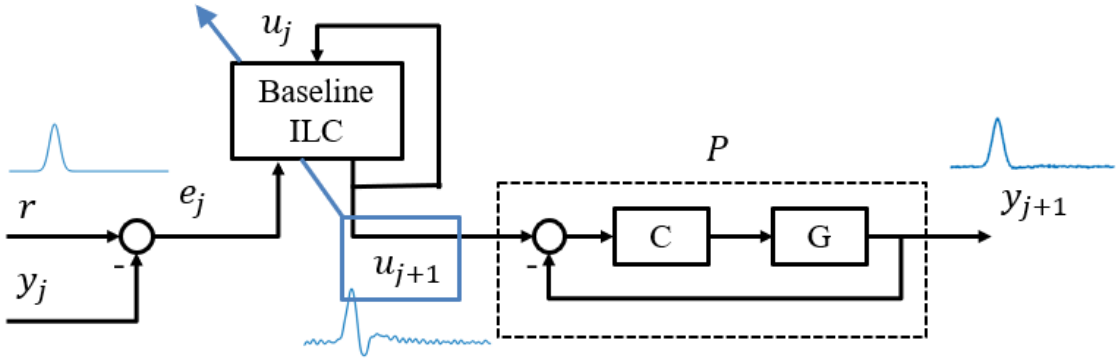


Figure 7.2: Block diagram of the ILC-based feedforward filter construction with progressive update.

The baseline ILC filter is constructed to track the impulse reference signal, and the converged system input can be used to approximate the system inverse. Meanwhile, the learning filter in the ILC gives ideal performance if it is the system inverse. Therefore, a framework of using the learned inverse filter to reconstruct the baseline ILC is proposed in [CT21], and the convergence conditions and the effects of the update period was analyzed.

The updated filter can be directly obtained from Eq. (7.4), which can be written as

$$L(z) = F(z) = z^{N_q} u_{j+1}(z) \tag{7.12}$$

Notice that, the progressive update of the learning filter can be done before the tracking of

the reference converges, depending on the noise-to-signal ratio and the tracking error. The convergence condition with various update periods is analyzed in [CT21].

7.3 Data-Driven ILC on z Direction

Same as the model-based ILC demonstrations,the data-driven ILC with the learning filter obtained by using another ILC filter to learn the impulse signal of the over-actuated platform is implemented on z direction in the first application, in order to address the downwash effects at vertical flights. First, the over-actuated platform maintains hovering pose, and a baseline ILC is implemented to learn the smooth impulse exerted on z direction iteratively. Next, after the learning of the impulse converges, the FIR filter of the system inverse is constructed from the converged input. The FIR filter of system inverse is then used to create a new ILC that is used to address the application trajectories involving repeated disturbances. The advantage of the data-driven learning process is similar to the system identification process: for the given over-actuated UAV platform, the learning of the system inverse needs to be done only once, and the learned filter can be directly applied to various application scenarios.

The reference signal as a smooth impulse $M(q)$ is Eq. (7.3), and $N_q = 200$. The smooth impulse is shown in Fig. 7.3 and the bode plot of the smooth impulse is shown in Fig. 7.4. The bandwidth of the smooth impulse is around 5 Hz, which is close to the bandwidth of the over-actuated system. If higher bandwidth is selected, the platform may be in saturation and cannot learn the impulse well.

In this section, simulation results of the process will be shown first. Then the experiment results will be shown.

7.3.1 Simulation Results

The data-driven initialization of the learning filter by reverse time filtering is shown in Fig. 7.5. The zero-phase fashion of the FIR filter is obvious. Since the filter is obtained from the output of the simulation model, the measurement noise is also included in the filter.

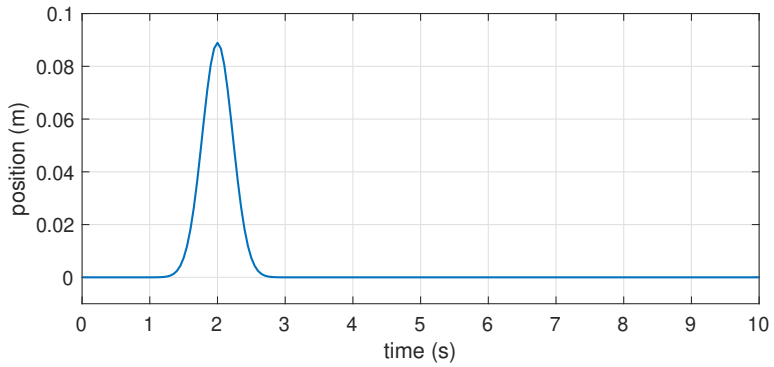


Figure 7.3: Smooth impulse $M(q)$ with $N_q = 200$.

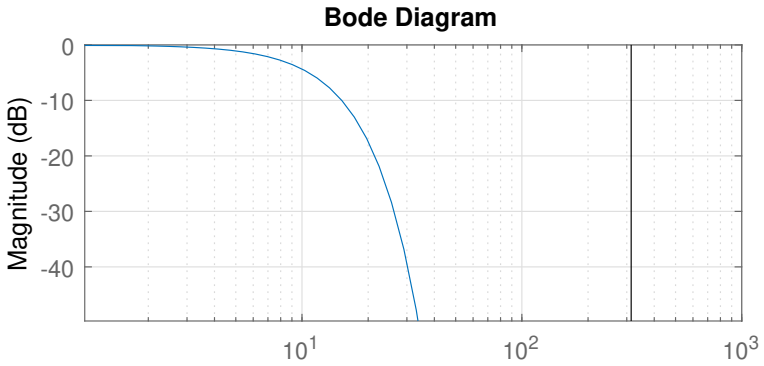


Figure 7.4: Bode plot of the smooth impulse $M(q)$.

Therefore, Q filter is still needed in this case. However, with the data-driven approach, the bandwidth of the learned filter can be compared with the smooth impulse, which also includes a low-pass filter already. Then the function of the Q filter is to filter out the high-frequency signals in the output data, and the cut-off frequency can be higher than the model-based ILC approach to allow better learning outcome of the mid-frequency part of the system. In this simulation, the Q filter is designed to be an ideal zero-phase low-pass filter with cut-off frequency 10 Hz.

The RMS error and max error of using the baseline ILC to learn the smooth impulse are shown in Fig. 7.6, and the tracking result of the final iteration $j = 15$ is shown in Fig. 7.7. The progressive update of the baseline ILC is not enabled yet. It is shown that with the baseline ILC, the tracking error converged, but the final tracking outcome is not accurate enough. There are still phase lag and oscillations after the impulse signal, and the

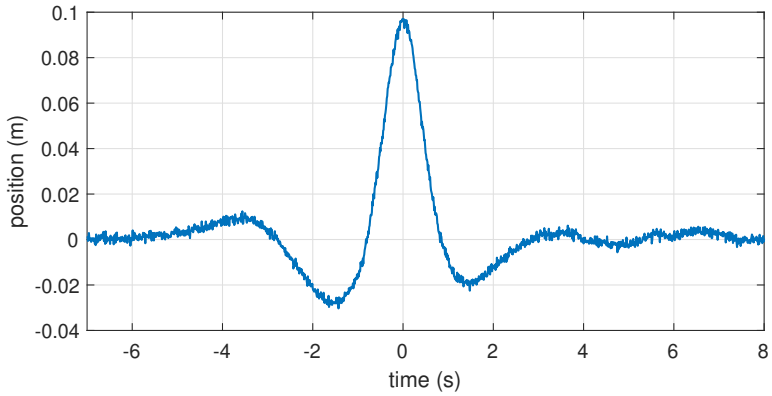


Figure 7.5: Simulation - FIR of the zero phase system adjoint in z direction by reverse time filtering.

convergence rate will be very slow thereafter.

To accelerate the learning of the reference signal, the learning filter of the baseline ILC is updated by the progressive update method. The update is conducted every 5 iterations. And the resulting RMS error and max error are shown in Fig. 7.8. The tracking result of the final iteration $j = 15$ is shown in Fig. 7.9. With the interleaving update of the baseline ILC, the convergence rate is accelerated and the final tracking error is greatly reduced even with the same number of iterations. The same result is expected and has been analyzed in [CT21]. And the input of the last iteration is then used as the FIR of system inverse and to address the downwash problem.

The simulation is then conducted using the FIR of system inverse obtained from the ILCFF with progressive updates to addreass the downwash problem. The trajectory of rotating the pitch angle of the over-actuated platform from 0 to 180 ° and remains at 90 ° for 3 seconds is used (Fig. 6.8), and an ILC using the constructed FIR filter as its learning filter is used. The simulation results of the RMS error and max error are shown in Fig. 7.10, and the tracking outcome in the 10th iteration is shown in Fig. 7.11. Compared with the simulation result of the model-based ILC, the data-driven approach reduced the converged RMS error from 0.02 to 0.005, maximum absolute error from 0.06 to 0.02, and the convergence rate is almost one iteration only.

