Développement d'un robot dirigeable pour opération intérieur

Yue Wang

N° d'ordre : 374

CENTRALE LILLE

THESE

Présentée en vue

d'obtenir le grade de

DOCTEUR

En

Spécialité : Automatique, Génie Informatique,
Traitement du Signal et des Images

Par

Yue WANG

DOCTORAT DELIVRE PAR CENTRALE LILLE

Titre de la thèse :

Développement d'un robot dirigeable pour opération en intérieur

Development of a blimp robot for indoor operation

Soutenue le 15 Mars 2019 devant le jury d'examen :

Président :	Philippe FRAISSE	Professeur, Université de Montpellier
Rapporteur :	Nicolas MARCHAND	Directeur de Recherche CNRS, GIPSA-lab
Rapporteur :	Sophie SAKKA	Maître de conférences, HDR, Centrale Nantes
Invité :	Dominique FARCY	Chercheur, ONERA Lille
Directeur de thèse :	Wilfrid PERRUQUETTI	Professeur, Centrale Lille
Co-Directeur de thèse :	Denis EFIMOV	Chargé de Recherche INRIA, HDR, INRIA Lille
Co-Directeur de thèse :	Gang ZHENG	Chargé de Recherche INRIA, HDR, INRIA Lille

Thèse préparée dans le Centre de Recherche en Informatique, Signal et Automatique de Lille,
CRIStAL, CNRS UMR 9189

Ecole Doctorale SPI 072

Order Number: 374

CENTRALE LILLE

THESIS

Presented in order to obtain the grade of

DOCTOR

In

Specialty: Control, Computer Science,

Signal and Image Processing

By

Yue WANG

DOCTORATE DELIVERED BY CENTRALE LILLE

Title of the thesis:

Development of a blimp robot for indoor operation

Defended on March 15, 2019 before the committee:

President:	Philippe FRAISSE	Professor, University of Montpellier
Referee:	Nicolas MARCHAND	CNRS Senior Researcher, GIPSA-lab
Referee:	Sophie SAKKA	Associate Professor, HDR, Centrale Nantes
Guest:	Dominique FARCY	Researcher, ONERA Lille
Supervisor:	Wilfrid PERRUQUETTI	Professor, Centrale Lille
Co- Supervisor:	Denis EFIMOV	INRIA Researcher, HDR, INRIA Lille
Co- Supervisor:	Gang ZHENG	INRIA Researcher, HDR, INRIA Lille

Thesis prepared in the Centre de Recherche en Informatique, Signal et Automatique de Lille,
CRIStAL, CNRS UMR 9189

Doctoral School SPI 072

Cette thèse a été préparée dans les laboratoires suivants:

INRIA Lille - Nord Europe

Parc scientifique de la Haute-Borne
40, avenue Halley - Bât A - Park Plaza
59650 Villeneuve d'Ascq - France

☎ (33)(0)3 59 57 78 00
Web Site https://www.inria.fr/centre/lille

Centrale Lille

Cité Scientifique - CS 20048
59651 Villeneuve d'Ascq - France

☎ 33 (0)3 20 33 54 87
Web Site https://centralelille.fr/

CRIStAL UMR 9189 CNRS

University Lille 1
Bâtiment M3 extension
Avenue Carl Gauss
59655 Villeneuve d'Ascq - France

☎ (33)(0)3 28 77 85 41
Web Site https://www.cristal.univ-lille.fr/

Table of Contents

List of Figures

List of Tables

Acronyms

A | B | C | D | E | F | G | H | I | J | L | M | N | P | R | S | T | U | V

A

AI Artificial Intelligence.
ANN Artificial Neural Network.

B

BDS BeiDou Navigation Statellite System.

C

CB Center of Buoyancy.
CCD Charge-Coupled Device.
CFD Computational Fluid Dynamics.
CG Center of Gravity.
CLF Control Lyapunov Function.
CMOS Complementary Metal-Oxide Semiconductor.
CV Center of Volume.

D

DC Direct Current.
DOF degree of freedom.

E

EKF Extended Kalman Filter.

F

FT Finite Time.

G

GLONASS Globalnaya navigatsionnaya sputnikovaya sistema (Russian).
GNSS Glocal Navigation Satellite System.
GP Gaussian Process.
GPIO General-Purpose Input/Output.
GPS Global Positioning System.

H

HG High-Gain (differentiator).
HOMD Homogeneous finite-time (differentiator).
HOSM High-order sliding mode (differentiator).
HTA Heavier-Than-Air (Aircrafts).

I

I2C Inter-Integrated Circuit.
IMU Inertial Measurement Units.
INS Inertial Navigation System.
IoT Internet of Things.
ISS Input-to-State Stability.

J

JTAG Joint Test Action Group.

L

LF Lyapunov Function.
LiPo Lithium Polymer (battery).
LMI Linear Matrix Inequality.
LQ Linear Quadratic (Controller).
LTA Lighter-Than-Air (Aircrafts).

M

MEMS Micro Electro-Mechanical Systems.
MOSFET Metal-Oxide Semiconductor Field Effect Transistor.
MPC Model Predictive Control.

N

NED North-East-Down (reference frame).
NON-A Non-asymptotic.

P

P/A passive/active (Sensors).
PC Personal Computer.
PC/EC proprioceptive/exteroceptive (Sensors).
PCB Printed Circuit Board.
PID Proportional Integral Derivative (Controller).
PLL Phase-Locked Loop.
PWM Pulse Width Modulation.

R

RB Rigid Body.
RFID Radio Frequency Identification Devices.
RS-232 Recommended Standard 232.

S

SfM Structure from Motion.
SLAM Simultaneous Localization And Mapping.
SNR Signal-to-Noise Ratio.
SPI Serial Peripheral Interface.

T

ToF Time-of-Flight.
TTL Transistor-Transistor Logic.

U

UAV Unmanned Aerial Vehicle.
UDP User Datagram Protocol.
US Ultrasonic.
USART Universal Synchronous/Asynchronous Receiver/Transmitter.

V

VTOL Vertical Take-Off and Landing.

Notations

$\mathbb{R}$ The set of real numbers.

$\mathbb{Z}$ The set of integers.

$|\cdot|$ Absolute value.

$u \in \mathbb{R}$ is a scalar denoted by the normal math font of a letter.

$\boldsymbol{u} \in \mathbb{R}^n$ is a n dimension vector denoted by the bold math font of a lower-case letter.

$\boldsymbol{M} \in \mathbb{R}^{n \times m}$ is a $n \times m$ matrix denoted by the bold math font of a upper-case letter.

$\mathcal{F}_b$ is a Cartesian right-hand orthonormal frame with origin $\boldsymbol{O_b}$ and axes $\boldsymbol{X_b Y_b Z_b}$.

$\boldsymbol{u}^b$ or $(\boldsymbol{u})^b$ The superscript b is used to denote the vector $\boldsymbol{u}$ with respect to frame $\mathcal{F}_b$.

$\mathbf{0}_{n \times m}$ The zero matrix of dimension $n \times m$.

$\mathrm{I}_{n \times n}$ The identity matrix of dimension $n \times n$.

$\boldsymbol{r}_{\boldsymbol{G}}^b$ The coordinate of the CG of the blimp in its body-fixed frame $\mathcal{F}_b$, with z_G the coordinate on the $\boldsymbol{Z}_b$-axis (page 32).

$\boldsymbol{\xi}^b$ The instantaneous linear and angular velocities vector of the blimp (page 32).

$v_x^b, v_y^b, v_z^b, \omega_x^b, \omega_y^b, \omega_z^b$ The components of $\boldsymbol{\xi}^b$, representing respectively the linear and angular velocities of the blimp in the $\boldsymbol{X}_b$-, $\boldsymbol{Y}_b$-, and $\boldsymbol{Z}_b$-axis direction (page 32).

$\boldsymbol{\eta}^n$ The position and orientation vector of the blimp (page 32).

$x^n, y^n, z^n, \phi, \theta, \psi$ The components of $\boldsymbol{\eta}^n$, representing respectively the position and orientation (roll, pitch, yaw) of the blimp with respect to the navigation frame $\mathcal{F}_n$ (page 32).

$\boldsymbol{R}_{\mathcal{F}}^{\mathcal{F}'}$ The rotation matrix from frame $\mathcal{F}$ to frame $\mathcal{F}'$ (page 32).

$\boldsymbol{M}$ The global inertia matrix of the blimp robot (page 36).

$\boldsymbol{C}(\boldsymbol{\xi}^b)$ The matrix of Coriolis and centripetal terms of the blimp (page 36).

$\boldsymbol{D}(\boldsymbol{\xi}^b)$ The damping matrix of the blimp (page 36).

$D_{v_x}, D_{v_y}, D_{v_z}, D_{\omega_x}, D_{\omega_y}, D_{\omega_z}, D_{v_x^2}, D_{v_y^2}, D_{v_z^2}, D_{\omega_x^2}, D_{\omega_y^2}, D_{\omega_z^2}$ The coefficients of $\boldsymbol{D}(\boldsymbol{\xi}^b)$, representing respectively the linear and quadratic damping coefficients of the blimp (page 39).

$\boldsymbol{g}(\boldsymbol{\eta}^n)$ The restoring forces and moments vector of the blimp (page 36).

f_G The norm of gravitational force vector $\boldsymbol{f_G}$ of the blimp (page 37).

f_B The norm of buoyancy force vector $\boldsymbol{f_B}$ of the blimp (page 37).

$\boldsymbol{\tau}^b$ The control inputs vector of the blimp (page 36).

$f_{px}, f_{py}, f_{pz}, \tau_{px}, \tau_{py}, \tau_{pz}$ The components of $\boldsymbol{\tau}^b$, representing respectively the forces and torques of the blimp (page 36).

$\boldsymbol{M}_{\text{RB}}$ The rigid body matrix of inertia of the blimp (page 40).

$\boldsymbol{M}_{\text{Added}}$ The added-inertia matrix of the blimp (page 40).

$m'_x, m'_y, m'_z, I'_x, I'_y, I'_z$ The diagonal elements of blimp global inertia matrix $\boldsymbol{M}$, representing respectively the apparent masses and moments of inertia of the blimp (page 41).

a_z, b_z, c_z The parameters of the blimp simplified altitude movement model (page 50).

u_z The control input of the blimp simplified altitude movement model (page 50).

$a_x, a_y, a_\psi, b, b_\psi$ The parameters of the blimp simplified planar movement model (page 53).

u, v The control inputs of the blimp simplified planar movement model (page 53).

Chapter 1

Introduction

1.1 Background and Motivation

Nowadays, with the development of society and economy, robotics has become a quickly developing area of science and technology. Compared with traditional manual operation, the application of robot can greatly improve productivity and production efficiency, reduce costs, thus free people from the tedious and low-tech repetitive work. In addition, robots can work in the severe environment where manpower can not reach, for instance handling dangerous industrial waste, exploring interstellar space, and investigating deep sea resources. Therefore, the research and application of robots have profound significance for improving people's quality of life and promoting the development of industry and service industry, it is also an important indicator reflecting human's science and technology development level.

Robots can be classified into two categories according to their workspace: industrial robots and service robots, see Figure 1.1.

Industrial robots are used for manufacturing, they improve the quality of work and productivity by replacing human for dangerous, tedious and dirty jobs [International Federation of Robotics, 2018]. Industrial robots are widely applied in the automotive industry, and there is an increasing demand in the electrical/electronics industry, the metal and machinery industry, the rubber and plastics industry and the food and beverage industry. For the past few

years, many countries have proposed their plans for the next generation of industrial revolution, for instance the "Industry 4.0" of Germany, the "Industrial Internet Consortium" of USA, and the "Made in China 2025" of China, all those programs are aimed at adding more intelligence to industry [Industrial Internet Consortium, 2014; Kennedy, 2015; Lasi et al., 2014]. Thus we will certainly witness a fast growth of industrial robots in the next decades. On the basis of International Federation of Robotics (IFR) forecast, from 2018 to 2020, the worldwide supply of industrial robots are estimated to increase by at least 15% on average per year [International Federation of Robotics, 2018]. Collaborative robots, Internet of Things (IoT) and Machine Learning/AI will lead robotics in the coming years.

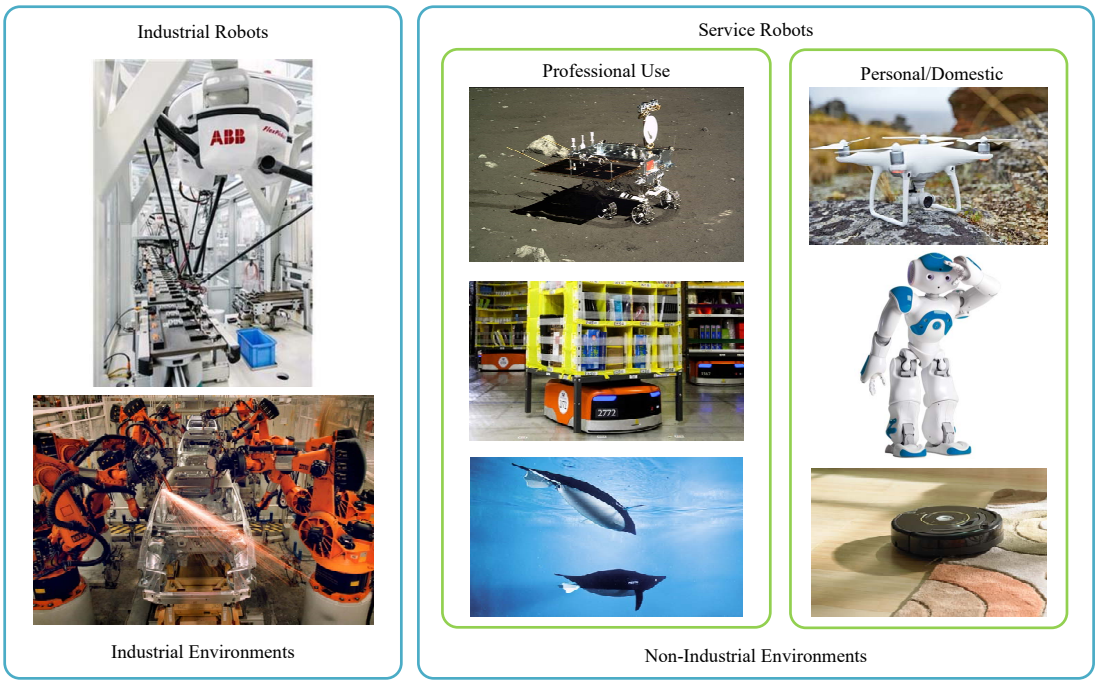

Figure 1.1 – Robot classification according to workspace
Picture source: ABB, KUKA, Yutu moon rover, Amazon Robotics, FESTO AquaPenguins, DJI Phantom, NAO humanoid robot, Roomba vacuuming robot

A service robot is a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations [International Standardization Organization (ISO),

2012]. Service robots can be categorized into professional use and personal/-domestic use according to application area. Concerning the service robots for professional use, they are mainly applied as logistic systems (for instance, autonomous ground vehicles used in factory or e-commerce warehouses [D'Andrea, 2012]), defense robots (for instance, remotely piloted vehicles for military operation [Glade, 2000]), field robots (for instance, milking robots), public relations robots (for instance, robots for mobile guidance and information), medical robots, and scientific/research platform. As for personal and domestic service robots, we can see an increasing trend of robots doing houseworks for human, like floor vacuuming, window cleaning, lawn-mowing etc. Moreover, the market of entertainment and leisure robots has also obtained an explosive growth for the past few years, for instance, people are using drones for aerial photography and racing etc. Based on the prediction of IFR, from 2018 to 2020, the market of service robots worldwide will increase by 20% - 30% on average per year, and the sales could reach almost 43 million units in this period [International Federation of Robotics, 2018].

Another approach to classify robots is through their ability of motion. Usually, industrial robots are often stationary, attached themselves to a fixed surface, using the jointed arm and end-effector to achieve manufacturing tasks. By contrast, mobile robots have the ability to move around in their application environment, and are not fixed to one location. Thus, mobile robots are more agile and can better serve human in various tasks. As a high-intelligence system, a mobile robot combines multiple functions such as environmental information perception, intelligent planning/decision, and motion control execution etc [Siegwart et al., 2011].

Flying machines have always attracted great attention of humans, and robotic researchers have shown a growing interest in Unmanned Aerial Vehicles (UAV), leading to various types of development and application.

In general, the aerial vehicles, or aircrafts, can be classified into Heavier-Than-Air (HTA) and Lighter-Than-Air (LTA) categories, more precisely they can be divided into sub-categories depending on the flying principle and propulsion mode, as shown in Figure 1.2 [Siegwart et al., 2011]. Aircrafts fly by gaining support from the air to counter the gravity, usually they use either static lift

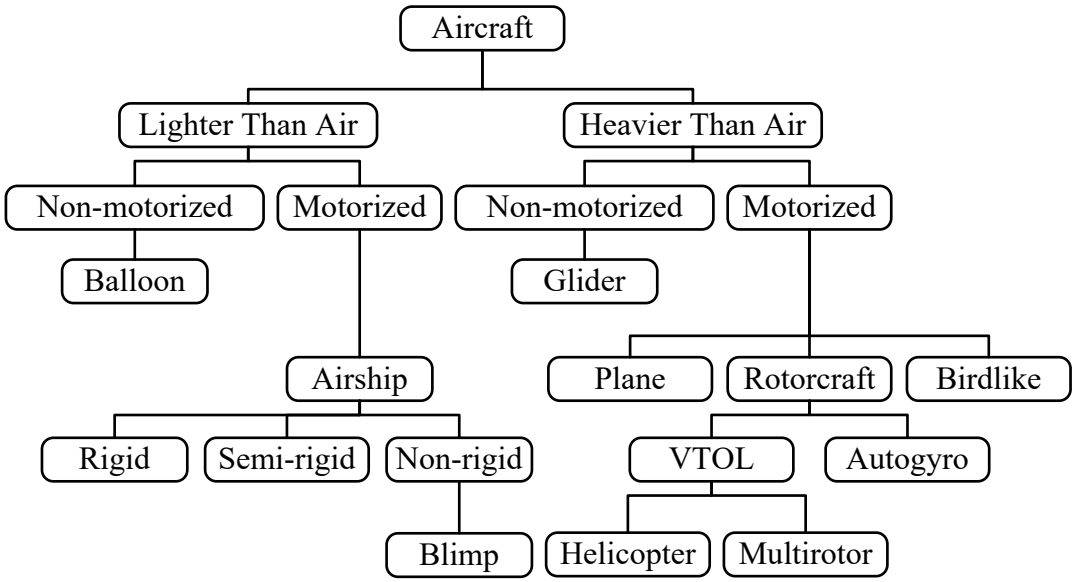

Figure 1.2 – General classification of aircrafts

(LTA) or dynamic lift of an airfoil to keep flying [Stevens et al., 2015]. Inspired by the work of [Siegwart et al., 2011], we try to give a non-exhaustive comparison of flying principles from the miniaturization point of view between different types of aircrafts, as shown in Table 1.1.

In this work, we focus on the blimp, which is a non-rigid airship. Different from semi-rigid and rigid airships (for instance Zeppelins), it maintains the shape by the pressure of the lifting gas inside envelope and the strength of the envelope [Crouch, 2009]. As a Vertical Take-Off and Landing (VTOL) system, blimps have the ability for vertical, stationary and low speed flight, this is a big advantage compare to airplanes. Moreover, in contrast to other VTOL system like helicopters and multirotors, the static lift produced by lifting gas (usually helium) makes it possible for blimp to stay in air for a long time without much fuel consumption. **This key advantage makes blimp an ideal platform for applications like surveillance and exploration.**

The history of airship can trace back to 18th century, where Jean Baptiste Marie Meusnier described a dirigible airship in a paper entitled "Mémoire sur l'équilibre des machine aérostatiques" (Memorandum on the equilibrium of

	Airplane	Helicopter	Multirotor	Bird	Blimp
Power cost	2	1	1	2	3
Control cost	2	1	2	1	3
Payload/weight	1	2	2	2	3
Maneuverability	2	3	3	3	1
Stationary flight	1	3	3	2	3
Low speed fly	1	3	3	2	3
Vulnerability	2	2	2	3	2
VTOL	1	3	3	2	3
Endurance	2	1	1	2	3
Miniaturization	2	3	3	3	1
Indoor usage	1	3	3	2	2
Total	17	25	26	24	27

Table 1.1 – Flying principle comparison (1=Bad, 3=Good)

aerostatic machines) [Meusnier, 1784]. It was a 79m long envelope equipped with 3 propellers and a rudder, but the airship was never aloft due to the lack of powerful and light weight engines [Li et al., 2011]. During the 19th century, scientists and engineers attempted different methods of propulsion to balloons. The first engine-powered flight of airship was made by Henri Giffard in 1852, he flew 27km in a steam-powered airship [Gerken, 1990]. In 1900, the first flight of the Luftschiff Zepplin LZ1 marked the beginning of the golden age of airships, and led to the most successful airships of all time: the Zeppelins [Stephenson, 2012]. The airships were quickly used in World War I as bombers, and powerful countries were building bigger and bigger airships for competition, mostly imitating the original Zeppelin design. But soon the fixed-wing airplanes showed great potential in wars, which slow down the study on airships. And then several incidents, including burning of the Hindenburg in 1937, the largest airship ever built, destroyed public confidence in airships, brought an end to the golden age of airships.

Despite the recession for decades in the research of airships, over the past few years, the progress in techniques such as composite materials, computational fluid dynamics (CFD), automatic control has brought a resurgence to this aircraft [Khoury, 2012; Li et al., 2011; Wilson, 2004]. Various applications are proposed for modern airships in civilian and military fields, such as surveillance [Dolce

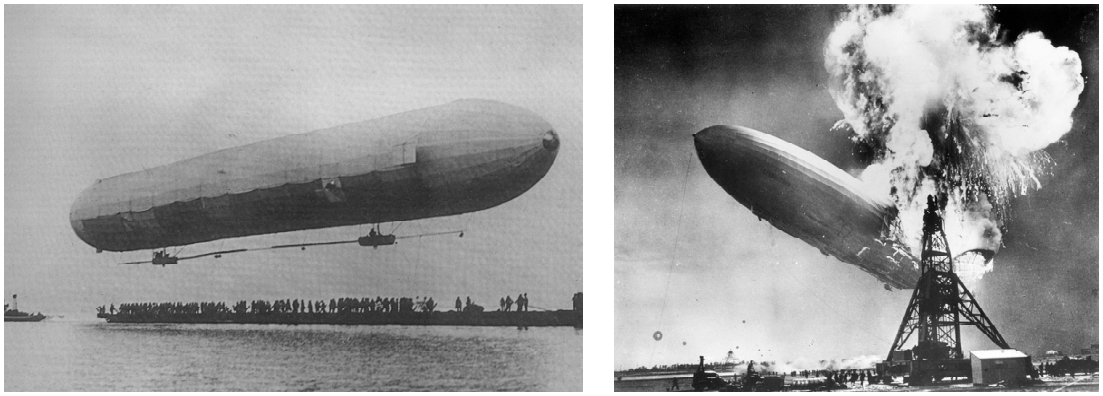

Figure 1.3 – the Luftschiff Zepplin LZ1 (left) and the Hindenburg (right)

and Collozza, 2005; Pan et al., 2015], tourism and advertising [Hansen, 1979], environment monitoring [Dorrington, 2005; Hygounenc et al., 2004; Kantor et al., 2001; Kulczycki et al., 2006], planatory exploration [Elfes et al., 2003; Kusagaya et al., 2006], cargo transportation [Huang and Dalton, 1976], stratospheric observation and telecommunication relay [Lee and Bang, 2007; Lee et al., 2006; Schmidt, 2007] and so on.

The aforementioned airships are all large scale, they can only be tested outdoor. But the researchers are always interested in the miniaturization of aerial vehicles and applying them to indoor operations. In the past few years, with the development of sensor, microprocessor, battery and wireless communication technologies, the hardwares for aerial robots have become more and more light weight, small size and cheap, making it easier for researchers to study miniature aerial vehicles, especially for civilian applications. However, the reduction on hardware cost and size is accompanying with limitations on performance. For instance, using Micro Electro-Mechanical Systems (MEMS) technology to produce Inertial Measurement Units (IMU) reduces the cost and size of IMU to a minimum, but the sensor is much less accurate than conventional ones due to noise and drift problems [Gardner and Varadan, 2001]. Therefore, the control and estimation of such micro aerial vehicles is still a challenging task.

Recently, researchers made various studies of blimps in indoor environments, such as localization of robot, obstacle avoidance algorithm, path planning and trajectory tracking control etc. Here a brief review of the existing works is

presented in chronological order. The work of [Wyeth and Barron, 1998] was an early attempt on the study of indoor blimp robot, they used information provided by sonar system and compared with the desired offsets delivered from the planner to design a controller for blimp, and realized landmark navigation in indoor environment, but the disturbances such as wind are not considered in this work. [Zhang and Ostrowski, 1999] used camera as sensor of blimp system, and studied the feasibility of transferring the dynamic parameters of the system into the image plane, and realized image-based control of robot to track a quasi-static object. Other researchers also tried blimp control using visual feedback, for instance, [Zwaan et al., 2000] designed algorithm to track image regions, and used PID controller for blimp docking and station keeping (keep image window at the center of frame), but when the robot drifts laterally, their controller didn't work well. [Fukao et al., 2003a,b] used image information and applied Extended Kalman Filter (EKF) to do the Structure from Motion (SFM) and got position of the blimp, they realized circling control of the blimp robot around a specified target. Later they extended their work to the design of path following controller based on velocity field using inverse optimal control [Fukao et al., 2007, 2008], and hovering of blimp against mild wind [Saiki et al., 2010]. [Green et al., 2005] also used camera on blimp to realize obstacle avoidance function without considering the dynamic of blimp robot. [Badia et al., 2005] studied an insect based neuronal model for collision avoidance and path following, their blimp robot can track a straight line and avoid collision, but the error was relatively big, and the parameters are trained based on obstacles. [Beji and Abichou, 2005] realized design and simulation of the tracking control of trim trajectories of a blimp robot, but their work didn't consider the influence of chattering input and actuator saturation, and had bad effect when tracking inclined straight line trajectory. The work of [Zufferey et al., 2006] used neuronal controllers whose parameters are trained in simulation to map visual input into motor commands to accelerate the movement of flying robot while avoiding collisions. [Fukushima et al., 2006, 2007] used Model Predictive Control (MPC) to handle the constraints of motor saturation and dead-zone, realized point reaching with straight trajectory. [Ko et al., 2007] combined their Gaussian Processes (GP) enhanced model to reinforcement learning and designed a controller for

blimp yaw and yaw rate control. [Rottmann et al., 2007a,b] tackled the control problem of blimp robot with a model-free approach, they applied Monte Carlo learning, and chose GP to approximate the state-action value function, and by using the ultrasonic (US) sensor information in online learning, their controller achieved blimp altitude control without knowledge of system dynamics or other environment parameters. However, the learning process is rather long. Later, they used the blimp as a platform and studied the problem of localization in indoor environment using US sensors [Müller et al., 2009], and autonomous navigation in mapped environments based on multi-stage path planning [Müller et al., 2011]. [Al-Jarrah and Roth, 2013a,b] used a 2 layer fuzzy controller to control the altitude of blimp and avoid collisions. In the paper of [Burri et al., 2013], an interesting spherical blimp capable of holonomic motion is designed.

To conclude this section of thesis background and motivation, it can be seen that as a UAV, the blimp is capable of realizing various operations and it is an ideal platform for scientific research such as motion control law design, path planning, navigation, sensor technology, image processing etc. Moreover, compared to the other types of aerial vehicles, blimp has the advantage of:

- Ability for VTOL, stationary and low speed flight;
- High payload-to-weight ratio;
- Long endurance in air;
- Low fuel consumption;
- Low acoustic noise level;
- Safe Human-Robot interaction.

Therefore, the blimp is ideal for various indoor applications like:

- Long-term surveillance and monitoring;
- Advertising and entertainment;
- Unknown environment exploration and mapping;

- Warehouse goods inventory;
- Pedestrian navigation in large facility;
- Scientific research and education platform.

However, unlike outdoor large scale airships, the blimp for indoor operation are limited in size and payload, thus only low weight sensors and actuators can be integrated in an embedded micro-system, which means the measurement of on-board sensors can not be very accurate, and the capacity of actuator is constrained. Moreover, although the structure of miniature sized blimp robot is not as complicated as those outdoor airships, and the indoor environment is more stable and have less perturbations, the workspace of indoor blimp is more cluttered and filled with obstacles. Thus, it demands a higher accuracy of blimp motion control to achieve indoor operations.

In general, it is still a challenging goal for the study of miniature aerial vehicles, and that is why we are motivated in the research entitled "Development of a Blimp Robot for Indoor Operation".

1.2 State of the art

Generally, the control of mobile robots involves several different aspects, from the perception of environmental information, to the localization in map, then according to mission, a path is planned for the robot to follow by applying motion control. This procedure rolls in a loop and forms the "See-Think-Act" control scheme for mobile robot systems [Siegwart et al., 2011], as shown in Figure 1.4.

In this section, we firstly present the related works of blimp modeling. Then concerning the problem of perception, a short presentation of sensors used nowadays for aerial vehicles is given. Finally, let us discuss the controller used by other researchers for their blimps to achieve motion control.

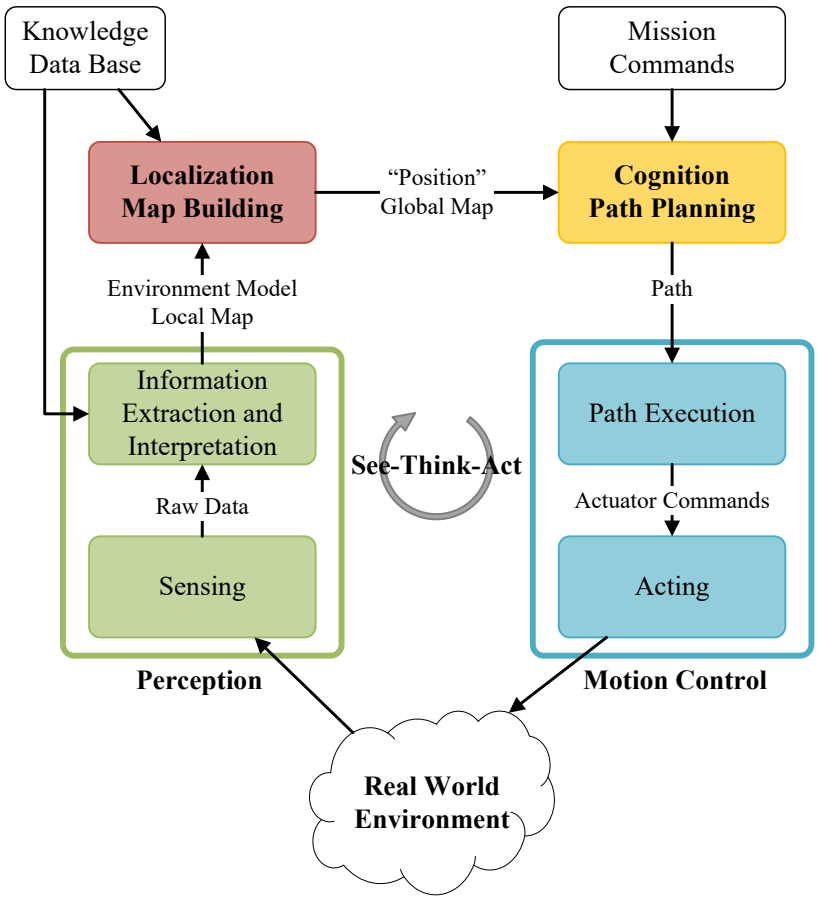

Figure 1.4 – General control scheme for mobile robot systems (from [Siegwart et al., 2011])

1.2.1 Modeling of blimp

As a non-rigid airship, the blimp has similar dynamics with airships, and many researchers have worked on the airship dynamic modeling for the last few decades, and provided theoretical and semi-empirical techniques to solve the problem of modeling.

Typically, an airship has a large streamlined ellipsoid shape hull filled with light gas like helium, actuated by thrusters (sometimes vectorable) which are mounted on a gondola, and equipped with controllable low aspect ratio tail fins [Li et al., 2011], as shown in Figure 1.5. The ballonet installed inside the hull are

used to control buoyancy and adjust the internal pressure of the hull.

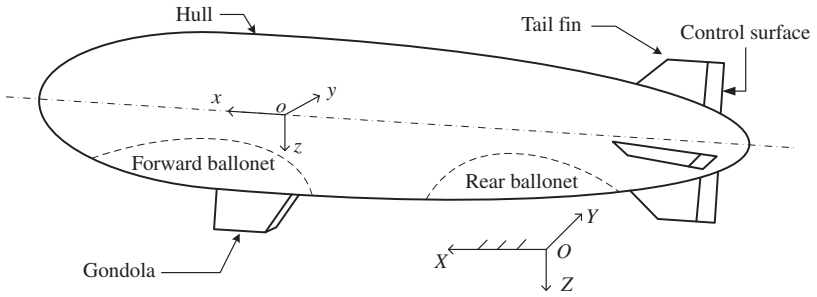

Figure 1.5 – A typical airship (from [Li et al., 2011])

The interaction forces between aircraft and air, including aerostatics and aerodynamics, determine the flight behavior of an airship. Researchers from some large projects have studied the aerodynamics of airship through wind-tunnel tests, such as YEZ-2A [Gomes, 1990] and Lotte [Funk et al., 2003]. And others have done theoretical works based on potential flow aerodynamics [Lamb, 1932; Munk, 1924; Newman, 2018], the airship moves in an unbounded heavy fluid has similar aerodynamic behavior as the hydrodynamics for submarines and underwater vehicles, and the resulting loading is usually called the added-mass force and moment [Fossen, 1994]. The viscous effect on hull, the hull-fin interaction, and axial drag are also studied in various literatures [Allen and Perkins, 1951; Hoerner, 1958; Jones and DeLaurier, 1983]. Lately thanks to powerful computers, CFD method has been applied to analyze the aerodynamic characteristics of airships [Kale et al., 2005; Lutz et al., 2002; Wong et al., 1985].

As for the flight dynamics of airship, researchers have proposed 6 degree of freedom (DOF) nonlinear models relating the inertial forces on the airship to the external forces and moments due to gravity, aerostatic, aerodynamic and control forces etc [Gomes, 1990]. While some others made the assumption that the motion of aircraft is near a trimmed equilibrium flight condition with small disturbances, and got the linear dynamics model such as [Cook et al., 2000; Kornienko, 2006; Schmidt, 2007]. Based on the flight model, stability of airship has been analyzed and researchers found some motion modes for conventional airships [Cook et al., 2000; Gomes, 1990; Kornienko, 2006; Li et al., 2008], they discovered that the stability of an airship depends on whether the fin force and

the restoring gravitational moment are powerful enough to balance the unstable Munk moment or not [Munk, 1924], and the airship speed has an great influence on its motion modes.

When studying the large scale airships which operate in the atmosphere, researchers also have to consider the influence of structural flexibility like deformation or winkles of the hull [Bessert and Frederich, 2005; Burgess and Starchild, 1927], the influence of atmospheric turbulence [Etkin, 1981; José et al., 2002], and the effects of ballonets to the model [Khoury, 2012; Li et al., 2011; Nakadate, 2005]. These topics are beyond the scope of this thesis.

Regarding small indoor blimp robot, it has simpler structure compared to the giant airships. For instance, it doesn't have ballonet to control buoyancy and internal pressure, and usually it doesn't have elevator and rudder to control the pitch and yaw angle because they barely have any influence when the speed of blimp is low. Moreover, the indoor environment is more static and has less disturbance than atmosphere, we can assume there is no deformation of hull due to pressure change, so the blimp can be regarded as a rigid body. Nonetheless, this does not mean that the modeling of indoor blimp robot is an easy task. Because the lack of ballonet for buoyancy adjusting, indoor blimps usually have actuators to change the resulting force in vertical direction, thus make the blimp an unconventional hybrid airship. It is noted that this type of hybrid airship has different flight dynamics due to the lift forces generated by propellers [Tischler et al., 1983].

Many researchers have worked on the modeling of small indoor blimp and the application to real robot [Bestaoui and Hima, 2001; Yamasaki and Goto, 2003; Zufferey et al., 2006], the model is basically a simplified form derived from the airship nonlinear model with some modifications [Gomes, 1990]. A detailed presentation will be carried out in Chapter 2 (Section 2.3).

1.2.2 Sensors used for UAVs

As it is shown in Figure 1.4, perception is an important task of an autonomous mobile robot, and sensors play the role of "eye" for a robot to "see" the world, in other words, to acquire knowledge about its environment [Everett, 1995].

There is a wide variety of sensors used for robots, some of them measure some simple values of the system, like internal temperature of an electronic chip or the rotational speed of the motors, while other sensors are capable to extract information about robot environment, which is then used to solve the motion state or localization problem of robot. Here we are mainly interested in the latter type of sensors.

The sensors can be classified using two functional axes: proprioceptive/exteroceptive (PC/EC) and passive/active (P/A) [Siegwart et al., 2011]. The first axis concerns the origin of information acquired. Proprioceptive sensors measure values from robot system itself, for example, motor speed, linear acceleration, and battery voltage etc. On the other hand, exteroceptive sensors acquire information from the robot environment, for instance, distance measurement to surrounding, light intensity, and sound amplitude etc. The measurements from exteroceptive sensors have to be interpreted so that meaningful environmental features can be extracted. The second axis regards the method for acquiring information. Passive sensors measure ambient environmental energy entering the sensor. Examples are temperature probes, CCD and CMOS cameras etc. While active sensors actively emit energy to environment and measure the environmental reaction, for instance US sensors, laser range finders etc. The most useful sensors for UAV with their classification are listed in Table 1.2 [Siegwart et al., 2011].

Let us discuss some of these sensors used for UAVs. Start with the heading sensors, they are used to determine the UAV orientation, which are called yaw, pitch and roll usually. Together with velocity information, they allow us to integrate the movement of robot and get an estimation of position, this procedure is called dead reckoning for wheeled robot and ship navigation, whose motion can be considered in a 2D plane. But for UAVs which move in 3D space, the localization in environment usually needs more sophisticated method and other sensors.

One of the heading sensors is compass or magnetometer, which measures the Earth's magnetic field [Lenz and Edelstein, 2006]. The history of compass can be dated back to 200 BC, in the Han dynasty of China [Lowrie, 2007], as shown in Figure 1.6. Recently, compasses have been miniaturized and incorporated in

General classification (typical use)	Sensor	PC/EC	P/A
Heading sensors (orientation of the robot in relation to a fixed reference frame)	Compass Gyroscopes	EC PC	P P
Acceleration sensor	Accelerometer	PC	P
Ground beacons (localization in a fixed reference frame)	GNSS Active optical or RFID beacons Active US beacons Reflective beacons	EC EC EC EC	A A A A
Active ranging (distance from robot to objects in its vicinity)	US sensor Lidar	EC EC	A A
Vision sensors	CCD/CMOS cameras	EC	P

Table 1.2 – Classification of sensors used in UAV applications (from [Siegwart et al., 2011])

integrated circuits at very low cost thanks to MEMS technology. In outdoor clear environment, the measurements from magnetometer can be used to calculate the heading angle relative to the direction of local magnetic field. However, a major drawback when applying compass for indoor robots is that its measurement is easily influenced by other magnetic objects and man-made structures.

Figure 1.6 – Heading sensor examples: Ancient Chinese compass (left) and Two-axis mechanical gyroscope (right)

Gyroscopes preserve their orientation in relation to a fixed reference frame, thus they provide an absolute measure for the heading of a mobile system. There are mechanical and optical gyroscopes. The concept of mechanical gyroscopes

is based on the inertial property of a fast-spinning gyro, which preserves its orientation in inertial frame due to the associated angular momentum, see Figure 1.6. Therefore with a carefully designed outer structure, they can be used to measure heading angle directly, but this type of gyroscope is usually very expensive and large [Lee, 1994]. Rate gyros measure the angular speeds instead of absolute orientation, and they can also be made by MEMS technology and become low-cost. The optical gyroscopes are more delicate, using the principle of Sagnac effect, they measure the frequency difference of two laser beam which are sent in opposite direction through an optical fiber [Post, 1967].

An accelerometer is a device used to measure all external forces acting on it. For the mechanical accelerometer, it can be modeled by a spring-mass-damper system, and the force is measured proportional to the displacement of mass [Dudek and Jenkin, 2008]. Other types of accelerometer are capacitive, piezoelectric etc.

Figure 1.7 – An example MEMS accelerometer produced by Sandia National Laboratories (left), A commercial IMU produced by Xsens (middle) and the inside of IMU (right)

Nowadays, on the robots, instead of individual gyroscope and accelerometer, the IMU is installed, see Figure 1.7. It uses gyroscopes and accelerometers to estimate relative position, orientation, velocity and acceleration of a moving object, it is an important component for the Inertial Navigation System (INS) [Britting, 1971]. It is worth to mention that, in order to derive position and orientation from the raw information acquired by sensors (usually angular velocities and linear accelerations), the integration is used, thus any error in measurement is also integrated over time. Therefore, drift of IMU is an unavoidable problem. There exist more expensive IMUs that drift relatively slowly, but after long

period of operation, the error will still be unacceptable. In order to counter the drift, some external measurement is required, and that's one reason which makes the last three types of sensors in Table 1.2 become extremely useful for robots. Researchers have used IMU in their studies of indoor blimp robot in the last few years [Burri et al., 2013; Fukushima et al., 2006; Müller et al., 2009; Rottmann et al., 2007b; Saiki et al., 2010].

One approach to solve the drift problem of INS and aid to improve localization accuracy of mobile robots is to use beacons, the interaction between on-board sensors and environmental beacons can be used to identify the position of robot.