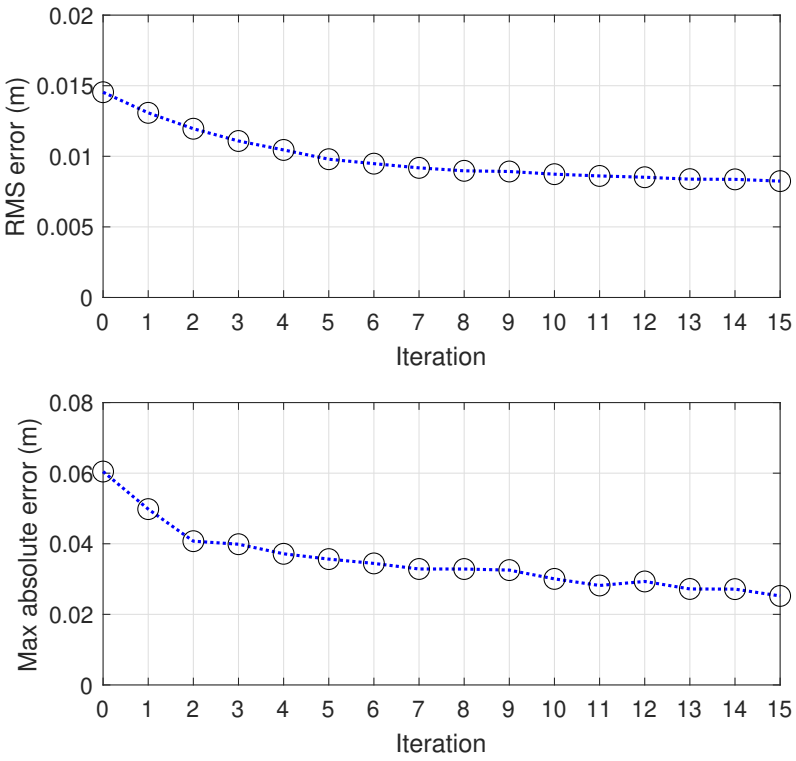


Figure 7.6: Simulation - RMS error and max absolute error of using baseline ILC to learn the smooth impulse on z direction, Q: 10 Hz.

7.3.2 Experiment Results

Real-world experiments are conducted after the effectiveness of ILCFF with progressive updates are verified by simulations. In the experiment, the Q filter is also designed to be an ideal zero-phase low-pass filter with cut-off frequency 10 Hz. To accelerate the learning of the reference signal, the learning filter of the baseline ILC is updated by the progressive update method. The update is conducted every 5 iterations.

The resulting RMS error and max error are shown in Fig. 7.12. The tracking result of the final iteration $j = 15$ is shown in Fig. 7.13. With the interleaving update of the baseline ILC, the convergence rate is accelerated. Right after each progressive update, the error starts to decrease again, which agrees with those shown in the simulation results. The final tracking error is converged to an RMS error of around 0.005 and a max error of around 0.01.

The experiment is then conducted using the FIR of system inverse obtained from the

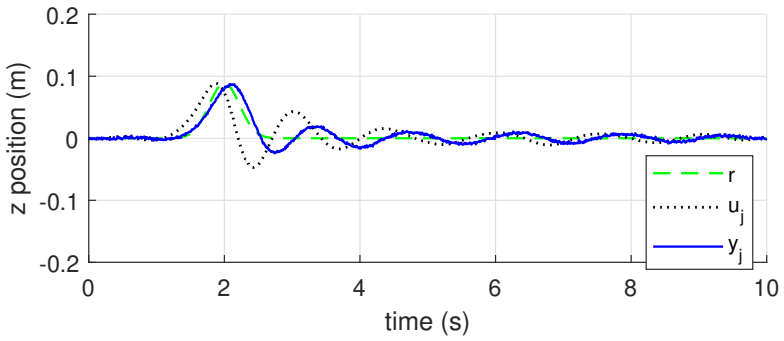


Figure 7.7: Simulation - Tracking result of using baseline ILC to learn the smooth impulse on z direction at $j = 15$.

ILCFF with progressive updates to addreass the downwash problem. The constructed system inverse in FIR format is shown in Fig. 7.14. The simulation results of the RMS error and max error are shown in Fig. 7.10, and the tracking outcome in the 10th iteration is shown in Fig. 7.11. Compared with the simulation result of the model-based ILC, the data-driven approach reduced the converged RMS error from 0.015 to 0.005, maximum absolute error from 0.05 to 0.01, and the convergence rate is almost one iteration only, which demonstrates the learning filter is a good approximation of the system inverse to make the ILC converge fast. Moreover, the Q filter used in the model-based ILC is 1 Hz, which largely limits the performance of the ILC to improve the tracking of signals with higher frequencies. But using the data-driven ILC, the requirement of using the Q filter is greatly released, and can largely improve the performance of the ILC.

7.4 Data-Driven ILC on All 6 DOF

The data-driven ILC algorithm consisting of ILCFF and the progressive update is then applied to all 6 DOF on the over-actuated UAV platform, because it has been demonstrated in Section 5.3 that the system can be seen as containing 6 SISO channels, and the control on each channel doesn't interfere with each other. Applying data-driven ILC on all 6 DOF makes the platform capable of addressing more types of aerodynamic effects and other repeatable disturbances. In this section, simulation results will be omitted because the simulation and

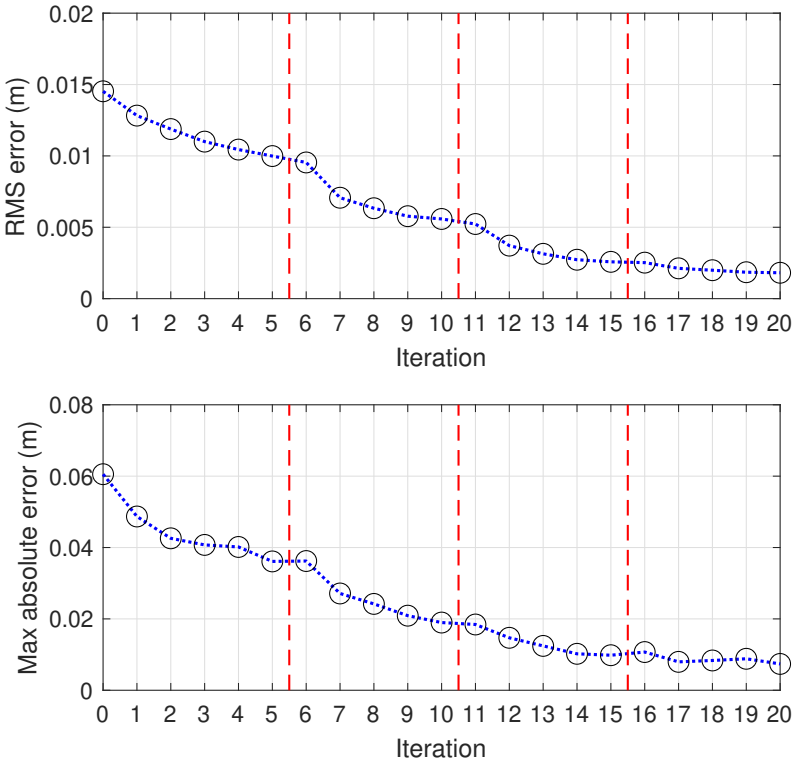


Figure 7.8: Simulation - RMS error and max absolute error of using baseline ILC to learn the smooth impulse with progressive updates every 5 iterations on z direction, Q: 10 Hz.

experiment will show very similar outcome, thanks to the accuracy of the simulation model.

The first step is to use the ILCFF with progressive updates to learn the smooth impulse in Fig. 7.3 on each 6 DOF separately. To avoid actuator saturation, the impulses are separated by 10 seconds with the adjacent one, as shown in Fig. 7.17. Also on this reference, the magnitude of the attitude references are larger than the magnitude of position references, because of the different SNR on position and attitude measurements. Data is also recorded and calculated for each DOF separately. Each DOF only uses the data within its 10-second interval for the ILC, and the input changes within the corresponding interval as well.

The resulting RMS error and max error are shown in Fig. 7.18. The tracking result of the final iteration $j = 15$ is shown in Fig. 7.19. A zoom-in view at the tracking result on x direction is shown in Fig. 7.20. The RMS errors in all the 6 DOF converge to a significantly lower level, and the 6 DOF inputs of the last iteration are then used as the FIR of system inverse and to improve the performance tracking with multiple aerodynamic effects in the

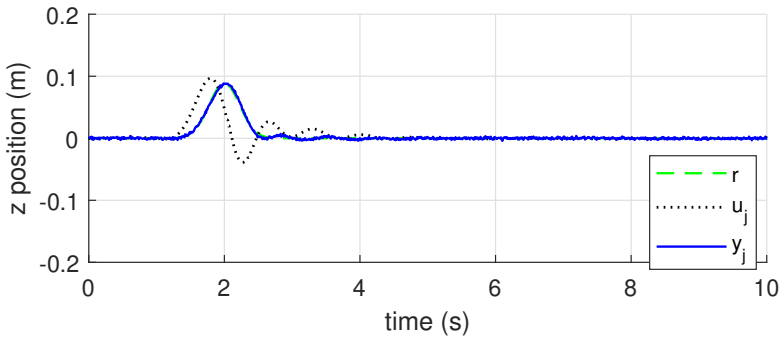


Figure 7.9: Simulation - Tracking result of using baseline ILC to learn the smooth impulse with progressive updates every 5 iterations on z direction at $j = 20$.

following sections.