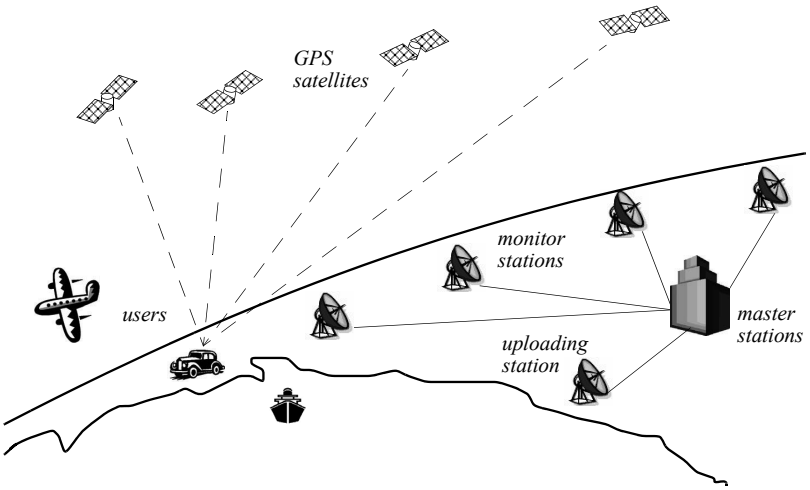

Figure 1.8 – Schematic diagram of positioning based on GPS (from [Siegwart et al., 2011])

One of such beacon system is Global Navigation Satellite System (GNSS), it has been proved to be extremely useful for military operation at the beginning, but now it is also available for civilian navigation. Many countries have developed their proper satellite navigation systems, such as Global Positioning System (GPS) of USA, BeiDou Navigation Satellite System (BDS) of China, Galileo system of European Union, and GLONASS system of Russia [Hofmann-Wellenhof et al., 2007; Yang, 2016]. GPS is the most successful and mature satellite navigation system, there are at least 24 operational satellites at all time, which orbit every 12 hours at a height of 20.190km [Kaplan and Hegarty, 2005]. Each satellite

continuously transmits its location and the current time data, therefore, GPS receivers installed on robots are completely passive and exteroceptive sensors. By combining the arrival time of information and location of satellites, the receiver can work out its own position. Information from at least four satellites are required, three position axes and one for time correction. An extension method is the differential GPS (DGPS), which uses the information from precisely located stationary ground station to improve the resolution of receiver position, as shown in Figure 1.8. The GPS system is extremely useful for outdoor flying robots, but at narrow places, challenges arise such as multi-path and loss of satellite signal [Kaplan and Hegarty, 2005]. Therefore, for indoor robots, we've rarely seen the use of GPS as a positioning sensor.

Other ways to build the ground beacons in indoor environment is to use active US or laser or Radio Frequency Identification Devices (RFID) beacons [Finkenzeller, 2010; Kleeman, 1992]. For such indoor beacon system, the difficulty lies in distinguishing the true signal from reflections of walls, smooth floors etc, and the blocking of obstacles for the direct signal.

The active ranging sensors is perhaps the most popular sensors used in robotics research, since they provide easily interpreted outputs: distance measurement from the robot to objects in its vicinity [Siegwart et al., 2011]. Generally, they use the principle of time-of-flight (ToF) to measure distance. Assume the propagation speed of sound (US sensor) or electromagnetic wave (Lidar) is constant in the environment, by measuring the time passed from the emission of signal to the receiving of reflection signal, the distance can be calculated. Lidar, or laser range finder, uses laser to scan the surroundings. It requires a precise determination of the exact time of arrival of the reflected signal and a ToF measurement for the Lidar to give an accurate distance measurement, because the speed of light is 0.3m/ns [Rioux, 1984]. Thus Lidar is usually expensive and weighs from several hundreds of gram to several kilogram, see Figure 1.9. In order to use it as a sensor on the UAVs, it demands a high payload carrying ability of the robot.

Another active range finder is the US sensor, or sonar, which transmit a packet of US waves at every measurement [Carullo and Parvis, 2001], see Figure 1.9. Most US sensors used by robots have a measurement range from 5 cm to 5 m and

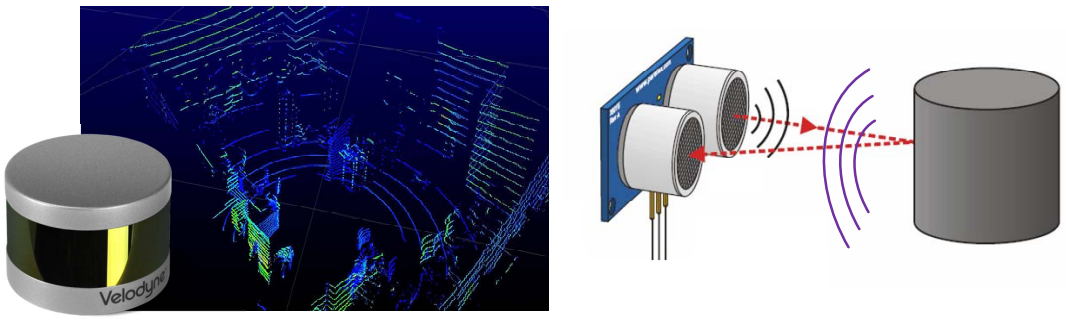

Figure 1.9 – Active ranging sensor examples: Lidar from Velodyne (left) and typical US sensor(right)

a resolution of approximately 1 cm, it is sufficient for normal indoor navigation purpose. But the main disadvantage of US sensor is that the directionality of US sound waves is not as good as laser, so there is a dispersal cone of the transmitted beam. Consequently, the sensor only tells the existence of an object at certain distance within the area of measurement cone instead of a depth data of a point. And the interaction with target such as surface absorption and multi-path problem makes the measurement rather noisy. Moreover the speed of sound is approximately 0.3m/ms, much slower than electromagnetic wave, so the frequency of measurement is limited. However, the US sensor is still low-cost, light weight and with acceptable performance for range measuring, which make it a popular sensor used for indoor blimp robot research [Al-Jarrah and Roth, 2013a; Müller et al., 2009; Rottmann et al., 2007a; Wyeth and Barron, 1998].

Vision sensors have become more and more popular in robotics in recent years for the enormous amount of environmental information they can provide in images. The difficulties consist of processing the digital image so as to get useful information like depth computation, motion detection, color tracking, feature detection, scene recognition and so on [Siegwart et al., 2011]. The analysis and processing of images lead to another scientific research field known as computer vision [Hartley and Zisserman, 2003; Ma et al., 2012; Szeliski, 2010; Trucco and Verri, 1998]. The two main types of camera imaging sensor used are CCD (charge-coupled device) and CMOS (complementary metal-oxide semicon-

Figure 1.10 – Vision sensor examples: omnidirectional camera from VSN Mobil (left) and stereo camera from Stereolabs (right)

ductor). They capture the light from real world which passes the camera lenses, and save the digital values in a matrix, which is known as image. Depending on the lenses used, the camera can be classified into pinhole camera, which is modeled as a perspective projection, and ominidirectional camera, which have a wide field of view of more than 180 degrees, see Figure 1.10. Normally, the pinhole cameras are used in robotics to reconstruct the structures of environment, provide spatial information and thus can be used to calibrate the drift of INS etc. This procedure is done by either taking several images of the scene from different view at the same time, which is called structure from stereo, see Figure 1.10, or taking different images from the same camera by changing its viewpoint to a different camera position, which is called structure from motion (SfM) [Hartley and Zisserman, 2003; Wilson and Ritter, 2000]. The processing of image and reconstruction of structure from image are beyond the scope of this section and will not be discussed. Researchers have implemented camera sensors in their blimp systems for a long time, such as [Badia et al., 2005; Burri et al., 2013; Fukao et al., 2003b; Müller et al., 2011; Saiki et al., 2010; Zhang and Ostrowski, 1999; Zufferey et al., 2006; Zwaan et al., 2000].

We have chosen the sensors used for our developed indoor blimp robot in consideration of the needed functionality, payload limit and budget, and it will be presented in Chapter 2 (Section 2.5).

1.2.3 Controller for blimp

From what we have seen from the literatures, other works of blimp motion control focus on three types of flight control technologies: stabilization, trajectory tracking and path following. The stabilization problem focus on stabilizing the blimp at desired state. The trajectory tracking problem is concerned with the design of control laws for the robot to reach and follow a time parameterized reference, while for the path following problem, the path to be followed is specified without a temporal law [Aguiar and Hespanha, 2007]. Taking into consideration the blimp low-maneuverability, the path following controller is more popular to be studied than the trajectory tracking controller since it doesn't have specific temporal constraint in the path [Zheng et al., 2013]. To design such motion controllers, researchers have to consider the highly nonlinear and usually under-actuated airship dynamics. Moreover, uncertainties in the model, parametric variations and disturbances also make it complex to design blimp motion controllers [Zheng and Sun, 2018].

In this review, we concentrate on the motion controllers developed for small indoor blimp robots, and it can be concluded that those controllers can be roughly divided into two categories. The first type is developed from the modern control theory, and those controllers rely on the model of blimp system. The second type is less traditional, using learning technologies to train the gains of controller, thus depending less on the model of blimp, sometimes the designing processes of controllers are even completely free from the knowledge of blimp dynamics.

Among the first category of controllers, one tool of nonlinear control design often used by other researchers is the feedback linearization (or dynamic inversion). Since the control task of blimp usually requires feedback and the models of blimp are highly nonlinear, if the feedback can be designed to cancel the nonlinear terms in blimp system and transforms the closed-loop control system into a linear one, then there are many ways to design a stabilizing state feedback control, this is the general idea for the feedback linearization approach [Khalil, 1996; Stevens et al., 2015]. In the work of [Zhang and Ostrowski, 1999], the authors used feedback linearization plus a simple PID controller to make the

blimp track a quasi-static object. In [Moutinho and Azinheira, 2005; Paiva et al., 2006], the authors used a dynamic inversion controller for their robotic airship path following task. However, to make the linearization approach work well, the perfect knowledge of the state equation is required, which in practice is almost impossible. Thus for the purpose of making the closed-loop system robust to model uncertainties and other perturbations, complementary designs have to be made to the controller [Khalil, 1996].

Another popular nonlinear design tool used for blimp motion control is backstepping method. By breaking the design problem of the full-system into a sequence of design problems for lower order subsystems and choosing properly Lyapunov functions (LF), the backstepping approach find a feedback controller which guarantees asymptotic stability of nonlinear system while assuring robustness to unmatched uncertainties [Azinheira et al., 2009; Khalil, 1996]. [Fukao et al., 2003b] used backstepping technique to design an image-based controller which flies around a set target automatically. In the work of [Hygounenc et al., 2004], the authors also used backstepping to design controllers, and proposed a global control strategy by switching between four sub-controllers corresponding to different flight phases. [Beji and Abichou, 2005] combined integrator backstepping approach to design tracking feedback control for trimmed ascent and descent flight. In [Azinheira and Moutinho, 2008; Azinheira et al., 2006], the authors used backstepping approach to solve the airship hover stabilization problem, then in [Azinheira et al., 2009] they designed controller for path tracking task. Although the backstepping is a powerful tool to design nonlinear controllers, it suffers from the problem that when the order of system increases, the controller becomes more and more complex to design, and the choice of LF also becomes difficult.

In the paper [Fukao et al., 2007, 2008], the authors proposed an inverse optimal tracking control for the blimp to move around a setting target, the design of feedback controller is based on the Control Lyapunov Function (CLF) of [Artstein, 1983] and Sontag's formula [Sontag, 1989]. Using the CLF, one can get the controller and verify the stability of system simultaneously, but again, the choice of LF can be tricky. For more details on the use of CLF with stronger stability, please refer to [Moulay and Perruquetti, 2006].

The controllers presented previously are mostly based on the theoretical analysis of the system, but sometimes taking the engineering judgment into the design is not a bad idea since they have to be applied to actual platforms. One of such controllers is the Linear Quadratic (LQ) controller. The control input of LQ controller has to minimize a quadratic cost (or performance index) which is regulated by some weighting matrices, and the matrices are selected according to engineering judgment [Stevens et al., 2015]. The LQ controller is widely used nowadays in aircraft systems since it can determine all the elements of controller gain simultaneously instead of trial-and-error method used for multivariable system stabilization in classical approaches, and guarantee the closed-loop system to be stable. In the work of [Fukushima et al., 2006, 2007], the authors used LQ controller to solve the controller gain, and used Model Predictive Control (MPC) to take into account additive uncertainties and constraints, finally they achieved tracking of a straight line for the blimp robot. [Müller et al., 2011] also used LQ controller to keep their robot on the desired trajectory and realized path following. One of the difficulties when applying LQ controller lies in the appropriate selection of the weighting matrices, which demands engineering expertise.

One of the controllers which is maybe the most popular in practice is the Proportional Integral Derivative (PID) controller, for its ease to design, and the ability to achieve asymptotic regulation in the presence of parametric uncertainties by including "integral action" in the controller [Khalil, 1996]. It should be classified somehow in between the two categories we defined before, since the gains can be tuned just according to the system tracking error even without any known system parameters, but obviously if the system parameters are known, the PID gains can be determined more precisely before any trial-and-error process. There are many related works of indoor blimp robot which applied PID controllers, for instance, [Wyeth and Barron, 1998] used information provided by sonar sensors in the controller to achieve blimp movement between set points. [Zwaan et al., 2000] used PID controller to realize the altitude stabilization control of blimp, the same team also used visual information in the design of PID controller to achieve position stabilization in [Zwaan et al., 2000]. [Fukao et al., 2003a] combined visual information in the controller and realized blimp

circling around static target. [Green et al., 2005] designed PID controller to achieve straight line following and obstacle avoidance based on camera information. And [Takaya et al., 2006] designed several PID controllers for different movements of blimp and achieved landing orbit motion for the robot. However, to make the PID controller works well in practice, the gains usually have to be tuned carefully via multiple tests, which can be time consuming.

Regarding the second category of controllers whose gains are trained via learning process, we can notice firstly the neural controller. An artificial neural network (ANN) can be trained to learn a plant's inverse dynamics directly by approximating the nonlinear dynamics with a generalization of linear regression, it can also be used to design controllers directly from states without explicit knowledge of system dynamics when the system is not complex [Psaltis et al., 1988]. For instance, [Badia et al., 2005] used insect based neuronal models to derive course stabilization and collision avoidance controllers for their blimp robot. In the work of [Zufferey et al., 2006], the authors designed neural controllers which map visual input into motor commands such that the blimp is steered forward as fast as possible while avoiding collisions in a patterned room, moreover the ANN parameters are trained in an evolutionary procedure with simulated data. In [Rao et al., 2007], the authors trained an ANN controller with flight data under manual control, and achieved yaw angle control during path following. Although ANN is powerful and has become a popular research area recently, it encounters several difficulties when applied as controller for real plants. Firstly, it is based on learning (or approximating) the dynamics from training data, thus if the training set does not cover all the possibilities (which is usually the case), it may not work well in reality. Moreover, when the system is complex, it is hard to find a proper architecture for the ANN, the collection of training data becomes tedious, and the learning may not lead to good results. Furthermore, unlike controllers based on control theories, the neural controller lacks of rigorous analysis of system, thus some important properties like stability of system is hard to be verified.

Some other researchers designed their controllers for indoor blimp robot based on reinforcement learning. Basically, the reinforcement learning supposes that the robot interacts with environment and gets rewards or penalties according

to the actions it performs, thus it should find the behaviors which can maximize the reward [Kaelbling et al., 1996]. In [Rottmann et al., 2007a,b], the authors used Monte Carlo reinforcement learning to learn the state-action value function approximated by Gaussian process (GP), and achieved the altitude control design for their blimp robot. In [Ko et al., 2007], the authors trained a GP on the residual between nonlinear model and ground truth data, and used so called GP-enhanced model together with reinforcement learning to design a yaw controller for the blimp.

We can also see the use of fuzzy logic controllers in other works on blimp robot. It provides a method of transforming a linguistic control strategy based on expert knowledge into an automatic control strategy [Lee, 1990]. In [Rao et al., 2005], the authors employed fuzzy controller which is optimized using an improved genetic algorithm, and the fuzzy controller works with an integrator for the heading control. In [Al-Jarrah and Roth, 2013a,b], the authors designed a 2 layer fuzzy controller to achieve altitude control and avoid collisions. One of the difficulties when applying fuzzy logic controller is the decision of fuzzy logic rules and fuzzy sets.

The controllers designed for our blimp motion control tasks are presented in Chapter 3 and Chapter 4.

1.3 Contribution

In this thesis, a blimp robot for indoor operation is developed. The work contains both theoretical and practical parts.

In the theoretical part, first a novel approach for the modeling of blimp system is proposed. As we can see from the previous works on modeling in Section 1.2.1, some of them are very complex (will be presented in Section 2.3 of Chapter 2), which need lots of accurate experiment data to identify the parameters in model, but there are still disturbances from the environment which cannot be modeled, and the others did not propose model for the blimp dynamics, they used learning based controllers for blimp motion control while not knowing blimp dynamics.

We want to build a model which can balance the accuracy and the complexity

while ensuring the performance of controller. Since an indoor blimp robot is considered in this work, then its size has to be sufficiently small, and its payload is limited. This limits possible quantity of sensors mounted on the robot and computational complexity of the control and estimation algorithms. That is why we intend to use a simple model for blimp motion control task, the model is considered as a nominal model and its parameters can be easily identified via the designed tests. Then the nominal model is complemented with disturbance term, which includes all the uncertainties in model, parameter identification inaccuracies and external perturbations etc. Next in the design of controller, the disturbance term is estimated then compensated in real-time, so that the accuracy of control is ensured while the complexity is reduced to a minimum.

After reasonable assumption, the motion of blimp is divided into two independent parts, the altitude and horizontal plane movement. Then the two types of movements are separately studied and controllers are designed. The restrictions on hardware (such as time-delay, motor defect etc) of blimp system observed during the parameter identification process are analyzed and solutions are proposed in the controller design. In addition, simulations are carried out to verify the effectiveness and performance of designed controller and disturbance estimation method.

In the practical part, a real blimp robot experiment platform is constructed. From the functionality analysis, to the robot structure design, sensors selection, electric circuit board design, through the construction and assemblage of blimp control board, to the calibration and debugging, eventually, a low-cost multi-functional indoor blimp robot experiment platform is established. Based on the platform, the designed control algorithms for blimp point stabilization and path following tasks are programmed for the robot and tested. Finally, satisfying test results are obtained. The platform can be investigated further on other complex tasks for blimp robot, like indoor navigation, unknown environment exploring and mapping, human-robot interaction applications etc.

1.4 Outline of the thesis

The thesis is organized as shown in Figure 1.11:

In this chapter, the background and motivation for the study of indoor blimp robot are introduced. Then a literature review regarding the blimp modeling, sensors and controllers used is briefly presented.

In Chapter 2, we focus on the problem of blimp modeling, the most commonly used blimp kinematic and dynamic model are presented in detail. Due to the complexity of this modeling method and the unsatisfying performance of the controllers designed based on this model in the related works, we proposed an alternative approach based on the use of a simplified nominal model plus disturbance term to represent the blimp dynamics. The blimp movement is divided on two separate ones: altitude and planar movement, analyzed and modeled. The model parameters are identified via designed tests.

In Chapter 3, the blimp altitude control problem is considered. To realize the state estimation, several differentiators are compared and used as observer. Then a predictor-based controller is conceived due to the time-delay occurred in the closed-loop system. A method to estimate the disturbance term in altitude model is proposed. Next we present the determination of controller gain in consideration of the parameter identification results. Finally simulation results for altitude control are given in this chapter.

In Chapter 4, we address the problem of blimp horizontal plane movement control. First, based on the analysis of blimp dynamics in planar movement, the under-actuated system is transformed to a simpler one for the ease of controller design. Then again the method to estimate disturbance term in model is proposed, in order to design the disturbance compensation based controller for point stabilization and trajectory tracking tasks. Finally, simulation results are discussed.

In Chapter 5, the development of an indoor blimp experiment platform is presented, from the hardware design, electric circuit design of blimp control board, to the testing environment setup including the use of camera tracking system. The control laws developed in Chapter 3 and Chapter 4 are implemented and tested on the blimp robot.

Finally, conclusion and perspectives come at the end.

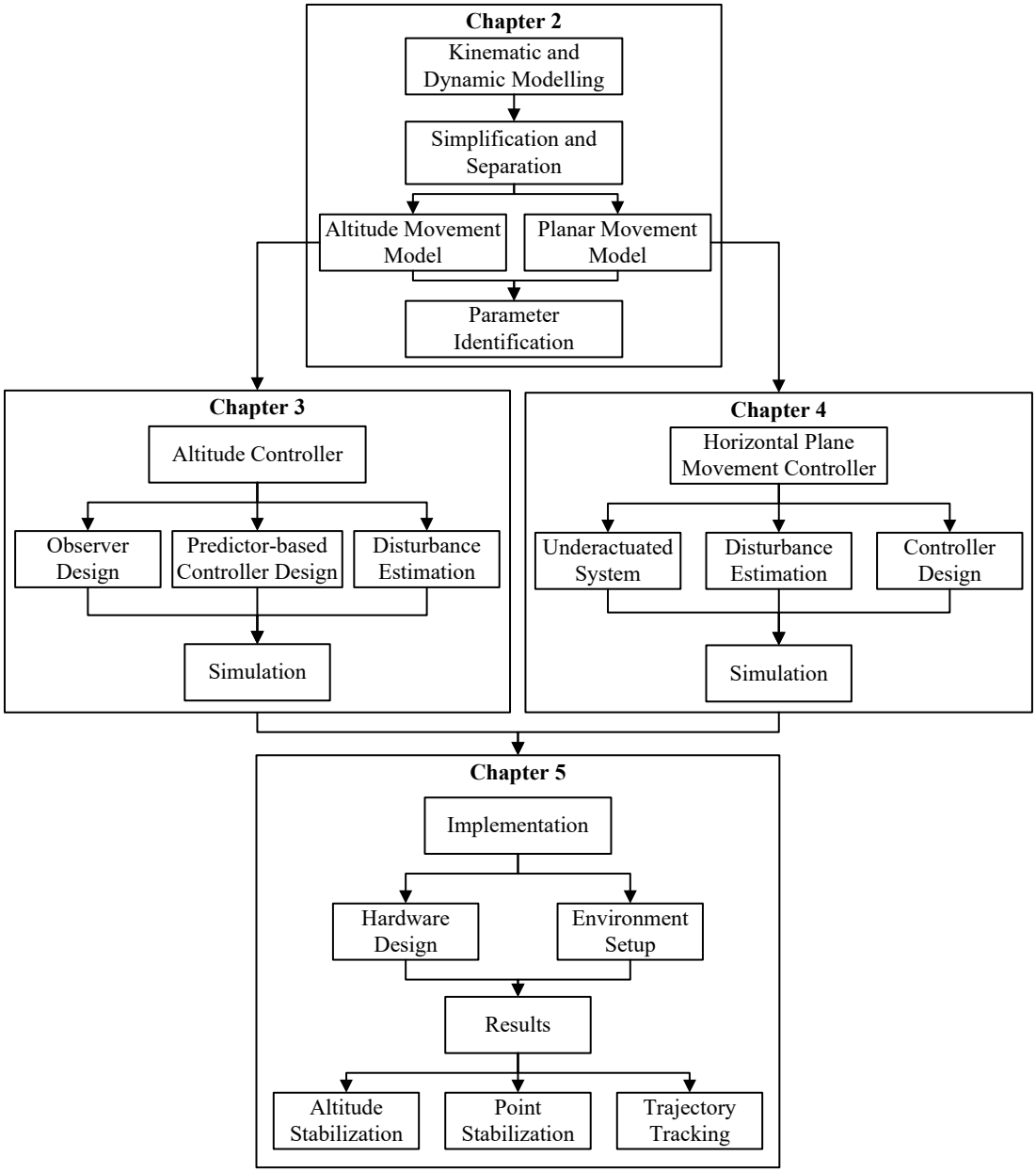

Figure 1.11 – Structure of the thesis

Chapter 2

Modeling and Parameter Identification

2.1 Introduction

The motion of the blimp robot can be described by its model. Many researchers have studied the modeling of small indoor blimp [Bestaoui and Hima, 2001; Yamasaki and Goto, 2003; Zufferey et al., 2006; Zwaan et al., 2000], where the model has basically a simplified form derived from the airship nonlinear model with some modifications [Fossen, 1994; Gomes, 1990].

However, due to the complexity of the airship nonlinear model, it needs lots of experiment data to accurately identify the model parameters. Under some reasonable assumptions, the complex model can be simplified and decoupled into independent parts, which will ease the design procedure of the motion controllers and estimators.

In addition, in order to validate the model, the parameters have to be identified based on test data observed on real robots, hence it is related to specified hardwares. In this work, two generations of robot are studied successively, the first one is named NON-A blimp prototype, where NON-A is the name of our research team which signifies non-asymptotic [NON-A, 2018]. It is modeled and tested for the altitude control law. Due to the hardware limits, later, a second version of the robot is created and named NON-A blimp V2. On the V2 robot,

nominal model parameters are identified via designed tests and motion control laws are conceived based on that.

This chapter starts with the modeling of the blimp (Section 2.2 and 2.3), then the simplification of the model (Section 2.4) and its parameter identification (Section 2.6) are given. Moreover, the sensors used for experiments are presented in Section 2.5.

2.2 General hypotheses

In this work, we study the modeling of an indoor blimp robot. Since the indoor environment is different from the outdoor atmosphere, several hypotheses are made regarding the environment and the blimp.

- The equivalent density of the blimp is approximately equal to the density of air, which means the blimp is able to stay aloft without (or with little) actuator actions.
- The aeroelastic phenomena applied on the hull of the blimp are neglected, which means the hull is free from any deformation or wrinkles, and it is regarded as a rigid body [Bessert and Frederich, 2005].
- The hull of the blimp is considered as an ellipsoid.
- The indoor blimp has a simpler structure compared to typical airship, it does not have ballonet inside the hull (see Figure 1.5) to adjust inner pressure and buoyancy. Therefore, the mass and volume of the blimp robot during operation are assumed to be constant.
- The phenomenon of internal added fluid due to the motion of helium molecules inside the hull is ignored [Hygounenc et al., 2004].
- The blimp center of buoyancy (CB) is assumed to coincide with the hull center of volume (CV).
- The blimp velocity is low.

- The air viscosity is considered constant. And the motion of blimp does not modify locally the density of air.

- As the blimp moves slowly in indoor environment, control surfaces such as rudder and elevator have poor efficiency for changing the yaw and pitch angle [Gomes, 1990]. Hence only thrusters (e.g. motor with propeller in our robot) are responsible for steering and propulsion of the blimp.

Based on the hypotheses, the dynamic and aerodynamic modeling of the blimp can be derived.

2.3 Kinematic and dynamic modeling

In this section, the most commonly used model for indoor blimp is presented, it is basically a simplified form derived from the airship nonlinear model with some modifications [Gomes, 1990].

2.3.1 Choice of Inertial and Body Frames

The reference frames for the blimp model are shown in Figure 2.1.

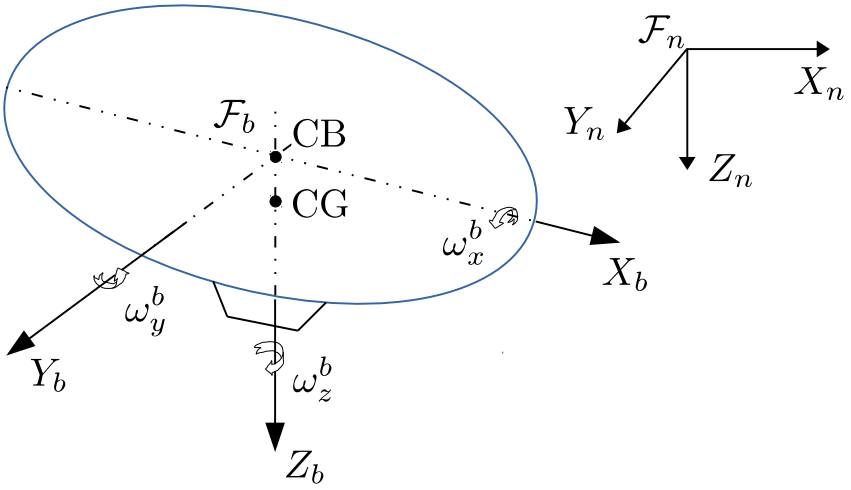

Figure 2.1 – Commonly used reference frames for indoor blimp robot

The frame $\mathcal{F}_n$ is the local navigation frame which is tangent to the Earth surface, its direction is North-East-Down (NED). Since we consider only the operation of blimp robot in indoor environment, the movement of the Earth is ignored, thus the navigation frame $\mathcal{F}_n$ is assumed to be an inertial frame (Galilean reference frame). We can also denote it as $\mathcal{F}_i$, but in order to distinguish the navigation frame which is only inertial in the specified application scenario, we will keep to use $\mathcal{F}_n$ in this chapter.

The body-fixed frame $\mathcal{F}_b$ locates its origin at the CB of the blimp, which is also the CV of the hull, the direction of $\mathcal{F}_b$ is forward-right-down.

Due to the fact that the gondola with actuators and other electrical components are mounted on the bottom of the hull, the center of gravity (CG) is located on the $\boldsymbol{Z_b}$ axis of body-fixed frame, therefore denote its coordinate in $\mathcal{F}_b$ as $\boldsymbol{r}_{\boldsymbol{G}}^{b} = \begin{bmatrix} 0 & 0 & z_G \end{bmatrix}^{\mathrm{T}}$.

The instantaneous linear and angular velocities of the blimp are described in $\mathcal{F}_b$ as

$$\boldsymbol{\xi}^b = \begin{bmatrix} (\boldsymbol{v}^b)^{\mathrm{T}} & (\boldsymbol{\omega}^b)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} v_x^b & v_y^b & v_z^b & \omega_x^b & \omega_y^b & \omega_z^b \end{bmatrix}^{\mathrm{T}}$$

whereas the position and orientation of the blimp with respect to $\mathcal{F}_n$ are expressed as

$$\boldsymbol{\eta}^n = \begin{bmatrix} (\boldsymbol{\eta}_1^n)^{\mathrm{T}} & (\boldsymbol{\eta}_2^n)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} x^n & y^n & z^n & \phi & \theta & \psi \end{bmatrix}^{\mathrm{T}}$$

where ϕ, θ and ψ are roll, pitch and yaw angle, respectively.

2.3.2 Choice of Tait-Bryan angles

Definition 2.1. $\boldsymbol{R}_{\mathcal{F}}^{\mathcal{F}'}$ of dimension 3×3 is the rotation matrix from frame $\mathcal{F}$ to frame $\mathcal{F}'$. It is a matrix whose columns are the vectors of the final frame expressed in the initial frame.

Here we choose the $z - y' - x''$ Tait-Bryan angle to transform from inertial frame $\mathcal{F}_n$ to body-fixed frame $\mathcal{F}_b$, which means first $\mathcal{F}_n$ is rotated by an angle ψ (yaw) around $\boldsymbol{Z_n}$-axis to get an intermediate frame $\mathcal{F}_1 = \boldsymbol{X'Y'Z'}$, the rotation

matrix from frame $\mathcal{F}_n$ to $\mathcal{F}_1$ is expressed as

$$\boldsymbol{R}_n^1 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

then this frame $\mathcal{F}_1$ is rotated by an angle θ (pitch) around the $\boldsymbol{Y}'$-axis to obtain another frame $\mathcal{F}_2 = \boldsymbol{X}''\boldsymbol{Y}''\boldsymbol{Z}''$, the rotation matrix from frame $\mathcal{F}_1$ to $\mathcal{F}_2$ is expressed as

$$\boldsymbol{R}_1^2 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

and finally the frame $\mathcal{F}_2$ is rotated by an angle ϕ (roll) around $\boldsymbol{X}''$-axis to get the body-fixed frame $\mathcal{F}_b$, the rotation matrix from frame $\mathcal{F}_2$ to $\mathcal{F}_b$ is expressed as

$$\boldsymbol{R}_2^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

Therefore the rotation matrix from $\mathcal{F}_n$ to $\mathcal{F}_b$ is

$$\boldsymbol{R}_n^b = \boldsymbol{R}_n^1 \boldsymbol{R}_1^2 \boldsymbol{R}_2^b \tag{2.1}$$

Thus for any vector $\boldsymbol{u}$, its coordinate transformation in the two frames can be obtained by the change of basis equation

$$\boldsymbol{u}^n = \boldsymbol{R}_n^b \boldsymbol{u}^b \tag{2.2}$$

2.3.3 Kinematic model

As presented in the Section 2.3.2, for the instantaneous linear velocity, the change of basis equation is

$$\boldsymbol{v}^n = \dot{\boldsymbol{\eta}}_1^n = \begin{bmatrix} \dot{x}^n & \dot{y}^n & \dot{z}^n \end{bmatrix}^{\mathrm{T}} = \boldsymbol{R}_n^b \boldsymbol{v}^b = \boldsymbol{R}_n^b \begin{bmatrix} v_x^b & v_y^b & v_z^b \end{bmatrix}^{\mathrm{T}}$$

in short

$$\dot{\boldsymbol{\eta}}_1^n = \boldsymbol{R}_n^b(\boldsymbol{\eta}_2^n)\boldsymbol{v}^b \tag{2.3}$$

concretely

$$\boldsymbol{R}_n^b(\boldsymbol{\eta}_2^n) = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\phi\sin\theta - \cos\phi\sin\psi & \sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta & \cos\phi\sin\psi\sin\theta - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{2.4}$$

The rotational kinematic equation relating robot angular velocity $\boldsymbol{\omega}$ to rotation matrix is [Barfoot, 2017]

$$\dot{\boldsymbol{R}}_n^b = \boldsymbol{R}_n^b S(\boldsymbol{\omega}^b)$$

where the operator $S(\cdot)$ is defined as follows:

Definition 2.2. The exterior product of two vectors $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^{\mathrm{T}}$ denoted by $x \wedge y$ is defined by:

$$\begin{aligned} x \wedge y &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \wedge \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{bmatrix} \\ &= -y \wedge x \\ &= S(x)y \end{aligned}$$

where the skew-symmetric (i.e. $S = -S^{\mathrm{T}}$) matrix S is defined by

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

Therefore, the angular velocity of frame $\mathcal{F}_b$ with respect to $\mathcal{F}_n$ and expressed in body-fixed frame is [Barfoot, 2017]

$$S(\boldsymbol{\omega}^b) = (\boldsymbol{R}_n^b)^{\mathrm{T}}\dot{\boldsymbol{R}}_n^b \tag{2.5}$$

where the property that the rotation matrix $\boldsymbol{R}_n^b$ belongs to the special orthogonal group

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

is used, thus $(\boldsymbol{R}_n^b)^{\mathrm{T}} = (\boldsymbol{R}_n^b)^{-1}$.

Using the equation (2.4) to resolve the right-hand side of (2.5), and then combining the coefficients with respect to $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ terms, we obtain

$$\begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.6}$$

From the inverse of this equation, it can be found that

$$\dot{\boldsymbol{\eta}}_2^n = \boldsymbol{T}_n^b(\boldsymbol{\eta}_2^n)\boldsymbol{\omega}^b = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \boldsymbol{\omega}^b \tag{2.7}$$

Note that $\boldsymbol{T}_n^b(\boldsymbol{\eta}_2^n)$ has singularities at $\theta = (2k+1)\frac{\pi}{2}, k \in \mathbb{Z}$, this is one of the disadvantages using Euler angles (Tait-Bryan angles) to represent the orientation. In practice, this problem can be avoided by using quaternion, a four parameter representation of the orientation, which is able to describe all possible orientations, and has good computational efficiency [Bestaoui and Hima, 2001]. However, the motion of blimp is considered to be mild, thus the pitch angle will not reach the singularity condition.

Summarizing from (2.3) and (2.7), the blimp kinematic equations can be expressed in vector form as

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_1^n \\ \dot{\boldsymbol{\eta}}_2^n \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_n^b(\boldsymbol{\eta}_2^n) & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{T}_n^b(\boldsymbol{\eta}_2^n) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^b \\ \boldsymbol{\omega}^b \end{bmatrix} \tag{2.8}$$

or

$$\dot{\boldsymbol{\eta}}^n = \boldsymbol{J}(\boldsymbol{\eta}^n)\boldsymbol{\xi}^b \tag{2.9}$$

2.3.4 Dynamic model

In this section the commonly used dynamic model designed for indoor blimp robot is presented. The blimp is depicted in Figure 2.1. To establish the relation between the blimp accelerations and the forces and moments acting on it, the Newton-Euler equation of motion is used [Fossen, 1994; Gomes, 1990; Zufferey et al., 2006].

Moreover due to the fact that the dynamics of the blimp is similar to the dynamics of underwater vehicles [Fossen, 1994; Gomes, 1990; Sagatun and Fossen, 1991], the added-inertia effects are taken into account, and it is shown that the 6-DOF nonlinear dynamic equations of the blimp motion can be expressed as [Fossen, 1994] (see Appendix A for the detail of derivation)

$$\boldsymbol{M}\dot{\boldsymbol{\xi}}^{\boldsymbol{b}} + \boldsymbol{C}(\boldsymbol{\xi}^b)\boldsymbol{\xi}^b + \boldsymbol{D}(\boldsymbol{\xi}^b)\boldsymbol{\xi}^b + \boldsymbol{g}(\boldsymbol{\eta}^n) = \boldsymbol{\tau}^b \tag{2.10}$$

where the terms are:

- $\boldsymbol{M}$: the inertia matrix, containing the blimp inertia, and added-inertia terms;
- $\boldsymbol{C}(\boldsymbol{\xi}^b)$: the matrix of the Coriolis and centripetal terms, which are fictitious forces due to the description of the blimp motion in non-inertial frame $\mathcal{F}_b$;
- $\boldsymbol{D}(\boldsymbol{\xi}^b)$: the damping matrix;
- $\boldsymbol{g}(\boldsymbol{\eta}^n)$: the vector of restoring forces and moments, including the gravity of the whole robot and the buoyancy generated by helium gas in the balloon, they are responsible for keeping the blimp upright;
- $\boldsymbol{\tau}^b$: the vector of control inputs, which is used to describe the propulsion forces and moments generated by actuators acting on the blimp in the body-fixed frame.

These terms are presented below.

2.3.4.1 Restoring forces and moments

The lifting force of the blimp is aerostatic, which means it is independent of the flight speed thanks to the helium gas inside the balloon. From Archimedes' principle, the buoyancy force of the blimp is equal to the weight of the air that the balloon displaces. In Figure 2.1, it is shown that, as a result of the gondola installation at the bottom of the balloon, the CG is below CB. In practice, the resultant force of buoyancy $\boldsymbol{f_B}$ and gravity $\boldsymbol{f_G}$ will keep the airship upright, thus it is called the restoring force.

In addition the gravitational force $\boldsymbol{f_G}$ acts on the CG which is at

$$\boldsymbol{r}_{\boldsymbol{G}}^{b} = \begin{bmatrix} 0 & 0 & z_G \end{bmatrix}^{\mathrm{T}}$$

of the blimp, and the buoyancy force $\boldsymbol{f_B}$ acts at the CB, which is the origin of $\mathcal{F}_b$, i.e. $\boldsymbol{r}_{\boldsymbol{B}}^{b} = \boldsymbol{0}_{3\times 1}$. By using the change of basis equation (2.2), in the body fixed frame there is:

$$\boldsymbol{f}_{\boldsymbol{G}}^{b} = \boldsymbol{R}_{\boldsymbol{b}}^{\boldsymbol{n}} \boldsymbol{f}_{\boldsymbol{G}}^{n} = (\boldsymbol{R}_{\boldsymbol{n}}^{\boldsymbol{b}})^{\mathrm{T}} \boldsymbol{f}_{\boldsymbol{G}}^{n} = (\boldsymbol{R}_{\boldsymbol{n}}^{\boldsymbol{b}})^{\mathrm{T}} \begin{bmatrix} 0 \\ 0 \\ f_G \end{bmatrix}$$

Similarly

$$\boldsymbol{f}_{\boldsymbol{B}}^{b} = (\boldsymbol{R}_{\boldsymbol{n}}^{\boldsymbol{b}})^{\mathrm{T}} \boldsymbol{f}_{\boldsymbol{B}}^{n} = (\boldsymbol{R}_{\boldsymbol{n}}^{\boldsymbol{b}})^{\mathrm{T}} \begin{bmatrix} 0 \\ 0 \\ -f_B \end{bmatrix}$$

with

$$\begin{aligned} f_G &= mg, \\ f_B &= \rho_{\text{air}} V g, \\ V &= \tfrac{4}{3}\pi a b^2 \end{aligned}$$

where m is the mass of the blimp, g is the Earth gravitational acceleration, ρ_{air} is the air density, and V is the volume of ellipsoid shape balloon with semi-axes a and b. Consequently, the restoring forces and moment vector in $\mathcal{F}_b$ is

$$\boldsymbol{g}(\boldsymbol{\eta}^{\boldsymbol{n}}) = -\begin{bmatrix} \boldsymbol{f}_{\boldsymbol{G}}^{\boldsymbol{b}} + \boldsymbol{f}_{\boldsymbol{B}}^{\boldsymbol{b}} \\ \boldsymbol{r}_{\boldsymbol{G}}^{\boldsymbol{b}} \wedge \boldsymbol{f}_{\boldsymbol{G}}^{\boldsymbol{b}} + \boldsymbol{r}_{\boldsymbol{B}}^{\boldsymbol{b}} \wedge \boldsymbol{f}_{\boldsymbol{B}}^{\boldsymbol{b}} \end{bmatrix} \tag{2.11}$$

Notice that the sign of $\boldsymbol{g}(\boldsymbol{\eta})$ must be changed since it appears on the left-hand side of the Newton's Second Law, see (2.10). Explicitly

$$\boldsymbol{g}(\boldsymbol{\eta}^{\boldsymbol{n}}) = -\begin{bmatrix} -(f_G - f_B)\sin\theta \\ (f_G - f_B)\cos\theta\sin\phi \\ (f_G - f_B)\cos\theta\cos\phi \\ -z_G f_G \cos\theta\sin\phi \\ -z_G f_G \sin\theta \\ 0 \end{bmatrix} \tag{2.12}$$

2.3.4.2 Propulsion forces and moments

As it has been discussed before, the propulsive forces of small indoor blimps are usually generated by motor with propellers, because at low flight speed, the control surface (like rudder and elevator) are not efficient. However, the number of thrusters and their mounting positions and orientations can vary from different applications. Therefore, the propulsion forces and moments vector $\boldsymbol{\tau}^b$ is usually different in different cases of blimp robot.

Generally, in other works on indoor blimp robots, the motors are assumed to be ideal, which means their effects are proportional to the commands. Moreover, in majority of the cases, the propeller fluxes and motor torques are ignored for the simplicity of modeling. As a consequence, the propulsion forces $\boldsymbol{\tau}^b$ depends only on the motor commands and the installation of motors. For now, we just use the following notation to denote the term $\boldsymbol{\tau}^b$

$$\boldsymbol{\tau}^b = \begin{bmatrix} f_{px} & f_{py} & f_{pz} & \tau_{px} & \tau_{py} & \tau_{pz} \end{bmatrix}^{\mathrm{T}} \tag{2.13}$$

2.3.4.3 Damping forces and moments

Due to air friction, the aerodynamic damping is depended on the velocity of the blimp. In general, there are two types of air frictions: the drag force proportional to the velocity for laminar flow and proportional to the squared velocity for turbulent flow [Curtiss et al., 1976; Hoerner, 1958]. In the work of [Fossen, 1994], the authors modeled the damping forces and moments of slowly moving underwater vehicle by ignoring the terms higher than second-order, and

proposed a diagonal structure of $\boldsymbol{D}(\boldsymbol{\xi}^{\boldsymbol{b}})$, there is

$$\boldsymbol{D}(\boldsymbol{\xi}^{\boldsymbol{b}}) = -\text{diag}\begin{bmatrix} D_{v_x} + D_{v_x^2}|v_x^b| \\ D_{v_y} + D_{v_y^2}|v_y^b| \\ D_{v_z} + D_{v_z^2}|v_z^b| \\ D_{\omega_x} + D_{\omega_x^2}|\omega_x^b| \\ D_{\omega_y} + D_{\omega_y^2}|\omega_y^b| \\ D_{\omega_z} + D_{\omega_z^2}|\omega_z^b| \end{bmatrix} \tag{2.14}$$

where D_{v_x}, D_{v_y}, D_{v_z}, D_{ω_x}, D_{ω_y}, D_{ω_z} are the linear damping coefficients, and $D_{v_x^2}$, $D_{v_y^2}$, $D_{v_z^2}$, $D_{\omega_x^2}$, $D_{\omega_y^2}$, $D_{\omega_z^2}$ are the quadratic damping coefficients. According to the authors, the uncoupled damping model (2.14) works well in case of low speed and highly symmetrical ellipsoid hull [Fossen, 1994].

In [Zufferey et al., 2006], the authors also proposed practical method to identify the twelve damping coefficients by giving known thrust, measuring the velocity (linear and angular) and fitting the model to test data. In our tests, we followed a similar approach to identify the drag coefficients, but we made further simplifications, and it will be presented in Section 2.6.

2.3.4.4 Inertia matrix

The inertia matrix $\boldsymbol{M}$ contains both the rigid body (RB) inertia $\boldsymbol{M}_{\text{RB}}$ and the added inertia $\boldsymbol{M}_{\text{Added}}$ (See Appendix A). The rigid body inertia matrix can be written as [Fossen, 1994]

$$\boldsymbol{M}_{\text{RB}} = \begin{bmatrix} m\mathbb{I}_{3\times 3} & -mS(\boldsymbol{r}_{\boldsymbol{G}}^{\boldsymbol{b}}) \\ mS(\boldsymbol{r}_{\boldsymbol{G}}^{\boldsymbol{b}}) & \boldsymbol{I}_{\text{RB}} \end{bmatrix}$$

where $\mathbb{I}_{3\times 3}$ is the identity matrix of dimension 3×3, $\boldsymbol{I}_{\text{RB}}$ is the moment of inertia matrix with respect to CB. Recall that $\boldsymbol{r}_{\boldsymbol{G}}^{\boldsymbol{b}}$ is the coordinate of CG in frame $\mathcal{F}_b$, and $S(\cdot)$ is the skew-symmetric matrix operator. Taking into account that the robot has two planes of symmetry ($xz-$ and $yz-$ planes), thus the rigid body matrix of inertia is simplified as [Fossen, 1994]

$$\boldsymbol{M}_{\mathrm{RB}} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & 0 \\ 0 & m & 0 & -mz_G & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -mz_G & 0 & I_x & 0 & 0 \\ mz_G & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{2.15}$$

Then let us focus on the added inertia of the blimp. In order to allow the blimp to move in the air, the robot has to push some volumes of the surrounding fluid aside, this phenomenon has a noticeable impact on the blimp, which is a buoyant vehicle and has similar density as air fluid. In result, the phenomenon acts as if the blimp has a bigger inertia than the measured one, which is not accounted in the standard rigid body inertia matrix (2.15).

The additional effect is modeled as added-inertia, including added-mass and added moment of inertia [Fossen, 1994]. Under the assumption that the indoor blimp robot moves slowly and it has three planes of symmetry for the ellipsoid shape hull, the added-inertia can be expressed as

$$\boldsymbol{M}_{\mathrm{Added}} = \mathrm{diag} \begin{bmatrix} m_{A_x} \\ m_{A_y} \\ m_{A_z} \\ I_{A_x} \\ I_{A_y} \\ I_{A_z} \end{bmatrix}$$

Then the global inertia matrix $\boldsymbol{M}$ is derived as the sum of $\boldsymbol{M}_{\mathrm{RB}}$ and added-

inertia matrix $\boldsymbol{M}_{\text{Added}}$ [Fossen, 1994]

$$\boldsymbol{M} = \boldsymbol{M}_{\text{RB}} + \boldsymbol{M}_{\text{Added}} = \begin{bmatrix} m'_x & 0 & 0 & 0 & mz_G & 0 \\ 0 & m'_y & 0 & -mz_G & 0 & 0 \\ 0 & 0 & m'_z & 0 & 0 & 0 \\ 0 & -mz_G & 0 & I'_x & 0 & 0 \\ mz_G & 0 & 0 & 0 & I'_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I'_z \end{bmatrix} \tag{2.16}$$

where

$$\begin{cases} m'_x = m + m_{A_x} \\ m'_y = m + m_{A_y} \\ m'_z = m + m_{A_z} \\ I'_x = I_x + I_{A_x} \\ I'_y = I_y + I_{A_y} \\ I'_z = I_z + I_{A_z} \end{cases}$$

It is worth to mention that the ellipsoid shape of the balloon suggests that $m_{A_x} < m_{A_y} \approx m_{A_z}$, $I_{A_x} \approx 0$ and $I_{A_y} \approx I_{A_z}$ [Munk, 1936].

The identification of $\boldsymbol{M}$ concerns mainly the diagonal elements, the parts from $\boldsymbol{M}_{\text{RB}}$ are not hard to identify from experiments, the remaining part is added-inertia. According to the work of [Munk, 1934], the added-mass is equal to the multiplication of the fluid density and a volume, where the volume only depends on the geometric outlines of the blimp. It results in the Lamb's k-factors, where k_1 and k_2 are the inertia coefficients of the fraction of the mass displaced by the hull, and k' is the ratio of the added moment of inertia to the moment of inertia of displaced air I_{z_h} [Lamb, 1932]. In the case where the hull is ellipsoid with semi-axes a and b ($a \geq b$), the I_{z_h} is given by [Fossen, 1994; Zufferey et al., 2006]

$$I_{z_h} = \frac{4}{15}\pi\rho a b^2 (a^2 + b^2)$$

Then the added-inertia can be calculated by using the Lamb's k-factors

$$\begin{cases} m_{A_x} = k_1 m \\ m_{A_y} = m_{A_z} = k_2 m \\ I_{A_x} = 0 \\ I_{A_y} = I_{A_z} = k' I_{z_h} \end{cases} \tag{2.17}$$

where the Lamb's k-factors are defined by

$$\begin{cases} k_1 = \frac{\alpha_0}{2-\alpha_0} \\ k_2 = \frac{\beta_0}{2-\beta_0} \\ k' = \frac{e^4(\beta_0-\alpha_0)}{(2-e^2)[2e^2-(2-e^2)(\beta_0-\alpha_0)]} \\ \alpha_0 = \frac{2(1-e^2)}{e^3}(\frac{1}{2}\ln\frac{1+e}{1-e} - e) \\ \beta_0 = \frac{1}{e^2} - \frac{(1-e^2)}{2e^3}\ln\frac{1+e}{1-e} \end{cases} \tag{2.18}$$

where e denotes the ellipsoid eccentricity

$$e = \sqrt{1 - \left(\frac{b}{a}\right)^2}$$

The k-factors as a function of the ellipsoid aspect ratio is shown in Figure 2.2.

Note that a spherical hull has 50% added-mass in all the directions and no added moment of inertia (when $a/b = 1$, $k_1 = k_2 = 0.5$, and $k' = 0$), and as the shape tends to be elongated, the longitudinal added-mass (k_1) decreases, the lateral added-mass and added moment of inertia (k_2 and k') increase.

2.3.4.5 Coriolis and centripetal forces and moments

The Coriolis and centripetal forces are fictitious forces due to the description of the blimp motion in non-inertial frame $\mathcal{F}_b$. The Coriolis force is proportional to the angular velocity and to the linear velocity, it acts in the direction perpendicular to the rotation axis and to the blimp velocity vector. The centripetal forces is proportional to the square of angular velocity, and to the distance of the CG

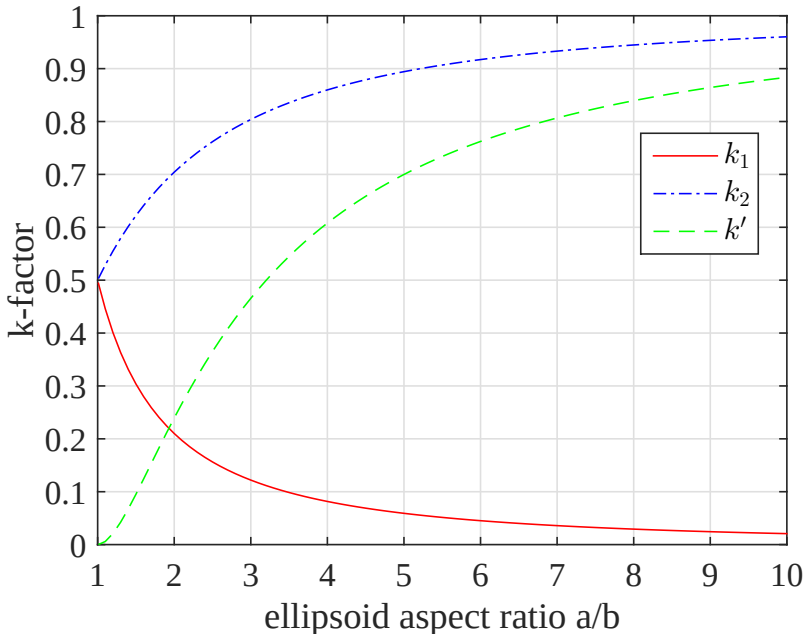

Figure 2.2 – Lamb's k-factors as a function of the ellipsoid hull aspect ratio (from [Zufferey et al., 2006])

from the axis of the rotating frame $\mathcal{F}_b$. The two forces can be expressed in the body-fixed frame and results in the term $\boldsymbol{C}(\boldsymbol{\xi}^{\boldsymbol{b}})\boldsymbol{\xi}^{\boldsymbol{b}}$, where $\boldsymbol{C}(\boldsymbol{\xi}^{\boldsymbol{b}})$ is called Coriolis matrix. From the result of [Sagatun and Fossen, 1991], the Coriolis matrix can be derived directly from the inertia matrix (See also Appendix A).

$$\boldsymbol{C}(\boldsymbol{\xi}^{\boldsymbol{b}}) = \begin{bmatrix} \mathbf{0}_{3\times 3} & -S(\boldsymbol{M}_{11}\boldsymbol{v}^{\boldsymbol{b}} + \boldsymbol{M}_{12}\boldsymbol{\omega}^{\boldsymbol{b}}) \\ -S(\boldsymbol{M}_{11}\boldsymbol{v}^{\boldsymbol{b}} + \boldsymbol{M}_{12}\boldsymbol{\omega}^{\boldsymbol{b}}) & -S(\boldsymbol{M}_{21}\boldsymbol{v}^{\boldsymbol{b}} + \boldsymbol{M}_{22}\boldsymbol{\omega}^{\boldsymbol{b}}) \end{bmatrix} \tag{2.19}$$

where $\boldsymbol{M}_{ij}(i,j=1,2)$ are the four 3×3 sub-matrices of the global inertia matrix $\boldsymbol{M}$. The explicit form of $\boldsymbol{C}(\boldsymbol{\xi}^{\boldsymbol{b}})$ is

$$\boldsymbol{C}(\boldsymbol{\xi}^{\boldsymbol{b}}) = -\begin{bmatrix} 0 & 0 & 0 & 0 & -m'_z v_z & m'_y v_y - mz_G\omega_x \\ 0 & 0 & 0 & m'_z v_z & 0 & -m'_x v_x - mz_G\omega_y \\ 0 & 0 & 0 & -m'_y v_y + mz_G\omega_x & m'_x v_x + mz_G\omega_y & 0 \\ 0 & -m'_z v_z & m'_y v_y - mz_G\omega_x & 0 & -I'_z\omega_z & mz_G v_x + I'_y\omega_y \\ m'_z v_z & 0 & -m'_x v_x - mz_G\omega_y & I'_z\omega_z & 0 & mz_G v_y - I'_x\omega_x \\ -m'_y v_y + mz_G\omega_x & m'_x v_x + mz_G\omega_y & 0 & -mz_G v_x - I'_y\omega_y & -mz_G v_y + I'_x\omega_x & 0 \end{bmatrix}$$

Note that the Coriolis matrix $\boldsymbol{C}(\boldsymbol{\xi}^{\boldsymbol{b}})$ includes the added-inertia terms auto-

matically because the inertia matrix $\boldsymbol{M}$ includes them. According to [Zufferey et al., 2006], the Coriolis and centripetal forces explain why an axial motion of the hull shaped blimp is intrinsically unstable. Because the difference between m'_x and m'_y results in the yaw moment that is induced by the Coriolis and centripetal forces vector $\boldsymbol{C}(\boldsymbol{\xi^b})\boldsymbol{\xi^b}$, any small angle between the $\boldsymbol{X_b}$-axis (blimp forward direction) and the direction of motion will tend to increase [Munk, 1936]. This unstable moment, in some literatures is referred to as Munk moment [Li et al., 2011], is proportional to the difference of lateral and longitudinal Lamb's k-factors $(k_2 - k_1)$ defined in equation (2.18).

In brief, the added-inertia phenomenon explains why the apparent inertia of the blimp is higher than the measured one $\boldsymbol{M}_{\mathrm{RB}}$, and it is also the reason for unwanted behaviors such as the yawing moments when the blimp moves forward [Zufferey et al., 2006].

At this point, we have presented the dynamic model (2.10) and its components including restoring forces (2.12), propulsion forces, damping matrix (2.14), inertia matrix (2.16) and Coriolis and centripetal matrix (2.19) for the indoor blimp robot.

To summarize, the 6-DOF dynamic model of the blimp is written as

$$
\begin{aligned}
m'_x\dot{v}^b_x + m\dot{\omega}^b_y z_G - \omega^b_z(m'_y v^b_y - m\omega^b_x z_G) + m'_z\omega^b_y v^b_z - v^b_x(D_{v_x} + D_{v^2_x}|v^b_x|) + (f_G - f_B)\sin\theta &= f_{px}\\
m'_y\dot{v}^b_y - m\dot{\omega}^b_x z_G + \omega^b_z(m'_x v^b_x + m\omega^b_y z_G) - m'_z\omega^b_x v^b_z - v^b_y(D_{v_y} + D_{v^2_y}|v^b_y|) - (f_G - f_B)\cos\theta\sin\phi &= f_{py}\\
m'_z\dot{v}^b_z + \omega^b_x(m'_y v^b_y - m\omega^b_x z_G) - \omega^b_y(m'_x v^b_x + m\omega^b_y z_G) - v^b_z(D_{v_z} + D_{v^2_z}|v^b_z|) - (f_G - f_B)\cos\theta\cos\phi &= f_{pz}\\
I'_x\dot{\omega}^b_x - m\dot{v}^b_y z_G + I'_z\omega^b_y\omega^b_z - v^b_z(m'_y v^b_y - m\omega^b_x z_G) - \omega^b_z(I'_y\omega^b_y + mv^b_x z_G) + m'_z v^b_y v^b_z&\\
- \omega^b_x(D_{\omega_x} + D_{\omega^2_x}|\omega^b_x|) + z_G f_G\cos\theta\sin\phi &= \tau_{px}\\
I'_y\dot{\omega}^b_y + m\dot{v}^b_x z_G - I'_z\omega^b_x\omega^b_z + v^b_z(m'_x v^b_x + m\omega^b_y z_G) + \omega^b_z(I'_x\omega^b_x - mv^b_y z_G) - m'_z v^b_x v^b_z&\\
- \omega^b_y(D_{\omega_y} + D_{\omega^2_y}|\omega^b_y|) + z_G f_G\sin\theta &= \tau_{py}\\
I'_z\dot{\omega}^b_z + v^b_x(m'_y v^b_y - m\omega^b_x z_G) - v^b_y(m'_x v^b_x + m\omega^b_y z_G) + \omega^b_x(I'_y\omega^b_y + mv^b_x z_G) - \omega^b_y(I'_x\omega^b_x - mv^b_y z_G)&\\
- \omega^b_z(D_{\omega_z} + D_{\omega^2_z}|\omega^b_z|) &= \tau_{pz}
\end{aligned}
\tag{2.20}
$$

2.4 Simplified model

As it can be seen from the commonly used indoor blimp dynamic model, it has a complex form (2.20), the researchers intend to build the model as close to the

real situation as possible, which requires a large amount of accurate experiment data to identify the parameters, but there are still terms that cannot be modeled precisely or that are not considered in the complex form.

Moreover, some of the hypotheses made to build the complex model are hardly to be true compared with real situation. Here we will discuss some of them. For instance, the assumption of the ellipsoid shape of the blimp hull is not accurate (in our case, the hull has a shorter semi-axis in the lateral direction than the one in vertical direction), thus the Lamb's k-factors for added-inertia calculation has to be determined for specified hull shape, which can be even more complex, and may require expensive equipments like the wind tunnel to make real tests and acquire reliable data. Another assumption made for the dynamic modeling is that the airflow generated by the motor propeller are ignored, which again in our case is not a good approximation for the real case, because the altitude movement of blimp is controlled by two vertically installed motor at the bottom of the hull and close to the surface of the hull (see Figure 2.3). Hence the propellers generate airflow that pass nearby the hull when they rotate, and it will definitely influences the aerodynamic effects like the drag forces appeared in the complex model. In addition, the temperature, pressure and density of air are assumed to be constant. However in indoor environment, as it is observed from the real tests, those properties of air can vary easily and create a big influence on the buoyancy force, hull internal pressure and other characteristics of the blimp robot. For instance, if the testing room is exposed to the sun, the temperature will increase easily (like a greenhouse), and the density of air will decrease, which leads to the loss of buoyancy force of blimp.

In addition, since an indoor blimp robot is considered in this work, then it has to be sufficiently small, and its payload is limited. This limits possible quantity of sensors mounted on the robot and computational complexity of the control and estimation algorithms.

Therefore, it can be seen that the commonly used model intends to build the dynamics for blimp as accurate as possible, which leads to a complex form and needs lots of precise experiment data and time to get an accurate parameter identification result, yet there are still terms and disturbances which are not considered. In contrast, we want to find a balance between the model complex-

ity and accuracy, a solution for the modeling method which is focused on the motion control problem of the robot, and thus it requires robustness to the disturbances. Moreover, we do not have access to expensive experiment equipments to accurately identify all the parameters.

To this end, in this work, further simplifications on the model are made so as to build a nominal model for the blimp motion control. The nominal model should be easy to identify, and can represent the motion with acceptable accuracy. Then a disturbance term is complemented to the nominal model, which represents the difference between nominal model and real situation, it is estimated and compensated in the controller in real-time. Hence, the accuracy of control is assured while the complexity is reduced to a minimum.

2.4.1 Hypotheses for simplified model

In addition to the general hypotheses presented in Section 2.2, further assumptions are made in order to simplify the model of blimp for its motion control.

The blimp is assumed to move in low speed, and it does not have violent motion, therefore the following assumption is made

Assumption 2.1. *The blimp roll ϕ and pitch θ angles are fixed to zero during movement.*

The Assumption 2.1 means that $\phi = \dot{\phi} = \theta = \dot{\theta} = 0$. Substituting the terms into equation (2.6), there is

$$\begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Therefore it is clear that under Assumption 2.1, the angular velocities $\omega_x^b = \omega_y^b = 0$. Hence two DOF out of six are constrained, only the motions v_x^b, v_y^b, v_z^b and ω_z^b are left to be studied.

Next, the propulsion forces and moments of our blimp robot have to be analyzed. The actuators used for the NON-A blimp V2 in our work are four motors with propellers mounted on the gondola, as shown in Figure 2.3.

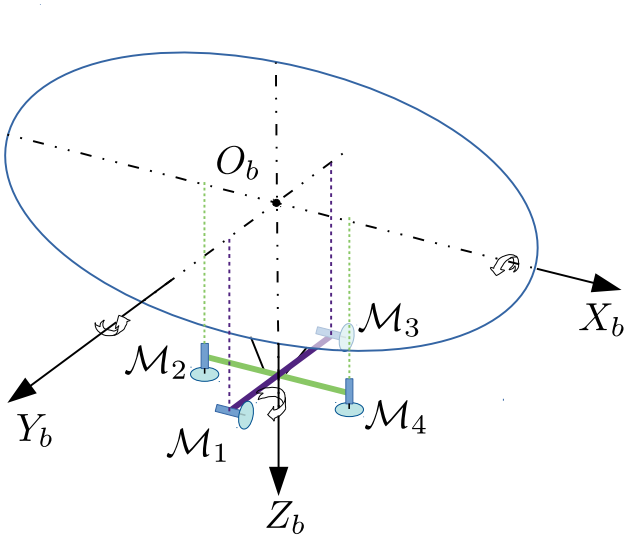

Figure 2.3 – Motor installation for NON-A blimp V2

Two of the motors ($\mathcal{M}_2$ and $\mathcal{M}_4$) are mounted vertically in the plane $\boldsymbol{O_bZ_bX_b}$, at symmetric position with respect to the $\boldsymbol{Z_b}$-axis, they are given same commands at any time in order to adjust the altitude of the blimp. The other two motors ($\mathcal{M}_1$ and $\mathcal{M}_3$) are mounted horizontally in a plane parallel to $\boldsymbol{O_bX_bY_b}$, and in the plane $\boldsymbol{O_bY_bZ_b}$, at symmetric position with respect to the $\boldsymbol{Z_b}$-axis.

Therefore, the propulsion forces of $\mathcal{M}_2$ and $\mathcal{M}_4$ are assumed to be equal in magnitude and always along the $\boldsymbol{Z_b}$-axis direction, the moment of the forces with respect to $\boldsymbol{X_b}$-axis (and $\boldsymbol{Y_b}$-axis, $\boldsymbol{Z_b}$-axis) is always zero. The propulsion forces of $\mathcal{M}_1$ and $\mathcal{M}_3$ are assumed to be always along the $\boldsymbol{X_b}$-axis direction. Hence, with Assumption 2.1, the following assumption is made.

Assumption 2.2. *Under Assumption 2.1, the propulsion forces of motor $\mathcal{M}_2$ and $\mathcal{M}_4$ only take effect in the $\boldsymbol{Z_n}$-axis direction (the vertical direction of inertial frame). In addition, the propulsion forces of motor $\mathcal{M}_1$ and $\mathcal{M}_3$ only take effect in the plane parallel to $\boldsymbol{O_nX_nY_n}$ (horizontal plane of inertial frame).*

From the 6-DOF dynamic model (2.10), with assumptions 2.1 and 2.2, the

model is simplified to

$$\begin{aligned}
m'_x\dot{v}^b_x - m'_y\omega^b_z v^b_y - v^b_x(D_{v_x} + D_{v^2_x}|v^b_x|) &= f_{px}\\
m'_y\dot{v}^b_y + m'_x\omega^b_z v^b_x - v^b_y(D_{v_y} + D_{v^2_y}|v^b_y|) &= f_{py}\\
m'_z\dot{v}^b_z - v^b_z(D_{v_z} + D_{v^2_z}|v^b_z|) + (f_B - f_G) &= f_{pz}\\
(m'_z - m'_y)v^b_y v^b_z - mz_G(\dot{v}^b_y + \omega^b_z v^b_x) &= \tau_{px}\\
(m'_x - m'_z)v^b_y v^b_z + mz_G(\dot{v}^b_x - \omega^b_z v^b_v) &= \tau_{py}\\
I'_z\dot{\omega}^b_z + (m'_y - m'_x)v^b_x v^b_y - \omega^b_z(D_{\omega_z} + D_{\omega^2_z}|\omega^b_z|) &= \tau_{pz}
\end{aligned} \tag{2.21}$$

The fourth and fifth equations can be ignored since we are not interested in the pitch and roll motion of the blimp.

In addition, the blimp is assumed to move slowly, thus the damping coefficients are approximated by only the linear term, which leads to the assumption:

Assumption 2.3. *The blimp moves slowly, thus the damping matrix (2.14) is approximated by the linear terms, i.e.*

$$\boldsymbol{D}(\boldsymbol{\xi^b}) = -diag(\begin{bmatrix} D_{v_x} & D_{v_y} & D_{v_z} & D_{\omega_x} & D_{\omega_y} & D_{\omega_z} \end{bmatrix}^T)$$

Thus the expression (2.21) is simplified to

$$\begin{aligned}
m'_x\dot{v}^b_x - m'_y\omega^b_z v^b_y - D_{v_x}v^b_x &= f_{px}\\
m'_y\dot{v}^b_y + m'_x\omega^b_z v^b_x - D_{v_y}v^b_y &= f_{py}\\
m'_z\dot{v}^b_z - D_{v_z}v^b_z + (f_B - f_G) &= f_{pz}\\
I'_z\dot{\omega}^b_z + (m'_y - m'_x)v^b_x v^b_y - D_{\omega_z}\omega^b_z &= \tau_{pz}
\end{aligned} \tag{2.22}$$

From the explicit form of the simplified model (2.22), the following remarks are made.

Remark 2.1. The movement in vertical direction (altitude movement) and the planar movement (in horizontal plane) of the blimp studied in this work can be decoupled, which means they are independent of each other.

Remark 2.2. The motion control of the indoor blimp robot can be separated into two sub-problems: altitude movement control and planar movement control,

they can be analyzed and solved independently, and then combined together to achieve complete motion control of the blimp.

Note that the inaccuracy caused by the simplification and decoupling of the blimp motion model will be considered in the added disturbance term, this term is estimated in real-time and will be compensated in the controller.

Based on Remark 2.2, the blimp robot altitude movement model and horizontal plane movement model are designed in Section 2.4.2 and 2.4.3 respectively. Then the controller design is presented in Chapter 3 and Chapter 4 respectively. Finally the combination of the controllers and implementation on real robot are carried out in Chapter 5.

2.4.2 Simplified altitude movement model

From Remark 2.1, it can be seen that the blimp movement in the vertical direction ($\boldsymbol{Z_n}$) can be considered independently, thus extracting from the equation (2.22), there is

$$m'_z \dot{v}_z^b - D_{v_z} v_z^b + (f_B - f_G) = f_{pz} \tag{2.23}$$

where m'_z is the apparent mass in the $\boldsymbol{Z_n}$ direction, it is not measured and the influence caused by the added-mass is considered in the parameter identification process.

In addition, from the kinematic model (2.9) and with the Assumption 2.1, the following relation is obtained

$$\dot{\boldsymbol{\eta}}_{\mathbf{1}}^{\boldsymbol{n}} = \begin{bmatrix} \dot{x}^n \\ \dot{y}^n \\ \dot{z}^n \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} \tag{2.24}$$

thus there is $\dot{z}^n = v_z^b$ and $\ddot{z}^n = \dot{v}_z^b$, recall that z^n is the altitude of the blimp to the reference plane $\boldsymbol{O_n X_n Y_n}$, and the propulsive force f_{pz} is supposed to be proportional to square of motor rotation speed, which is modeled as the input. Then the equation (2.23) can be transformed to get the simplified altitude movement dynamic model:

$$\ddot{z} = a_z \dot{z} + b_z u_z + c_z \tag{2.25}$$

where c_z represents the resultant effect of buoyancy and gravity of the blimp, a_z is the coefficient related to air drag, b_z is the coefficient of the input u_z, the superscript $(\cdot)^n$ for the altitude z is omitted for simplicity of notation. Note that the command signal for vertical motors u_z is just a quantity, it is dimensionless, which means the parameter b_z has the same unit as $\ddot{z}$.

Therefore the equation (2.25) represents the nominal model for the blimp altitude motion control problem, parameters a_z, b_z and c_z have to be identified in order to validate the nominal model (Section 2.6.1). Later the disturbance term is complemented to the nominal model to include all the inaccuracies in the simplified model, it is estimated and compensated in the controller to assure control performance, it will be presented in Chapter 3.

2.4.3 Simplified planar movement model

The blimp decoupled movement in horizontal plane is modeled by the rest part of (2.22):

$$\begin{aligned} m'_x \dot{v}_x^b - m'_y \omega_z^b v_y^b - D_{v_x} v_x^b &= f_{px} \\ m'_y \dot{v}_y^b + m'_x \omega_z^b v_x^b - D_{v_y} v_y^b &= f_{py} \\ I'_z \dot{\omega}_z^b + (m'_y - m'_x) v_x^b v_y^b - D_{\omega_z} \omega_z^b &= \tau_{pz} \end{aligned} \tag{2.26}$$

On the horizontal plane, the blimp has a configuration vector

$$\eta_{\text{Horiz}}^n = \begin{bmatrix} x^n & y^n & \psi \end{bmatrix}^{\mathrm{T}}$$

and the instantaneous velocities vector

$$\xi_{\text{Horiz}}^b = \begin{bmatrix} v_x^b & v_y^b & \omega_z^b \end{bmatrix}^{\mathrm{T}}$$

where v_x^b and v_y^b can also be called the lateral velocity and longitudinal velocity respectively.

Note that from the kinematic model (2.9) and with the Assumption 2.1, there is the equality between the time-derivative of yaw angle and the angular velocity

ω_z^b, i.e. $\omega_z^b = \dot{\psi}$. Thus the kinematic equation of the blimp simplified horizontal movement is

$$\dot{\boldsymbol{\eta}}_{\text{Horiz}}^n = \boldsymbol{J}(\boldsymbol{\eta}_{\text{Horiz}}^n)\boldsymbol{\xi}_{\text{Horiz}}^b \tag{2.27}$$

where

$$\boldsymbol{J}(\boldsymbol{\eta}_{\text{Horiz}}^n) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Next, let us analyze the propulsion forces and moments f_{px}, f_{py}, τ_{pz}, they are generated by the two motors $\mathcal{M}_1$ and $\mathcal{M}_3$ mounted horizontally as shown in Figure 2.3. Assume $\mathcal{M}_1$ and $\mathcal{M}_3$ generate propulsive forces f_{right} and f_{left} respectively. Then with Assumption 2.2 there is

$$\begin{cases} f_{px} = f_{\text{left}} + f_{\text{right}} \\ f_{py} = 0 \\ \tau_{pz} = (f_{\text{left}} - f_{\text{right}})l \end{cases}$$

where l is the half distance between the two motors. In reality, the force cannot be measured in real time during operations, instead, the relation between the motor commands and the blimp motion is identified via tests.

Rearranging the blimp planar movement dynamic model (2.26) we get

$$\begin{cases} m_x' \dot{v}_x^b = m_y' v_y^b \omega_z^b + D_{v_x} v_x^b + f_{\text{left}} + f_{\text{right}} \\ m_y' \dot{v}_y^b = -m_x' v_x^b \omega_z^b + D_{v_y} v_y^b \\ I_z' \dot{\omega}_z^b = (m_x' - m_y') v_x^b v_y^b + D_{\omega_z} \omega_z^b + (f_{\text{left}} - f_{\text{right}})l \end{cases} \tag{2.28}$$

recall that m_x', m_y' and I_z' are the apparent mass and moment with respect to different axis; D_{v_x}, D_{v_y} and D_{ω_z} are the corresponding damping coefficients.

In fact, after simplification and decoupling, the blimp movement in the horizontal plane (2.28) is similar to a slider which moves on a 2D horizontal plane and whose velocities are not restricted (unlike nonholonomic wheeled vehicles) [D'Andréa-Novel and Thorel, 2016; Fantoni et al., 1999; Reyhanoglu, 1997].

Remark 2.3. In the third equation, it can be seen that the difference between apparent mass in the lateral and longitudinal direction will cause an unstable yaw moment during the moving forward motion. To simplify the dynamic model, recall that the blimp is supposed to move slowly, and the global mass terms m'_x and m'_y are assumed to be approximately equal, i.e. $m'_x \approx m'_y = m_{\text{Horiz}}$, thus this unstable yaw moment term disappears. Once more, the inaccuracy caused by the hypothesis will be considered in the complemented disturbance term and compensated in the designed controller.

From equation (2.27) there is:

$$\begin{cases} \dot{x}^n = c_\psi v_x^b - s_\psi v_y^b \\ \dot{y}^n = s_\psi v_x^b + c_\psi v_y^b \end{cases}$$

where $c_\psi = \cos\psi$, $s_\psi = \sin\psi$ for simplicity of notation. Taking time-derivative on both sides:

$$\begin{cases} \ddot{x}^n = -s_\psi \dot{\psi} v_x^b + c_\psi \dot{v}_x^b - c_\psi \dot{\psi} v_y^b - s_\psi \dot{v}_y^b \\ \ddot{y}^n = c_\psi \dot{\psi} v_x^b + s_\psi \dot{v}_x^b - s_\psi \dot{\psi} v_y^b + c_\psi \dot{v}_y^b \end{cases}$$

Substituting $\dot{v}_x^b$ and $\dot{v}_y^b$ from (2.28), and under the assumption that the blimp moves slowly thus $m'_x \approx m'_y = m_{\text{Horiz}}$, it becomes:

$$\begin{cases} \ddot{x}^n = c_\psi \frac{f_{\text{left}} + f_{\text{right}}}{m_{\text{Horiz}}} + c_\psi a_x v_x^b - s_\psi a_y v_y^b \\ \ddot{y}^n = s_\psi \frac{f_{\text{left}} + f_{\text{right}}}{m_{\text{Horiz}}} + c_\psi a_y v_y^b + s_\psi a_x v_x^b \end{cases}$$

where $a_x = D_{v_x}/m_{\text{Horiz}}$, $a_y = D_{v_y}/m_{\text{Horiz}}$. Then from the inverse of kinematic model (2.27), substituting the terms v_x^b and v_y^b into $\dot{x}^n$ and $\dot{y}^n$, there is

$$\begin{cases} \ddot{x}^n = c_\psi \frac{f_{\text{left}} + f_{\text{right}}}{m_{\text{Horiz}}} + (a_x c_\psi^2 + a_y s_\psi^2)\dot{x}^n + (a_x c_\psi s_\psi - a_y c_\psi s_\psi)\dot{y}^n \\ \ddot{y}^n = s_\psi \frac{f_{\text{left}} + f_{\text{right}}}{m_{\text{Horiz}}} + (a_y c_\psi^2 + a_x s_\psi^2)\dot{y}^n + (a_x c_\psi s_\psi - a_y c_\psi s_\psi)\dot{x}^n \end{cases}$$

For the angular movement, the kinematic relation (2.27) tells that $\dot{\psi} = \omega_z^b$, under the assumption that the blimp moves slowly thus $m'_x \approx m'_y = m_{\text{Horiz}}$. Hence the

third equation of (2.28) becomes:

$$\ddot{\psi} = \frac{(f_{\text{left}} - f_{\text{right}})l}{I_z'} + a_\psi \dot{\psi}$$

where $a_\psi = D_{\omega_z}/I_z'$.

In summary, we obtain

$$\begin{cases} \ddot{x} = c_\psi \frac{f_{\text{left}}+f_{\text{right}}}{m_{\text{Horiz}}} + (a_x c_\psi^2 + a_y s_\psi^2)\dot{x} + (a_x c_\psi s_\psi - a_y c_\psi s_\psi)\dot{y} \\ \ddot{y} = s_\psi \frac{f_{\text{left}}+f_{\text{right}}}{m_{\text{Horiz}}} + (a_y c_\psi^2 + a_x s_\psi^2)\dot{y} + (a_x c_\psi s_\psi - a_y c_\psi s_\psi)\dot{x} \\ \ddot{\psi} = \frac{(f_{\text{left}}-f_{\text{right}})l}{I_z'} + a_\psi \dot{\psi} \end{cases}$$

Note that the superscript $(\cdot)^n$ for x and y is omitted for simplicity of notation.

For clarity of the expression, rewrite the blimp simplified planar movement model as:

$$\begin{cases} \ddot{x} = c_\psi b u + \kappa_1(a_x, a_y, \psi)\dot{x} + \kappa_3(a_x, a_y, \psi)\dot{y} \\ \ddot{y} = s_\psi b u + \kappa_2(a_x, a_y, \psi)\dot{y} + \kappa_3(a_x, a_y, \psi)\dot{x} \\ \ddot{\psi} = b_\psi v + a_\psi \dot{\psi} \end{cases} \tag{2.29}$$

where u and v are the two control inputs, with $u = u_{\text{left}} + u_{\text{right}}$ and $v = u_{\text{left}} - u_{\text{right}}$; u_{right} and u_{left} are respectively the value of command signal for right and left motors; b and b_ψ are the coefficients related to the control inputs; coefficients $\kappa_1(a_x, a_y, \psi) = a_x c_\psi^2 + a_y s_\psi^2$, $\kappa_2(a_x, a_y, \psi) = a_y c_\psi^2 + a_x s_\psi^2$ and $\kappa_3(a_x, a_y, \psi) = a_x c_\psi s_\psi - a_y c_\psi s_\psi$.

Note that the command signal for right and left motors u_{right} and u_{left} are just quantities, they are dimensionless, which means the parameter b has the same unit as $\ddot{x}$ and the parameter b_ψ has the same unit as $\ddot{\psi}$, and l (the half distance between the two motors) is included in parameter b_ψ.

The equation (2.29) can be considered as the nominal model for blimp planar motion control problem, the parameters a_x, a_y, a_ψ, b and b_ψ should be estimated to validate the nominal model (Section 2.6.2). Then the disturbance term is complemented to the nominal model to represent the difference between simplified nominal model and real situation, it is estimated and compensated in the controller to assure motion control performance, this part will be presented in Chapter 4.

2.5 Sensors

Before presenting the parameter identification of the blimp altitude and planar movement nominal model (2.25) and (2.29), let us take a look at the sensors used for the blimp system which provide measurements for state estimation and control process.

The choice of sensors for the blimp robot system depends on various factors, including the desired operations for the robot to achieve, the hardware restrictions (e.g. weight limit, energy limit, installation position and method), etc. Therefore it puts forward specified requirements on the measuring method, precision and frequency of the sensors.

On the other way, the information provided by the sensors also influences the observer and controller design for the blimp system, and leads to the success or failure of the blimp motion control task.

In this work, the studied blimp robot has a balloon which has an ellipsoid-like shape, with a length of 105cm, a width of 55cm, and the height is 71cm, it has a volume of about 0.2m^3, when filled with helium, the balloon can carry a total weight of about 200 grams. The size of the blimp is miniature compared to other airships in the related works.

Notice that the 200 grams of payload has to include all the hardwares including gondola structure to fix the micro-controller board to the hull, the motors with propellers as the system actuators, the battery for power supply and wireless communication devices for the possibility of data exchange with host computer. Therefore only low weight sensors and actuators can be integrated in an embedded micro-system, which means the measurement of on-board sensors cannot be very accurate.

At first, for the design of the blimp robot prototype, the following sensors are chosen, as shown in Figure 2.4.

The IMU MPU-6050 (on the left of Figure 2.4) combines a MEMS 3-axis accelerometer and a 3-axis gyroscope in a miniature package, which can be used to estimate relative position, orientation, velocity and acceleration of the blimp [InvenSense, 2018]. But due to the integration of measurement (including error) to get position, the result suffers from drifting problem. Thus it is preferable to

Figure 2.4 – Sensors used for NON-A blimp prototype: IMU MPU-6050 (left), US range finder LV-MaxSonar-EZ1 (middle), wireless camera ALM-2451G (right)

only use the IMU measurements for orientation estimation, and relative position estimation during a short period.

It is worth to mention that the MPU-6050 is a low cost and light weight sensor, although digital filter is implemented inside the chip to eliminate high-frequency noise, the accuracy of the measurements is still limited. Hence, when using the sensor for yaw angle estimation (as the roll and pitch movements are ignored), the result is not satisfactory especially after long time. As one of the main advantages of the blimp robot is its long endurance in air and autonomous operation time, if the IMU MPU-6050 is the only source of information to determine the pose of the blimp, it will be hard to design powerful observer and controller to assure the performance of blimp motion control.

The US range finder is an active exteroceptive sensor which can measure directly the distance from the robot to its surrounding within the dispersal cone of the sensor. Thus it is an ideal complementary sensor to the IMU, and if possible, we can install multiple US sensors around the blimp robot pointing to different directions and get the relative position estimation of robot inside the environment.

However, the acceptable payload of the balloon only allows one of such sensor to be mounted, thus in the prototype of our blimp robot, one LV-MaxSonar-EZ1 US range finder (on the middle of Figure 2.4) is installed vertically downward on the control board to measure the distance from robot to the ground (or other obstacles below the robot). The sensor gives readings from 0 to 255

inches (0 to 6.45m) with a resolution of 1 inch (2.54cm) [MaxBotix, 2018]. In the parameter identification process, the US sensor is used to give altitude measurements (Section 2.6.1). But it is worth to mention that the dispersal cone of the sensor limits the precision of measurements and creates possible jumps during successive measurements, and the speed of sound limits the frequency of acquisition and introduces delay into the control loop.

As it is mentioned before, the visual sensors like cameras are powerful sensors which can provide enormous amount of environmental information in images. Hence, in the prototype of blimp robot, the wireless camera ALM-2451G (on the right of Figure 2.4) is mounted horizontally towards front on the gondola [Aliveal, 2018]. It is supposed to take pictures of the environment in front of the robot and send it to PC for processing and extracting useful informations. Together with up to date technologies like SLAM (simultaneous localization and mapping), the robot can localize itself in unknown environment and achieve complex tasks. But during tests it is observed that the images transmitted wirelessly to the PC are obscure and distorted, with low frequency, thus in this work, the camera is not used afterwards. However, it is worth to note that if the blimp is supposed to operates completely autonomously in any indoor environments, the camera is probably the best choice to provide environmental information.