7.4.1 Tracking of Circular Trajectory

The presented over-actuated UAV platform shows potential in inspections and physical interactions with aircraft components. The circular trajectory is designed to emulate the process of inspecting a turbine engine. The circular trajectory as shown in Fig. 7.21, is a circle in $x - z$ plane with 0.4 m radius. The platform is required to move along the circle, while rotate the body so that the main frame is always facing the center of the circle. During this process, the platform will be subject to the downwash effects. The downwash effects will affect the tracking performance more than on z direction, because the position is also changing in this case.

The over-actuated UAV platform tracking of the circular trajectory in experiment without any ILC compensation is shown in Fig. 7.22. A side view of the tracking result is shown in Fig. 7.23. The position error is obvious on position z direction, and the 3 DOF attitude as shown in the visualization.

ILC with the learning filter constructed by the data-driven approach is applied on this trajectory. The Q filter is also selected to be an ideal low-pass filter with cut-off frequency of 10 Hz. The resulting RMS error and max error are shown in Fig. 7.24. The converged tracking result selected from $j = 6$ is shown in Fig. 7.25. The tracking error converges within

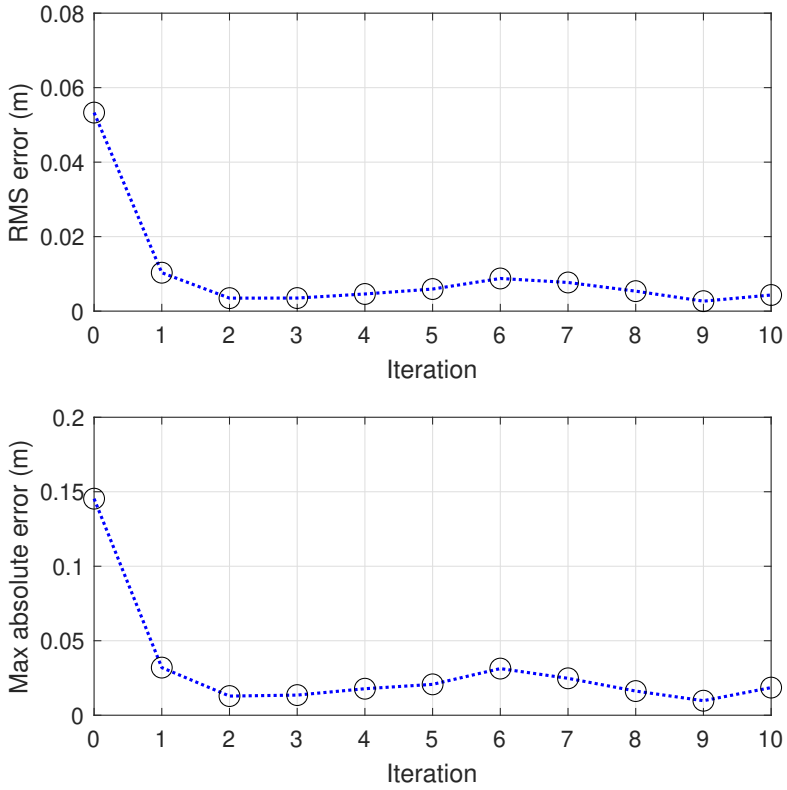


Figure 7.10: Simulation - RMS error and max absolute error of using data-driven ILC on z direction to address the downwash effects in 180 ° pitch rotation, Q: 10 Hz.

the first few iterations. The variations of the errors in later iterations are possibly caused by the saturation of the actuators. As analyzed in Section 4.1, when the over-actuated platform is in vertical flight, the gimbal actuators needs more amount of thrusts to regulate the α joint angle. In such condition, if the platform is controlled to accelerate upwards, the nominal magnitude of thrust is higher, and the low-level thrust of each propeller will be more likely to reach its saturation. Fig. 7.25 also shows that the converged tracking has larger error is at around 270 ° pitch angle from around 25 to 30 seconds. This can be solved by switching to a more powerful gimbal actuator with quadcopter in the future works.

7.4.2 Regulation Under Wind Field Disturbance

The other application case to demonstrate the effectiveness of the data-driven ILC algorithm is flying through a wind field. As shown in Fig. 7.26, the over-actuated UAV platform

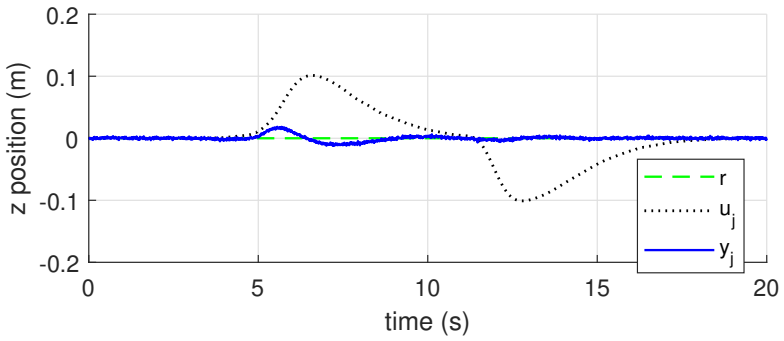


Figure 7.11: Simulation - Tracking result of using data-driven ILC on z direction to address the downwash effects in 180 ° pitch rotation at $j = 10$.

is controlled to fly through a wind field. The wind field in the experiment is realized by an air blower that generates wind forces in x and z directions. The platform is flying along y direction and maintains horizontal hovering. The platform will pass through the wind field twice, going back and forth. Without ILC, the tracking error is most significant in position x and y directions, and roll, yaw directions. The platform flies along y direction and the tracking error is supposed to be large; the error in the other directions is obviously caused by the wind disturbance.

ILC with the learning filter constructed by the data-driven approach is applied on this trajectory. The Q filter is also selected to be an ideal low-pass filter with cut-off frequency of 10 Hz. The resulting RMS error and max error are shown in Fig. 7.28. The error drops in position x direction and roll, yaw directions significantly. The converged tracking result selected from $j = 1$ is shown in Fig. 7.29. With the platform maintains horizontal hovering during the tracking process, the tracking error converges within only one iteration, demonstrating that the learning filter is a good approximation of the system at horizontal hovering. The variations of the errors in later iterations are possibly caused by the turbulent flow generated by the blower, which is not identical in different iterations, and the disturbances that do not repeat will reduce the performance of an ILC.

The aforementioned applications of the data-driven ILC, including vertical flight under downwash effects, circular trajectory tracking and flying over wind fields, can be used to construct an accurate model of the aerodynamic effects under these trajectories. By using

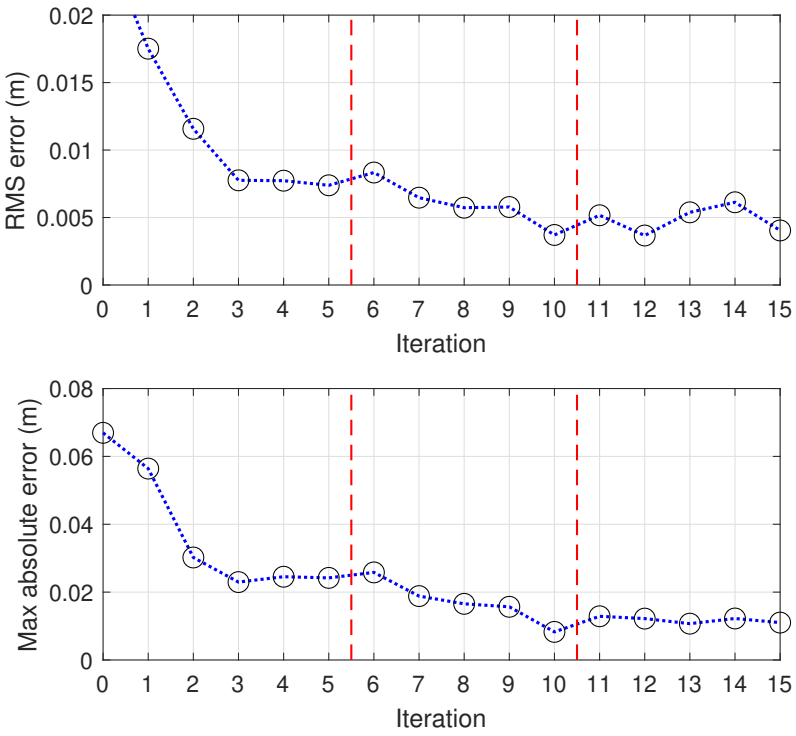


Figure 7.12: Experiment - RMS error and max absolute error of using baseline ILC to learn the smooth impulse with progressive updates every 5 iterations on z direction, Q: 10 Hz.

the converged inputs from ILC, the feedforward force and torque command can be related to the pose and position of the platform, and thus the magnitude of force and torque needed to compensate the aerodynamic effects at every pose and position can be calculated. This will be an invaluable future direction of the ILC application on the over-actuated platform.