After the first trials on the blimp prototype and summary of experiences, it is finally decided to implement a camera capturing system OptiTrack in the testing room to track the robot and obtain its position and orientation measurements. The OptiTrack system uses infrared waves to capture the reflective markers mounted on blimp control board, and solves the pose of the robot at a rate of 100 frames per second, and the precision for position measurement is 1mm [NaturalPoint, 2018]. A schema for OptiTrack system is shown in Figure 2.5. In fact, the camera capturing system cannot be called as a sensor for the robot, the image processing and pose estimation are achieved by the camera system and then the result is sent to the blimp control system. The OptiTrack enhanced blimp control system will be presented in detail later in Chapter 5.

The advantages of using the camera capturing system are that it provides high precision localization result and orientation measurements of the robot,

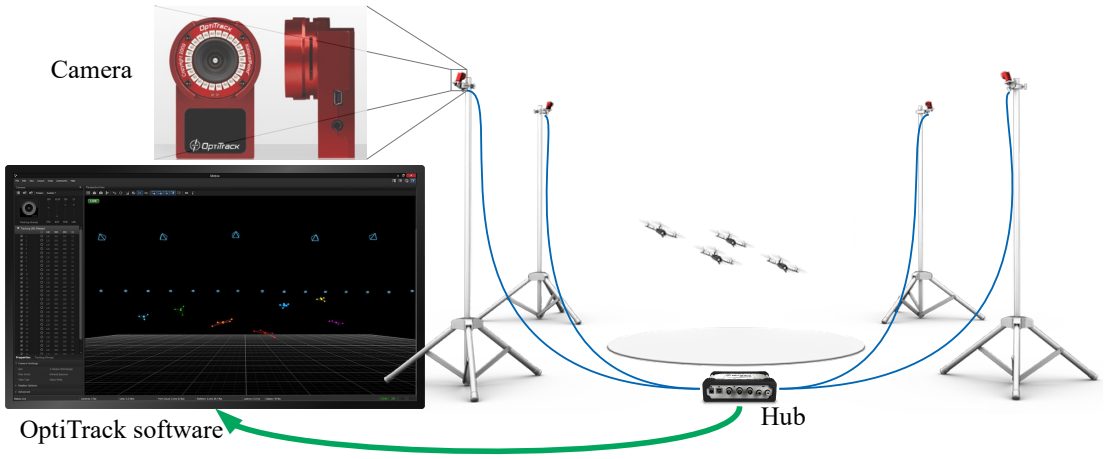

Figure 2.5 – Schema of OptiTrack camera capturing system in robotics application (from [NaturalPoint, 2018])

which do not drift with time, and the frequency is high enough even for some violent movements. But the disadvantages are also remarkable, such that the system is fixed in the environment, which means the robot can only be localized in the testing room, it is not autonomous. In addition, the system is expensive, which limits its usage. However, in this work, the study of the blimp motion control problem is focused, thus the autonomous localization and navigation of blimp in unknown environment will be considered in the future, for now, the OptiTrack system is used for the parameter identification of the NON-A blimp V2 planar movement nominal model (Section 2.6.2).

2.6 Parameter identification

It is worth to mention that the parameters to be identified (i.e. a_z, b_z and c_z of altitude model (2.25), and a_x, a_y, a_ψ, b and b_ψ of planar movement model (2.29)) are composed of physically measurable quantities such as the rigid body mass, the length, and the force (via a force meter) etc. But instead of measuring independently those quantities and calculating the parameters, we chose to identify the parameters directly from some designed tests. The reasons are listed as follows:

- The added-inertia of the balloon is hard to be determined;
- We do not have the equipments to measure the forces or the damping coefficients in the lab;
- There are fewer parameters to be identified via tests than the physical quantities to be measured directly;
- The parameters identified via tests are reliable since they are measured from the robot actual performance.

The parameter identification process for altitude movement nominal model (2.25) and planar movement nominal model (2.29) are carried out separately.

As it has been mentioned before, during the development of our blimp robot, first, the prototype of robot is designed and tested for the altitude control task (presented in Chapter 3). Then, due to the hardware restrictions observed throughout the experiments, which lead to difficulties in controller design, a new version of the blimp robot is conceived, which is referred to as NON-A Blimp V2 (presented in detail in Chapter 5). Although the model remains the same, the parameter identification should be proceeded once more on the NON-A Blimp V2 for the altitude movement nominal model. In addition the horizontal plane movement model is tested and validated on the NON-A Blimp V2, and controller is designed for planar movement (presented in Chapter 4).

Therefore in this section, first, the parameter identification for altitude nominal model is discussed both on the prototype and V2 of the blimp robot, then considering the planar movement nominal model, the parameters are identified only for the V2 of the blimp robot.

2.6.1 For altitude movement nominal model

The basic idea to identify the parameters a_z, b_z and c_z in the nominal model (2.25) is to give the altitude control motor a step input signal and measure the blimp altitude change by either on-board US sensor (for blimp prototype) or camera capturing system (for blimp V2), then based on the relationship between altitude and input, the parameters can be identified.

However, the nominal model (2.25) has to be modified a little in order to incorporate the hardware restrictions observed during real tests.

2.6.1.1 On NON-A Blimp Prototype

The NON-A blimp prototype is installed with an US range finder which points vertically downward to the ground to measure the altitude of the robot in indoor environment. Due to the sound speed of 0.3m/ms, the frequency of measurement of US sensor is limited, and during our parameter identification process, the sensor measures altitude at a rate of 10Hz. The raw data of the sensor is proceeded by the on-board micro-controller Arduino FIO, then the solved altitude result is sent to PC via wireless communication module XBee. The control board of NON-A blimp prototype is shown in Figure 2.6.

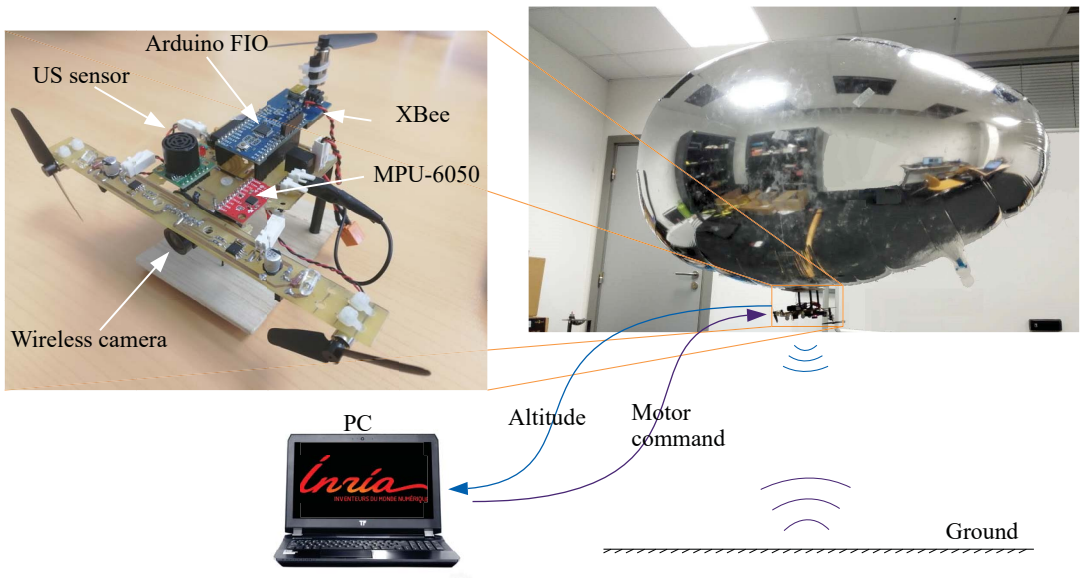

Figure 2.6 – Control board of NON-A Blimp Prototype and its communication loop

In the tests for blimp prototype, a time-delay in the control loop is observed and cannot be ignored, thus the altitude nominal model (2.25) has to be modified so as to incorporate the time-delay. Moreover, due to hardware defect, the chosen coreless DC motor with propeller demonstrates different efficiency when

it rotates in clockwise/counter-clockwise direction with the same absolute motor command values. Besides, the air resistance coefficient differs when the blimp moves upward and downward on account of the installation of control board at bottom of balloon.

Therefore the altitude nominal model is modified, according to the blimp prototype hardware, into

$$\ddot{z}(t) = a_{z\sigma}\dot{z}(t) + b_{z\sigma}u_z(t-\tau) + c_z \tag{2.30}$$

where σ is used to indicate that the coefficient is switched for different cases, τ is the time-delay. Note that in practice, for NON-A blimp prototype robot (respectively V2 robot), the command value u_z is quantized in the range of $[-255, 255]$ (respectively $[-1000, 1000]$), and $u_z \in \mathbb{Z}$.

Using Laplace transform for the differential equation (2.30), we get:

$$s^2Z(s) - sz(0) - \dot{z}(0) = a_{z\sigma}(sZ(s) - z(0)) + b_{z\sigma}U_z(s)e^{-\tau s} + c_z s^{-1}$$

In the step input tests, the blimp prototype is carefully adjusted such that the buoyancy is approximately equal to the gravity, which means $c_z \approx 0$. In addition, assume that at the beginning, the blimp is static, i.e. $\dot{z}(0) = 0$. The Laplace transformation of the step input is

$$U_z(s) = \frac{N}{s}$$

where N is the step value for the vertical motor of the blimp prototype, and it is an integer in the range of $[-255, 255]$. Then the output becomes:

$$Z(s) = \frac{z(0)}{s} + \frac{Nb_{z\sigma}}{s^2(s - a_{z\sigma})}e^{-\tau s}$$

Using inverse Laplace transform, there is:

$$z(t) = z(0) + N\left(\frac{b_{z\sigma}}{a_{z\sigma}^2}e^{a_{z\sigma}(t-\tau)} - \frac{b_{z\sigma}}{a_{z\sigma}^2} - \frac{b_{z\sigma}}{a_{z\sigma}}(t-\tau)\right) \tag{2.31}$$

Here in the equation (2.31), the time-delay τ is measured manually from

the jump moment of the step input to the robot altitude change. Then after τ is known, the Levenberg-Marquardt nonlinear least squares algorithm can be used to determine the coefficients $a_{z\sigma}$ and $b_{z\sigma}$ such that the curve $z(t)$ defined by (2.31) best fits measured data [Seber and Wild, 2003]. This is done in MATLAB by the NLINFIT function.

Note that this procedure has to be carried out multiple times for different step input value N (also for negative N), and then take the average of the results to get a better identification of the parameters.

Some of the results of the parameter identification for the blimp prototype under positive step input are shown in Figure 2.7.

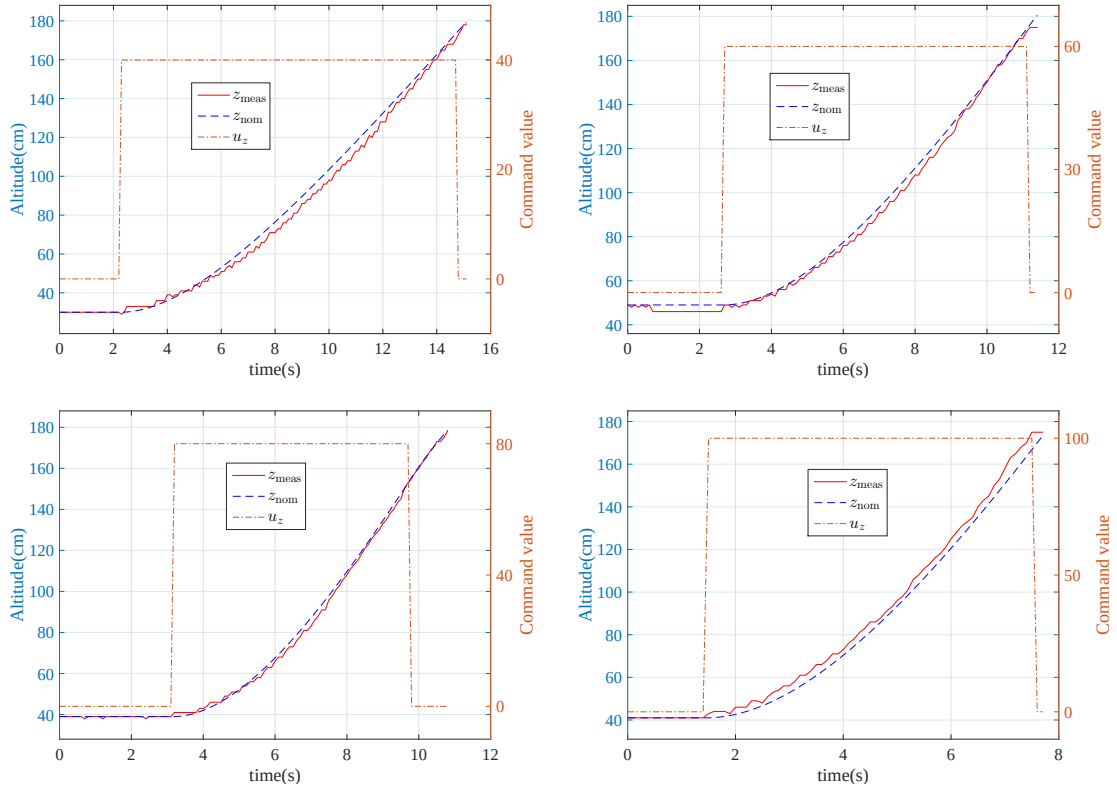

Figure 2.7 – Altitude nominal model parameter identification result comparison

Note that here the time-delay has been estimated and used to shift the curves such that the altitude start to change from the moment step input is given. Moreover, for the ease of comprehension, the curve is shown in a way such that

when the command is positive, the blimp actual altitude with respect to the ground increases. The value N of the step input are respectively 40 (top left), 60 (top right), 80 (bottom left) and 100 (bottom right), z_{meas} curves are the altitude results measured by US sensor, and z_{nom} are the results generated by nominal model with same identified parameters $a_{z\sigma}$ and $b_{z\sigma}$.

It is shown that the parameter identification result is satisfactory and reflects the real performance (altitude response) of blimp under different step inputs.

The same procedure is performed again for different command values but for negative signs, which means for the blimp to descend.

Finally, the parameter identification result for altitude movement nominal model on the NON-A blimp prototype is given below.

The nominal time-delay in the control loop is:

$$\tau_{\text{nom}} = 0.6\text{s}$$

as for the $a_{z\sigma}$ and $b_{z\sigma}$, results are shown in Table 2.1

Case	$\dot{z} < 0$	$\dot{z} \geq 0$
$u_z < 0$	$a_{z\sigma} = -0.28412$	$a_{z\sigma} = -0.34316$
	$b_{z\sigma} = 0.11214$	$b_{z\sigma} = 0.11214$
$u_z \geq 0$	$a_{z\sigma} = -0.28412$	$a_{z\sigma} = -0.34316$
	$b_{z\sigma} = 0.06149$	$b_{z\sigma} = 0.06149$

Table 2.1 – Altitude nominal model parameter identification result for NON-A blimp prototype

Remark 2.4. It is worth to mention that in the parameter identification process, we did not measure any of the physical values such as the rigid body mass, or the propulsion force etc. Instead, a relation between the time-derivatives of altitude and the input is established and the parameters are identified via real test measurements. Therefore, the nominal model can reflect sufficiently well the actual performance of the blimp when it is in motion, and the complex modeling terms such as the added-mass are also included in the results.

As it is indicated by the parameter identification result, there is delay in the system and the nominal model parameter is switched according to its state and the input, hence depending on that, the controller for altitude movement is designed and will be presented in Chapter 3.

2.6.1.2 On NON-A Blimp V2

Since on the prototype of NON-A blimp, only one US range finder is used to measure the blimp altitude, and it is not possible to mount more of them due to the payload limit, thus the position information is not complete to achieve planar and full movement control for the robot. Furthermore, the time-delay increases the difficulties in controller design and degrades the performance. That is why in the second version of the blimp design, we decided to use the motion capturing system OptiTrack for the localization of blimp in indoor environment and develop the control law.

Therefore on the new hardware NON-A Blimp V2, the altitude movement nominal model parameters have to be re-identified. Because the OptiTrack solves the pose information of the blimp and send it to the PC via Ethernet UDP connection, the time-delay is greatly reduced to 30ms in the system for blimp V2. It is then reasonable to ignore it in the loop because the blimp is assumed to move slowly. The hardware design and OptiTrack enhanced control system will be presented in Chapter 5.

The parameter identification process is same to the one presented before for the blimp prototype, and the equation (2.31) is simplified for $\tau = 0$. There are two motors in charge of the altitude control for the blimp V2 ($\mathcal{M}_2$ and $\mathcal{M}_4$ as shown in Figure 2.3), they are given same step input and the value N is an integer in the range $[-1000, 1000]$. Similar to the blimp prototype motors, the motors for blimp V2 also have different efficiencies in clockwise/counter-clockwise rotation given same input value N but opposite signs. Thus multiple tests are made with different N values for both positive and negative signs.

The altitude is measured by OptiTrack with a rate of 100Hz. Then the parameters $a_{z\sigma}$ and $b_{z\sigma}$ can be identified by the nonlinear least squares algorithm. The result is given in Table 2.2.

Case	$\dot{z} < 0$	$\dot{z} \geq 0$
$u_z < 0$	$a_{z\sigma} = -0.035$	$a_{z\sigma} = -0.0366$
	$b_{z\sigma} = 0.0628$	$b_{z\sigma} = 0.0628$
$u_z \geq 0$	$a_{z\sigma} = -0.035$	$a_{z\sigma} = -0.0366$
	$b_{z\sigma} = 0.0424$	$b_{z\sigma} = 0.0424$

Table 2.2 – Altitude nominal model parameter identification result for NON-A blimp V2

The detailed hardware design about the NON-A blimp V2 will be presented in Chapter 5.

2.6.2 For planar movement nominal model

The parameters a_x, a_y, a_ψ, b and b_ψ should be estimated to validate the blimp planar movement nominal model (2.29). In the paper of [Zufferey et al., 2006], the authors proposed a method to estimate the damping coefficients (a_x, a_y, a_ψ in our model), they gave a known thrust in the forward direction and measure the constant forward velocity which the blimp can reach. Then with the assumption that the propulsion force (which is known) is equal to the damping force, the damping coefficients can be estimated. But for our test conditions, this identification process is not easy to perform, since there is no available device to measure the propulsion force of the two motors in horizontal plane ($\mathcal{M}_1$ and $\mathcal{M}_3$ as shown in Figure 2.3), and it is hard to keep the blimp move only in forward direction by giving same command to the motors, and also hard to judge whether the blimp has reached constant velocity or not.

For the purpose of simplifying the parameters identification process, the two damping coefficients in lateral and longitudinal direction are assumed to be approximately equal, i.e. $a_x = a_y$, thus in equation (2.29), the terms $\kappa_1(a_x, a_y, \psi) = \kappa_2(a_x, a_y, \psi) = a_x = a_y$, and $\kappa_3(a_x, a_y, \psi) = 0$. The identification process is performed by two different types of tests, where the first one is for identifying a_x, a_y, b by the forward and backward movement of the blimp without turning, and the second one is for identifying a_ψ and b_ψ by the spinning motion without changing position.

In the first type of tests, the blimp is first manually pushed forward and moves along a straight line in its longitudinal direction, and no command is sent to the horizontal motors $\mathcal{M}_1$ and $\mathcal{M}_3$, i.e. $u = 0$, $v = 0$, the robot decelerates by the damping forces, its position is measured by OptiTrack motion capturing system at a rate of 100Hz. The equation is

$$\begin{cases} \ddot{x} = a_x \dot{x} \\ \ddot{y} = a_y \dot{y} \end{cases}$$

Then the differentiator is used to get the first and second-order time-derivatives of the position $\dot{x}$, $\ddot{x}$, $\dot{y}$ and $\ddot{y}$. The detailed presentation of the used differentiator will be carried out in Section 3.3 of Chapter 3. By applying least squares method, the parameter $a_x = a_y$ can be identified. The result is

$$a_x = a_y = -0.24$$

In the next step, the identified parameters for air friction is used in the nominal model, and constant command is given at same time to the motors $\mathcal{M}_1$ and $\mathcal{M}_3$, i.e. $u > 0$ and $v = 0$, the propulsion forces make the blimp move forward, and measurements of position and orientation in the straight line part (where ψ can be assumed invariant) of the trajectories are used to estimate b. The equation is

$$\begin{cases} \ddot{x} = c_\psi b(\text{sign}(u))u + a_x \dot{x} \\ \ddot{y} = s_\psi b(\text{sign}(u))u + a_y \dot{y} \end{cases}$$

Here the coefficient for motor input b is denoted by $b(\text{sign}(u))$ due to the fact that the blimp horizontal motor $\mathcal{M}_1$ and $\mathcal{M}_3$ have different efficiencies when they rotate in clockwise/counter-clockwise direction given same input value but opposite signs.

Again the differentiator is used to get the first and second-order time-derivatives of the position $\dot{x}$, $\ddot{x}$, $\dot{y}$ and $\ddot{y}$. The ψ is measured by OptiTrack and in the straight line part, ψ is approximately constant, then by least squares method, the parameter b can be identified. Note that this procedure has to be done multiple times for different input value u, in order to get a better accuracy on the result. Some

of the identification results of b are shown in Table 2.3.

u	50	80	100	120	140	160	200	250
b	0.0858	0.0816	0.0823	0.0825	0.0853	0.0871	0.0802	0.0787

Table 2.3 – Planar nominal model parameter b identification results for different inputs u

It is shown that the estimated values of b are close with different input u, thus the average of the result is taken as the final result. The same procedure is executed for the negative u values ($u < 0$ and $v = 0$). And finally, the first type of tests give the following parameter identification results.

$$\begin{aligned} a_x &= a_y = -0.24 \\ b(u) &= \begin{cases} 0.0822 & \text{if} \quad u \geq 0 \\ 0.0527 & \text{if} \quad u < 0 \end{cases} \end{aligned} \tag{2.32}$$

In the second type of tests, the coefficients a_ψ and b_ψ are identified. Similarly, the blimp is first given a manual torque to spin around its $\boldsymbol{Z_b}$-axis without change of position, the horizontal motors are given zero commands. The yaw angle is measured by the camera system, and the blimp decelerates by the damping torque $a_\psi \dot{\psi}$. Then $\ddot{\psi}$ and $\dot{\psi}$ are calculated by differentiator and then parameter a_ψ is identified as

$$a_\psi = -0.20$$

Then after a_ψ is identified and substituted in the yaw dynamic equation in nominal model, the motors $\mathcal{M}_1$ and $\mathcal{M}_3$ are given step input with same value but opposite signs at same moment, i.e. $u = 0$ and $v > 0$ (or $v < 0$), to make the blimp rotating while not changing its position in horizontal plane. Multiple tests are accomplished for different values of v, the results are shown in Table 2.4.

Note that the two horizontal motors $\mathcal{M}_1$ and $\mathcal{M}_3$ have good consistency such that when giving same command value but opposite sign, they rotate in opposite direction and generate torque to make the blimp spinning, from the Table 2.4 it can be seen that the parameters identified for opposite v values are close to each

v	-100	-130	-160	-200	-240
b_ψ	0.0702	0.0729	0.0669	0.0621	0.0603
v	100	130	160	200	240
b_ψ	0.0751	0.0733	0.0661	0.0612	0.0599

Table 2.4 – Planar nominal model parameter b_ψ identification results for different inputs v

other. So the final parameter identification results for the yaw dynamics is

$$\begin{aligned} a_\psi &= -0.2 \\ b_\psi &= 0.0668 \end{aligned} \tag{2.33}$$

Therefore, the NON-A blimp V2 planar movement nominal model parameter identification result is given by (2.32) and (2.33). It is worth to mention that some simplifications are made during the identification process, and some modifications are added to the nominal model so as to incorporate with the real robot hardware constraints. The parameter identification inaccuracy will be included in the disturbance term and will be presented in Chapter 4.

2.7 Conclusion

In this Chapter, the problem of blimp modeling and its parameter identification are focused.

First, some general hypotheses are proposed for the indoor blimp robot studied in this work. Then, the most commonly used blimp kinematic model and 6-DOF dynamic model are presented in detail.

Due to the complexity of the 6-DOF model and the difficulties it raised when designing control laws, an alternative approach based on the use of a simplified nominal model augmented by a disturbance term to represent the blimp dynamics is proposed for the modeling. Some reasonable hypotheses based on real indoor blimp robot motion characteristics are made so as to derive the simplified model, and decouple it into altitude movement and planar movement parts.

After the nominal models for the two independent motions are established,

the sensors used in the tests, their advantages and limits are discussed for both NON-A blimp prototype and V2 robots.

Finally, the parameter identification procedures and results are presented separately for altitude movement nominal model and planar movement nominal model, which is the basis for the controller design presented in the coming chapters.

Chapter 3

Altitude Control

3.1 Introduction

From the analysis presented in Chapter 2, under reasonable assumptions, the blimp motion is decoupled into two independent parts. In this chapter, we focus on the controller design for the blimp altitude motion.

However, due to the hardware restrictions observed during parameter identification process, the altitude movement nominal model of NON-A blimp prototype is switched and with time-delay, which increase the difficulty for controller design. Moreover, in order to assure the accuracy of control, a disturbance term is complemented to the simplified nominal model, and it has to be estimated in real-time and compensated in the controller.

Therefore in this chapter, first a complete description for the blimp altitude movement model is given (Section 3.2). Then, for the purpose of designing an output feedback controller, an observer is designed for state and switching signal estimation (Section 3.3). Next, a predictor-based controller with disturbance compensation is conceived (Section 3.4). Finally simulations are made to verify the designed controller performance and disturbance estimation result (Section 3.5).

3.2 System description

For the NON-A blimp prototype robot, the parameters identified in Section 2.6.1 can be used to establish a nominal model for blimp altitude control system. Then we intend to use a disturbance term to represent the errors between nominal model and real one, which includes the errors caused by:

- Nominal model parameter identification inaccuracy;
- Difference between blimp buoyancy force and gravity;
- Helium leak of the balloon;
- Airflow perturbation to the balloon;
- Variation of time-delay in the control loop;
- Temperature change in testing environment;
- Other environmental disturbances which are impossible to be accurately modeled.

Thus the altitude nominal model (2.30) complemented with disturbance term becomes:

$$\ddot{z}(t) = a_{z\sigma}\dot{z}(t) + b_{z\sigma}\left(u_z(t-\tau_{\text{nom}}) + d_z(t)\right) \tag{3.1}$$

Note that according to the dimensional analysis, since the command signal for vertical motors u_z is dimensionless, and the parameter b_z has the same unit as $\ddot{z}$, thus the disturbance d_z is also dimensionless. Therefore, the blimp altitude control system studied in this work is considered as a *switched system with a constant time-delay complemented with uncertain bounded disturbances* [Guerra et al., 2015]. Blimp altitude z and velocity in vertical axis $\dot{z}$ are chosen as the state vector $X = \begin{bmatrix} z & \dot{z} \end{bmatrix}^{\mathrm{T}}$, the system can be written in state space form as the combination of nominal model and real-time estimated disturbance term:

$$\begin{cases} \dot{X}(t) = A_{z\sigma}X(t) + B_{z\sigma}u_z(t-\tau_{\text{nom}}) + B_{z\sigma}d_z(t) \\ y(t) = C_zX(t) \end{cases} \tag{3.2}$$
$$\sigma \in \mathcal{P} = \{1,2,3,4\}$$

where

$$A_{z\sigma} = \begin{bmatrix} 0 & 1 \\ 0 & a_{z\sigma} \end{bmatrix}, \quad B_{z\sigma} = \begin{bmatrix} 0 \\ b_{z\sigma} \end{bmatrix}, \quad C_z = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

with parameters identified for nominal model, τ_{nom} is the nominal delay in control loop, $d_z(t)$ is the disturbance term which is estimated on-line. Aiming to facilitate calculation, a coefficient $B_{z\sigma}$ is added to $d_z(t)$, $B_{z\sigma}d_z(t)$ is assumed to be small, bounded and smooth.

As it has been shown in the parameter identification process, the switching signal is governed by:

$$\sigma = \begin{cases} 1, & u_z \geq 0 \text{ and } \dot{z} \geq 0 \\ 2, & u_z \geq 0 \text{ and } \dot{z} < 0 \\ 3, & u_z < 0 \text{ and } \dot{z} \geq 0 \\ 4, & u_z < 0 \text{ and } \dot{z} < 0 \end{cases} \tag{3.3}$$

From the results in Section 2.6.1, for the NON-A blimp prototype robot, there is (Table 2.1):

$$\begin{aligned} a_{z1} &= a_{z3} = -0.34316 \\ a_{z2} &= a_{z4} = -0.28412 \\ b_{z1} &= b_{z2} = 0.06149 \\ b_{z3} &= b_{z4} = 0.11214 \end{aligned} \tag{3.4}$$

For the purpose of solving the altitude stabilization problem for the system (3.2), an output feedback controller should be designed. For this purpose, we firstly consider an integral controller, which is popular in industrial applications and easy to be implemented.

3.2.1 Integral control - a first approach

For simplicity of controller design, the time-delay is ignored here, i.e. $\tau_{\text{nom}} = 0$ in (3.2). The regulation error $e_z = C_z e = C_z(X - X_{\text{ref}})$ is integrated to introduce integral action:

$$\dot{X}_I = e_z$$

without loss of generality, assume that $X_{\text{ref}} = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\text{T}}$, then we augment the integrator with the state equation (3.2) to obtain:

$$\begin{cases} \dot{X}(t) = A_{z\sigma}X(t) + B_{z\sigma}u_z(t) + B_{z\sigma}d_z(t) \\ \dot{X}_I(t) = C_z X(t) \end{cases}$$

Consider a linear feedback control law of the form:

$$u_z = -KX - K_I X_I$$

where $K = \begin{bmatrix} K_P & K_D \end{bmatrix}$, then the closed-loop system can be written as:

$$\bar{X} = (\bar{A} - \bar{B}\bar{K})\bar{X} + \bar{B}d_z \tag{3.5}$$

where

$$\bar{X} = \begin{bmatrix} X \\ X_I \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A_{z\sigma} & 0 \\ C_z & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B_{z\sigma} \\ 0 \end{bmatrix}, \quad \bar{K} = \begin{bmatrix} K & K_I \end{bmatrix}$$

the controller gain $\bar{K}$ is chosen to make the matrix $\bar{A} - \bar{B}\bar{K}$ Hurwitz. Let us look at the second equation of the system (3.5), it is:

$$\ddot{z} = a_{z\sigma}\dot{z} + b_{z\sigma}(-K_P z - K_D \dot{z} - K_I X_I + d_z) \tag{3.6}$$

Since $X_{\text{ref}} = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\text{T}}$, there is $e_z = z - z_{\text{ref}} = z$, and $X_I = \int z \mathrm{d}t$, then (3.6) can be written as:

$$\ddot{e}_z - (a_{z\sigma} - b_{z\sigma}K_D)\dot{e}_z + b_{z\sigma}K_P e_z + b_{z\sigma}K_I \int e_z \mathrm{d}t = b_{z\sigma}d_z$$

Taking time-derivative on both sides, we obtain:

$$e_z^{(3)} - (a_{z\sigma} - b_{z\sigma}K_D)\ddot{e}_z + b_{z\sigma}K_P\dot{e}_z + b_{z\sigma}K_I e_z = b_{z\sigma}\dot{d}_z \tag{3.7}$$

If the disturbance term d_z is constant, then by choosing controller gain properly, the origin of error dynamic (3.7) is exponentially stable, and the integral action

will make the steady-state error tends to zero. But as we have mentioned before, the disturbance term d_z represents the errors between nominal model and real one, it includes (at least) the following parts: system parameter identification inaccuracies, slowly time-varying external disturbances such as the airflow generated by air conditioner, the coupling of the two "separated" motions, and the dynamics of the actuators (which are ignored during modeling process). For the first two aforementioned disturbances, the integral controller is able to compensate them, but for disturbances such as the last two which are time-varying, the integral controller is not able to cancel them and achieve zero steady-state error [Khalil, 1996].

3.2.2 Disturbance compensation based controller - the selected approach

As it has been analyzed in previous section, since the integral controller fails to reject the aforementioned time-varying disturbances, thus in this work, a disturbance compensation based controller is designed, which includes the following steps:

1) Design an observer which can estimate state and switching signal of time-delay switched system (3.2).

2) Design a real-time estimator of the disturbance term.

3) Design a controller with disturbance compensation which stabilizes the closed-loop system.

The procedures are presented in order in the following sections.

3.3 Observer design

As the sensors in system (ultrasonic sensor for blimp prototype, OptiTrack for blimp V2 robot) can only measure blimp altitude, in order to use state feedback to design the closed-loop system, both components of the state z and $\dot{z}$ have to

be known. Moreover, the switching signal σ has to be also estimated to give commutation information about the system. As σ is determined by the velocity $\dot{z}$ and command u (see criteria (3.3)), so $\dot{z}$ has to be calculated.

The studied system is *switched with a constant time-delay complemented with uncertain bounded disturbances* as described in subsection 3.2. A Luenberger observer can be used to estimate all the states, but the switching feature of system increases the complexity of observer design and the precision of estimation result cannot be guaranteed. Considering the fact that required state is the first-order derivative of measurement, a differentiator is more suitable for the state estimation task.

For this purpose, three different differentiators are considered: HG (High-Gain), HOSM (High-order sliding mode), and HOMD (Homogeneous finite-time) differentiator. Firstly a brief overview of these differentiators is given, then tests are made to compare their performances in our application scenarios.

3.3.1 High gain differentiator

Consider a single-input-single-output nonlinear system, having a uniform relative degree equal to the dimension of the state vector [Dabroom and Khalil, 1997], then it can be transformed into the following normal form [Isidori, 2013]:

$$\begin{aligned} \dot{x} &= Ax + B[a(x)u + b(x)] \\ y &= Cx \end{aligned} \tag{3.8}$$

where (A, B, C) are canonical form matrices, and $a, b : \mathbb{R}^n \to \mathbb{R}$ are Lipschitz continuous functions. Let $u : \mathbb{R}_+ \to \mathbb{R}$ be a bounded known input, then the observer equations take the form:

$$\dot{\hat{x}} = A\hat{x} + B[a_0(\hat{x})\psi(\hat{x}, t) + b_0(\hat{x})] + H(y - C\hat{x}) \tag{3.9}$$

where $a_0(x)$ and $b_0(x)$ are nominal models of nonlinear functions $a(x)$ and $b(x)$, H is the observer gain, the output feedback control is taken as $u = \psi(\hat{x}, t)$. It is

shown in [Esfandiari and Khalil, 1992] that when the observer gain is chosen as:

$$H^T = \begin{bmatrix} \frac{\alpha_1}{\epsilon} & \frac{\alpha_2}{\epsilon^2} & \dots & \frac{\alpha_n}{\epsilon^n} \end{bmatrix} \tag{3.10}$$

then the state reconstruction is achieved, where ϵ is a small positive parameter, and the positive constants α_i are chosen to make the roots of

$$s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1} s + \alpha_n = 0 \tag{3.11}$$

having negative real parts [Dabroom and Khalil, 1999]. The choice of H sets the eigenvalues of $(A - HC)$ at $1/\epsilon$ times the roots of (3.11). According to [Esfandiari and Khalil, 1992], the estimation error will decay to $O(\epsilon)$ after a short transient period.

3.3.2 High-order sliding mode differentiator

The HOSM differentiator is proposed by [Levant, 2003]. Let input signal $f(t)$ be consisting of a bounded Lebesgue-measurable noise with unknown features, and a unknown base signal $f_0(t)$ with the n-th derivative having a known Lipschitz constant $L > 0$. Then a recursive scheme can be used to get the n-th order differentiator:

$$\begin{aligned}
\dot{z}_0 &= v_0, & v_0 &= -\lambda_0 |z_0 - f(t)|^{n/(n+1)} \operatorname{sign}(z_0 - f(t)) + z_1, \\
\dot{z}_1 &= v_1, & v_1 &= -\lambda_1 |z_1 - v_0|^{(n-1)/n} \operatorname{sign}(z_1 - v_0) + z_2, \\
&\vdots & &\vdots \\
\dot{z}_{n-1} &= v_{n-1}, & v_{n-1} &= -\lambda_{n-1} |z_{n-1} - v_{n-2}|^{1/2} \operatorname{sign}(z_{n-1} - v_{n-2}) + z_n \\
\dot{z}_n &= -\lambda_n \operatorname{sign}(z_n - v_{n-1})
\end{aligned} \tag{3.12}$$

Note that for $l < k$, the k-th order differentiator provides for a better accuracy of l-th order derivative, than the l-th order differentiator. So in this work, a second-order differentiator is used to estimate the first-order derivative of

blimp's altitude as proposed by [Levant, 2003], for input f with $|\dddot{f}| \leq L$:

$$\begin{aligned} \dot{z}_0 &= v_0, & v_0 &= -3L^{1/3}|z_0 - f|^{2/3}\,\mathrm{sign}(z_0 - f) + z_1 \\ \dot{z}_1 &= v_1, & v_1 &= -1.5L^{1/2}|z_1 - v_0|^{1/2}\,\mathrm{sign}(z_1 - v_0) + z_2 \\ \dot{z}_2 &= -1.1L\,\mathrm{sign}(z_2 - v_1) \end{aligned} \tag{3.13}$$

3.3.3 Homogeneous finite-time differentiator

Consider a nonlinear system of the form:

$$\begin{aligned} \dot{\xi} &= \eta(\xi, u) \\ y &= h(\xi) \end{aligned} \tag{3.14}$$

where ξ is the state, u is control input which is sufficiently smooth, and y is the output. Assume that (3.14) is locally observable and there exists a local change of coordinates (a diffeomorphism) which transforms the nonlinear system (3.14) into the following canonical form:

$$\begin{aligned} \dot{x}_1 &= x_2 + f_1(x_1, u, \dot{u}, \ldots, u^{(r)}) \\ \dot{x}_2 &= x_3 + f_2(x_1, u, \dot{u}, \ldots, u^{(r)}) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, u, \dot{u}, \ldots, u^{(r)}) \\ y &= x_1 \end{aligned} \tag{3.15}$$

where $x \in \mathbb{R}^n$ is the new state and $r \in \mathbb{N}$. The observer is designed as follows:

$$\begin{aligned} \frac{\mathrm{d}\hat{x}_1}{\mathrm{d}t} &= \hat{x}_2 + f_1(x_1, u, \dot{u}, \ldots, u^{(r)}) - \chi_1(x_1 - \hat{x}_1) \\ \frac{\mathrm{d}\hat{x}_2}{\mathrm{d}t} &= \hat{x}_3 + f_2(x_1, u, \dot{u}, \ldots, u^{(r)}) - \chi_2(x_1 - \hat{x}_1) \\ &\vdots \\ \frac{\mathrm{d}\hat{x}_n}{\mathrm{d}t} &= f_n(x_1, u, \dot{u}, \ldots, u^{(r)}) - \chi_n(x_1 - \hat{x}_1) \end{aligned} \tag{3.16}$$

where the functions χ_i will be defined in such a way that the observation error $e = x - \hat{x}$ tends to zeros in *finite time* (FT) [Perruquetti et al., 2008]:

$$\chi_i(e_1) = -k_i \lceil e_1 \rfloor^{\alpha_i} \tag{3.17}$$

where for any real number $x \in \mathbb{R}$:

$$\lceil x \rfloor^{\alpha} = |x|^{\alpha} \operatorname{sign}(x) \tag{3.18}$$

and the parameters α_i are chosen as:

$$\begin{aligned} \alpha_1 &= \alpha \in \left(\tfrac{n-1}{n}, 1\right), \\ \alpha_2 &= 2\alpha - 1, \\ \alpha_3 &= 3\alpha - 2, \\ &\vdots \\ \alpha_n &= n\alpha - (n-1) \end{aligned} \tag{3.19}$$

Then the error dynamics of observer (3.16) become:

$$\begin{aligned} \dot{e}_1 &= e_2 - k_1 \lceil e_1 \rfloor^{\alpha} \\ \dot{e}_2 &= e_3 - k_2 \lceil e_1 \rfloor^{2\alpha-1} \\ &\vdots \\ \dot{e}_n &= -k_n \lceil e_1 \rfloor^{n\alpha-(n-1)} \end{aligned} \tag{3.20}$$

Consider a dilation with weights $(r_1, r_2, \ldots, r_n) = (1, \alpha_1, \ldots, \alpha_{n-1})$, system (3.20) is *homogeneous* of degree $d = \alpha - 1$ with respect to these weights [Bernuau et al., 2014]. With the aim of proving stability and convergence of this differentiator, the following Lyapunov function can be used:

$$\begin{aligned} V_{\alpha}(e) &= \sigma^T P \sigma, \\ \sigma &= \begin{bmatrix} \lceil e_1 \rfloor^{1/r_1} & \lceil e_2 \rfloor^{1/r_2} & \ldots & \lceil e_n \rfloor^{1/r_n} \end{bmatrix}^T \\ &= \begin{bmatrix} e_1 & \lceil e_2 \rfloor^{1/\alpha} & \ldots & \lceil e_n \rfloor^{1/((n-1)\alpha-(n-2))} \end{bmatrix}^T \end{aligned} \tag{3.21}$$

where P is the solution of the Lyapunov equation:

$$A_0^T P + P A_0 = -I$$
$$A_0 = \begin{bmatrix} -k_1 & 1 & 0 & \dots & 0 \\ -k_2 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -k_{n-1} & 0 & \dots & \dots & 1 \\ -k_n & 0 & \dots & \dots & 0 \end{bmatrix} \tag{3.22}$$

and the gains k_i are chosen such that A_0 is Hurwitz. Then there exists $\epsilon \in [1 - \frac{1}{n-1}, 1)$, such that, for all $\alpha \in (1-\epsilon, 1)$, the system (3.20) is *globally finite-time stable* [Perruquetti et al., 2008].

3.3.4 Comparison of differentiators

The HG, HOSM and HOMD differentiator can be written in a similar formulation according to [Perruquetti and Floquet, 2007]. In our case, the second-order differentiator can be written as:

$$\begin{aligned} \dot{x}_1 &= -k_1 \lceil x_1 - y \rfloor^{\alpha} + x_2 \\ \dot{x}_2 &= -k_2 \lceil x_1 - y \rfloor^{2\alpha-1} + x_3 \\ \dot{x}_3 &= -k_3 \lceil x_1 - y \rfloor^{3\alpha-2} \end{aligned} \tag{3.23}$$

where y represents the measurement, x_1, x_2, x_3 respectively represent the zero-, first-, and second-order derivative estimation.