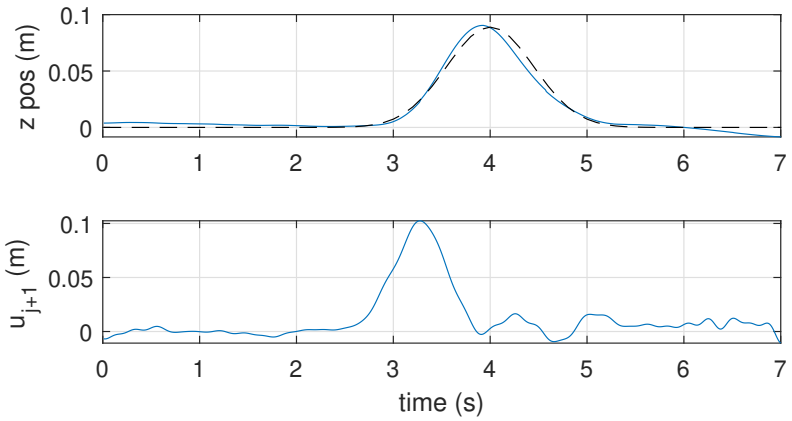


Figure 7.13: Experiment - Tracking result of using baseline ILC to learn the smooth impulse with progressive updates every 5 iterations on z direction at $j = 15$.

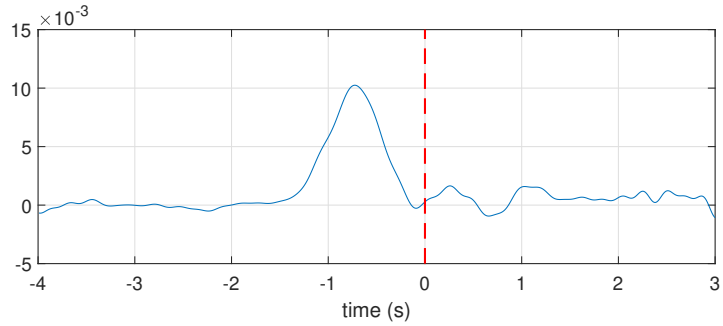


Figure 7.14: Experiment - FIR filter of system inverse constructed from ILCFF with progressive update process.

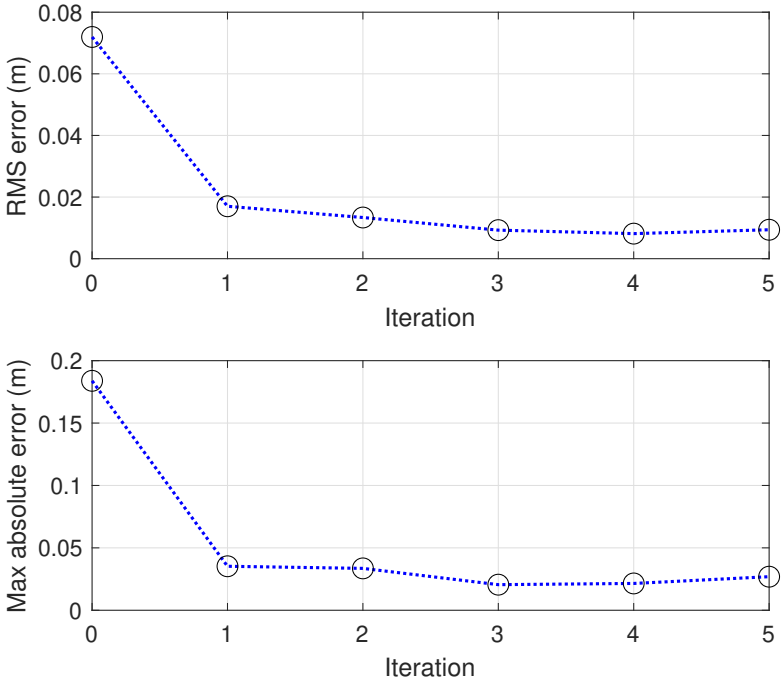


Figure 7.15: Experiment - RMS error and max absolute error of using data-driven ILC on z direction to address the downwash effects in 180 ° pitch rotation, Q: 10 Hz.

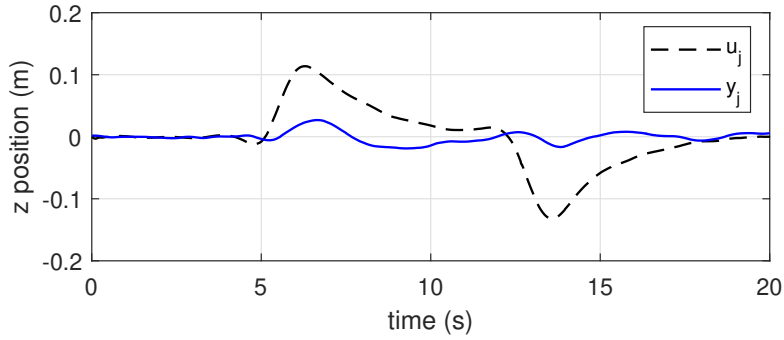


Figure 7.16: Experiment - Tracking result of using data-driven ILC on z direction to address the downwash effects in 180 ° pitch rotation at $j = 5$.

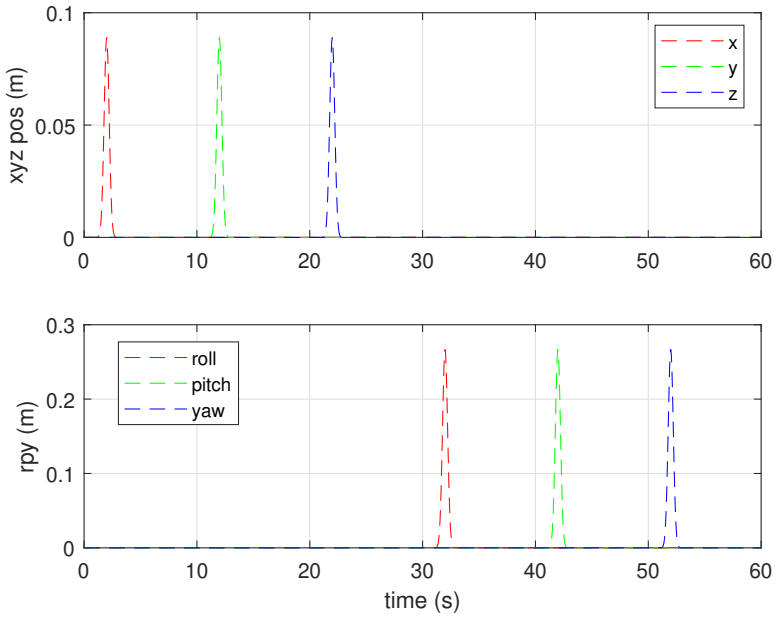


Figure 7.17: Reference of smooth impulses on all 6 DOF, triggered at separate time intervals.

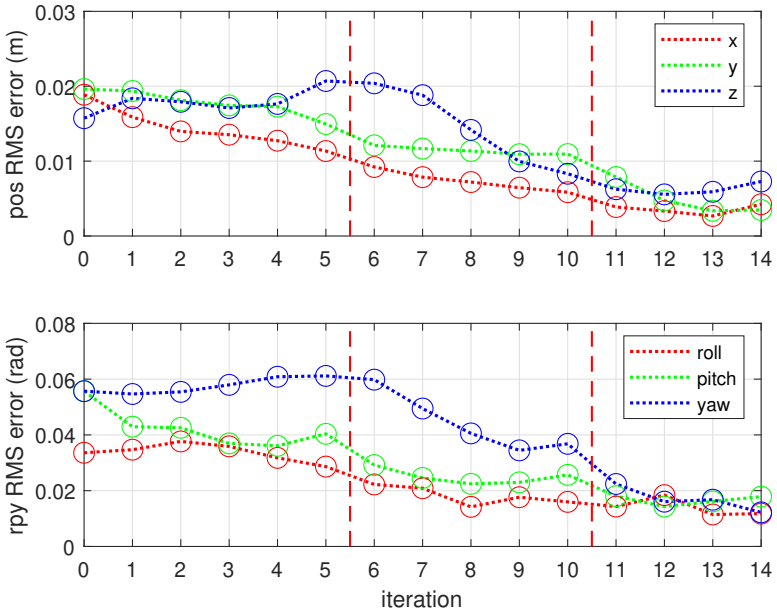


Figure 7.18: Experiment - RMS error of using baseline ILC to learn the smooth impulse with progressive updates every 5 iterations on all 6 DOF.

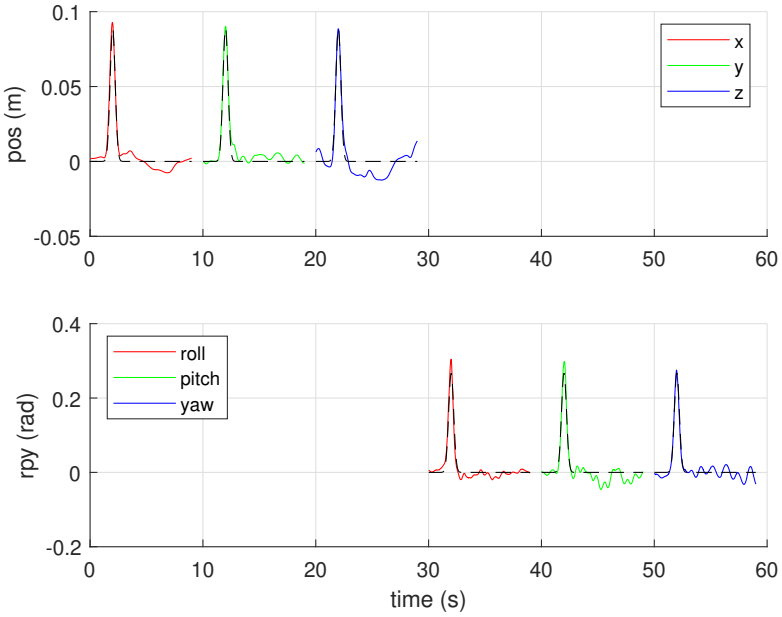


Figure 7.19: Experiment - Tracking result of using baseline ILC to learn the smooth impulse with progressive updates every 5 iterations on all 6 DOF at $j = 14$.