The gains k_1, k_2 and k_3 are selected so as to ensure that the corresponding polynomial is Hurwitz, and:

- For HG differentiator $\alpha = 1$;
- For HOMD differentiator $\alpha \in (\frac{2}{3}, 1)$;
- For HOSM differentiator $\alpha = \frac{2}{3}$.

This formula is easy to be implemented, and the parameter of HOMD differentiator is chosen as $\alpha = \frac{5}{6}$ in tests. Step altitude control is applied to compare the performances of the three differentiators.

Firstly it is the comparison of the zero-order derivative estimation result in Figure 3.1:

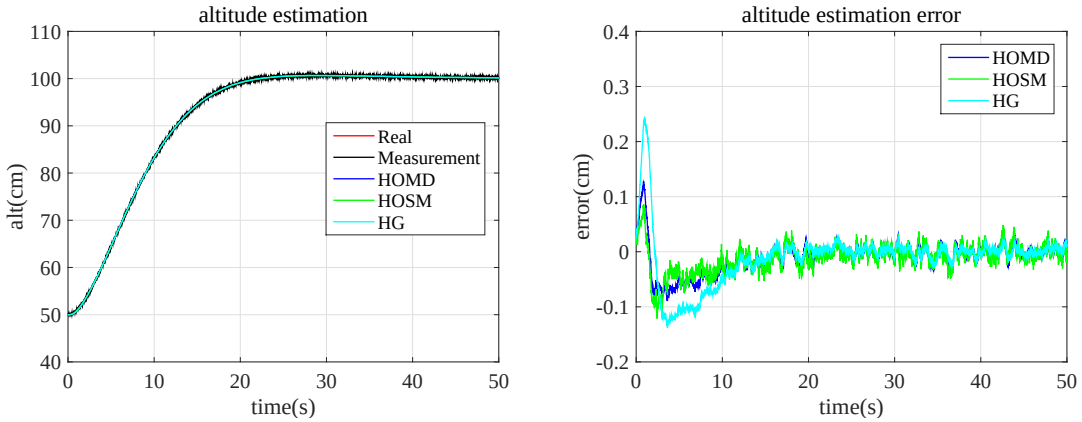

Figure 3.1 – Altitude estimation comparison.

Then it is the comparison of the first-order derivative estimation result in Figure 3.2:

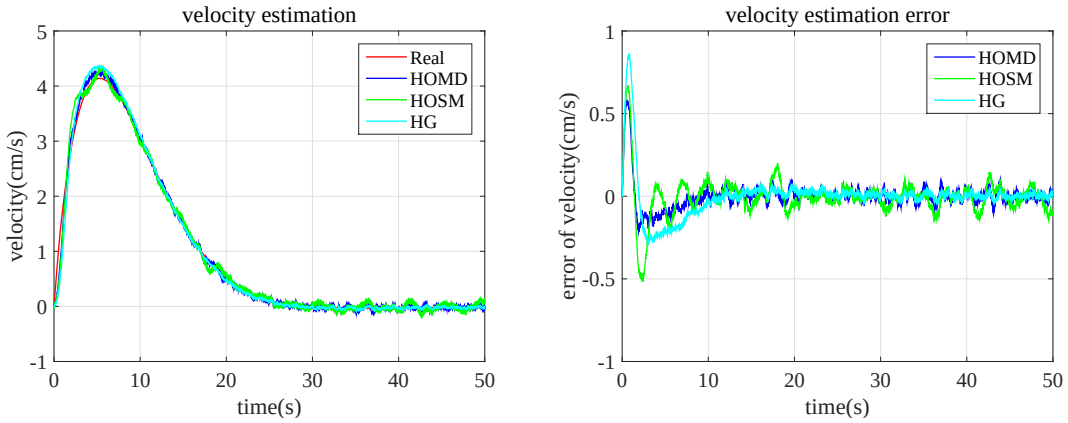

Figure 3.2 – Velocity estimation comparison.

The "Real" curve is generated using nominal model with parameters identified in Section 2.6.1, the sampling rate is 100Hz. In the "Measurement" of altitude a white noise is added to simulate measuring error, which has a Signal-to-Noise Ratio (SNR) of 25dB.

From Figure 3.1, it is clear that all the three differentiators can estimate the altitude signal after certain periods, but the HG differentiator has bigger

estimation error than the others.

As shown in Figure 3.2, the HOMD, HOSM and HG differentiator have similar performances for the first-order derivative estimation. But it is worth to mention that HOSM has bigger fluctuation than the others after they converge, it means that HOSM is less robust to noises. Moreover, as the blimp NON-A prototype system studied in this chapter has a time-varying delay which is non-ignorable, thus it is preferable to use a differentiator which converges faster. According to [Perruquetti et al., 2008], HOMD differentiator, compared to HG differentiator, demonstrates a finite-time convergence rate of differentiation error for globally Lipschitz signals $y(t)$, and for any twice continuously differentiable y and bounded measurement noises the error dynamics possesses input-to-state stability (ISS) property.

Therefore in consideration of the convergence time and robustness to noises, the HOMD differentiator is used in this chapter for $\dot{z}$ estimation. Then the value of $\hat{\sigma}$ can be evaluated by the sign of u_z and $\dot{z}$ using the following criteria:

$$\begin{aligned} \hat{X} &= \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \end{bmatrix}^{\mathrm{T}} \\ \hat{\sigma} &= \begin{cases} 1, & u_z \geq 0 \text{ and } \hat{x}_2 \geq 0 \\ 2, & u_z \geq 0 \text{ and } \hat{x}_2 < 0 \\ 3, & u_z < 0 \text{ and } \hat{x}_2 \geq 0 \\ 4, & u_z < 0 \text{ and } \hat{x}_2 < 0 \end{cases} \end{aligned} \tag{3.24}$$

In addition, as mentioned in the Section 2.6, the HOMD differentiator is also used in parameter identification process for the estimation of position and yaw angle measurement derivatives.

3.4 Controller design

As it has been mentioned in the Section 3.2, the blimp altitude control system is considered as a *switched system with a constant time-delay complemented with uncertain bounded disturbances*, thus the controller has to compensate the disturbance estimated in real-time, deal with the time-delay and stabilize the switched system.

3.4.1 Disturbance estimation

The disturbance term $d_z(t)$ in system description (3.2) represents the error between nominal model and blimp real situation.

With the intention of estimating the disturbance $d_z(t)$ in real-time, a filter is designed:

$$\begin{cases} \dot{X}_{\text{fil}}(t) = A_{z\hat{\sigma}} X_{\text{fil}}(t) + B_{z\hat{\sigma}} u_z(t - \tau_{\text{nom}}) + L(y(t) - y_{\text{fil}}(t)) \\ y_{\text{fil}}(t) = C_z X_{\text{fil}}(t) \end{cases} \tag{3.25}$$

where $L = \begin{bmatrix} l_1 & l_2 \end{bmatrix}^{\text{T}}$ is the gain of filter. It is obvious that the filter has a similar form as a Luenberger observer for nominal model, i.e. (3.2) without disturbance term.

Let $e(t) = X(t) - X_{\text{fil}}(t)$ be the error between state vector of (3.2) and that of filter (3.25). Taking the time derivative of $e(t)$, we get:

$$\dot{e}(t) = (A_{z\hat{\sigma}} - LC)e(t) + B_{z\sigma} d_z(t) + \delta(t) \tag{3.26}$$

where $\delta(t) = (A_{z\sigma} - A_{z\hat{\sigma}})X(t)$. Thanks to the finite-time convergence of differentiator, after a finite time T, we have $\hat{\sigma}(t) = \sigma(t)$, there is

$$\begin{cases} \dot{e}_1(t) = -l_1 e_1(t) + e_2(t) \\ \dot{e}_2(t) = -l_2 e_1(t) + a_{z\sigma} e_2(t) + b_{z\sigma} d_z(t) \end{cases}$$

Taking time-derivative on both sides for the first equation, we obtain:

$$\ddot{e}_1(t) = -l_1 \dot{e}_1(t) + \dot{e}_2(t)$$

Substituting $\dot{e}_2(t)$ by the second equation, and rearranging the terms, there is

$$\hat{d}_z(t) = \frac{\ddot{e}_1(t) + l_1 \dot{e}_1(t) + l_2 e_1(t) - a_{z\sigma} e_2(t)}{b_{z\sigma}}$$

With the relation $e_2(t) = \dot{e}_1(t) + l_1 e_1(t)$, we finally get the expression of disturbance

estimation:

$$\hat{d}_z(t) = \frac{\ddot{e}_1(t) + (l_1 - a_{z\sigma})\dot{e}_1(t) + (l_2 - a_{z\sigma} l_1)e_1(t)}{b_{z\sigma}} \tag{3.27}$$

Note that $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, i.e. $y(t) = x_1(t)$, so e_1 is the difference between altitude measurement and filter output, and it is available from measurements. The calculation of $\dot{e}_1(t)$ and $\ddot{e}_1(t)$ is also realized by HOMD differentiator (3.23).

Aiming to determine the gain L of the filter, using the fact that the filter matrix $A_{z\hat{\sigma}} - LC$ needs to be Hurwitz, then we need to seek a symmetric and positive definite matrix P and gain L satisfying the following LMIs (Linear Matrix Inequalities) [Geromel et al., 1998; Khalil, 1996]

$$\begin{cases} P \succ 0 \\ (A_{z\hat{\sigma}} - LC)^{\mathrm{T}} P + P(A_{z\hat{\sigma}} - LC) \prec 0, \quad \forall \sigma \in \mathcal{P} \end{cases} \tag{3.28}$$

where the set $\mathcal{P}$ is defined in (3.2).

Let $W = PL$, there is

$$\begin{cases} P \succ 0 \\ A_{z\hat{\sigma}}^{\mathrm{T}} P - C^{\mathrm{T}} W^{\mathrm{T}} + P A_{z\hat{\sigma}} - WC \prec 0, \quad \forall \sigma \in \mathcal{P} \end{cases} \tag{3.29}$$

A feasible solution of P and L can be solved by YALMIP toolbox of MATLAB [Löfberg, 2004]:

$$\begin{cases} P = \begin{bmatrix} 213.427 & -30.826 \\ -30.826 & 229.629 \end{bmatrix} \\ L = \begin{bmatrix} 0.653 & 1.059 \end{bmatrix}^{\mathrm{T}} \end{cases} \tag{3.30}$$

3.4.2 Predictor-based controller design

Many researchers have studied the problem of stability and stabilization of systems with control input delay, there are many possible approaches to deal with the problem. They can be classified into memoryless and memory controllers, the first type of controllers have feedback of the current state only, while the second one employs a feedback of the past control history as well as the current state [Moon et al., 2001].

Among the memoryless controllers, researchers have used different ap-

proaches to design the feedback, for instance, [Luo and Chung, 2002] proposed control based on the optimal control for delay-free linear system with quadratic performance index, [Kojima et al., 1994] used H_∞ control theory to investigate the robust stabilization problem for uncertain input-delay system, the authors of [Roh and Oh, 1999] proposed a sliding mode controller for the stabilization of uncertain input-delay systems with nonlinear parameter perturbations.

As for the memory controllers, [Cheres et al., 1990] designed a min-max control by using Razumikhin method which can handle a system with a fast time-varying delay, [Niculescu, 2001] used integral quadratic constraint method to design the memory controller but it is only available for a system with a constant delay, [Kwon and Pearson, 1980] designed a delayed feedback control by employing the reduction method which reduced the original system to a delay-free one, then [Moon et al., 2001] investigated the robustness of the reduction method controller by LMIs method, [Yue and Han, 2005] applied Lyapunov-Krasovskii functional approach to study the stability of uncertain system with time-varying delay and reduction method designed memory controller.

Since in our work the time-varying time-delay of the blimp NON-A prototype system is varying slowly, thus it is approximated by the constant nominal time-delay which is identified in tests as $\tau_{\text{nom}} = 0.6$s. Then we chose to use the predictor-based controller which includes a Smith predictor to compensate the fixed nominal time-delay and transformed the system to a delay-free closed-loop system. The uncertainties caused by the time-delay approximation are included in the disturbance term and compensated in the controller. The predictor-based controller has two parts:

1) Predict state at time $t + \tau_{\text{nom}}$ with Smith predictor [Smith, 1959]:

$$\begin{aligned}\hat{X}(t+\tau_{\text{nom}}) = e^{A_{z\hat{\sigma}}\tau_{\text{nom}}}\hat{X}(t) &+ \int_{-\tau_{\text{nom}}}^{0} e^{-A_{z\hat{\sigma}}s}B_{z\hat{\sigma}}u_z(t+s)\mathrm{d}s \\ &+ \int_{-\tau_{\text{nom}}}^{0} e^{-A_{z\hat{\sigma}}s}B_{z\hat{\sigma}}\hat{d}_z(t+\tau_{\text{nom}}+s)\mathrm{d}s\end{aligned} \tag{3.31}$$

where $\hat{\sigma}$ is estimated according to (3.24), and $\hat{d}_z$ is an estimate of the disturbance obtained by (3.27).

2) Assign the controller output based on predictor result and the estimated disturbance term:

$$u_z(t) = -K_{z\hat{\sigma}}\hat{X}(t+\tau_{\text{nom}}) - \hat{d}_z(t+\tau_{\text{nom}}) \tag{3.32}$$

From (3.31) and (3.32), it is clear that the disturbance term $d_z(t)$ has to be estimated in real-time and predicted for the time interval $[t, t+\tau_{\text{nom}}]$. As it has been observed in experiments, the estimated disturbance signal is rather noisy, in order to decrease the "chattering", we choose to use a time polynomial to fit $\hat{d}_z(t)$ in a sliding window, then the polynomial is used to predict $\hat{d}_z(t+\tau_{\text{nom}})$.

Remark 3.1. It is worth to mention that for the blimp NON-A V2 system, where the OptiTrack camera capturing system is implemented, the delay is greatly reduced and can be ignored with the assumption that blimp moves slowly. Thus the altitude controller for blimp V2 robot can be simply obtained by taking the $\tau_{\text{nom}} = 0$ in the equation (3.32), and replacing the nominal model parameters by the identification result shown in Table 2.2.

3.4.3 Determination of controller gain

As the system (3.2) is time-delayed with an uncertain bounded disturbances, a predictor-based controller (3.32) with disturbance compensation is designed. With the assumption that the switching signal is perfectly estimated $\hat{\sigma}(t) = \sigma(t)$, the problem remains to determine a gain of controller $K_{z\hat{\sigma}}$ which can make the closed-loop system $\dot{X}(t) = (A_{z\sigma} - B_{z\sigma}K_{z\hat{\sigma}})X(t)$ stable.

Assume the gains of controller of switched system are chosen to be the same for all σ, i.e. $K_{z\hat{\sigma}} = K_z$, to simplify calculation.

According to [Liberzon and Morse, 1999], if there exists a matrix P and gain K_z, with $P = P^{\mathrm{T}}$, such that:

$$\begin{cases} P \succ 0 \\ (A_{z\sigma} - B_{z\sigma}K_z)^{\mathrm{T}}P + P(A_{z\sigma} - B_{z\sigma}K_z) \prec 0, \forall \sigma \in \mathcal{P} \end{cases} \tag{3.33}$$

is satisfied, then the closed-loop switched system is stable. Transform (3.33) to:

$$\begin{cases} P^{-1} \succ 0 \\ P^{-1}(A_{z\sigma} - B_{z\sigma}K_z)^{\mathrm{T}} + (A_{z\sigma} - B_{z\sigma}K_z)P^{-1} \prec 0, \forall \sigma \in \mathcal{P} \end{cases} \tag{3.34}$$

Let $W = K_z P^{-1}$:

$$\begin{cases} P^{-1} \succ 0 \\ P^{-1}A_{z\sigma}^{\mathrm{T}} - W^{\mathrm{T}}B_{z\sigma}^{\mathrm{T}} + A_{z\sigma}P^{-1} - B_{z\sigma}W \prec 0, \forall \sigma \in \mathcal{P} \end{cases} \tag{3.35}$$

where decision variables are P^{-1} and W. If there exists solution for LMI (3.35), then the switched system (3.2) is globally uniformly exponentially stable.

A feasible solution of P and K_z can be solved by YALMIP toolbox of MATLAB [Löfberg, 2004]:

$$\begin{cases} P = \begin{bmatrix} 5.431 & 11.559 \\ 11.559 & 46.052 \end{bmatrix} \\ K_z = \begin{bmatrix} 0.332 & 1.540 \end{bmatrix} \end{cases} \tag{3.36}$$

3.5 Simulation

Simulations are made via MATLAB Simulink to validate the designed controller and verify the performance of disturbance estimation and compensation method. The designed block diagram is shown in Figure 3.3, it is a simplified version for the ease of comprehension.

The blocks are presented as following

- The block "real system" is used to simulate the real blimp system, with parameters slightly differed from the nominal model, and a time-varying delay.
- In the "Observer" block, the HOMD differentiator (3.23) is implemented to estimate the state $\hat{X}$ and switching signal $\hat{\sigma}$ according to (3.24).
- In "Filter" block, X_{fil} is calculated according to (3.25), which uses the nominal model parameter identification results.

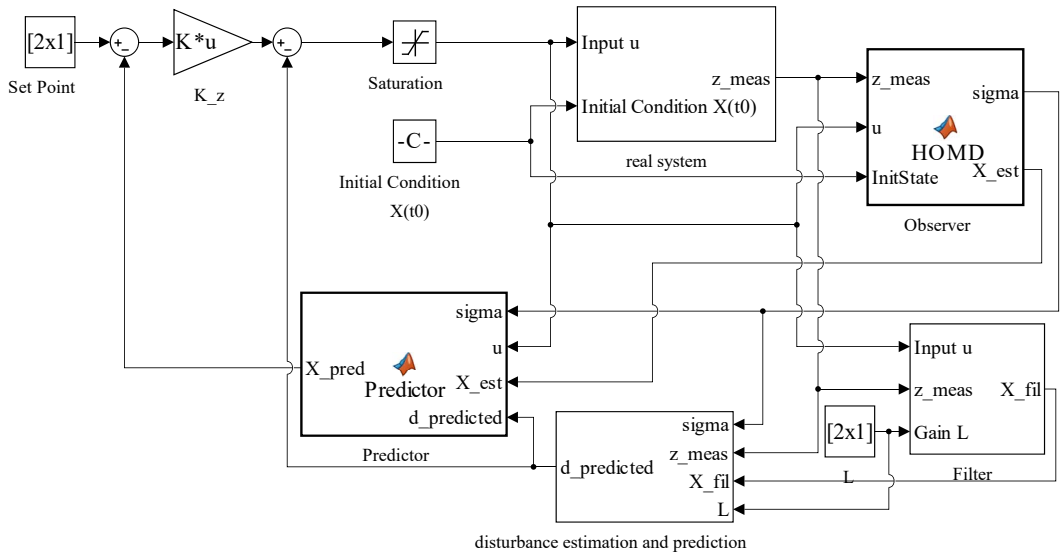

Figure 3.3 – Altitude controller Simulink block diagram

- Then X_{fil} is sent to "disturbance estimation and prediction" block where $\hat{d}_z$ is estimated according to (3.27) and $\hat{d}_z(t + \tau_{\text{nom}})$ is predicted by the time polynomial.

- Next in the "Predictor" block, the state is predicted for moment $t + \tau_{\text{nom}}$ by (3.31).

- Finally the predicted state $\hat{X}(t + \tau_{\text{nom}})$ and disturbance estimation $\hat{d}_z(t + \tau_{\text{nom}})$ are used in the controller (3.32) and close the control loop.

3.5.1 Simulation parameter setting

The parameters in the blocks are as follows:

- the time step is set as 0.1s, to simulate the measurement frequency of the blimp NON-A prototype which is 10Hz.

- for the "real system", parameters are:

$$\begin{cases} a_{r1} = a_{r3} = -0.4 \\ a_{r2} = a_{r4} = -0.2 \\ b_{r1} = b_{r2} = 0.05 \\ b_{r3} = b_{r4} = 0.12 \\ \tau_r(t) = 0.6 + 0.2\sin(t) \end{cases}$$

  Note that they are differed from the identified nominal model parameters (Table 2.1) to simulate the identification inaccuracies.

  Moreover, disturbances can be added in the "real system" block to simulate external disturbances.

- for the "Observer", the HOMD differentiator gains are set as:

$$k_1 = 5, \quad k_2 = 10, \quad k_3 = 5$$

  which make the matrix A_0 of (3.22) Hurwitz. Note that the second-order HOMD differentiator is used to estimate the zero- and first-order derivatives of the altitude measurement, because it provides a better accuracy than the first-order differentiator.

- for the "Filter" block, the parameters are set as the nominal model identification result (3.4).

- for the "disturbance estimation and prediction" block, the HOMD differentiator used to estimate $\dot{e}_1$ and $\ddot{e}_1$ has the gains:

$$k_1 = 10, \quad k_2 = 40, \quad k_3 = 80, \quad k_4 = 10$$

  which also make the matrix A_0 of (3.22) Hurwitz. And here a third-order HOMD differentiator is used for similar reason. In addition, a first-order time polynomial is used to fit the disturbance estimation $\hat{d}_z$ in a sliding window of size 6s, then predict the $\hat{d}_z(t + \tau_{\text{nom}})$.

- for the "Predictor" block, the delay is set as the nominal value $\tau_{\text{nom}} = 0.6$s

- gain L for filter is set as the result solved from LMIs:

$$L = \begin{bmatrix} 0.653 & 1.059 \end{bmatrix}^{\mathrm{T}}$$

- gain K_z for controller is set as the result from common Lyapunov function:

$$K_z = \begin{bmatrix} 0.332 & 1.540 \end{bmatrix}$$

- saturation of the input u_z is set as $[-255, 255]$ as for the real robot.

Remark 3.2. Since in Chapter 2, during the modeling process, the positive direction of $\boldsymbol{Z_n}$ axis is chosen as the down direction, which is not intuitive for understanding. Therefore for the remaining part of this section, in order to ease the comprehension, the figures are plot in a way such that the positive direction of altitude is the up direction, and the physical meaning of the altitude z of the blimp is the distance from the ground plane of the testing room to the control board of the blimp robot. And the control input and disturbance term are also reversed to be compatible.

3.5.2 Simulation test 1

For the first simulation test, the initial condition and desired state of the blimp are set as:

$$X_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$$
$$X_{\mathrm{set}} = \begin{bmatrix} 200 & 0 \end{bmatrix}^{\mathrm{T}}$$

which means the blimp is supposed to start from the ground and stabilize itself at an altitude of 200cm. A constant disturbance $d_z = 5$ is added to the "real system" block, which, in the physical world, can be considered as the difference between the buoyancy force and robot gravity force for instance, and the positive sign here means the buoyancy force is bigger than the gravity force. Or it can be interpreted as a constant wind acts on the blimp.

First, the comparison of the altitude response for the controller without and with disturbance compensation is shown in Figure 3.4.

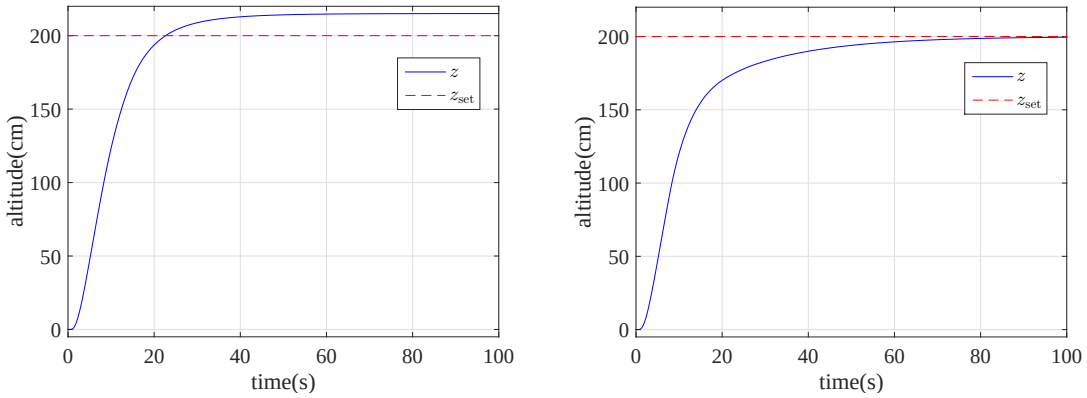

Figure 3.4 – Simulation Test1-Altitude control simulation result for controller without (left) and with (right) disturbance compensation

It can be seen that when disturbance compensation is not added in the controller (on the left of Figure 3.4), the blimp fails to reach desired altitude, it has a static error of about 15cm. While the designed controller (3.32) successfully stabilizes the robot at desired altitude in the presence of a constant disturbance d_z.

The result of switching signal estimation is shown in Figure 3.5.

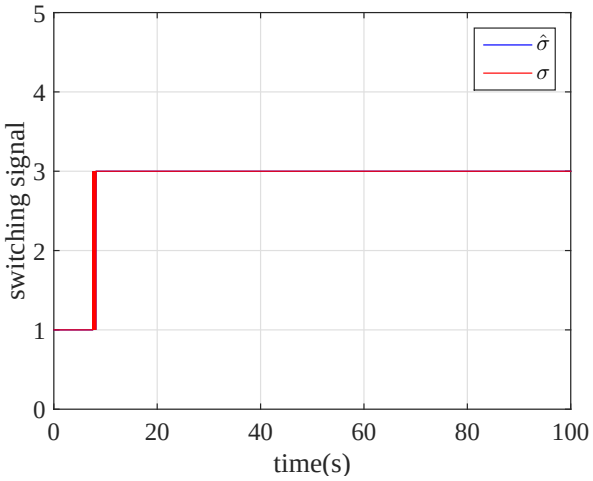

Figure 3.5 – Simulation Test1-Switching signal estimation result

It can be seen that the estimation of switching signal coincides with the real one almost everywhere except for some moments when there is error on the

vertical velocity $\dot{z}$ estimation. Thus the assumption that $\hat{\sigma}(t) = \sigma(t)$ is reasonable.

The comparison of disturbance estimation and the real one is shown in Figure 3.6. It is worth to mention that the disturbance d_z is dimensionless as analyzed before (page 70), thus there is no unit for the ordinate of Figure 3.6.

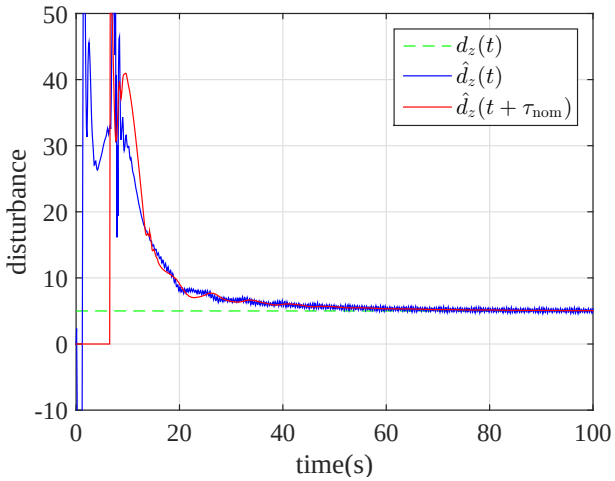

Figure 3.6 – Simulation Test1-Disturbance estimation result

Note that the estimated disturbance represents the difference between the nominal model and the real situation, thus in the figure the $d_z(t)$ shows only the constant disturbance we added manually in the "real system" block, but other perturbations such as the difference of model parameters, influence of time-varying time-delay in the "real system" block are not included in this term.

From 3.6, we can only say that the estimated disturbance term $\hat{d}_z$ tends to the constant d_z after the blimp reached desired altitude. Moreover, the prediction of disturbance term $\hat{d}_z(t + \tau_{\text{nom}})$ in the figure is shifted τ_{nom} seconds so as to align with the $\hat{d}_z(t)$ curve, the disturbance prediction has the effect of smoothing the chattering of the estimated signal, and the first-order time-polynomial works well for the prediction of a slowly varying disturbance term.

The predictor result is shown in Figure 3.7 and Figure 3.8.

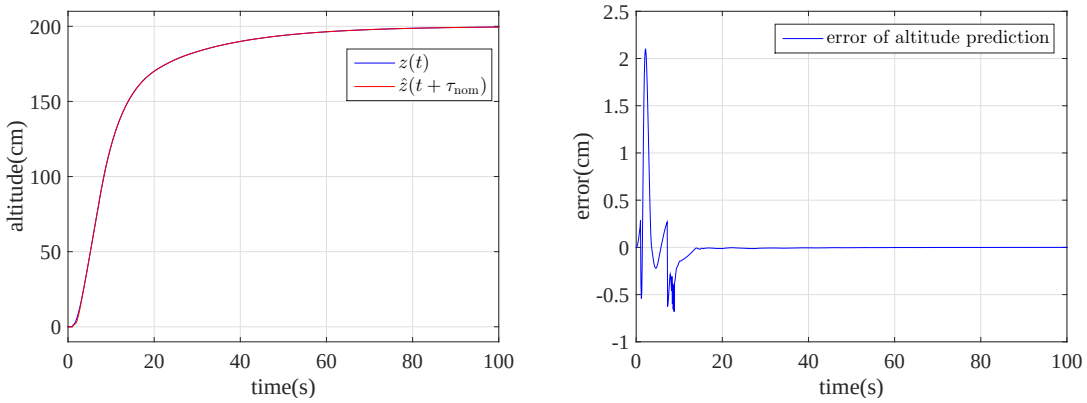

Figure 3.7 – Simulation Test1-Predictor result for altitude

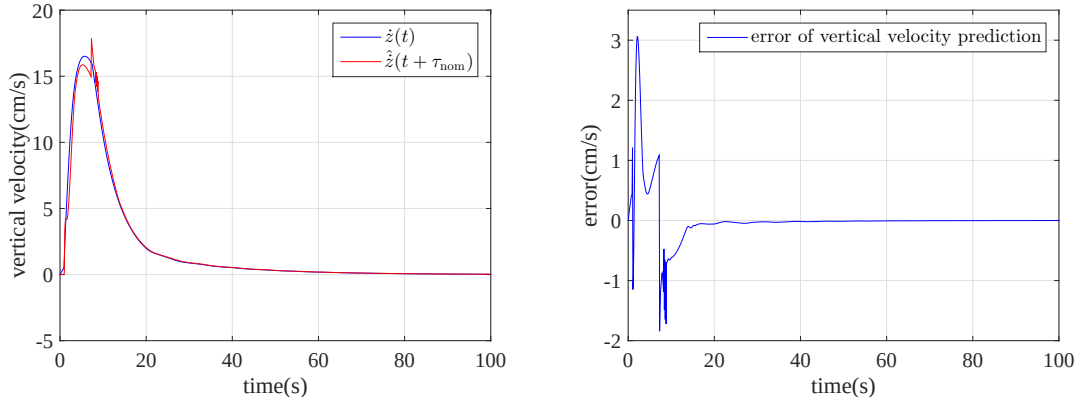

Figure 3.8 – Simulation Test1-Predictor result for vertical velocity

The predictor evaluates the state $\hat{X}(t+\tau_{\text{nom}})$ with equation (3.31). Note that in the figures, $\hat{z}(t+\tau_{\text{nom}})$ and $\hat{\dot{z}}(t+\tau_{\text{nom}})$ are shifted by τ_{nom} seconds to align with the real state. It can be shown that the predicted state, compare to the real one, has acceptable accuracy, the error between them is probably caused by the time-varying time-delay $\tau_r(t)$ set in the "real system" block.

3.5.3 Simulation test 2

For the second simulation test, the initial condition and desired state of the blimp are the same as in the test 1, but for the added disturbance, it is set as

$$d_z(t) = \begin{cases} 5, & t < 100 \\ -10, & t \geq 100 \end{cases}$$

which can be interpreted physically as a sudden change of total mass of the robot at moment 100s. Or it can be interpreted as a sudden wind direction change.

The result is shown in Figure 3.9.

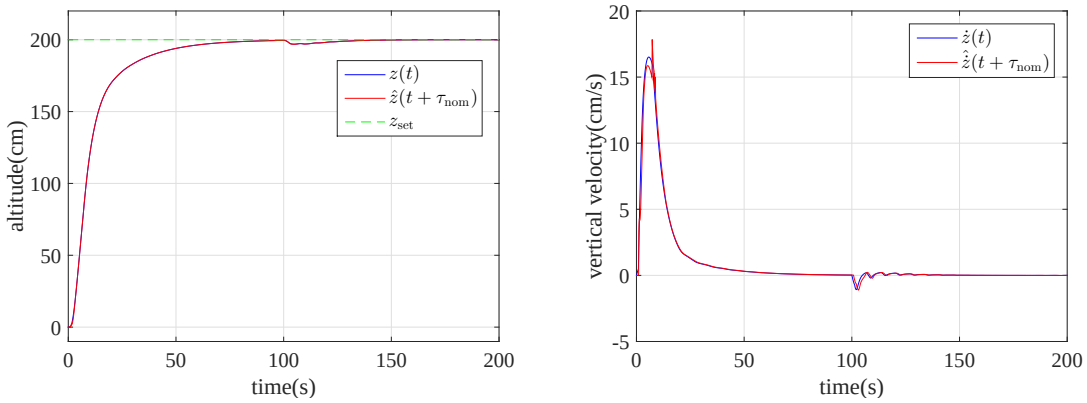

Figure 3.9 – Simulation Test2-Predictor result for altitude and vertical velocity

It can be seen that after the disturbance change, the blimp returns to desired altitude and is stabilized there. Moreover, the predicted state remains close to the real state. Note that in the figures, $\hat{z}(t+\tau_{\text{nom}})$ and $\hat{\dot{z}}(t+\tau_{\text{nom}})$ are shifted by τ_{nom} seconds to align with the real state.

The result of switching signal estimation is shown in Figure 3.10.

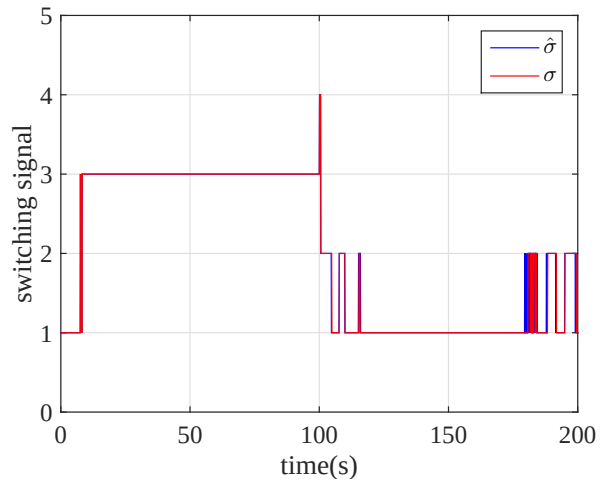

Figure 3.10 – Simulation Test2-Switching signal estimation result

It can be seen that the switching signal is perfectly estimated when there is a sudden change of the disturbance term. But when the vertical velocity $\dot{z}$ is around zero, there is chattering on the switching signal.

For the disturbance estimation result, it is shown in Figure 3.11.

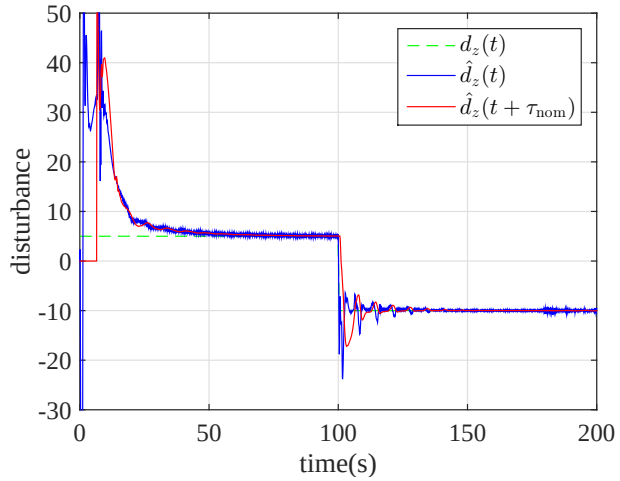

Figure 3.11 – Simulation Test2-Disturbance estimation result

It is shown that the designed disturbance estimation method responds quickly to the sudden change of $d_z(t)$ and converges finally to the exact value of added disturbance.

3.5.4 Simulation test 3

In this simulation test, the settings are the same as before except for the added disturbance, it is set as a slowly time-varying signal:

$$d_z(t) = 2 + 8\sin(0.1t)$$

The result of altitude and its error is shown in Figure 3.12.

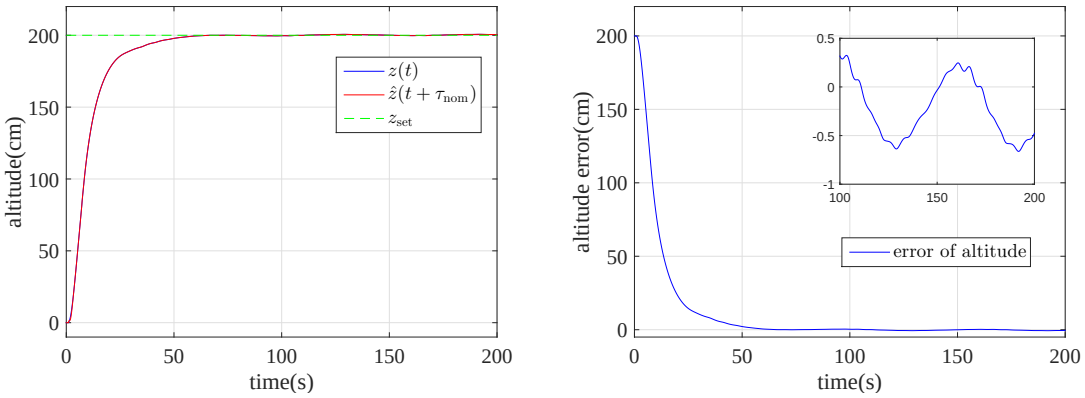

Figure 3.12 – Simulation Test3-Altitude and error of altitude result

It is shown that under time-varying disturbance d_z, the designed controller successfully stabilizes robot at desired altitude, and the error of altitude oscillates only in a small region near the origin, which is acceptable in our application scenario.

As for the switching signal and disturbance term estimation results, they are shown in Figure 3.13.

It can be seen that the switching signal is well estimated, and the disturbance estimation result follows the time-varying signal after it has converged.

In general, the simulation results prove that the designed predictor-based controller with disturbance compensation can successfully stabilize the NON-A blimp prototype robot at desired altitude in the presence of various perturbations (system parameter identification inaccuracy, time-varying delay, external constant or slowly varying disturbance). Later it will be implemented on both the NON-A blimp prototype and NON-A blimp V2 (by setting $\tau_{\text{nom}} = 0$) and the

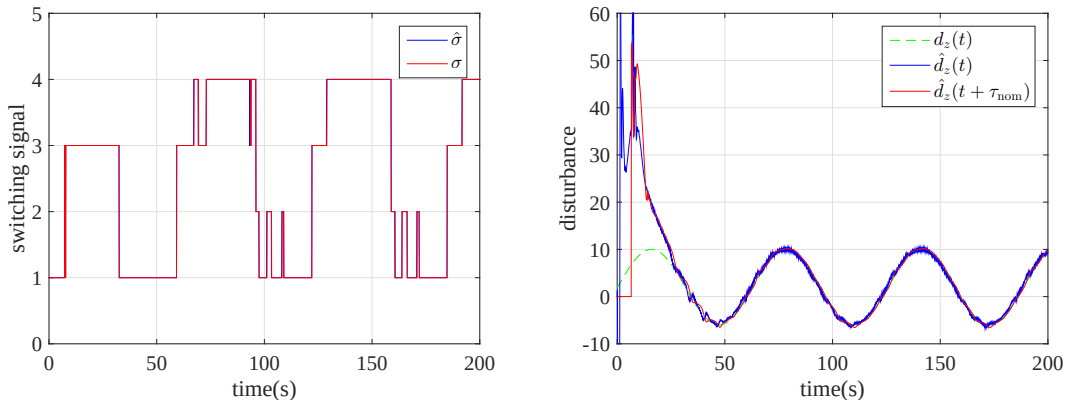

Figure 3.13 – Simulation Test3-Switching signal and disturbance estimation result

experiment result will be presented in Chapter 5.

3.6 Conclusion

In this chapter, we concentrate on one of the blimp decoupled dynamics, which is the altitude stabilization control.

First, the blimp altitude control nominal model is complemented with a disturbance term and the system is expressed in state space form. Based on the parameter identification results, the altitude control system for blimp NON-A prototype is considered as a *switched system with a constant time-delay complemented with uncertain bounded disturbances*.

Then for the purpose of designing an output feedback controller for the system, we first design an observer for state and switching signal estimation. To this end, HG, HOSM and HOMD differentiators are presented briefly and compared. HOMD differentiator is finally chosen for the estimation task.

Next, for the controller design, based on the error between filter output and altitude measurement, a real-time disturbance estimator is conceived. With the aim of compensating the time-delay of NON-A prototype robot, the state is predicted by Smith predictor, then in the controller the predicted state is used together with the disturbance compensation term. The controller gain is determined by a common Lyapunov function approach for the switched

parameters.

Finally, simulations are carried out by MATLAB Simulink to validate the designed controller and verify the performances of disturbance estimation and compensation method. The results demonstrate that designed controller is ready to be applied on the robot for real tests, which will be presented in Chapter 5.

Chapter 4

Horizontal Plane Movement Control

4.1 Introduction

From the analysis presented in Chapter 2, under reasonable assumptions, the blimp motion is decoupled into two independent parts. In Chapter 3, the blimp altitude motion controller is conceived. In this chapter, we focus on the controller design for the blimp motion in the horizontal plane.

First, a complete description for the blimp planar movement model is given (Section 4.2), moreover, two approaches are discussed to transform the under-actuated system to a simpler form for the ease of controller design. Then, for the purpose of compensating the disturbance in controller, a method to estimate perturbations in real-time is proposed (Section 4.3). Next, a disturbance compensation based robust controller is designed for tracking a predefined trajectory (Section 4.4). Finally, simulations are made to verify the designed controller performance and disturbance estimation result (Section 4.5).

4.2 System description

Similar to the altitude control system, for the NON-A blimp V2 robot, the parameter identified (with the help of OptiTrack camera system) in Section 2.6.2 can be used to establish a nominal model for the blimp planar movement control system. Then in order to ensure the accuracy of control, disturbance

terms are added to the planar movement nominal model, which represent the errors between nominal model and real one, they include the errors caused by:

- Nominal model parameter identification inaccuracy;
- Airflow perturbation to the balloon;
- Temperature change in testing environment (which influences the buoyancy force of the balloon);
- Ignored motor dynamics during modeling process;
- Disturbance caused by the altitude movement;
- Other environmental disturbances which are impossible to be accurately modeled.

Thus the planar movement nominal model (2.29) complemented with disturbance terms becomes:

$$\begin{cases} \ddot{x} = c_\psi bu + \kappa_1(a_x, a_y, \psi)\dot{x} + \kappa_3(a_x, a_y, \psi)\dot{y} + d_x \\ \ddot{y} = s_\psi bu + \kappa_2(a_x, a_y, \psi)\dot{y} + \kappa_3(a_x, a_y, \psi)\dot{x} + d_y \\ \ddot{\psi} = b_\psi v + a_\psi \dot{\psi} + d_\psi \end{cases} \tag{4.1}$$

where d_x, d_y and d_ψ are the disturbance terms which are estimated on-line, they are assumed to be small, bounded and smooth. Moreover, a dimensional analysis shows that d_x and d_y has the same unit as $\ddot{x}$ (or $\ddot{y}$), while d_ψ has the same unit as $\ddot{\psi}$.

Recall that u and v are the two control inputs, with $u = u_{\text{left}} + u_{\text{right}}$ and $v = u_{\text{left}} - u_{\text{right}}$; u_{right} and u_{left} are respectively the value of command signal for right and left motors, which are dimensionless quantities (see also (2.29) on page 53); b and b_ψ are the coefficients related to the control inputs; coefficients $\kappa_1(a_x, a_y, \psi) = a_x c_\psi^2 + a_y s_\psi^2$, $\kappa_2(a_x, a_y, \psi) = a_y c_\psi^2 + a_x s_\psi^2$ and $\kappa_3(a_x, a_y, \psi) = a_x c_\psi s_\psi - a_y c_\psi s_\psi$. As we have mentioned before, the two damping coefficients in lateral and longitudinal direction are assumed to be approximately equal, i.e. $a_x = a_y$, thus the terms $\kappa_1(a_x, a_y, \psi) = \kappa_2(a_x, a_y, \psi) = a_x = a_y$, and $\kappa_3(a_x, a_y, \psi) = 0$. The inaccuracy caused by this assumption is also included in disturbance terms, and can be compensated once d_x, d_y and d_ψ are estimated.

4.2.1 Under-actuated system

Note that the system (4.1) is under-actuated, since it has 3 configurations x, y and ψ but only two control inputs u, v. According to Brockett's Theorem:

Theorem 4.1 ***[Brockett's Theorem]*** *(Brockett, 1983). Let $\dot{x} = f(x,u)$ be given with $f(x_0,0) = 0$ and $f(\cdot,\cdot)$ continuously differentiable in a neighborhood of $(x_0,0)$. A necessary condition for the existence of a continuously differentiable control law which makes $(x_0,0)$ asymptotically stable is that:*

(i) the linearized system should have no uncontrollable modes associated with eigenvalues whose real part is positive.

(ii) there exists a neighborhood N of $(x_0,0)$ such that for each $\xi \in N$ there exists a control $u_\xi(\cdot)$ defined on $[0,\infty)$ such that this control steers the solution of $\dot{x} = f(x,u_\xi)$ from $x = \xi$ at $t = 0$ to $x = x_0$ at $t = \infty$

(iii) the mapping

$$\gamma : A \times \mathbb{R}^m \to \mathbb{R}^n$$

defined by $\gamma : (x,u) \mapsto f(x,u)$ should be onto an open set containing 0.

A quick verification shows that the system (4.1) doesn't satisfy the necessary condition proposed by Brockett's theroem. Therefore, there exists no continuous time-independent static state feedback controller which makes the origin of (4.1) asymptotically stable.

Fortunately, in practice, we do not have to stabilize the blimp exactly at desired positions, as long as it is stabilized in a small region near the goal position, it is sufficient for most of our application scenarios. However, the under-actuated system is expected to be transformed to a simpler form which eases the design of the control laws, for this purpose, two approaches are investigated and presented below.

4.2.1.1 Dynamic extension

For the first method, the dynamic extension is applied to transform the under-actuated system to a more simple system for stabilizing control design. To this

end, the simplified case where $\kappa_1 = \kappa_2 = a_x = a_y$ and $\kappa_3 = 0$ is considered, then the first two equations of (4.1) are differentiated again and there is:

$$\begin{cases} x^{(3)} = -s_\psi \dot{\psi} b u + c_\psi b \dot{u} + a_x \ddot{x} + \dot{d}_x \\ y^{(3)} = c_\psi \dot{\psi} b u + s_\psi b \dot{u} + a_y \ddot{y} + \dot{d}_y \end{cases}$$

the term $\ddot{\psi}$ doesn't appear in the equations thus the control input v cannot be included, differentiate again and there is:

$$\begin{cases} x^{(4)} = -s_\psi b u \ddot{\psi} - c_\psi \dot{\psi}^2 b u - 2 s_\psi b \dot{u} \dot{\psi} + c_\psi b \ddot{u} + a_x x^{(3)} + \ddot{d}_x \\ y^{(4)} = c_\psi b u \ddot{\psi} - s_\psi \dot{\psi}^2 b u + 2 c_\psi b \dot{u} \dot{\psi} + s_\psi b \ddot{u} + a_y y^{(3)} + \ddot{d}_y \end{cases}$$

Substituting $\ddot{\psi}$ from the third equation of (4.1), we obtain:

$$\begin{cases} x^{(4)} = c_\psi b \ddot{u} - s_\psi b b_\psi u v - c_\psi \dot{\psi}^2 b u - s_\psi b \dot{\psi}(2\dot{u} + a_\psi u) + a_x x^{(3)} + \ddot{d}_x - s_\psi b u d_\psi \\ y^{(4)} = s_\psi b \ddot{u} + c_\psi b b_\psi u v - s_\psi \dot{\psi}^2 b u + c_\psi b \dot{\psi}(2\dot{u} + a_\psi u) + a_y y^{(3)} + \ddot{d}_y + c_\psi b u d_\psi \end{cases} \tag{4.2}$$

or it can be written in a compact form as:

$$\begin{bmatrix} x^{(4)} \\ y^{(4)} \end{bmatrix} = \Theta \begin{bmatrix} \ddot{u} \\ v \end{bmatrix} + \Delta + D \tag{4.3}$$

where

$$\Theta = \begin{bmatrix} c_\psi b & -s_\psi b b_\psi u \\ s_\psi b & c_\psi b b_\psi u \end{bmatrix}$$

$$\Delta = \begin{bmatrix} -c_\psi \dot{\psi}^2 b u - s_\psi b \dot{\psi}(2\dot{u} + a_\psi u) + a_x x^{(3)} \\ -s_\psi \dot{\psi}^2 b u + c_\psi b \dot{\psi}(2\dot{u} + a_\psi u) + a_y y^{(3)} \end{bmatrix}$$

$$D = \begin{bmatrix} \ddot{d}_x - s_\psi b u d_\psi \\ \ddot{d}_y + c_\psi b u d_\psi \end{bmatrix}$$

Δ includes terms which are known under measurements and thus can be canceled in the controller, D is related to added disturbance terms, it has to be estimated then compensated in the controller.

For the system (4.3), the control input is chosen as $\ddot{u}$ and v. Given a desired

position (or trajectory) $\begin{bmatrix} x_{\text{ref}} & y_{\text{ref}} \end{bmatrix}^{\text{T}}$, the state error is

$$e = \begin{bmatrix} x - x_{\text{ref}} \\ y - y_{\text{ref}} \end{bmatrix}$$

If a state feedback controller is designed as:

$$\begin{bmatrix} \ddot{u} \\ v \end{bmatrix} = \Theta^{-1} \left(-\Delta - D + \begin{bmatrix} x_{\text{ref}}^{(4)} \\ y_{\text{ref}}^{(4)} \end{bmatrix} - K_0 e - K_1 \dot{e} - K_2 \ddot{e} - K_3 e^{(3)} \right) \tag{4.4}$$

Then the closed-loop system is linearized and the error dynamic is:

$$e^{(4)} = -K_0 e - K_1 \dot{e} - K_2 \ddot{e} - K_3 e^{(3)}$$

Then by choosing the controller gains K_0, K_1, K_2, K_3 such that the polynomial roots have negative real parts, the linearized system is stabilized at desired state.

It is worth to mention that the matrix Θ is singular when the command $u = 0$, however in this case it corresponds to a *removable singularity* (in French: singularité apparente), which means by choosing properly the flat outputs and designing trajectory for the system to follow, one can still make the closed-loop system asymptotically stable without any singularity.

This dynamic extension method is still complex to apply and difficulties rise when designing a proper trajectory to avoid the singularity problem, therefore in this work, we followed another approach to transform the under-actuated system to a simpler one for controller design as presented below.

4.2.1.2 Coordinate transformation

The second approach to transform the system to a simpler form is via a coordinate transformation process [Guerra et al., 2016]. Considering the dynamics of a point Q on the $\boldsymbol{X}_b$-axis of the blimp body-fixed frame (see Figure 4.1), and apply control to regulate its position. Specifically, the distance between Q and O_b is q, denote the coordinates of the point Q in the horizontal plane of navigation frame $\mathcal{F}_n$ as (s, r), knowing that the coordinates of O_b in the horizontal plane is

(x, y), there is:

$$\begin{bmatrix} s \\ r \end{bmatrix} = \begin{bmatrix} x + q\cos\psi \\ y + q\sin\psi \end{bmatrix} \tag{4.5}$$

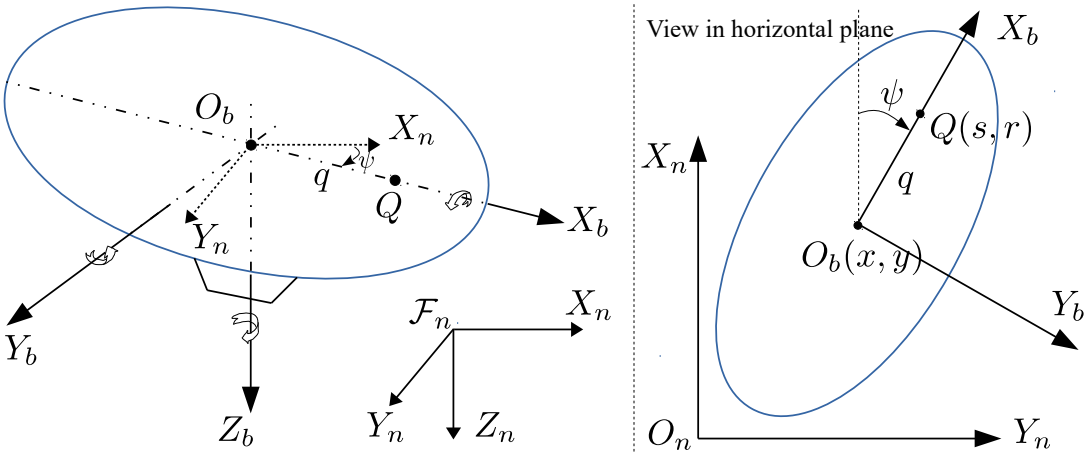

Figure 4.1 – The blimp coordinates in horizontal plane

Taking the second-order time derivative on both sides of (4.5), we obtain:

$$\begin{bmatrix} \ddot{s} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} \ddot{x} - q\sin\psi\ddot{\psi} - q\cos\psi\dot{\psi}^2 \\ \ddot{y} + q\cos\psi\ddot{\psi} - q\sin\psi\dot{\psi}^2 \end{bmatrix}$$

Substituting $\ddot{x}$, $\ddot{y}$ and $\ddot{\psi}$ from (4.1), there is

$$\begin{bmatrix} \ddot{s} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} \kappa_1\dot{x} + \kappa_3\dot{y} + c_\psi bu - qc_\psi\dot{\psi}^2 - qs_\psi(a_\psi\dot{\psi} + b_\psi v) + d_x - qs_\psi d_\psi \\ \kappa_2\dot{y} + \kappa_3\dot{x} + s_\psi bu - qs_\psi\dot{\psi}^2 + qc_\psi(a_\psi\dot{\psi} + b_\psi v) + d_y + qc_\psi d_\psi \end{bmatrix} \tag{4.6}$$

where $c_\psi = \cos\psi$, $s_\psi = \sin\psi$, and the arguments (a_x, a_y, ψ) of κ_1, κ_2 and κ_3 are omitted for compactness of expression.

Substituting $\dot{x}$ and $\dot{y}$ in (4.6) by:

$$\dot{x} = \dot{s} + qs_\psi\dot{\psi}$$
$$\dot{y} = \dot{r} - qc_\psi\dot{\psi}$$

we get:

$$\begin{bmatrix} \ddot{s} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} (\kappa_1\dot{s} + \kappa_3\dot{r}) + \left(c_\psi bu - qs_\psi b_\psi v\right) \\ \quad + \left(\kappa_1 qs_\psi\dot{\psi} - \kappa_3 qc_\psi\dot{\psi} - qc_\psi\dot{\psi}^2 - qs_\psi a_\psi\dot{\psi}\right) + \left(d_x - qs_\psi d_\psi\right) \\ (\kappa_2\dot{r} + \kappa_3\dot{s}) + \left(s_\psi bu + qc_\psi b_\psi v\right) \\ \quad + \left(-\kappa_2 qc_\psi\dot{\psi} + \kappa_3 qs_\psi\dot{\psi} - qs_\psi\dot{\psi}^2 + qc_\psi a_\psi\dot{\psi}\right) + \left(d_y + qc_\psi d_\psi\right) \end{bmatrix} \tag{4.7}$$

Let

$$U = \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \Theta \begin{bmatrix} u' \\ v \end{bmatrix} = \Theta \begin{bmatrix} b(u)u \\ v \end{bmatrix} \tag{4.8}$$

with

$$\Theta = \begin{bmatrix} c_\psi & -qs_\psi b_\psi \\ s_\psi & qc_\psi b_\psi \end{bmatrix}$$

Note that the coefficient b with respect to control input u is written as $b(u)$ according to the parameter identification result (2.32) for blimp V2 robot.

Remark 4.1. The matrix Θ is invertible for $q \neq 0$, which means instead of stabilizing the center of body-fixed frame O_b (point (x, y)) in the horizontal plane, the point Q which has a distance q from O_b is stabilized (and the yaw angle ψ is ignored). The point O_b lies nearby point Q but they can never be coincided. In practice, the q is chosen as a small constant but not too small, so that the matrix Θ is not close to its singularity, in our tests the distance is set as $q = 5$cm. In fact, since the size of the blimp balloon (length 110cm) is much bigger than the chosen q, thus the position error (between point O_b and point Q) is acceptable for our applications.

Remark 4.2. With the proposed coordinate transformation method, the problem of controlling the exact position (x, y) and orientation ψ of the robot is transformed to a practical control problem of the position of point $Q(s, r)$ which is close to the robot body-fixed frame center O_b. Therefore, when the position of point Q is regulated to the desired location, the robot center (x, y) lies on a circle which is centered at (s, r) and with a radius q.

Remark 4.3. Note that for this coordinate transformation approach, neither the states nor the disturbance terms are differentiated to higher orders than the order

of system (4.1), thus it is more suitable to be implemented in real applications compared to the first approach.

For $q \neq 0$, the matrix Θ is invertible thus $\begin{bmatrix} u' & v \end{bmatrix}^{\mathrm{T}}$ can be solved from U, then depending on the sign of u', u can be obtained by:

$$u = u'/b(u) \tag{4.9}$$

$b(u)$ is chosen from (2.32), note that $b(u)$ is always positive.

Denote

$$\Delta = \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = \begin{bmatrix} \kappa_1 q s_\psi \dot{\psi} - \kappa_3 q c_\psi \dot{\psi} - q c_\psi \dot{\psi}^2 - q s_\psi a_\psi \dot{\psi} \\ -\kappa_2 q c_\psi \dot{\psi} + \kappa_3 q s_\psi \dot{\psi} - q s_\psi \dot{\psi}^2 + q c_\psi a_\psi \dot{\psi} \end{bmatrix} \tag{4.10}$$

and disturbance term

$$D = \begin{bmatrix} d_s \\ d_r \end{bmatrix} = \begin{bmatrix} d_x - q s_\psi d_\psi \\ d_y + q c_\psi d_\psi \end{bmatrix} \tag{4.11}$$

A dimensional analysis shows that d_s and d_r both have the same unit as $\ddot{s}$ (or $\ddot{r}$), in our case, it is cm/s^2.

From (4.7), we choose the state vector $X = \begin{bmatrix} s & \dot{s} & r & \dot{r} \end{bmatrix}^{\mathrm{T}}$, hence the linearized system for blimp planar movement can be written in state space form as:

$$\begin{cases} \dot{X} = AX + B(U + \Delta + D) \\ y = CX \end{cases} \tag{4.12}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \kappa_1 & 0 & \kappa_3 \\ 0 & 0 & 0 & 1 \\ 0 & \kappa_3 & 0 & \kappa_2 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Remark that under the assumption $a_x = a_y$, the terms $\kappa_1 = \kappa_2 = a_x = a_y$, and $\kappa_3 = 0$, and the matrix A is constant. Therefore, the NON-A blimp V2 robot horizontal plane movement control system studied in this work (4.12) is considered as a *nonlinear system complemented with uncertain bounded disturbances* [Guerra et al., 2015].

4.3 Disturbance estimation

For the purpose of achieving the robust control of the blimp in horizontal plane, an output feedback controller based on disturbance compensation is designed. Therefore, the disturbance terms need to be firstly estimated.

The disturbance term D in system description (4.12) represents the error between nominal model and blimp real situation. With the aim of estimating it, a filter is designed:

$$\begin{cases} \dot{X}_{\text{fil}} = AX_{\text{fil}} + B(U + \Delta) + L(y - y_{\text{fil}}) \\ y_{\text{fil}} = CX_{\text{fil}} \end{cases} \tag{4.13}$$

where L is the gain of filter. Let $e_{\text{fil}} = X - X_{\text{fil}}$, there is:

$$\dot{e}_{\text{fil}} = (A - LC)e_{\text{fil}} + BD \tag{4.14}$$

Assume

$$L = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \\ l_{31} & l_{32} \\ l_{41} & l_{42} \end{bmatrix}$$

With the intention of ensuring the stability, the filter matrix $A - LC$ needs to be Hurwitz, and the gain L can be chosen to make the error converges at least 5-6 times faster than (4.12).

From (4.14), there is:

$$\begin{cases} \dot{e}_1 = -l_{11}e_1 + e_2 - l_{12}e_3 \\ \dot{e}_2 = -l_{21}e_1 + \kappa_1 e_2 - l_{22}e_3 + \kappa_3 e_4 + d_s \\ \dot{e}_3 = -l_{31}e_1 - l_{32}e_3 + e_4 \\ \dot{e}_4 = -l_{41}e_1 + \kappa_3 e_2 - l_{42}e_3 + \kappa_2 e_4 + d_r \end{cases} \tag{4.15}$$

From the first and third equation, there is:

$$\begin{cases} e_2 = \dot{e}_1 + l_{11}e_1 + l_{12}e_3 \\ e_4 = \dot{e}_3 + l_{31}e_1 + l_{32}e_3 \end{cases}$$

Taking time-derivative on both sides:

$$\begin{cases} \dot{e}_2 = \ddot{e}_1 + l_{11}\dot{e}_1 + l_{12}\dot{e}_3 \\ \dot{e}_4 = \ddot{e}_3 + l_{31}\dot{e}_1 + l_{32}\dot{e}_3 \end{cases}$$

Substituting e_2, e_4, $\dot{e}_2$ and $\dot{e}_4$ in the second and fourth equation of (4.15), the expression of disturbance estimation is obtained:

$$\begin{cases} \hat{d}_s = \ddot{e}_1 + (l_{11} - \kappa_1)\dot{e}_1 + (l_{21} - \kappa_1 l_{11} - \kappa_3 l_{31})e_1 + (l_{12} - \kappa_3)\dot{e}_3 + (l_{22} - \kappa_1 l_{12} - \kappa_3 l_{32})e_3 \\ \hat{d}_r = \ddot{e}_3 + (l_{32} - \kappa_2)\dot{e}_3 + (l_{42} - \kappa_2 l_{32} - \kappa_3 l_{12})e_3 + (l_{31} - \kappa_3)\dot{e}_1 + (l_{41} - \kappa_2 l_{31} - \kappa_3 l_{11})e_1 \end{cases} \tag{4.16}$$

As blimp body-fixed frame origin O_b position (x, y) can be measured by the OptiTrack system, with the relation (4.5), position of control point Q can be evaluated directly, thus $e_1 = s - s_{\text{fil}}$, $e_3 = r - r_{\text{fil}}$ are known. Therefore only $\dot{e}_1$, $\ddot{e}_1$, $\dot{e}_3$ and $\ddot{e}_3$ need to be evaluated to get the estimation of disturbance $\hat{d}_s$ and $\hat{d}_r$ [Wang et al., 2018].

For this purpose, the HOMD differentiator (3.23) is applied again as the observer for $\dot{e}_1$, $\ddot{e}_1$, $\dot{e}_3$ and $\ddot{e}_3$. Here a third-order HOMD differentiator is used aiming to get higher estimation accuracy.

4.4 Controller design

As it is mentioned before, the exact control problem of the robot position (x, y) and orientation ψ is transformed to a practical control of the position of point $Q(s, r)$ which is nearby the robot body-fixed frame center O_b. The relation between them is given by equation (4.5).

In practice, for the robust control of blimp in horizontal plane, let $\begin{bmatrix} s_{\text{ref}} \\ r_{\text{ref}} \end{bmatrix}$ be the reference point (or trajectory) with the error term:

$$e = \begin{bmatrix} e_s \\ e_r \end{bmatrix} = \begin{bmatrix} s - s_{\text{ref}} \\ r - r_{\text{ref}} \end{bmatrix} \tag{4.17}$$

The following theorem is proposed:

Theorem 4.2. *For system (4.12), if the disturbance compensation based controller is chosen as*

$$U = \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} -\Delta_{c1} - \hat{d}_s - K_p e_s - K_d \dot{e}_s + \ddot{s}_{\text{ref}} \\ -\Delta_{c2} - \hat{d}_r - K_p e_r - K_d \dot{e}_r + \ddot{r}_{\text{ref}} \end{bmatrix} \tag{4.18}$$

where $\hat{d}_s$, $\hat{d}_r$ are estimated with equation (4.16), $\dot{e}_s = \hat{\dot{s}} - \dot{s}_{\text{ref}}$, $\dot{e}_r = \hat{\dot{r}} - \dot{r}_{\text{ref}}$, the estimates $\hat{\dot{s}}$ and $\hat{\dot{r}}$ are also obtained by HOMD differentiator, K_p, K_d are the gains of controller, and

$$\Delta_c = \begin{bmatrix} \Delta_{c1} \\ \Delta_{c2} \end{bmatrix} = \begin{bmatrix} \kappa_1 \dot{x} + \kappa_3 \dot{y} + (-q c_\psi \dot{\psi} - q s_\psi a_\psi)\dot{\psi} \\ \kappa_3 \dot{x} + \kappa_2 \dot{y} + (-q s_\psi \dot{\psi} + q c_\psi a_\psi)\dot{\psi} \end{bmatrix}$$

then the position of control point Q in horizontal plane converges to the desired one with exponential convergence rate.

Proof. Take the second-order time-derivative of equation (4.17), and with (4.6) there is:

$$\ddot{e} = \begin{bmatrix} \ddot{e}_s \\ \ddot{e}_r \end{bmatrix} = \begin{bmatrix} \ddot{s} - \ddot{s}_{\text{ref}} \\ \ddot{r} - \ddot{r}_{\text{ref}} \end{bmatrix} = \Delta_c + U + D - \begin{bmatrix} \ddot{s}_{\text{ref}} \\ \ddot{r}_{\text{ref}} \end{bmatrix}$$

Using the controller proposed in Theorem 4.2, the error dynamics become:

$$\begin{cases} \ddot{e}_s + K_d \dot{e}_s + K_p e_s = d_s - \hat{d}_s \\ \ddot{e}_r + K_d \dot{e}_r + K_p e_r = d_r - \hat{d}_r \end{cases}$$

As presented in Section 3.3.4, with HOMD differentiator, the estimation of d_s and d_r converge in finite-time. Therefore, by adjusting the controller gains K_p and K_d, we can set the poles of error dynamic equation anywhere we want, which guarantee the exponential convergence of e_s and e_r to zero. ■

Therefore, with the disturbance compensation based controller designed in Theorem 4.2, the blimp planar movement closed-loop system is linearized and disturbance terms are compensated in finite-time, thus the control point Q on the blimp is supposed to be successfully stabilized at desired position or following a slowly time-varying trajectory in the horizontal plane.

For the blimp to track a trajectory, the problem remains to plan a path from the initial position to the final position. One available method is to use time-polynomial as trajectory, and decide the parameters by boundary conditions. For instance, the blimp is initialized (at moment t_i) in the horizontal plane at position (s_i, r_i), and the goal is to reach position (s_f, r_f) at moment t_f. Thus the reference trajectory for the robot to follow can be set as the time-polynomials:

$$\begin{cases} s_{\text{ref}} = \Sigma(t) \\ r_{\text{ref}} = P(t) \end{cases}$$

Without loss of generality, the robot is supposed to start from static state and when it arrives at goal position it should also be static. Therefore, the boundary conditions of the desired trajectory can be expressed as:

$$\begin{cases} \Sigma(t_i) = s_i, & \dot{\Sigma}(t_i) = 0 \\ \Sigma(t_f) = s_f, & \dot{\Sigma}(t_f) = 0 \\ P(t_i) = r_i, & \dot{P}(t_i) = 0 \\ P(t_f) = r_f, & \dot{P}(t_f) = 0 \end{cases}$$

The boundary conditions can be used to decide the parameters of these two third-order polynomials $\Sigma(t)$ and $P(t)$. For $\Sigma(t)$ (similarly $P(t)$), there is:

$$\begin{aligned} &\Sigma(\tau) = \sigma_0 + \sigma_1\tau + \sigma_2\tau^2 + \sigma_3\tau^3, \quad \tau = \tfrac{t-t_i}{t_f-t_i} \\ &\dot{\Sigma}(\tau) = \dot{\tau}(\sigma_1 + 2\sigma_2\tau + 3\sigma_3\tau^2), \qquad \dot{\tau} = \tfrac{1}{t_f-t_i} \\ &\ddot{\Sigma}(\tau) = \dot{\tau}^2(2\sigma_2 + 6\sigma_3\tau) \end{aligned}$$

The boundary conditions gives:

$$\begin{bmatrix} s_i \\ 0 \\ s_f \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \dot{\tau} & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & \dot{\tau} & 2\dot{\tau} & 3\dot{\tau} \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix}$$

We can solve:

$$\begin{bmatrix} \sigma_0 \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} = \begin{bmatrix} s_i \\ 0 \\ 3(s_f - s_i) \\ -2(s_f - s_i) \end{bmatrix}$$

Therefore, the reference trajectory for the robot is obtained:

$$\begin{aligned} &\tau = \tfrac{t-t_i}{t_f-t_i} \\ &s_{\text{ref}}(t) = \Sigma(\tau(t)) = s_i + (s_f - s_i)\tau^2(3-2\tau) \\ &r_{\text{ref}}(t) = P(\tau(t)) = r_i + (r_f - r_i)\tau^2(3-2\tau) \end{aligned} \tag{4.19}$$

The first- and second-order derivatives of the reference trajectory ($\dot{s}_{\text{ref}}$, $\dot{r}_{\text{ref}}$, $\ddot{s}_{\text{ref}}$, $\ddot{r}_{\text{ref}}$) are also easily obtained from the expression of time-polynomials which can be then used in the controller (4.18).

Note that for the blimp to move in 3D space, similar procedure can be followed to determine a reference trajectory z_{ref} for the altitude motion, then s_{ref}, r_{ref}, z_{ref} together form the 3D trajectory of the robot. Therefore when the blimp is regulated to the goal position (s_f, r_f, z_f), for the body-fixed frame center O_b, its altitude is exactly regulated, but on the horizontal plane its position lies practically on a circle centered at (s_f, r_f) and with a radius q.

4.5 Simulation

Simulations are made via MATLAB Simulink to validate the designed disturbance compensation based controller for the blimp planar movement control. The designed block diagram is shown in Figure 4.2, it is a simplified version for the ease of comprehension.

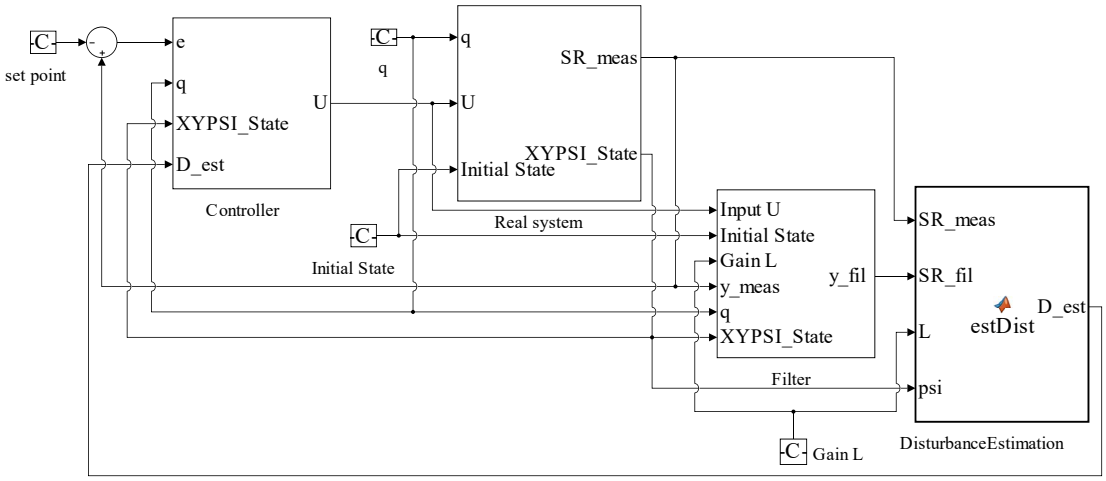

Figure 4.2 – planar movement controller Simulink block diagram

Since the original system is transformed to a simpler one via coordinate transformation, only the position of the point Q is controlled instead of the original state x, y and ψ, thus the yaw angle ψ is ignored during the blimp planar movement. Nonetheless, the original state and their derivatives are required for solving the control input U as shown in its expression. In reality, the position and yaw angle for the NON-A blimp V2 robot are measured by the OptiTrack camera system and sent to the Simulink block program for further calculation, and derivatives $\dot{x}$, $\dot{y}$, $\dot{\psi}$ are also evaluated by HOMD differentiator. But for the simulation, the original configuration x, y and ψ should be generated manually.

Therefore, the designed Simulink block diagram has in fact a nested structure, for the outer layer shown in Figure 4.2, the state and control input passing through blocks are from the system (4.12), i.e. the state of point $Q(s,r)$ and control $U = \begin{bmatrix} \tilde{u} & \tilde{v} \end{bmatrix}^{\mathrm{T}}$. Meanwhile inside the "Real system" block is the inner layer, which performs the following tasks:

1) Receives control input U, and calculates input u and v by (4.8) and (4.9)

2) Implements the original system (4.1) to simulate the states

$$\begin{bmatrix} x & \dot{x} & y & \dot{y} & \psi & \dot{\psi} \end{bmatrix}^{\mathrm{T}}$$

and output it as "XYPSI_State"

3) Transform the coordinate to s and r by (4.5), and output it as "SR_meas"

In the "Real system" block, the disturbance terms d_x, d_y and d_ψ can be adjusted to simulate different kinds of perturbations.

The rest of the blocks are presented as following

- In "Filter" block, X_{fil} is calculated according to (4.13). In addition, the term Δ is evaluated by (4.10) thanks to the information of ψ and $\dot{\psi}$ provided by "XYPSI_State".
- In "DisturbanceEstimation" block, $\hat{d}_s$ and $\hat{d}_r$ are estimated by (4.16), and output as "D_est".
- Then $\hat{d}_s$ and $\hat{d}_r$ are compensated in the "Controller" block where control input U is solved by (4.18). Moreover, Δ_c term is evaluated also with information from "XYPSI_State".

4.5.1 Simulation parameter setting

The parameters in the blocks are as follows:

- the time step is set as 0.01s, to simulate the measurement frequency of the blimp NON-A V2 which is 100Hz.
- the parameter q is set as 5, which means the control point Q locates 5cm away from the blimp body-fixed frame center O_b.
- for the "Filter" block, the parameters are set as the nominal model identification result (2.32) and (2.33), which indicates $\kappa_1 = \kappa_2 = a_x = a_y$ and $\kappa_3 = 0$.
- for the "DisturbanceEstimation" block, the HOMD differentiator used to estimate $\dot{e}_1$, $\ddot{e}_1$, $\dot{e}_3$ and $\ddot{e}_3$ has the gains:

$$k_1 = 50, \quad k_2 = 400, \quad k_3 = 200, \quad k_4 = 10$$

which make the matrix A_0 of (3.22) Hurwitz. And here a third-order HOMD differentiator is used for first- and second-order time-derivative estimation for better accuracy.

- the HOMD differentiator for $\dot{s}$ and $\dot{r}$ estimation (used in controller) has the gains:

$$k_1 = 50, \quad k_2 = 400, \quad k_3 = 200$$

which make the matrix A_0 of (3.22) Hurwitz. And here a second-order HOMD differentiator is used for first-order time-derivative estimation for similar reason.

- the controller gain is chosen as

$$K_p = 0.1, \quad K_d = 0.7$$

which makes the closed-loop system has two poles of $p_1 = -0.2$, $p_2 = -0.5$. The poles are not too big because of the assumption that the blimp moves slowly.

- the gain L for the filter is set as:

$$L = \begin{bmatrix} 3.76 & 0 \\ 3.0976 & 0 \\ 0 & 3.76 \\ 0 & 3.0976 \end{bmatrix}$$

which makes the poles of filter to be $\begin{bmatrix} -2 & -2 & -2 & -2 \end{bmatrix}$, which is approximately 8 times faster than system (4.12).

Next, we are going to simulate two types of tasks for the blimp horizontal plane movement control: point stabilization and tracking of slowly varying trajectories, to validate the designed disturbance compensation based controller.

4.5.2 Point stabilization

In this simulation, the blimp is supposed to reach set point

$$s_{\text{ref}} = 200, \quad r_{\text{ref}} = 200$$

and stabilized there. The initial state of blimp is set to be

$$s_0 = r_0 = 0, \quad \psi_0 = -\pi/4$$

which means the control point Q is initialized at origin, and the initial direction of blimp is perpendicular to the line connecting initial and the final points.

4.5.2.1 Simulation test 1

For this first simulation test of point stabilization, the disturbance terms are set as:

$$d_x(t) = \begin{cases} 2, & t < 100 \\ -1, & t \geq 100 \end{cases}$$
$$d_y(t) = \begin{cases} -1, & t < 50 \\ 2, & t \geq 50 \end{cases}$$
$$d_\psi(t) = 0$$

Moreover a white-noise generated by "Gaussian Noise Generator" block of Simulink with variance set as 0.001 is added to both d_x and d_y. According to relation (4.11), as $d_\psi = 0$, there is $d_s = d_x$ and $d_r = d_y$. Note that according to the dimensional analysis (page 104), the unit of d_s and d_r is cm/s^2.

Physically, the constant disturbance d_x and d_y can be interpreted as wind act on the blimp robot. And there is sudden change of the wind intensity at moment 50s and 100s.

For the "Real system" block in Figure 4.2, the parameters of real system is set to be the same as the nominal model, which means there is no parameter identification inaccuracy for this test.

The results of point Q position under controller WITHOUT and WITH disturbance compensation are shown in Figure 4.3.

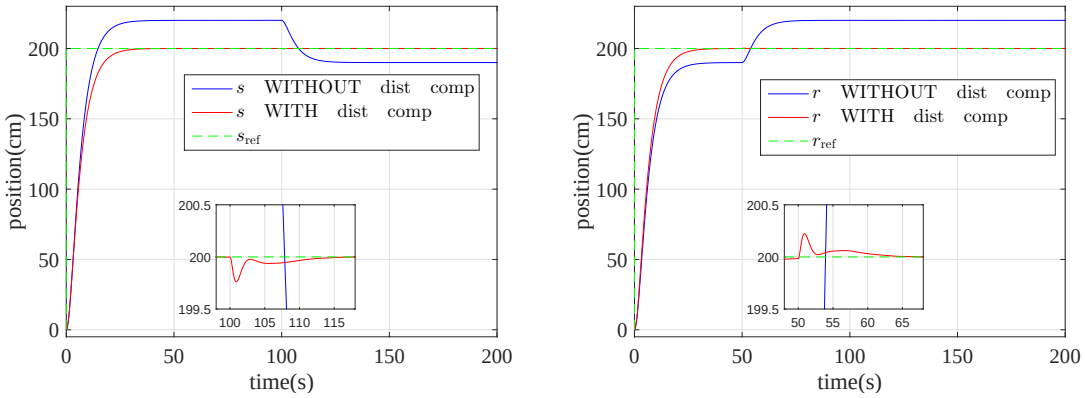

Figure 4.3 – Point stabilization simulation Test1-Q position result for controller WITHOUT and WITH disturbance compensation

It is clear that under constant disturbance, the controller without disturbance compensation fails to stabilize the position at desired one, while the designed disturbance compensation based controller succeeds. In addition, when the external perturbation has sudden changes, the position of the blimp in horizontal plane does not move far from the desired one, which shows the robustness of the designed controller.

As for the disturbance estimation results, they are shown in Figure 4.4 and 4.5.

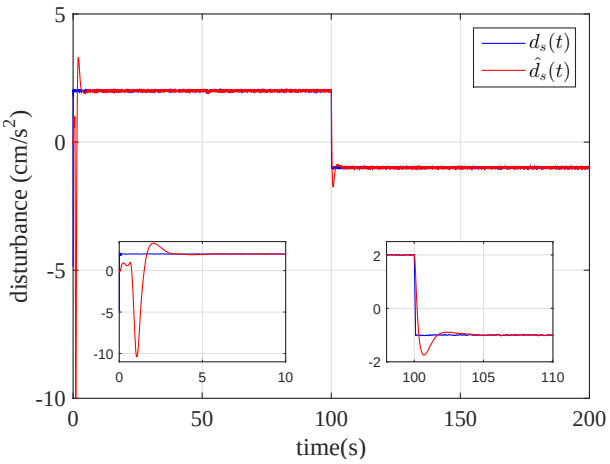

Figure 4.4 – Point stabilization simulation Test1-Disturbance d_s estimation result

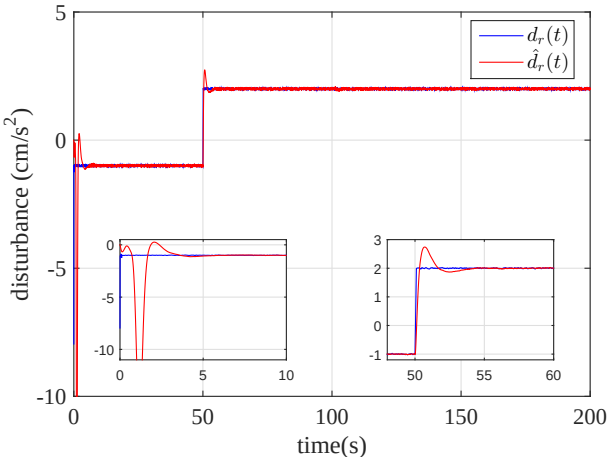

Figure 4.5 – Point stabilization simulation Test1-Disturbance d_r estimation result

It is shown that for both $\hat{d}_s$ and $\hat{d}_r$, they converge to the real value within 5 seconds, and when there is a sudden change in the added perturbation, the estimation follows quickly the change, and converge also within 5 seconds. Moreover, although the real d_s and d_r signals are noisy (because of the added Gaussian noise), the designed disturbance estimation method still gives correct estimates of the signal.

Even though the yaw angle ψ is not controlled by the designed controller, let us take a look at its result during the process as shown in Figure 4.6.

As it is mentioned before, the constant disturbances d_x and d_y can be regarded physically as the wind in X_n and Y_n direction respectively. The yaw angle ψ result shows that the blimp eventually pointed itself to the direction against the perturbation direction in the horizontal plane, as depicted on the right of Figure 4.6. And when the perturbations changes at moment 50s and 100s, the blimp also rotates itself to the direction against the wind.

4.5.2.2 Simulation test 2

For the second point stabilization simulation test, the "Real system" block in Figure 4.2 also has parameters same as the nominal model, which means there is no parameter identification inaccuracy for this test. But the disturbance terms

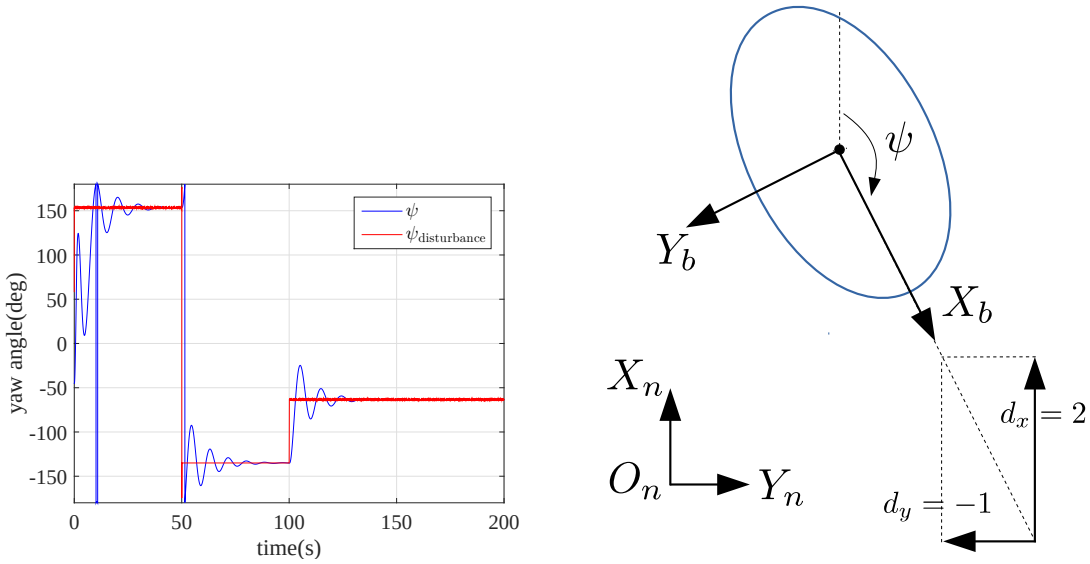

Figure 4.6 – Point stabilization simulation Test1-Yaw angle ψ result

are set as:

$$d_x(t) = \begin{cases} 2\sin(0.2t)+1, & t<150 \\ -4\sin(0.1t)-2, & t\geq 150 \end{cases}$$
$$d_y(t) = \begin{cases} -4\sin(0.1t)-2, & t<100 \\ 2\sin(0.2t)+1, & t\geq 100 \end{cases}$$
$$d_\psi(t) = 0$$

Moreover a white-noise generated by "Gaussian Noise Generator" block of Simulink with variance set as 0.001 is added to both d_x and d_y. According to relation (4.11), as $d_\psi = 0$, there is $d_s = d_x$ and $d_r = d_y$.