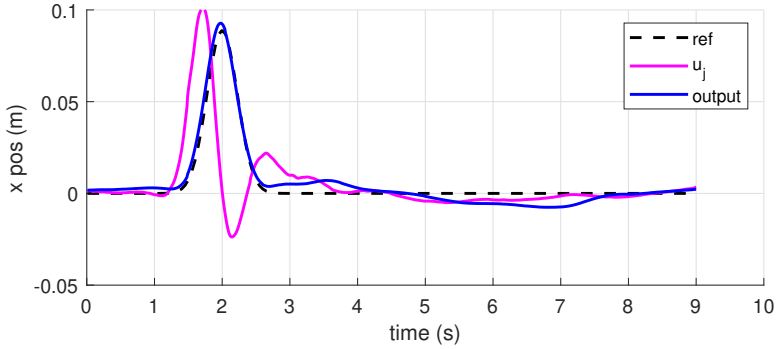


Figure 7.20: Experiment - a closer look at the tracking result of 6 DOF ILCFF on x direction at $j = 14$.

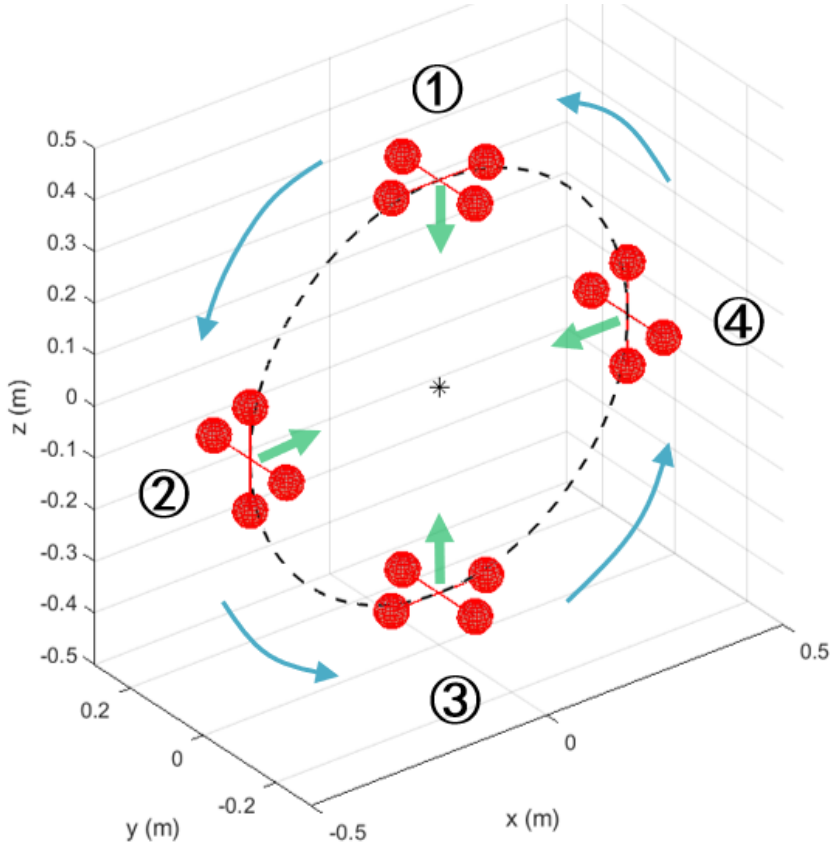


Figure 7.21: Circular trajectory.

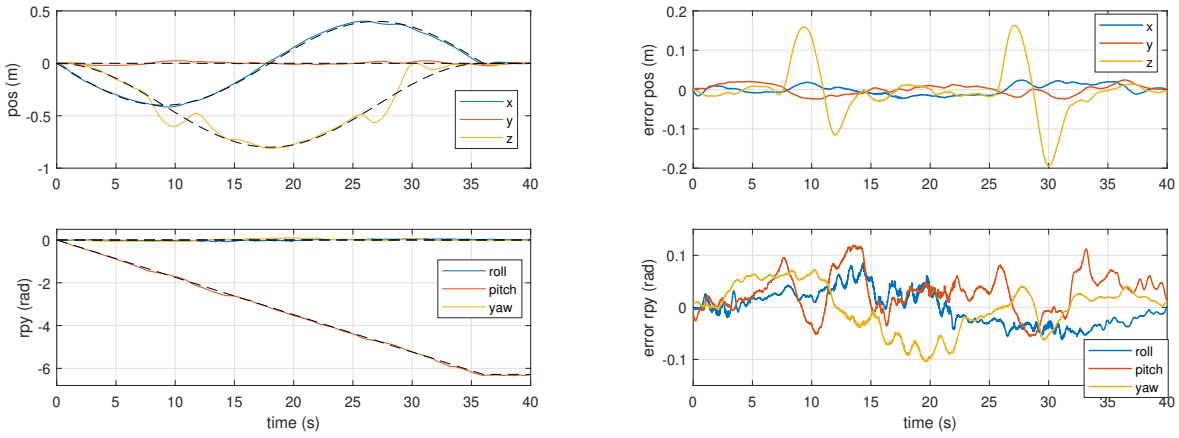


Figure 7.22: Experiment result of circular trajectory tracking without ILC.

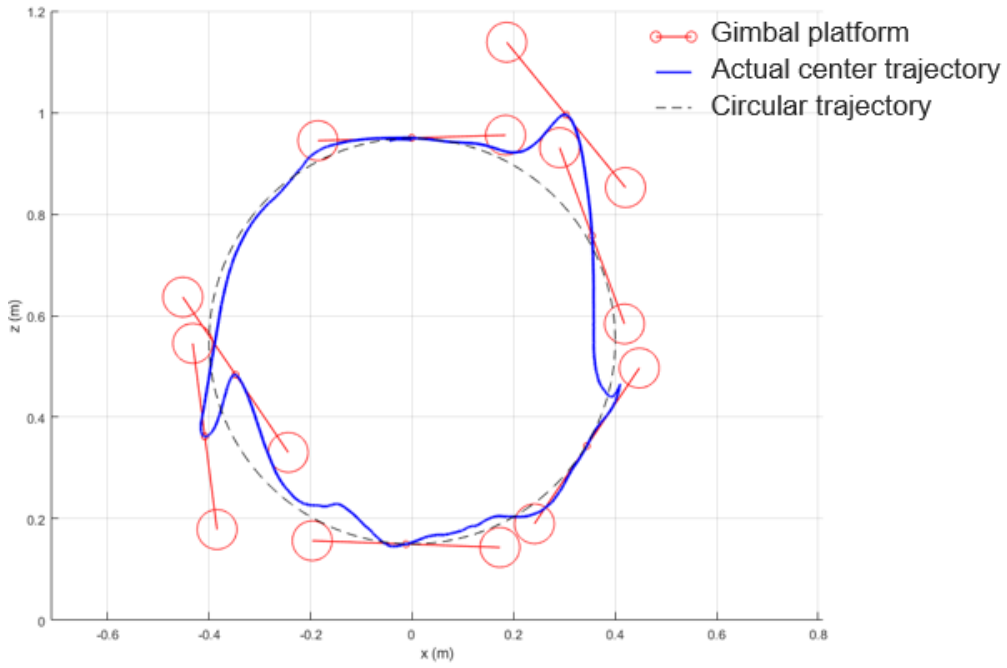


Figure 7.23: Tracking of the circular trajectory, side view.

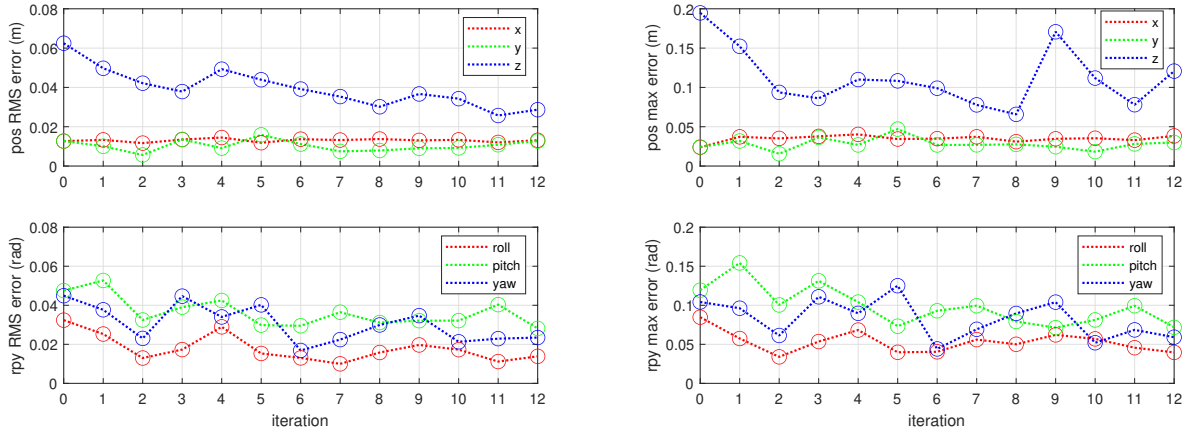


Figure 7.24: Experiment - Tracking result of using data-driven ILC on all 6 DOF to improve the circular trajectory tracking performance.