The results of point Q position errors e_s and e_r are shown in Figure 4.7.

It is shown that under time-varying disturbance d_x, d_y, the designed controller successfully stabilizes robot at desired position, and the error of position oscillates only in a small region near the origin, which is acceptable in our application scenario. In addition, when the external perturbation has sudden changes, the position of the blimp in horizontal plane does not move far from the desired one, which shows the robustness of the designed controller.

As for the disturbance estimation results, they are shown in Figure 4.8 and 4.9.

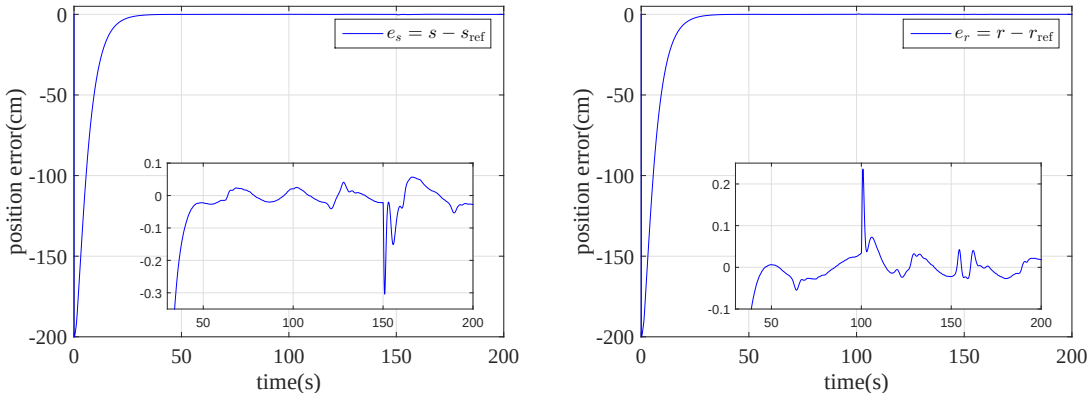

Figure 4.7 – Point stabilization simulation Test2-Q position error result

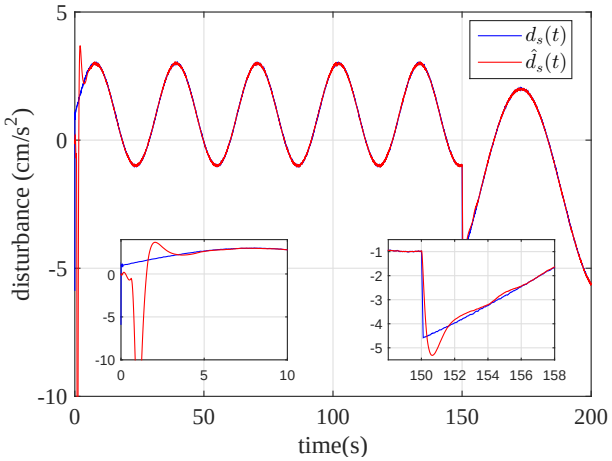

Figure 4.8 – Point stabilization simulation Test2-Disturbance d_s estimation result

It is shown that for both $\hat{d}_s$ and $\hat{d}_r$, they converge to the real value within 5 seconds and follow the slowly varying signal afterwards, and when there is a sudden change in the added perturbation at moment 100s and 150s, the estimation follows quickly the change, and converge also within 5 seconds. Moreover, although the real d_s and d_r signals are noisy (because of the added Gaussian noise), the designed disturbance estimator still gives correct estimates of the signal.

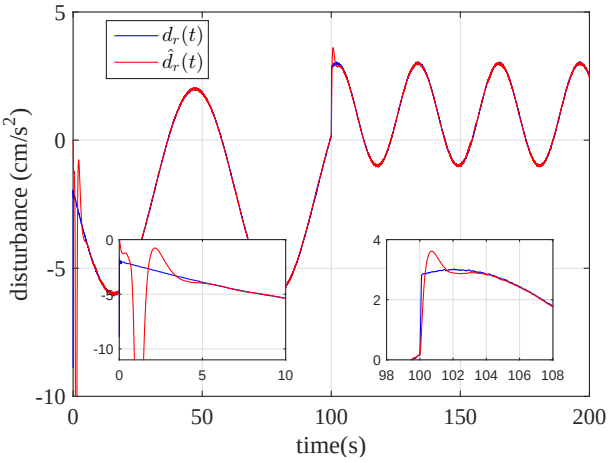

Figure 4.9 – Point stabilization simulation Test2-Disturbance d_r estimation result

4.5.2.3 Simulation test 3

For the third point stabilization simulation test, the disturbance term is set as the same as in test 2. But the "Real system" block in Figure 4.2 has parameters:

$$
\begin{aligned}
&a_x = a_y = -0.26 \\
&a_\psi = -0.16 \\
&b(u) = \begin{cases} 0.09 & \text{if} \quad u \geq 0 \\ 0.045 & \text{if} \quad u < 0 \end{cases} \\
&b_\psi = 0.06
\end{aligned}
$$

which are different to the nominal model parameters (2.32) and (2.33), i.e. there is parameter identification inaccuracy.

The results of point Q position errors e_s and e_r are shown in Figure 4.10.

It is shown that under time-varying disturbance d_x, d_y, and parameter identification inaccuracy, the designed controller successfully stabilizes robot at desired position, and the error of position oscillates only in a small region near the origin, which is acceptable in our application scenario. In addition, when the external perturbation has sudden changes (at moment 100s and 150s), the position of the blimp in horizontal plane does not move far from the desired one, which shows the robustness of the designed controller.

The disturbance estimation results are shown in Figure 4.11 and 4.12.

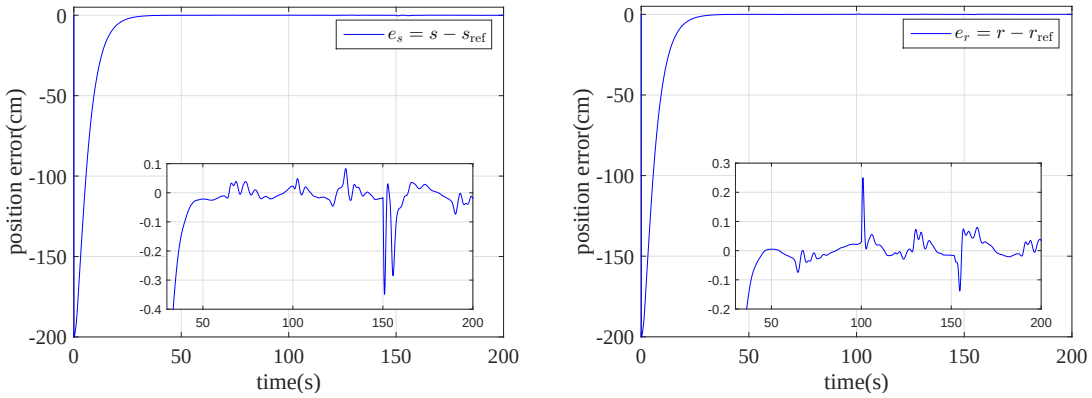

Figure 4.10 – Point stabilization simulation Test3-Q position error result

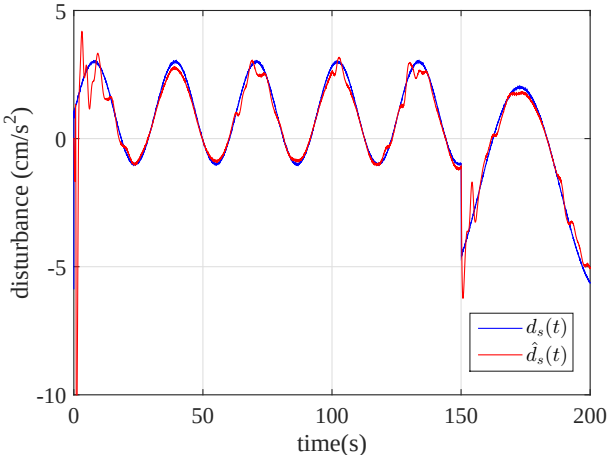

Figure 4.11 – Point stabilization simulation Test3-Disturbance d_s estimation result

It is worth to mention that the estimated disturbances $\hat{d}_s$ and $\hat{d}_r$ reflect the difference between the nominal model and the real one, in this test, they include not only the added perturbation d_s and d_r, but also the difference due to parameter identification inaccuracy, that is why in the Figure 4.11 and 4.12, the curves are not completely superposed. However, we can observe that when the parameters of "Real system" do not differ largely from the nominal model, the estimated $\hat{d}_s$ and $\hat{d}_r$ follow the variation trend of the added d_s and d_r, even with the sudden change at moment 100s and 150s.

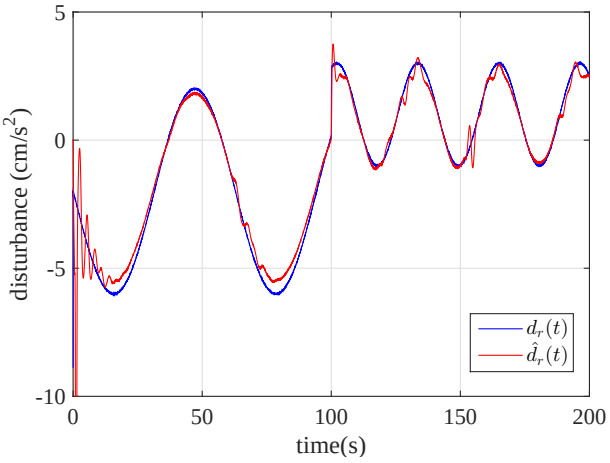

Figure 4.12 – Point stabilization simulation Test3-Disturbance d_r estimation result

From these three simulation tests on the designed disturbance compensation based controller, we can conclude that the controller is able to stabilize the blimp robot at desired position, and it is robust to constant and slowly varying disturbances and system parameter identification inaccuracy.

4.5.3 Trajectory tracking

In this simulation test, the blimp is supposed to track a reference trajectory generated by the time-polynomial method. The initial state of blimp is set to be same as before:

$$s_i = r_i = 0, \quad \psi_i = -\pi/4$$

and at $t_f = 30$s, it is supposed to reach the goal position $s_f = -200, \quad r_f = 200$ then stabilized there. The trajectory $s_{\text{ref}}(t)$, $r_{\text{ref}}(t)$ are generated by (4.19). The disturbance terms are set as:

$$d_x(t) = -1, \quad d_y(t) = 2, \quad d_\psi(t) = -0.2$$

Moreover a white-noise generated by "Gaussian Noise Generator" block of Simulink with variance set as 0.001 is added to d_x, d_y and d_ψ. The relation to obtain d_s and d_r is given by (4.11).

For the "Real system" block in Figure 4.2, the parameters of real system are set to be the same as the nominal model, which means there is no parameter identification inaccuracy for this test.

The results of point Q tracking the reference trajectory are shown in Figure 4.13.

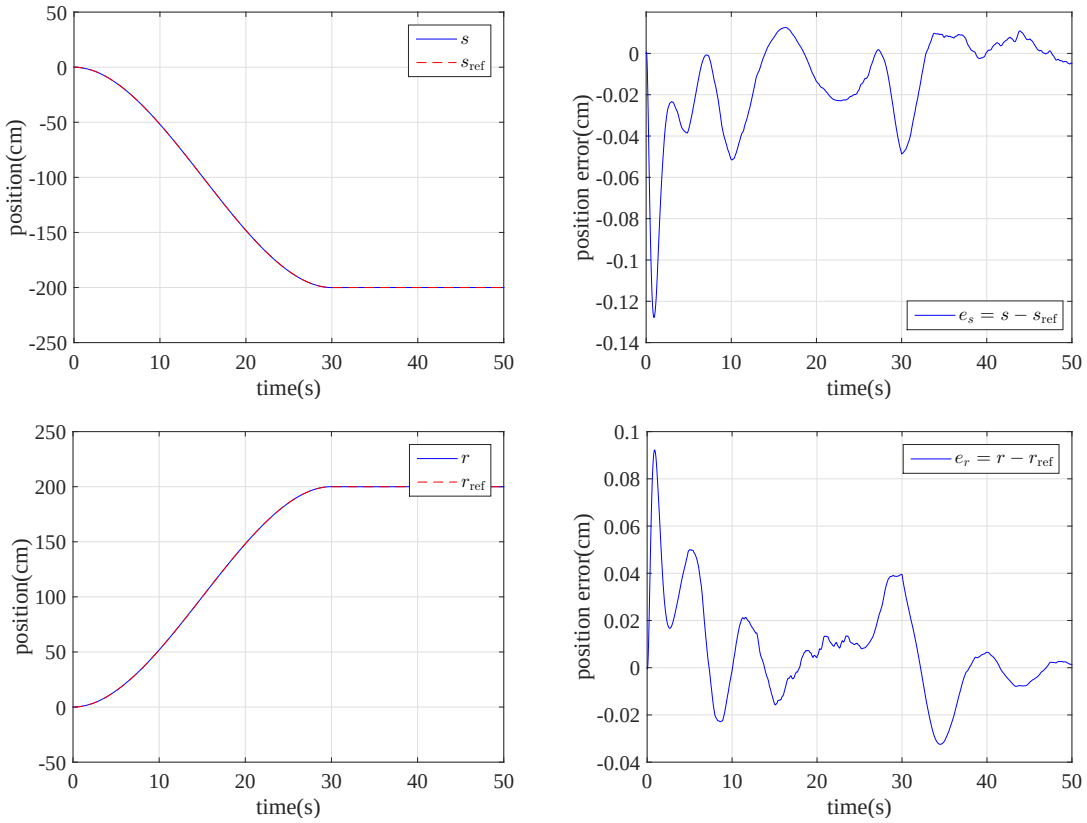

Figure 4.13 – Trajectory tracking simulation test-Q position and error result

It is shown that the blimp follows the desired trajectory with an satisfying accuracy, and even there is disturbances, the controller robustly finishes its work.

The disturbance estimation result is given in Figure 4.14.

It can be seen that when introducing the disturbance on the yaw angle d_ψ, the disturbances d_s and d_r depend also on the yaw angle, thus have more complex form, but the conceived disturbance estimator successfully evaluates the value, and helps to improve the controller robustness against perturbations.

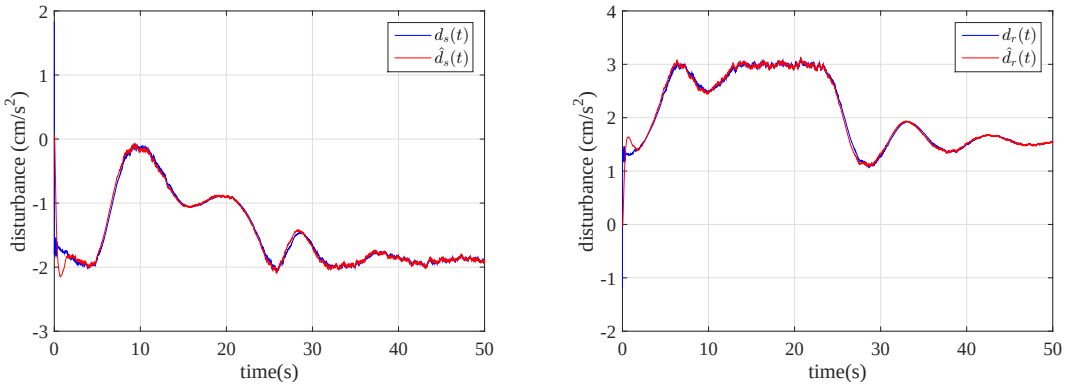

Figure 4.14 – Trajectory tracking simulation test-disturbance estimation result

From the simulation tests on point stabilization and trajectory tracking presented before, we can conclude that the designed disturbance compensation based controller is able to achieve trajectory tracking of blimp on the horizontal plane and it is robust against small bounded disturbances. It is ready to be implemented on real blimp robot for validation.

4.6 Conclusion

In this chapter, we concentrate on the other blimp decoupled dynamics, which is the horizontal plane movement control.

First, the blimp planar movement nominal model is complemented with disturbance terms. Since the system is under-actuated, it needs to be transformed to a simpler one for controller design. Therefore two approaches are discussed separately, the difficulties of applying the dynamic extension approach are analyzed, hence we decided to use a coordinate transformation to simplify the system, and only practically control the position of a point Q nearby the blimp body-fixed frame center O_b instead of exactly controlling the position and orientation (x, y, ψ) of the robot. The planar movement control system for NON-A V2 blimp robot is considered as a *nonlinear system complemented with uncertain bounded disturbances*.

Then for the purpose of achieving robust control of blimp in horizontal plane, a method to estimate the disturbance term in real-time is proposed.

Next a disturbance compensation based controller for trajectory tracking or point stabilization is conceived. Moreover, a simple path planning based on time-polynomials is presented briefly.

Finally, point stabilization and trajectory tracking simulations are made via MATLAB Simulink to validate the designed controller and verify the performance of disturbance estimation and compensation method. The results demonstrate that designed controller is ready to be applied on the robot for real tests, which will be presented in Chapter 5.

Chapter 5

Implementation and Results

5.1 Introduction

The simulations in Chapter 3 and Chapter 4 show good results, but it is never easy to implement algorithms in practice. In this chapter, we focus on the implementation and try to transform the theories into reality.

First, based on the analysis of blimp robot system functionalities requirements, an idea on how to design the hardware of the robot is decided and realized (Section 5.2). Then the testing environment is investigated, and a motion capturing system is implemented to enhance the performance of blimp system (Section 5.3). Next, the controllers designed for altitude motion and planar movement are tested on real robots (Section 5.4 and Section 5.5). Finally, the performance of the controller is validated and a conclusion is given.

5.2 Hardware design

The hardware design of the blimp robot depends on the desired functionalities for the robot to accomplish. As we have mentioned in the first chapter, in this work, we intend to create a blimp robot for indoor environment application such as indoor long-term surveillance by stabilizing itself at a fixed location or patrolling along a predefined path. Thus it puts forward some requirements on the blimp robot hardware design such as the robot size, weight, acoustic noise

level, and autonomous operation time.

Meanwhile, in this dissertation, we focus on the motion control problem of the blimp robot, decouple the motion of blimp into two independent parts, and design motion control law for the altitude and planar movement separately (in Chapter 3 and 4). Therefore, the created blimp robot hardware should be able to serve as an experiment platform to verify the efficiency and performance of the conceived controllers.

5.2.1 Blimp robot system overall analysis

In general, the navigation and control system for an autonomous mobile blimp robot should have the following parts as shown in Figure 5.1.

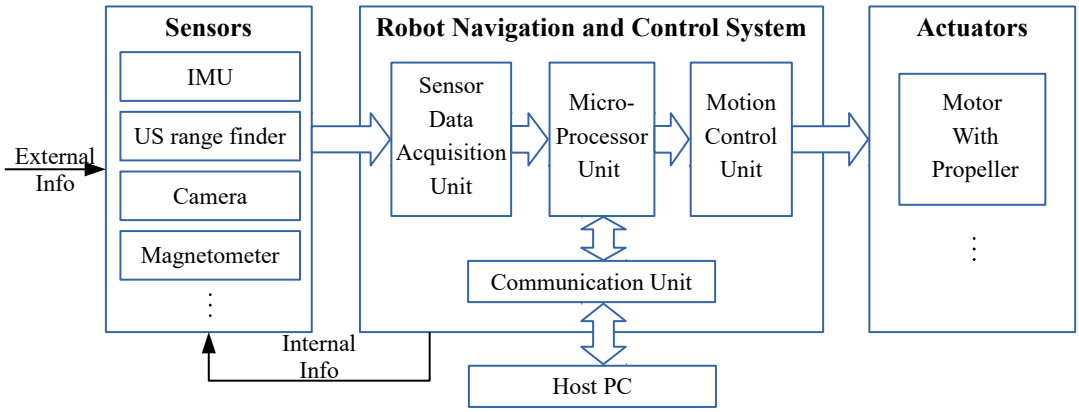

Figure 5.1 – Autonomous blimp robot system schema

The sensors are responsible for the perception of the robot, they are the "eye" for the robot to "see" the world. For our indoor blimp robot, useful sensors are the IMU, US range finder, camera etc. But due to the payload limitation of the balloon, the choice of the sensors is restricted, for the NON-A blimp prototype, the chosen sensors are already presented in Section 2.5.

The "Robot Navigation and Control System" has several different units in charge of different functionalities:

- The "Sensor Data Acquisition Unit" communicates with the proprioceptive and exteroceptive sensors and manage the raw data acquisition process of

those sensors, then transmits the raw data to micro-processor for further calculation;

- The micro-processor extracts useful information from the raw data acquired by sensors, and performs the robot localization in the map. Depending on the knowledge database, the localization may be carried out simultaneously with the map building process. Then according to the mission commands set by "Host PC" (e.g. a goal position for the robot to reach), the on-board micro-processor makes decision and plans the path to achieve the mission.

- The "Motion Control Unit" combines the planned path and real-time robot localization information, and solves actuators commands according to the implemented control laws.

- The "Communication Unit" serves as the contact interface between "Host PC" and the on-board robot control system. It receives mission commands from the "Host PC" and transmits useful robot system information to PC for display.

Finally, the actuators receive commands and drive the robot to achieve the required missions. For the blimp robot, as we have discussed before, the chosen actuators are the motors with propellers. The rudder, elevator or tail fins are not mounted due to their negligible efficiency in low speed.

After analyzing the blimp robot control system composition, we can start to select specific hardware and design the electric circuit according to application demands. For the electric circuit design, instead of combining all the functionalities together in one board, we follow a modular designing idea, dividing the whole complex robot system into several simple sub-systems dedicated to specific functions. For each one of the sub-systems, we perform the design, test, debug, modify loop, and after all the sub-systems have fulfilled the requirements, we can integrate them together as a whole system for blimp robot navigation and control purpose. Moreover, the modular design has the advantage that when there is problem occurred in one of the sub-systems, it is easy to maintain or update the broken part.

In our design, the autonomous blimp robot system is divided into four subsystems, which are:

- Main control board module
- Motor driver module
- Wireless communication module
- Sensors module

For the main control board, there are several different parts, including the microprocessor and its auxiliary circuit, the power supply and conversion circuit, the interfaces for sensors and communication, and the interface with the motor driver board.

The motor driver board is conceived separately from the main control board on account of the fact that usually it requires a high current output capability to drive the motors, and the big current in the circuit is better to be laid far away from the weak current signal (e.g. the sensor raw data) in the main control board to avoid perturbations.

The wireless communication module and sensors modules are usually available from market, but we need to design interfaces on the main control board to fit with those commercial modules for easy "Plug and Play".

5.2.2 NON-A blimp robot: Two generations

For our NON-A blimp project, we have created two versions of hardware of the robot, the first one is called prototype, and the second one is called V2. A brief comparison of these two versions is shown in Table 5.1.

The NON-A blimp prototype robot is designed and built by our former team engineer, it is a straightforward implementation of the basic functions required by our project. The hardware is shown in Figure 5.2.

The prototype uses an Arduino Fio micro-controller as its processor, which has the I/O port resources (I2C interface, input interrupt and PWM output) just sufficient for the chosen sensors and motor drivers [Arduino, 2018], which

Term	Prototype	V2
Sensors	IMU: **MPU-6050** US: **LV-MaxSonar-EZ1** Camera: **ALM-2451G**	**OptiTrack** motion capturing system SPI, I2C & GPIO **interfaces reserved for sensors**
Micro-controller	**ATmega328P** on Arduino Fio Board: • 8-bit CPU • 32Kb Flash memory, 2Kb SRAM • 14 digital I/O and 8 Analog Input pins • One I2C, USART, SPI interface • 3 timers	**STM32F103VET6**: • ARM 32-bit CPU, 72MHz maximum frequency • 512Kb Flash memory, 64Kb SRAM • 80 fast GPIO ports • Multiple I2C, USART, SPI interfaces • 11 timers
Communication	**XBee** wireless communication	**XBee** wireless communication Multiple **RS-232** serial com
Motor and driver	**L293D motor driver chip** for vertical motor: 600mA max output current **LB1930MC motor driver chip** for 2 horizontal motors: 1A max output current	**BTS7960B MOSFET chip in motor driver circuit** for four **8520 (8.5×20mm) coreless motors** (2 in vertical direction, 2 in horizontal plane): 43A max output current

Table 5.1 – Comparison of NON-A blimp prototype and V2 hardware

means it will be hard to extend the functionalities of the prototype robot since there is no more available pin resources of the micro-controller.

On the prototype robot, we have tested the altitude stabilization task, including the model parameter identification, the controller implementation and validation. However, from what we observed during tests, the micro-processor on the prototype robot demonstrates a time varying delay, although it is only programmed to collect sensor data from IMU and US range finder, communicate with Host PC and control the three motors. The time-delay raises the controller design difficulties (which has been discussed in Chapter 3).

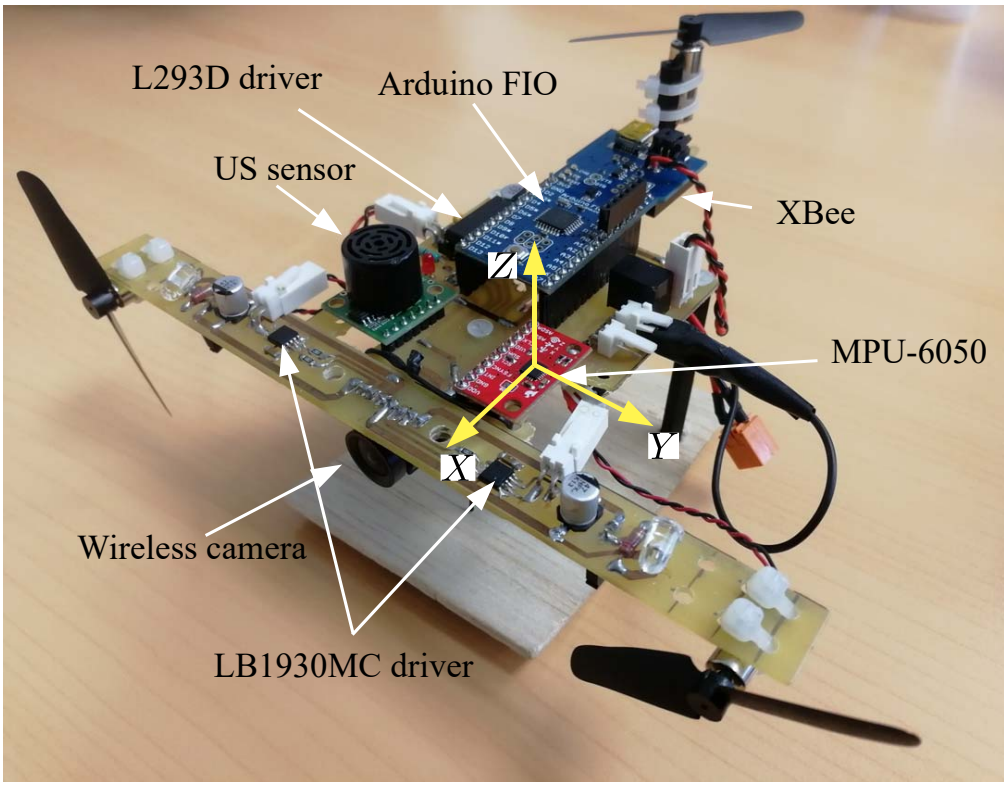

Figure 5.2 – NON-A blimp prototype robot (without balloon)

Some other restrictions due to the prototype hardware selection and design are also observed during tests. For instance, although the motor driver chip L293D and LB1930MC for the three DC motors are easy to implement and use, they lack the capability to drive the motors rotating at higher speed due to the output current limitation (600mA and 1A, see Table 5.1), and they generates immense heat due to the energy loss on the transistors inside the chip, which could be harmful for potential long term operation applications of the blimp robot [ONSemiconductor, 2018; TexasInstruments, 2018].

Moreover, the prototype only uses one motor in the vertical direction for altitude regulation, in the tests it is observed that when the motor rotates, the torque generated by the rotation of the propellers will cause the robot to rotate in the counter direction, which is not appreciated, because we divide the motion of blimp into two separate parts in the model simplification process (Chapter 2).

In addition, since the prototype is just a first quick trial on the basic idea of

indoor blimp robot, the PCB (Printed Circuit Board) is made manually in the team. As it can be seen in Figure 5.2, without overlay, the circuits are exposed to air and the copper is oxidized after long time.

In general, thanks to the experiences obtained from the creation and tests on NON-A blimp prototype robot, we are able to improve the hardware design and rebuild a second version of the robot with more powerful computation capability, more interfaces reserved for further function expansion, more reasonable layout of motors and more efficient motor driver circuit, which came out as the NON-A blimp V2 robot.

5.2.3 NON-A blimp V2 robot: Electric circuit design

As it has been introduced before, the blimp V2 robot system has four sub-systems, and for main control board and motor driver board, there are no available choices in market, so we have to design the electric circuit for this two boards.

The most important part of the blimp robot control system is the micro-controller used on the main control board. On this second version of robot hardware, we chose a powerful micro-processor which is STM32F103VET6 from STMicroelectronics [STMicroelectronics, 2018b]. As listed in Table 5.1, it has a ARM 32-bit Cortex-M3 CPU, the computation can reach 72MHz of maximum frequency, the memories are also bigger than the chip on prototype robot, and more importantly, it provides plenty of GPIO (General-Purpose Input/Output) ports which support multiple different functions such as I2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface) and USART (Universal Synchronous/Asynchronous Receiver/Transmitter) interfaces, thus it is sufficient to connect with the chosen sensors, and there are still many ports left available to be extended with other sensors. In addition, the STM32 chip integrates 11 timers which can be used as PWM (Pulse Width Modulation) wave generator, incremental encoder input, external interrupt timekeeping (e.g. the US range finder input), etc. In a word, this micro-processor is powerful enough for our robot control system design requirements, and in the future if we want to extend its functionality and make the robot more autonomous, this chip still has the potential to achieve the goal.

Depending on the function requirements, the main control board should have the following connections with external components as shown in Figure 5.3.

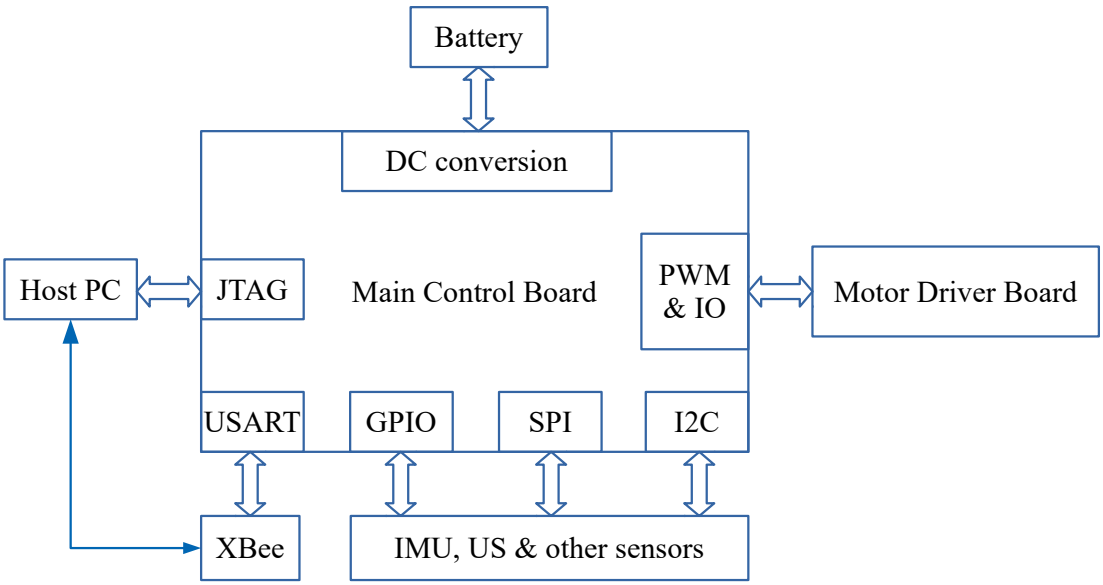

Figure 5.3 – NON-A blimp V2 main control board interfaces

Note that the JTAG interface is used for programming the micro-controller from the PC software and on-line debugging. The XBee wireless communication module is used to transmit and receive messages with Host PC wirelessly. And aiming to facilitate the extension of functions in the future, all the available GPIO ports are led out to headers.

The design of electric circuit follows the modular design idea, different parts are designed, tested, debugged separately and combined together in the end to form the complete main control board. For the clarity of writing, the design of circuit schematics and PCB are put in Appendix B. Finally, the created main control board is shown in Figure 5.4.

As for the motor driver board, it is in charge of receiving commands from main control board then driving the motors with high current. It is separated from the main control board because we want to reduce the influence of the big currents in motor driving circuit to the weak current signals in main control board. The conceived electric circuit schematics and PCB of motor driver board

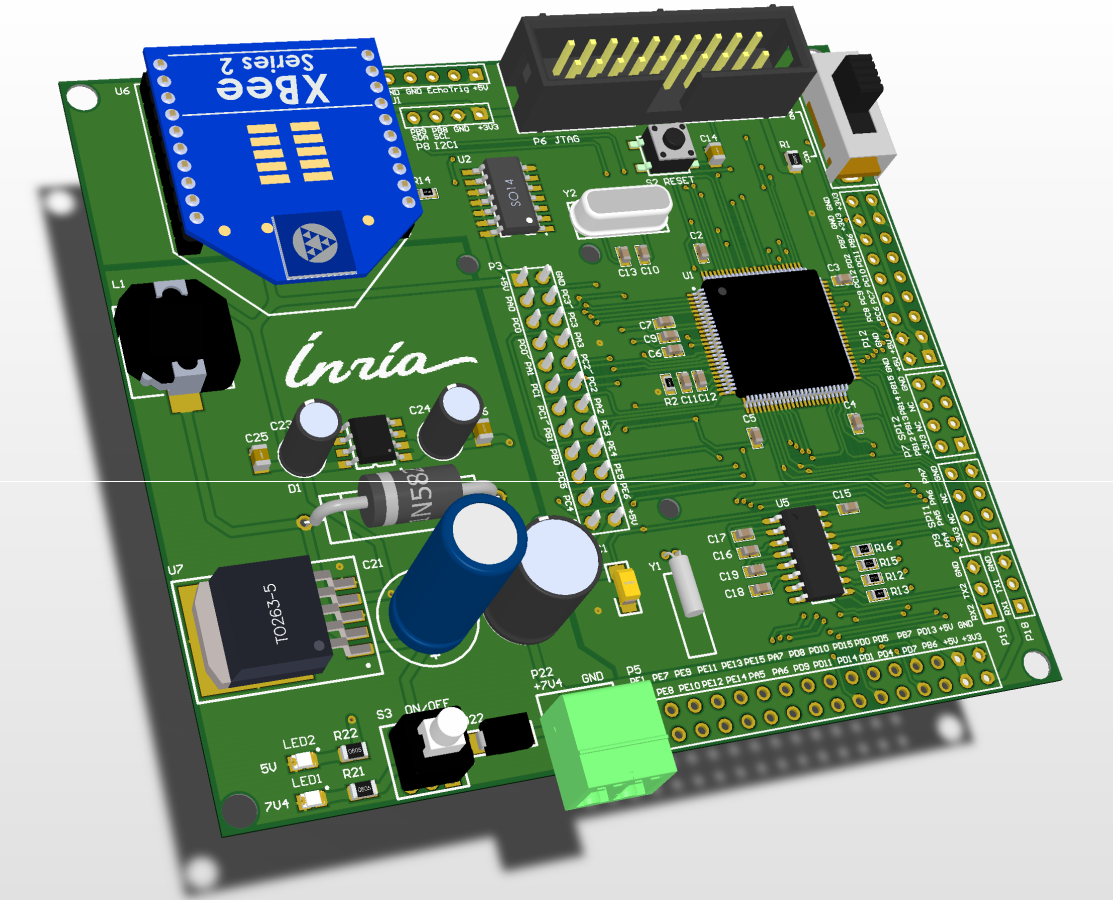

Figure 5.4 – NON-A blimp V2 main control board design 3D view

are also put in the Appendix B. Finally, the created motor driver board is shown in Figure 5.5.

In this first version of designed motor driver board, we used the Metal-Oxide Semiconductor Field Effect Transistor (MOSFET) (Metal Oxide Semiconductor Field Effect Transistor) half bridge chip BTS7960B to build a maximum output current 43A motor driver which has the advantage of high efficiency and low energy waste, and even if we change to a bigger blimp robot and bigger motors, the BTS7960B chip still has the capability to drive the motors normally [InfineonTechnologies, 2018]. Despite those advantages, unfortunately in practice, we encountered the problem that the chosen balloon with helium gas does not have enough payload to carry this motor driver board, it overweighted by several

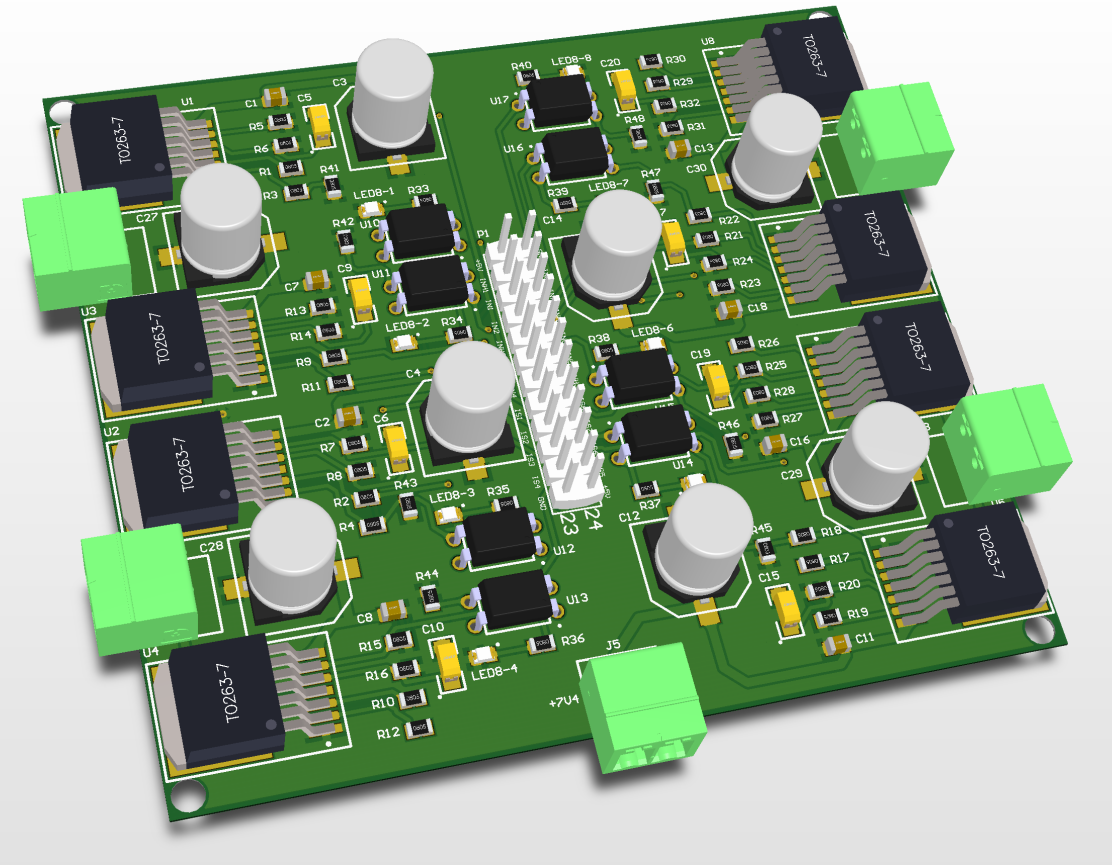

Figure 5.5 – NON-A blimp V2 motor driver board (first version) design 3D view

dozens of grams.

To solve this problem, considering that the chosen 8520 (8.5×20mm for diameter and length) coreless motors usually works at a current level less than 2A, we decided to use an alternative motor driver chip L298N which supports a maximum output of 2A and re-design a lighter version of the motor driver board [STMicroelectronics, 2018a]. Although this chip has lower efficiency when driving motors due to the use of transistor inside, but we have to compromise with the maximum payload the balloon can carry.

The designed electric circuit schematics and PCB of this second version motor driver board are put in the Appendix B. Finally, the created motor driver board with lighter weight is shown in Figure 5.6.