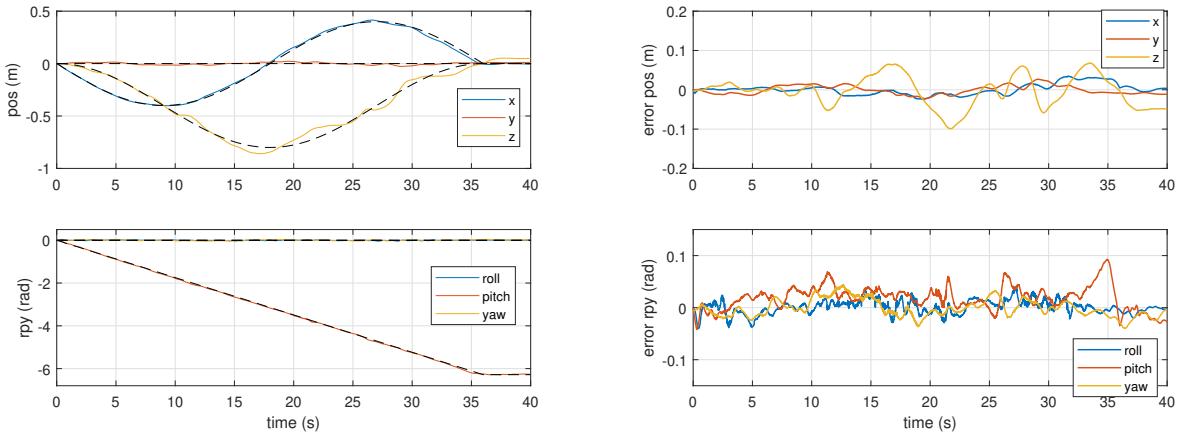


Figure 7.25: Experiment - Converged result of circular trajectory tracking using data-driven ILC on all 6 DOF.

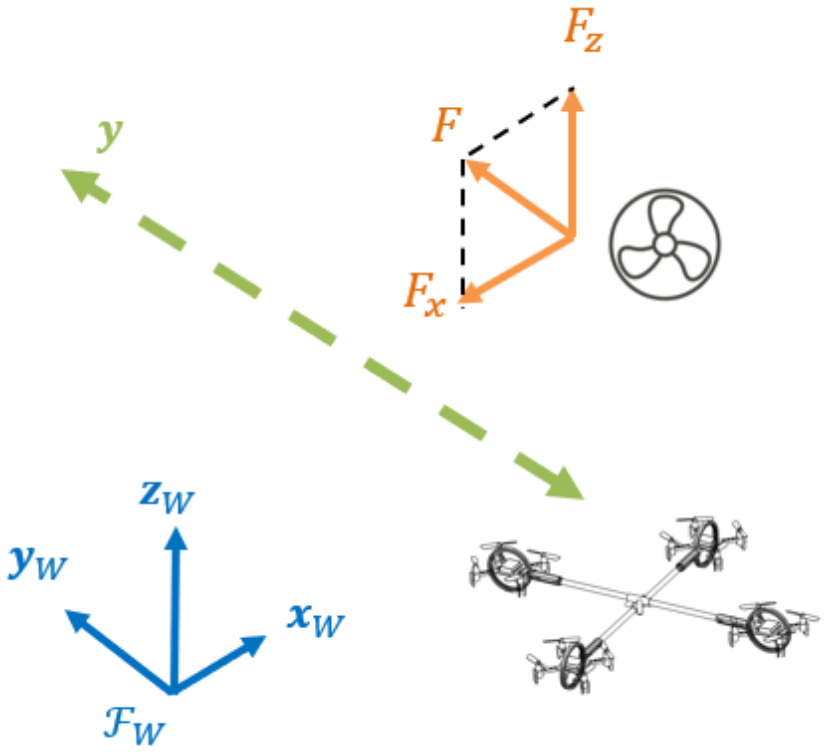


Figure 7.26: Over-actuated platform flying through a wind field.

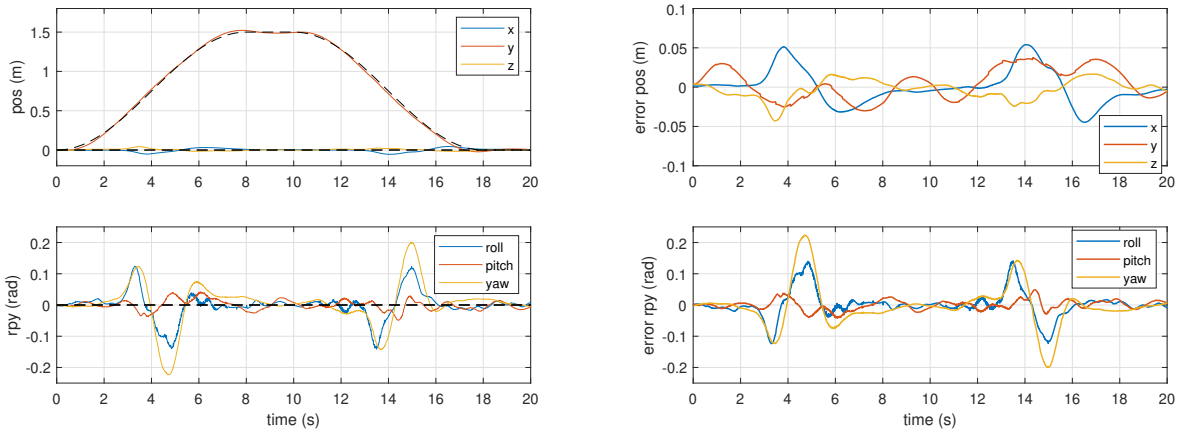


Figure 7.27: Experiment result of wind field trajectory tracking without ILC.

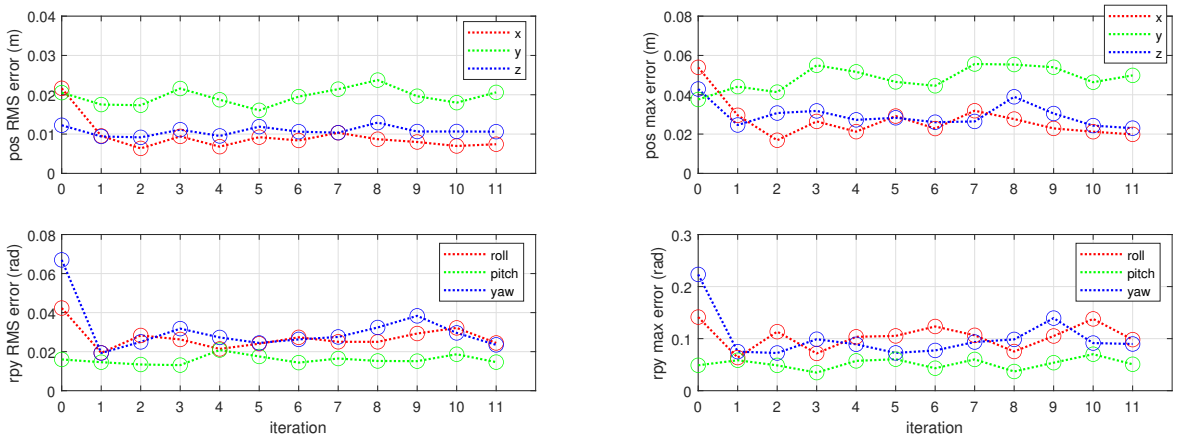


Figure 7.28: Experiment - Tracking result of using data-driven ILC on all 6 DOF to improve the wind field trajectory tracking performance.

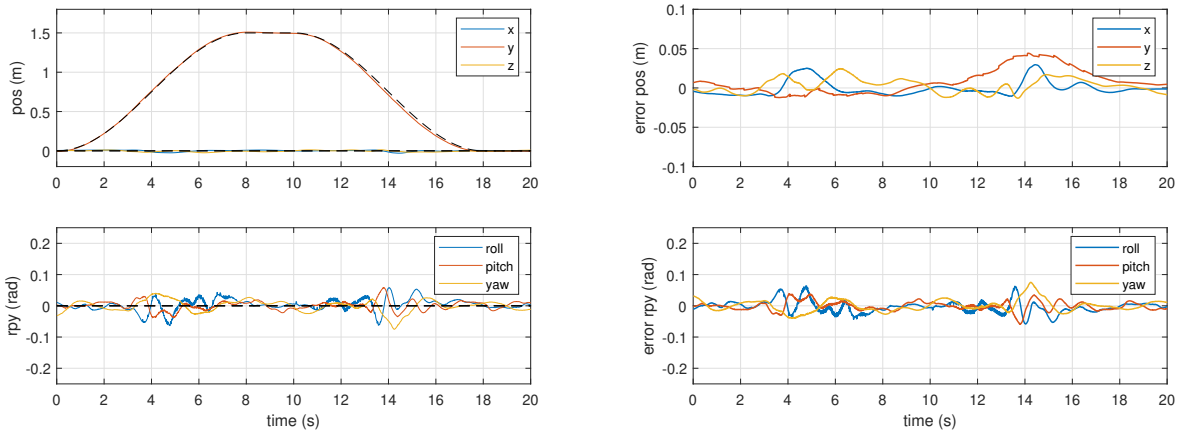


Figure 7.29: Experiment - Converged result of wind field trajectory tracking using data-driven ILC on all 6 DOF.

CHAPTER 8

Conclusions

This thesis have proposed a novel over-actuated aerial vehicle using quadcopters and passive gimbals as omni-directional vectoring thrust actuators. Both the thrust actuators and the UAV platform have no rotational constraints and can reach any arbitrary orientation. Hierarchical controller based on simple yet accurate dynamics model with singularity-handling solutions are constructed. High-fidelity simulation model is built, and reliable software architecture is established. The ability to track independent 6-DOF trajectories and disturbance rejection is also demonstrated. To the best of our knowledge, the presented platform is the first UAV platform to experimentally achieve any arbitrary attitude with uniformly high thrust efficiency.

This thesis have presented the applications of both model-based and data-driven ILC algorithms on the over-actuated aerial platform and shows great improvements against the aerodynamic effects. It has been demonstrated that the proposed UAV platform can have six independent control channels. The formulation of the closed-loop dynamics has enabled add-on feedforward and/or feedback controllers to be implemented on any of the control channels. Simulations and real-world experiments have demonstrated the authenticity of the SISO format of the closed loop system and the effectiveness of the ILC algorithms against various aerodynamic effects.

Some future research directions are given related to the proposed over-actuated UAV platform: The gimbal actuator can be used as a modular unit to build aerial platforms for more sophisticated applications. Singularity issue introduced by the use of gimbal actuators can be addressed in a more systematic manner. The aerodynamic effects involved on the

platform by the interaction of the air flows by different propellers can be addressed and compensated.

Future directions related to the ILC applications: The ILC and the converged tracking results can be used to model the aerodynamic effects and/or design feedforward compensators to improve the system performance without repeating the same trajectory over the ILC framework. The data-driven ILC can also be used to identify the low-level dynamics of the over-actuated platform which involves nonlinearity and friction.

BIBLIOGRAPHY

[ABN80] James Sacra Albus, Anthony J Barbera, and Roger N Nagel. *Theory and practice of hierarchical control*. National Bureau of Standards, 1980.

[ÅE71] Karl Johan Åström and Peter Eykhoff. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

[AKM84] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. *Journal of Robotic systems*, 1(2):123–140, 1984.

[AML20] Robert B Anderson, Julius A Marshall, and Andrea L'Afflitto. Constrained robust model reference adaptive control of a tilt-rotor quadcopter pulling an unmodeled cart. *IEEE Transactions on Aerospace and Electronic Systems*, 57(1):39–54, 2020.

[AZ20] Revant Adlakha and Minghui Zheng. An optimization-based iterative learning control design method for uav's trajectory tracking. In *2020 American Control Conference (ACC)*, pages 1353–1359. IEEE, 2020.

[BAB+15] Georg Bareth, Helge Aasen, Juliane Bendig, Martin Leon Gnyp, Andreas Bolten, András Jung, René Michels, and Jussi Soukkamäki. Low-weight and uav-based hyperspectral full-frame cameras for monitoring crops: Spectral comparison with portable spectroradiometer measurements. *Photogrammetrie-Fernerkundung-Geoinformation*, 2015(1):69–79, 2015.

[BBP+19] Karen Bodie, Maximilian Brunner, Michael Pantic, Stefan Walser, Patrick Pfändler, Ueli Angst, Roland Siegwart, and Juan Nieto. An omnidirectional aerial manipulation platform for contact-based inspection. *arXiv preprint arXiv:1905.03502*, 2019.

[BCMC14] John Bartholomew, Andrew Calway, and Walterio Mayol-Cuevas. Learning to predict obstacle aerodynamics from depth images for micro air vehicles. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4967–4973. IEEE, 2014.

[BCSM16] JXJ Bannwarth, ZJ Chen, KA Stol, and BA MacDonald. Disturbance accomodation control for wind rejection of a quadcopter. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 695–701. IEEE, 2016.

[BK13] Kira Barton and Derek Kingston. Systematic surveillance for uavs: A feedforward iterative learning control approach. In *2013 American Control Conference*, pages 5917–5922. IEEE, 2013.

[BMSP09] Pierre-Jean Bristeau, Philippe Martin, Erwan Salaün, and Nicolas Petit. The role of propeller aerodynamics in the model of a quadrotor uav. In *2009 European control conference (ECC)*, pages 683–688. IEEE, 2009.

[BPA12] Jeffrey A Butterworth, Lucy Y Pao, and Daniel Y Abramovitch. Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems. *Mechatronics*, 22(5):577–587, 2012.

[BTA06] Douglas A Bristow, Marina Tharayil, and Andrew G Alleyne. A survey of iterative learning control. *IEEE control systems magazine*, 26(3):96–114, 2006.

[BTKS18] Karen Bodie, Zachary Taylor, Mina Kamel, and Roland Siegwart. Towards efficient full pose omnidirectionality with overactuated mavs. In *International Symposium on Experimental Robotics*, pages 85–95. Springer, 2018.

[CBQ21] Darius J Carter, Lauren Bouchard, and Daniel B Quinn. Influence of the ground, ceiling, and sidewall on micro-quadrotors. *AIAA Journal*, 59(4):1398–1405, 2021.

[CKBM06] David W Casbeer, Derek B Kingston, Randal W Beard, and Timothy W McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6):351–360, 2006.

[Cra09] John J Craig. *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.

[CRT19] Cheng-Wei Chen, Sandeep Rai, and Tsu-Chin Tsao. Iterative learning of dynamic inverse filters for feedforward tracking control. *IEEE/ASME Transactions on Mechatronics*, 25(1):349–359, 2019.

[CSM11] Nalin A Chaturvedi, Amit K Sanyal, and N Harris McClamroch. Rigid-body attitude control. *IEEE control systems magazine*, 31(3):30–51, 2011.

[CT21] Cheng-Wei Chen and Tsu-Chin Tsao. Accelerated convergence interleaving iterative learning control and inverse dynamics identification. *IEEE Transactions on Control Systems Technology*, 30(1):45–56, 2021.

[dOMPR12] Sebastian d'Oleire Oltmanns, Irene Marzolff, Klaus Peter, and Johannes Ries. Unmanned aerial vehicle (uav) for monitoring soil erosion in morocco. *Remote Sensing*, 4(11):3390–3416, 2012.

[DQR+18] Xunhua Dai, Quan Quan, Jinrui Ren, Zhiyu Xi, and Kai-Yuan Cai. Terminal iterative learning control for autonomous aerial refueling under aerodynamic disturbances. *Journal of Guidance, Control, and Dynamics*, 41(7):1577–1584, 2018.

[dRO19] Robin de Rozario and Tom Oomen. Data-driven iterative inversion-based control: Achieving robustness through nonlinear learning. *Automatica*, 107:342–352, 2019.

[DSJ19] Eric Dyer, Shahin Sirouspour, and Mohammad Jafarinasab. Energy optimal control allocation in a redundantly actuated omnidirectional uav. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5316–5322. IEEE, 2019.

[DYO18] Molong Duan, Deokkyun Yoon, and Chinedum E Okwudire. A limited-preview filtered b-spline approach to tracking control–with application to vibration-induced error compensation of a 3d printer. *Mechatronics*, 56:287–296, 2018.

[FL13] Li Fu and Pengfei Li. The research survey of system identification method. In *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 397–401. IEEE, 2013.

[FLW20] Husam A Foudeh, Patrick Luk, and James Whidborne. Application of norm optimal iterative learning control to quadrotor unmanned aerial vehicle for monitoring overhead power system. *Energies*, 13(12):3223, 2020.

[FMKK11] Jonathan Fink, Nathan Michael, Soonkyum Kim, and Vijay Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *The Int. Journal of Robotics Research*, 30(3):324–334, 2011.

[GMG+08] Michael A Goodrich, Bryan S Morse, Damon Gerhardt, Joseph L Cooper, Morgan Quigley, Julie A Adams, and Curtis Humphrey. Supporting wilderness search and rescue using a camera-equipped mini uav. *Journal of Field Robotics*, 25(1-2):89–110, 2008.

[GN01] Svante Gunnarsson and Mikael Norrlöf. On the design of ilc algorithms using optimization. *Automatica*, 37(12):2011–2016, 2001.

[GSW+17] Wojciech Giernacki, Mateusz Skwierczyński, Wojciech Witwicki, Paweł Wroński, and Piotr Kozierski. Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 37–42. IEEE, 2017.

[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):061013, 2018.

[GTM94] Eric Gross, Masayoshi Tomizuka, and William Messner. Cancellation of discrete time unstable zeros by feedforward control. *Journal of Dynamic Systems, Measurement, and Control*, 1994.

[HL08] Andrew Hazell and David JN Limebeer. An efficient algorithm for discrete-time h∞ preview control. *Automatica*, 44(9):2441–2448, 2008.

[HZX14] Minh Quan Huynh, Weihua Zhao, and Lihua Xie. L1 adaptive control for quadcopter: Design and implementation, ieee control automation robotics & vision (icarcv). In *2014 13th International Conference, Singapore*, 2014.

[IGGL20] Davide Invernizzi, Mattia Giurato, Paolo Gattazzo, and Marco Lovera. Comparison of control methods for trajectory tracking in fully actuated unmanned aerial vehicles. *IEEE Transactions on Control Systems Technology*, 29(3):1147–1160, 2020.

[KDDCSH+18] Helge Klippstein, Alejandro Diaz De Cerio Sanchez, Hany Hassanin, Yahya Zweiri, and Lakmal Seneviratne. Fused deposition modeling for unmanned aerial vehicles (uavs): a review. *Advanced Engineering Materials*, 20(2):1700552, 2018.

[KJ74] J Kormylo and V Jain. Two-pass recursive digital filter with zero phase shift. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 22(5):384–387, 1974.

[KNC13] Waqas Khan, Meyer Nahon, and Ryan Caverly. Propeller slipstream model for small unmanned aerial vehicles. In *AIAA modeling and simulation technologies (MST) conference*, page 4907, 2013.

[KSA02] Koji Kinosita, Takuya Sogo, and Norihiko Adachi. Iterative learning control using adjoint systems and stable inversion. *Asian Journal of Control*, 4(1):60–67, 2002.

[KVE+18] 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.

[KZ12] Kyong-Soo Kim and Qingze Zou. A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems. *IEEE/ASME Transactions on Mechatronics*, 18(6):1767–1777, 2012.

[LAN11] Jack W Langelaan, Nicholas Alley, and James Neidhoefer. Wind field estimation for small unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 34(4):1016–1030, 2011.

[Lie77] Alain Liegeois. Automatic supervisory control of the configuration and behaviour of multibody mechanisms. *IEEE Transactions on systems, man and cybernetics*, 7(12):868–871, 1977.

[LLK+20] Seung Jae Lee, Dongjae Lee, Junha Kim, Dabin Kim, Inkyu Jang, and H Jin Kim. Fully-actuated autonomous flight of thruster-tilting multirotor. *IEEE/ASME Transactions on Mechatronics*, 2020.

[ML21] Salvatore Meraglia and Marco Lovera. Smoother-based iterative learning control for uav trajectory tracking. *IEEE Control Systems Letters*, 6:1501–1506, 2021.

[MMLK10] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. The grasp multiple micro-uav testbed. *IEEE Robotics & Automation Magazine*, 17(3):56–65, 2010.

[Moo12] Kevin L Moore. Iterative learning control for deterministic systems. 2012.

[NG02] Mikael Norrlöf and Svante Gunnarsson. Time and frequency domain convergence properties in iterative learning control. *International Journal of Control*, 75(14):1114–1126, 2002.

[NPC19] Michelangelo Nigro, Francesco Pierri, and Fabrizio Caccavale. Preliminary design, modeling and control of a fully actuated quadrotor uav. In *2019 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, pages 1108–1116. IEEE, 2019.

[NPPL18] 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.

[PD09] Oliver Purwin and Raffaello D'Andrea. Performing aggressive maneuvers using iterative learning control. In *2009 IEEE international conference on robotics and automation*, pages 1731–1736. IEEE, 2009.

[PLA+18] 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.

[PO13] Pong-in Pipatpaibul and PR Ouyang. Application of online iterative learning tracking control for quadrotor uavs. *International Scholarly Research Notices*, 2013, 2013.

[PRY+21] Chen-Huan Pi, Lecheng Ruan, Pengkang Yu, Yao Su, Stone Cheng, and Tsu-Chin Tsao. A simple six degree-of-freedom aerial vehicle built on quadcopters. In *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pages 329–334. IEEE, 2021.

[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.

[RCS19a] 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.

[RCS19b] Ramy Rashad, Federico Califano, and Stefano 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.

[RLO18] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 3(3):1957–1964, 2018.

[RMP+17] 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)*, pages 5190–5195. IEEE, 2017.

[RPL09] Brian P Rigney, Lucy Y Pao, and Dale A Lawrence. Nonminimum phase dynamic inversion for settle time applications. *IEEE Transactions on Control Systems Technology*, 17(5):989–1005, 2009.

[RRBF15] 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)*, pages 4006–4013. IEEE, 2015.

[Rua20] Lecheng Ruan. *Independent Position and Attitude Control on Multirotor Aerial Platforms*. Ph.D. dissertation, Univ. of California, Los Angeles, 2020.

[SA94] T Sogo and N Adachi. A design method of the learning control algorithm for linear systems. In *Proc. Jpn-USA Symp. Flexible Autom*, pages 681–684, 1994.

[SA13] F. Senkul and E. Altug. Modeling and control of a novel tilt — roll rotor quadrotor uav. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1071–1076, 2013.

[SCHO17] Pedro Sanchez-Cuevas, Guillermo Heredia, and Anibal Ollero. Characterization of the aerodynamic ground effect and its influence in multirotor control. *International Journal of Aerospace Engineering*, 2017, 2017.

[SHV+06] Mark W Spong, Seth Hutchinson, Mathukumalli Vidyasagar, et al. *Robot modeling and control*, volume 3. wiley New York, 2006.

[SRY+21] Yao Su, Lecheng Ruan, Pengkang Yu, Chen-Huan Pi, Matthew Gerber, and Tsu-Chin Tsao. A fast and efficient attitude control algorithm of a tilt-rotor aerial platform using inputs redundancies. *IEEE Robotics and Automation Letters*, 2021.

[SST+20] Dario Sanalitro, Heitor J Savino, Marco Tognon, Juan Cortés, and Antonio Franchi. Full-pose manipulation control of a cable-suspended load with multiple uavs under uncertainties. *IEEE Robotics and Automation Letters*, 5(2):2185–2191, 2020.

[Su21] Yao Su. *Compensation and Control Allocation with Input Saturation Limits and Rotor Faults for Multi-Rotor Copters with Redundant Actuations*. Ph.D. dissertation, Univ. of California, Los Angeles, 2021.

[SYG+21] Yao Su, Pengkang Yu, Matthew J Gerber, Lecheng Ruan, and Tsu-Chin Tsao. Nullspace-based control allocation of overactuated uav platforms. *IEEE Robotics and Automation Letters*, 6(4):8094–8101, 2021.

[Tom87] Masayoshi Tomizuka. Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*, 1987.

[TT87] Tsu-Chin Tsao and Masayoshi Tomizuka. Adaptive zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*, 1987.

[TT88] Tsu-Chin Tsao and Masayoshi Tomizuka. Adaptive and repetitive digital control algorithms for noncircular machining. In *1988 American Control Conference*, pages 115–120. IEEE, 1988.

[TT15] Kuo-Tai Teng and Tsu-Chin Tsao. A comparison of inversion based iterative learning control algorithms. In *2015 American Control Conference (ACC)*, pages 3564–3569. IEEE, 2015.

[vZO18] Jurgen van Zundert and Tom Oomen. On inversion-based approaches for feedforward and ilc. *Mechatronics*, 50:282–291, 2018.

[Wam86] Charles W Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):93–101, 1986.

[WW09] Steven Waslander and Carlos Wang. Wind disturbance estimation and rejection for quadrotor position control. In *AIAA Infotech@ Aerospace conference and AIAA unmanned... Unlimited conference*, page 1983, 2009.

[WWW+18] Tao Wang, Jianqin Wang, Chao Wu, Min Zhao, and Tong Ge. Disturbance-rejection control for the hover and transition modes of a negative-buoyancy quad tilt-rotor autonomous underwater vehicle. *Applied Sciences*, 8(12):2459, 2018.

[WYZ14] Danwei Wang, Yongqiang Ye, and Bin Zhang. Reverse time filtering based ilc. In *Practical Iterative Learning Control With Frequency Domain Design and Sampled Data Implementation*, pages 75–102. Springer, 2014.

[WZWT16] Cong Wang, Minghui Zheng, Zining Wang, and Masayoshi Tomizuka. Robust two-degree-of-freedom iterative learning control for flexibility compensation of industrial robot manipulators. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2381–2386. IEEE, 2016.

[YDBJ+17] Seokkwan Yoon, P Ventura Diaz, D Douglas Boyd Jr, William M Chan, and Colin R Theodore. Computational aerodynamic modeling of small quadcopter vehicles. In *American Helicopter Society (AHS) 73rd Annual Forum Fort Worth, Texas*, 2017.

[YSG+21] Pengkang Yu, Yao Su, Matthew J Gerber, Lecheng Ruan, and Tsu-Chin Tsao. An over-actuated multi-rotor aerial vehicle with unconstrained attitude angles and high thrust efficiencies. *IEEE Robotics and Automation Letters*, 6(4):6828–6835, 2021.

[YSPA15] Derrick Yeo, Elena Shrestha, Derek A Paley, and Ella M Atkins. An empirical model of rotorcrafy uav downwash for disturbance localization and avoidance. In *AIAA Atmospheric Flight Mechanics Conference*, page 1685, 2015.

[YSPS15] Derrick Yeo, Nitin Sydney, Derek A Paley, and Donald Sofge. Onboard flow sensing for downwash detection and avoidance with a small quadrotor helicopter. In *AIAA Guidance, Navigation, and Control Conference*, page 1769, 2015.

[YW72] Peter Colin Young and JC Willems. An approach to the linear multivariable servomechanism problem. *International journal of control*, 15(5):961–979, 1972.

[YZB+17] Yuzhe Yang, Zijie Zheng, Kaigui Bian, Lingyang Song, and Zhu Han. Real-time profiling of fine-grained air quality index distribution using uav sensing. *IEEE Internet of Things Journal*, 5(1):186–198, 2017.