In general, the designed main control board and motor driver board for

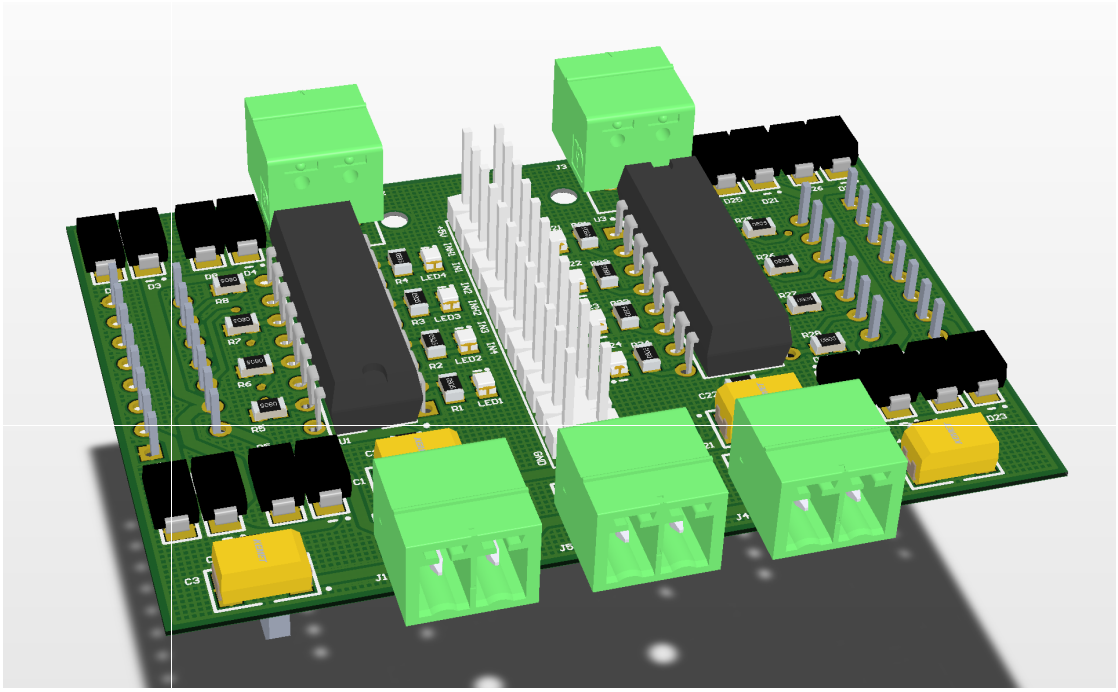

Figure 5.6 – NON-A blimp V2 motor driver board (second version) design 3D view

blimp V2 are fabricated by professional machines thus can be used and tested for longer duration. More importantly, the main control board is left with plenty of potential for extended functions. And the modular design idea facilitates the procedure of maintenance, replacement of broken parts and upgrade of hardware.

5.2.4 NON-A blimp V2 robot: Structure design

Since on the prototype of blimp robot, we only use three motors as actuators, one for altitude regulation and the other two for the motion control in horizontal plane. And during the altitude stabilization tests, it can be observed that the torque of the propeller rotation of the vertical motor will cause the blimp to spin around Z_n-axis in the reverse direction, which means the motion in vertical direction will influence the motion in horizontal plane.

Therefore, in order to cancel the unwanted spinning, we decided to add another motor for the altitude regulation. The two vertical motors are located

symmetrically, they have reversed rotation direction when given same commands, when installed propellers which are also reversed, they will generate propulsive force in the same direction. In addition, the two vertical motors are always given same commands at any moment, thus ideally, the resultant propulsion force generated by the two motors is always in the vertical direction, and the torques of the propellers rotation will cancel each other because they rotate in the opposite direction.

The motors installation is depicted in Figure 2.3, where $\mathcal{M}_2$ and $\mathcal{M}_4$ are the two motors in charge of altitude stabilization, $\mathcal{M}_1$ and $\mathcal{M}_3$ are the two motors mounted in the horizontal plane, responsible for the planar motion control.

The main control board (shown in Figure 5.4) and motor driver board (shown in Figure 5.5 and 5.6) both have a header in the middle of the PCB, which serves as the interface between the two boards, the headers can be plugged together. Then we put screws through the mounting holes (located at the four corners of the main control board and the first version of motor driver board, or around the center header of the main control board and the second version of the motor control board) to fix the two boards as one.

Next, the four motors are mounted as depicted in Figure 2.3, supported by carbon fiber tubes, and fixed to the control board. The two PCB boards and four motors together compose the major part of the blimp robot. The real hardware with first version motor driver board is shown in Figure 5.7. It weights 140 grams (without sensors and battery).

The real hardware with second version motor driver board is shown in Figure 5.8. It weights 123 grams (without sensors and battery).

Finally the control board with motors are fixed to the bottom of the balloon which is filled with helium gas. The balloon has an ellipsoid-like shape, with a length of 105cm, a width of 55cm, and the height is 71cm, it has a volume of about 0.2m^3, when filled with helium, the balloon can carry a total weight of about 200 grams. The NON-A blimp V2 robot is shown in Figure 5.9.

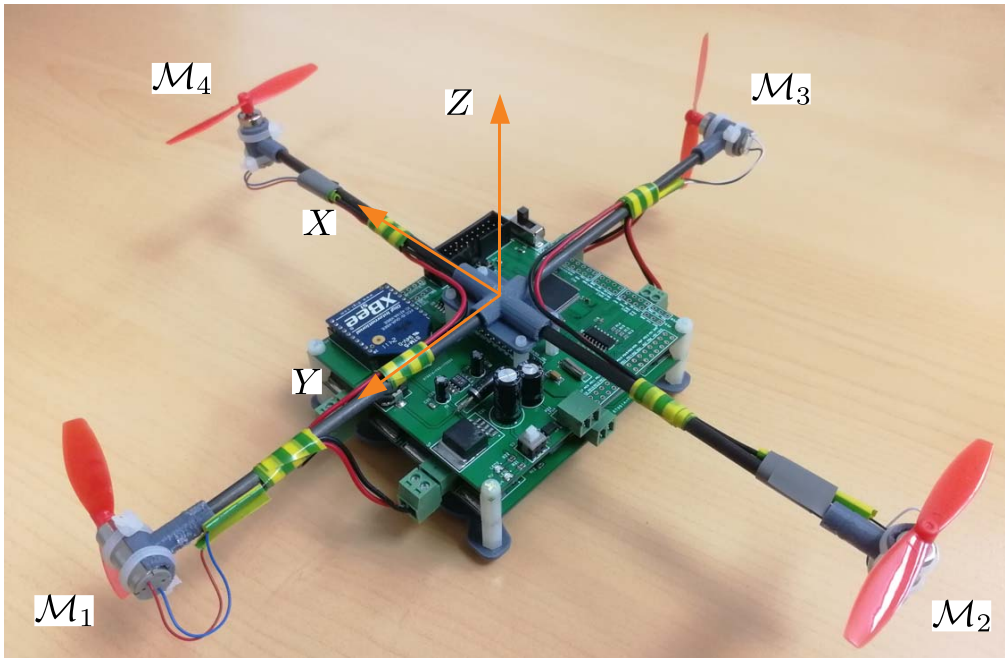

Figure 5.7 – NON-A blimp V2 with first version motor driver board

5.3 Testing environment setup

The tests are taken place in indoor environment such as normal office room and warehouse-like room. The testing room is filled with obstacles for the robot such as tables, chairs, cabinet and floor lamps on the ground, which can cause problem if the blimp measures its altitude with an US sensor faced downward. Moreover, although the ambiance air is rather stable in the indoor environment, it is observed that several reasons will cause airflow perturbation to the blimp robot, such as the airflow generated by air-conditioner, people walking around the room, temperature change inside the room and temperature difference in different parts of the room (e.g. in sunny days, the places closer to glass window has quicker temperature growth than further places). In brief, the indoor testing environment is cluttered and filled with obstacles, and there are many resources of perturbation for the blimp robot. Thus, it demands a higher accuracy of blimp motion control to achieve indoor operations, and it is also preferred that we can obtain an accurate blimp localization information.

Since the best balloon we can find in market has a payload of about 200

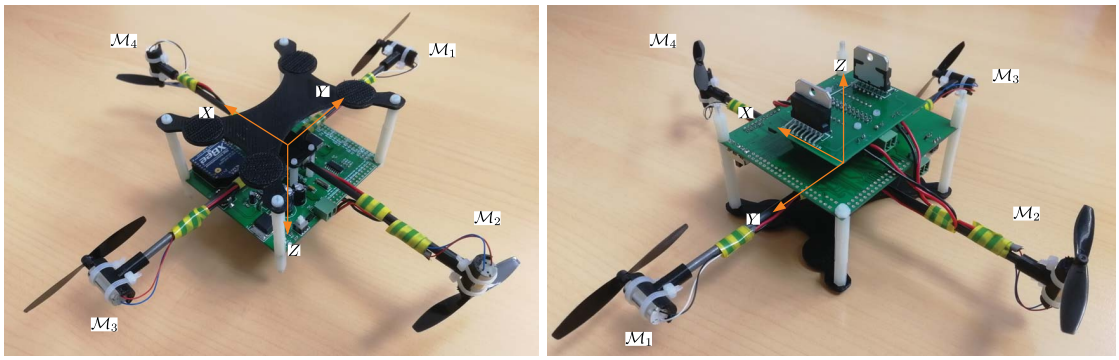

Figure 5.8 – NON-A blimp V2 with second version motor driver board

grams, and the blimp control board with motors has a weight of 123 grams, there only left several dozens of grams for the battery and sensors to be carried on the robot. Hence only low weight sensors can be chosen for robot autonomous localization task, usually the low weight sensors are not very accurate and suffers from measurement noise and drifting problem. That is why after first trials on the blimp prototype robot, we finally decided to use an external camera motion capturing system OptiTrack for the localization of blimp robot inside the testing room.

5.3.1 Implementation with OptiTrack

As it is presented before, the OptiTrack system uses infrared waves to capture the reflective markers mounted on blimp control board, and solves the pose of the robot at a rate of 100 frames per second, and the precision for position measurement is 1mm [NaturalPoint, 2018]. A schema for OptiTrack system is shown in Figure 2.5.

In our testing room, five infrared cameras are installed on ground to form a circle, with their optic axis inclined upward to the vertical axis which passes the center of the circle. The blimp floats inside the circle formed by the camera system, and when it moves it does not leave the view of cameras. The cameras are calibrated and the navigation reference frame $\mathcal{F}_n$ is set in the OptiTrack software before use.

The utilization of OptiTrack system releases the blimp main control board

Figure 5.9 – NON-A blimp V2 robot

from the work of sensor data acquisition, processing and localization calculation, however, it is a separate system with respect to the blimp robot control system, thus it needs to be integrated in order to work well. The OptiTrack-enhanced blimp control system is shown in Figure 5.10.

It works as follows:

- First the OptiTrack system captures and tracks the blimp in indoor testing room, then solves blimp position and orientation and transmits the result via Ethernet to the host PC.

- Next, on host PC, we can see the same Simulink block diagram designed in Chapter 3 and Chapter 4 where the blimp motion controller is implemented. However, the simulated "real system" is replaced by two interfaces: the one receives blimp pose information packet from OptiTrack, decodes the packet and extracts pose info; the other one packs motors commands into packet and sends it via wireless communication module XBee to blimp

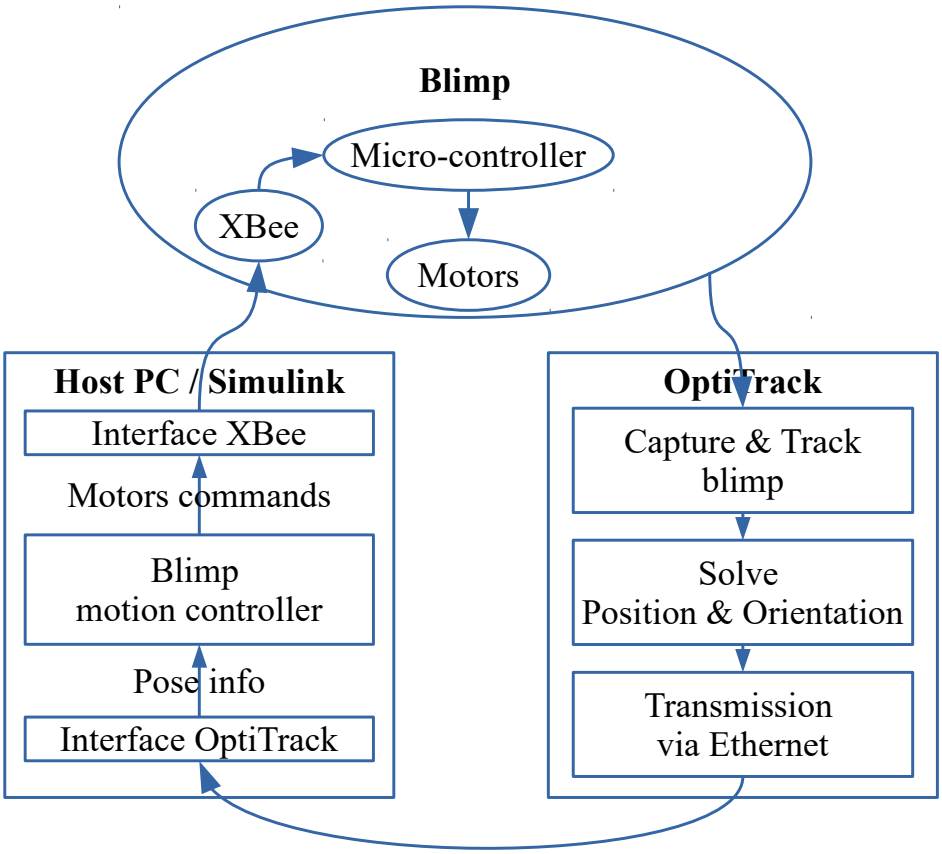

Figure 5.10 – Scheme of Optitrack-enhanced blimp control system

robot on-board STM32 micro-controller.

- Then, on blimp robot control board, the XBee module receives packet from host PC, the micro-controller parses the packet and drives the motors by PWM waves with the help of motor driver board.

- Finally, the blimp is driven by the motors to reach desired goal, and its motion is always captured by OptiTrack system thus closes the system loop.

It is worth to mention that this scheme reuses the Simulink program designed during simulation for simplicity of testing and debugging. In real blimp indoor applications, we should implement all the conceived controllers into the robot

on-board micro-controller, and the host PC only sends the mission commands to blimp. As presented in Chapter 3 and Chapter 4, the designed motion controllers are not complex; they both have only several hundreds lines of code, and do not require high computational capability to solve the command signal. Moreover, the size of program is less than the Flash memory of chosen STM32 micro-controller. Therefore, the Simulink program can be easily transformed to be used on embedded microprocessor.

Although the OptiTrack system is easy to be used and provides high precision measurement of the robot pose, it has the problem of low flexibility, expensive and makes the robot not completely autonomous. In the future, we want to make the blimp robot more autonomous, localizing itself in unknown indoor environment only with on-board sensors. But at this moment, the OptiTrack-enhanced blimp control system is implemented for the validation of the designed control laws.

5.4 Altitude stabilization control

The altitude stabilization tests have been carried out on both the two generations of blimp robot.

5.4.1 On NON-A blimp prototype

For the blimp prototype, the US range finder is used to measure the altitude of robot from the ground of testing room, one of the major problem encountered during tests is the time-varying delay for the control input, that is why we designed a predictor-based controller in Chapter 3. In order to verify the performance of the designed controller, the following tests are made.

5.4.1.1 Real test 1

In this first test, the blimp total weight is carefully adjusted to be approximately equal to the buoyancy force of helium balloon, which means the blimp can almost stay floating with no control input. And with the assumption that there is no external disturbances like airflow perturbation, we temporarily disable

the disturbance compensation term in the controller (3.32), and only check the performance of the predictor-based controller.

The altitude stabilization test results are shown in Figure 5.11.

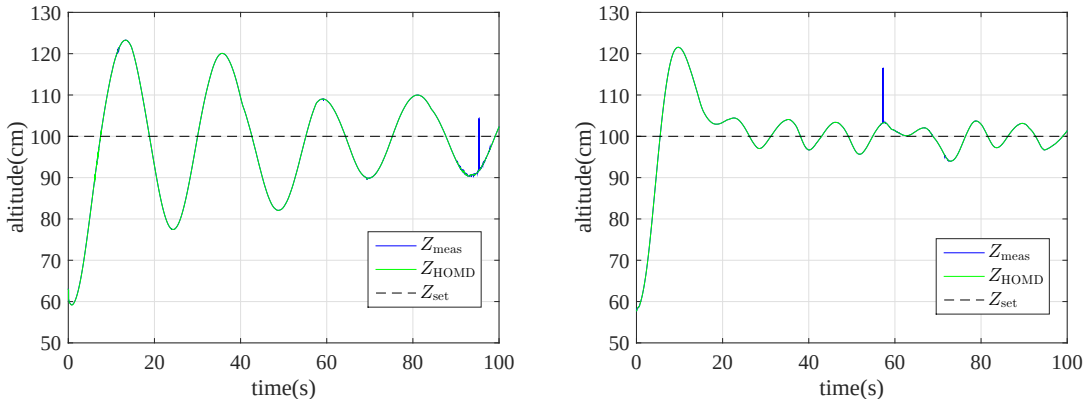

Figure 5.11 – Blimp prototype robot altitude stabilization test 1 - without (left) and with (right) predictor

In the figure, the Z_{set} curve is the desired altitude for the robot to reach, the Z_{meas} curve is the measurement and the Z_{HOMD} curve is the estimated altitude given by HOMD differentiator (zero-order). Notice that the measurement given by ultrasonic range finder has some false detection results due to the reflection of ultrasonic waves at unwanted surfaces instead of the ground. (For the measurement given by OptiTrack system, there are also false detections of the altitude due to the infrared rays emitted from the cameras are reflected on the reflective balloon surface instead of the markers mounted on the control board.) However, thanks to the preprocessing and filter setting in our program, the false detections are filtered and do not influence the estimation results Z_{HOMD} as shown in Figure 5.11.

As it is shown in Figure 5.11, without predictor, the blimp keeps oscillating while trying to reach desired altitude, although the curve seems to have a converging trend, the result is not acceptable. Whereas when we use the predictor-based controller (without disturbance compensation), the blimp can reach to desired altitude with an error less than 10cm within 10 seconds. Then the altitude show some small oscillation around Z_{set}, probably due to the ignored disturbances which come from the time-varying time-delay, parameter

identification inaccuracy, etc.

Briefly, the results show that the predictor is useful to compensate the time-delay and makes the blimp stay in a small region near the desired altitude.

5.4.1.2 Real test 2

In the second real test on blimp prototype robot, the blimp total weight is reduced a little bit from the previous test, which means now the buoyancy force is bigger than gravity force of the robot system, if no control input is given, the robot will keep going upwards (until reaches the ceiling).

The difference between buoyancy force and gravity force can be considered as a constant disturbance to the system. Thus in this test, the disturbance compensation term is added in the controller (3.32), it is supposed to compensate the perturbation and stabilize the robot altitude at desired level. The results are shown in Figure 5.12.

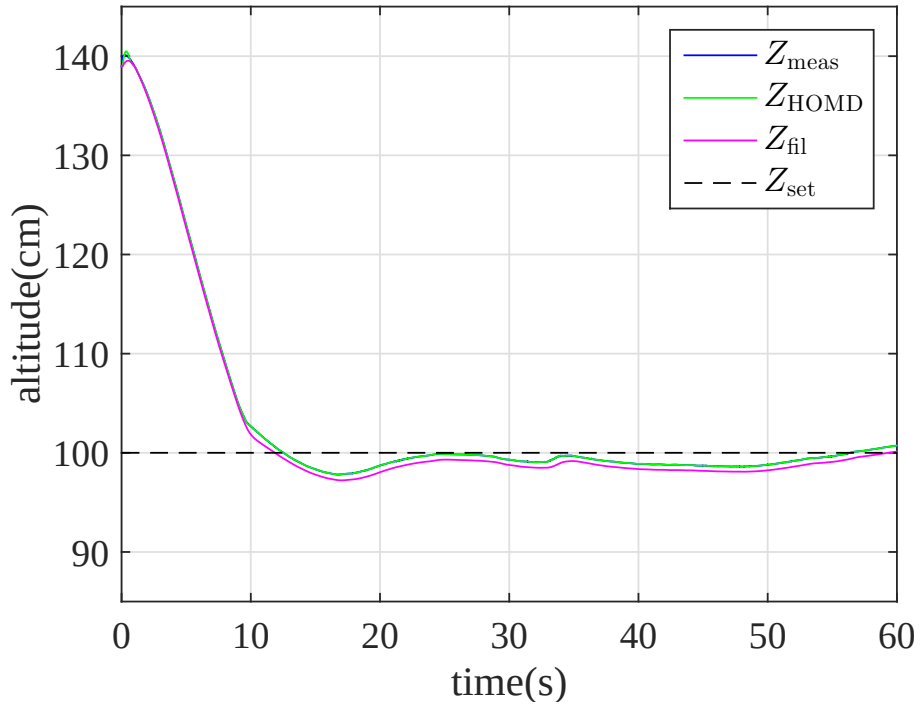

Figure 5.12 – Blimp prototype robot altitude stabilization test 2 - altitude result

The Z_{set}, Z_{meas} and Z_{HOMD} curves have the same meaning as the test 1, and the Z_{fil} curve is the output of the filter (3.25). As it can be seen from the result, the blimp reaches the desired altitude with an error less than 5cm within 10 seconds, and although there is a constant external disturbance, the altitude is stabilized afterwards.

It is worth to mention that there is a difference between Z_{meas} and Z_{fil}, which is used to estimate the disturbance term as shown in equation (3.27).

As for the controller command value and disturbance estimation results, they are shown in Figure 5.13. Notice that the command u_z and disturbance estimation $\hat{d}_z$ are both dimensionless as analyzed before (page 70), thus the ordinates in Figure 5.13 are without units.

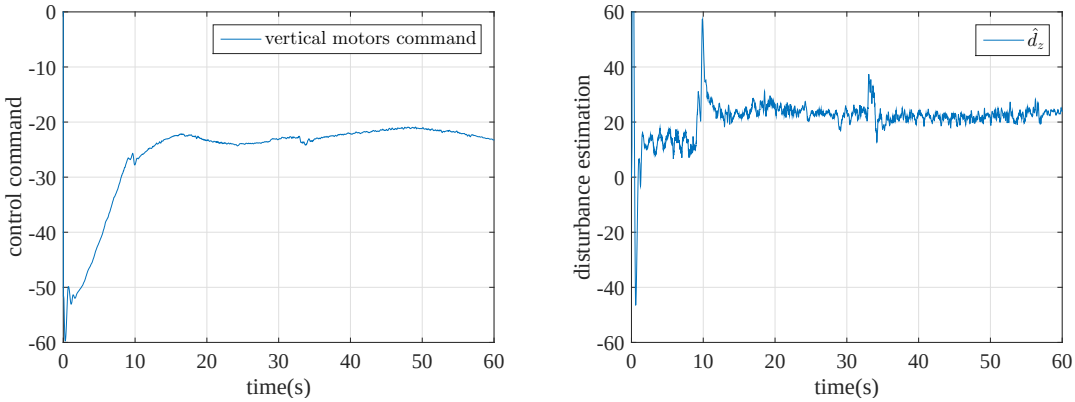

Figure 5.13 – Blimp prototype robot altitude stabilization test 2 - vertical motors command (left) and disturbance estimation (right)

It is observed that the motor command is smooth and not saturated, which is appreciable for the motor, and when the blimp reaches the desired altitude (after 10 seconds), the motor command is approximately equal to the estimated disturbance $\hat{d}_z$. And the disturbance estimation algorithm proposed in Chapter 3 successfully evaluates the constant disturbance (although with chattering) caused by the difference of buoyancy and gravity force.

In general, the test results prove that the designed predictor-based controller with disturbance compensation can stabilize the blimp prototype robot at desired altitude, even with some external disturbances.

5.4.2 On NON-A blimp V2

As it is discussed before, for the NON-A blimp V2 robot, we finally chose to use the OptiTrack camera capturing system for the robot localization, thus the altitude controller also needs to be tested on the V2 robot hardware, by setting the nominal time-delay as $\tau_{\text{nom}} = 0$ (or 0.03s, as there is a 30ms delay in the communication loop of OptiTrack-enhanced blimp control system Figure 5.10) in the altitude controller (3.32).

In this test, the blimp is first initialized with buoyancy force bigger than gravity, then at 105s moment, an additional weight is added to the robot to make the robot heavier than the payload of balloon. Moreover, to test the robustness of the designed altitude controller, we manually push the blimp away at some moments.

The altitude stabilization test result on blimp V2 robot is shown in Figure 5.14.

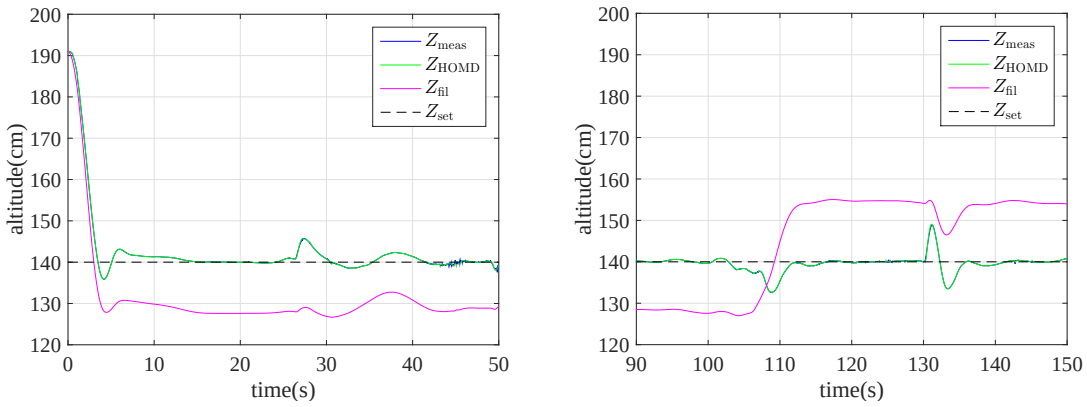

Figure 5.14 – Blimp V2 robot altitude stabilization test - altitude result

The disturbance estimation result is shown in Figure 5.15.

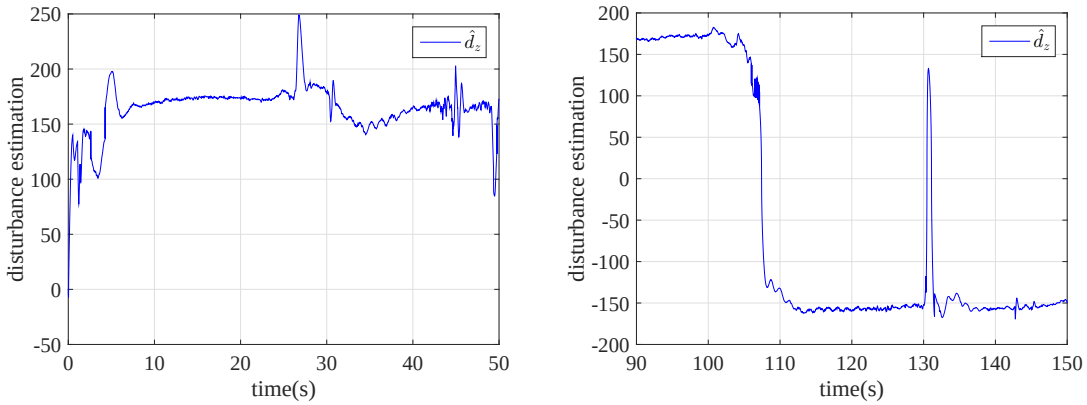

Figure 5.15 – Blimp V2 robot altitude stabilization test - disturbance estimation result

Note that we only extract two parts from the whole test, from 0s to 50s and from 90s to 150s, for clarity of exhibition. As it is shown in the figures, the designed predictor-based altitude controller with disturbance compensation successfully stabilizes the V2 robot at desired altitude. And when the blimp is manually pushed away vertically at moment 26s and 130s, which can be considered as instantaneous external disturbances, it returns to the desired altitude quickly. In addition, when the system weight is changed at 105s moment, which can be considered as an internal system parameters change, the blimp V2 robot also stabilizes itself within 5 seconds.

As for the disturbance estimation results, it can be seen from Figure 5.15 that the designed disturbance estimation method successfully estimates the constant disturbances at the beginning when buoyancy force is bigger than gravity, and at moment 26s and 130s, it responds quickly to the instantaneous external disturbances. Moreover, when the system weight is changed at moment 105s, and the external constant disturbance changes sign, the $\hat{d}_z$ also follows the change.

A video of this experiment can be found here: `https://drive.google.com/drive/folders/1YmksewRW0odrUsALI--CyPM4qWGIUWv6?usp=sharing`.

The experiment results show that the predictor-based altitude controller with disturbance compensation conceived in Chapter 3 works well on both the blimp prototype and V2 robot, and it is robust against disturbances such as

system parameter identification inaccuracy, exterior constant or instantaneous perturbations.

5.5 Validation of the complete motion controller

Finally, the altitude controller designed in Chapter 3 and the planar movement controller designed in Chapter 4 can be combined together to achieve the complete motion control of the blimp robot.

Two types of tests are made to validate the robot motion controller. They are carried out on the NON-A blimp V2 robot, and the OptiTrack-enhanced control system (Figure 5.10) is applied.

The first one is to stabilize the blimp at a point in the space, which can be considered as a special case of trajectory tracking with constant s_{ref}, r_{ref} and z_{ref}. The reference point is set as (units: cm):

$$\begin{cases} s_{\text{ref}} = 60 \\ r_{\text{ref}} = -50 \\ z_{\text{ref}} = 300 \end{cases}$$

As it is discussed before, we achieve a practical control of the point Q instead of the exact control of blimp body-fixed frame center O_b. Therefore when the blimp is regulated to the goal position $(s_{\text{ref}}, r_{\text{ref}}, z_{\text{ref}})$, for the point O_b, its altitude is exactly regulated but on the horizontal plane its position lies practically on a circle centered at $(s_{\text{ref}}, r_{\text{ref}})$ and with a radius q (5cm in our setting).

The result of blimp robot point stabilization is shown in Figure 5.16.

It can be seen that the blimp reaches the goal point with position error less than 10cm within 20 seconds, which is acceptable considering the slow dynamics of the robot. Moreover, when we manually push the blimp horizontally at moment 50s and 110s, it returns to the set point. In addition, the system buoyancy force is bigger than gravity, but the altitude is still stabilized face to this constant disturbance.

The disturbance estimation results are shown in Figure 5.17. Notice that the $\hat{d}_s$ and $\hat{d}_r$ both have the unit of cm/s^2, but the $\hat{d}_z$ is dimensionless as analyzed

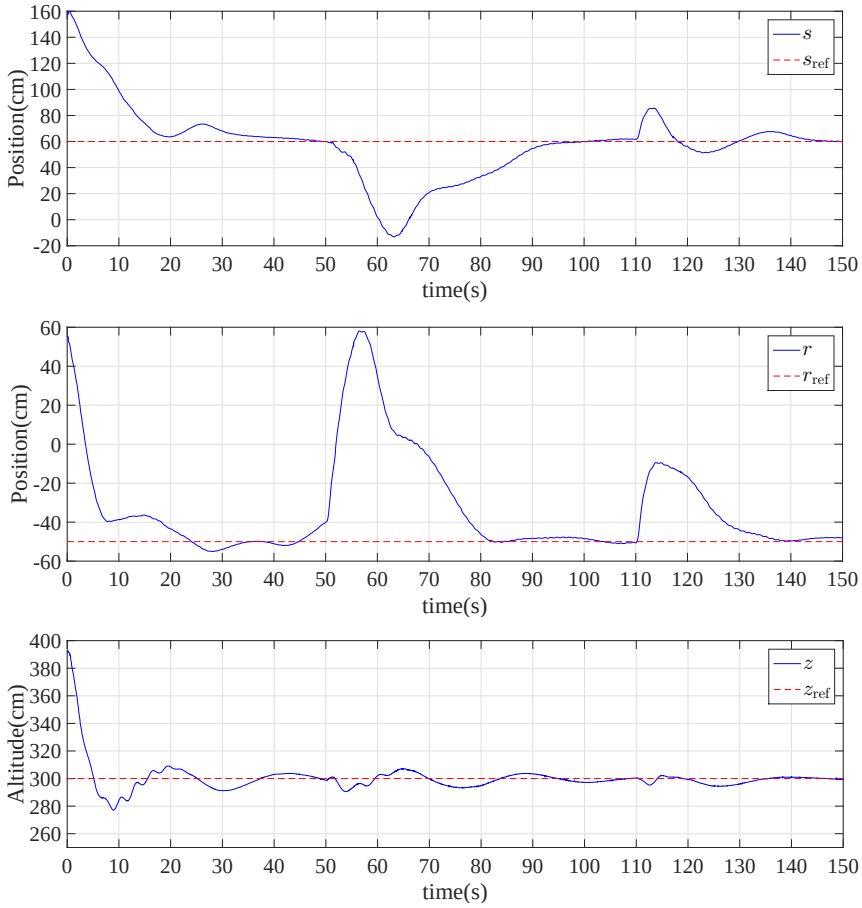

Figure 5.16 – Blimp V2 robot point stabilization test - position of control point $Q(s, r, z)$

before (page 70 and page 104).

It is clear that the estimation $\hat{d}_s$ and $\hat{d}_r$ respond quickly to the external disturbances at moment 50s and 110s, thus help the controller to regulate the blimp back to its desired position, and the $\hat{d}_z$ also correctly estimates the constant external disturbances. It is worth to notice that although theoretically the altitude motion and planar movement of the blimp robot are decoupled, in reality they are not. The influences of one motion to another is also estimated by the designed algorithm, which can be observed at the first 30 seconds during the reaching phase of the blimp to its goal position.

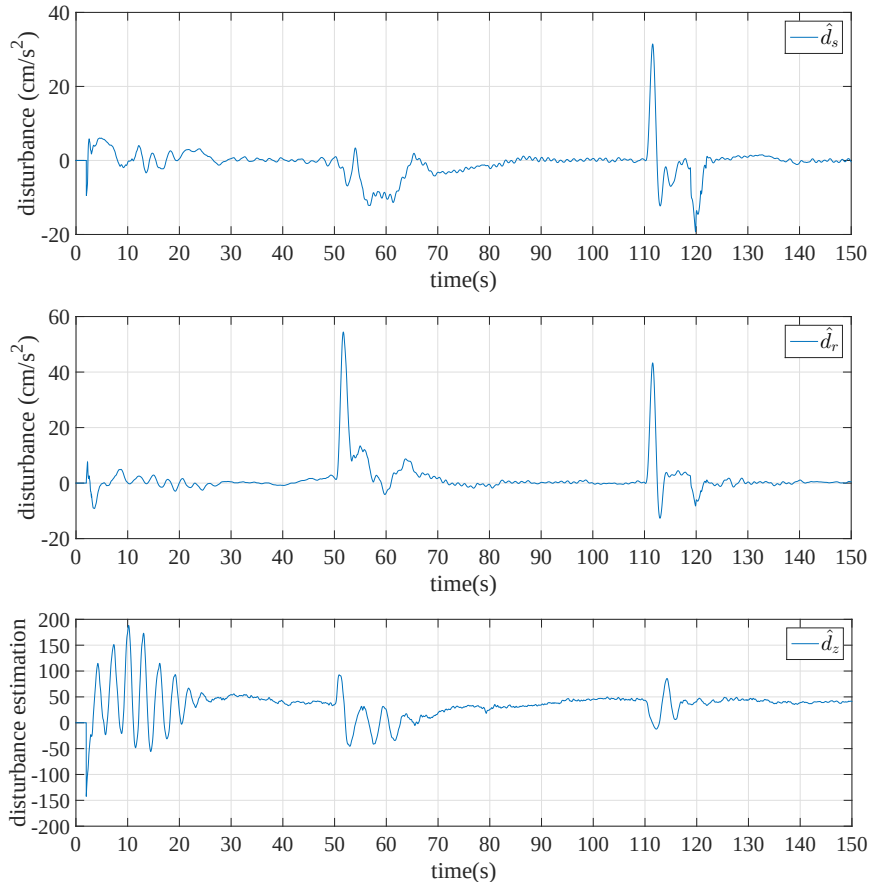

Figure 5.17 – Blimp V2 robot point stabilization test - disturbance estimation $\hat{d}_s$ (top), $\hat{d}_r$ (middle), $\hat{d}_z$ (bottom)

The point stabilization test proves the efficiency and robustness of the designed blimp robot motion controllers. It can be seen as a simulation scenario of the blimp surveilling a target room at a fixed location.

A video of this experiment can be found here: `https://drive.google.com/drive/folders/1Z0f-ZEppNcTuCIsqHWaAdpYQib6jpjF1?usp=sharing`.

In the second test, we want to make the blimp robot follow a designed path by choosing several intermediate points on the desired trajectory, when the blimp approaches closely enough the current way point, the goal reference point is set to the next intermediate point, and the controller continuously makes the blimp reaching the goal point. If an infinite number of intermediate points chosen, the

process is same as the trajectory tracking. But in our test, only few intermediate points are chosen for the blimp to move in a simulated office room.

The intermediate points are chosen as (units: cm):

$$\begin{cases} W_1(s,r,z) = (30,0,180) \\ W_2(s,r,z) = (30,0,310) \\ W_3(s,r,z) = (130,0,310) \\ W_4(s,r,z) = (130,0,200) \\ W_5(s,r,z) = (130,-50,160) \\ W_6(s,r,z) = (130,-100,120) \end{cases}$$

The path following result is shown in Figure 5.18. Only the part from 20s to 77s is shown in the figure. For practical reasons, the threshold to judge whether the robot has reached a way point or not is set as 10cm, it is enough for our application scenario.

A video of this experiment can be found here: `https://drive.google.com/drive/folders/1rzP5nhhFL9jfXn_ipbE29lJ8-QfrcaPW?usp=sharing`.

This test is a simulation scenario of the blimp following a predefined path and surveilling a room. In practice, the blimp is set to pass through narrow space such as windows. In order to achieve that operation, for the blimp horizontal plane movement, in fact two controllers are used, one of them is presented before, and the other one is a simple yaw angle controller. The two controllers operate alternatively, which means when the blimp is far from the current reference point, the position regulation controller works; when it approaches the reference point with an acceptable tolerance, the yaw angle controller is switched on to stabilize the yaw angle at desired direction. Then if both the position and heading angle are well stabilized, the blimp can start to move to its next goal, otherwise, the two controllers works alternatively to achieve the mission. The video shows that the blimp moves smoothly and follows the designed way points with satisfying accuracy. Therefore in real application scenarios, it should be able to finish the missions successfully.

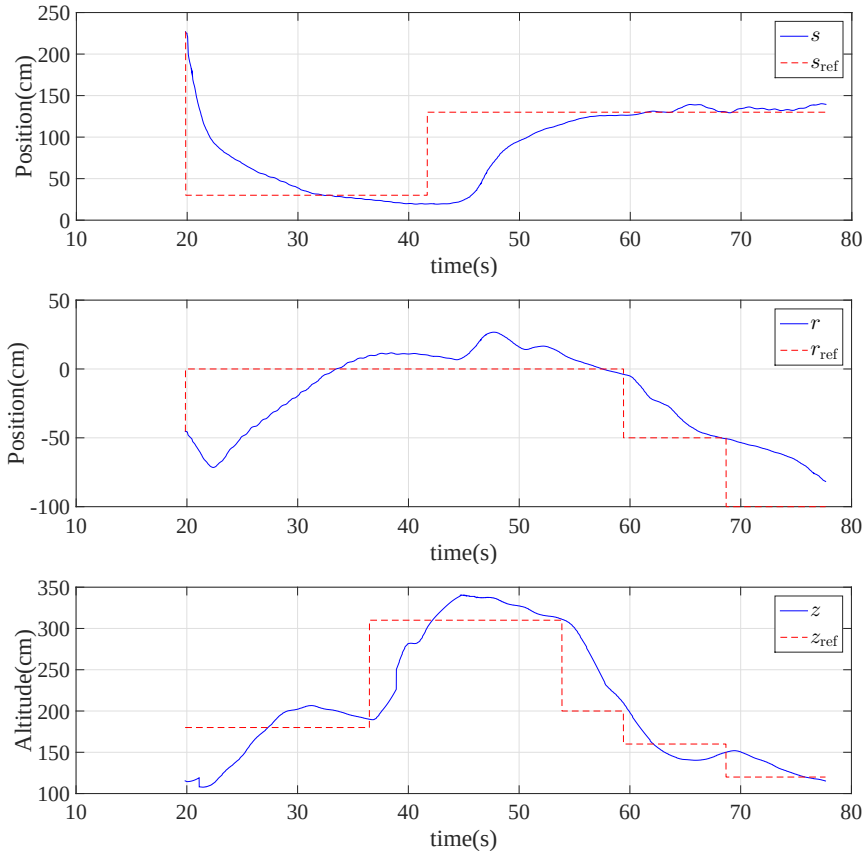

Figure 5.18 – Blimp V2 robot path following test - position of control point $Q(s, r, z)$

5.6 Conclusion

In this chapter, we summarize all the previous theoretical works on the modeling of blimp, controllers design for robot decoupled altitude and planar motion, and try to implement them on the real robot by doing the practical part of this work.

First, the blimp robot system necessary functionalities are analyzed, and depending on the analysis, we proposed a division of the system hardware and decided to follow a modular design procedure for each of the sub-systems. For the blimp robot hardware, it passed though two generations of design and creation, we presented the first generation - the NON-A blimp prototype robot briefly, followed by the detailed introduction on the NON-A blimp V2 robot,

including the control board electric circuit design (for clarity, some details are put in Appendix B), and the robot structure design.

Then, based on the experiences gained from tests, we decided to implement the OptiTrack camera motion capturing system into our blimp robot system. The scheme of the implementation of OptiTrack into the robot system is presented, advantages and inconveniences are analyzed.

Next, we want to test the motion controllers conceived in Chapter 3 and Chapter 4. Altitude stabilization tests are carried out both on the prototype and V2 robot, and the results prove the efficiency and robustness of the predictor-based altitude controller with disturbance compensation on the real robot.

Finally, the planar movement controller is combined with altitude controller in the complete motion controller and it is validated by point stabilization and path following tests.

Videos are taken to show more intuitively the results obtained and the performance of controllers, which prove that the developed blimp robot is able to achieve indoor operations such as long-term surveillance by stabilizing itself at a fixed position or by following a predefined path. In addition, the created NON-A blimp V2 robot can be served as a scientific research and education platform.

Conclusion and Perspectives

In this thesis, a blimp robot is developed for indoor operations. The obtained results can be divided into two groups: theoretical and practical results.

In Chapter 1, a brief introduction is given regarding the actual situation of the robotics field, and a special attention is given to aerial robots, among which the blimp attracts the researchers interests for its various advantages compared to other aircrafts. Then a review on the state of the art of blimp and airships researches is presented, including the modeling, sensor technologies, and controller designs. We are motivated by this subject because of the various potential applications of blimp robot in indoor environment, and there are many challenges remain to be solved.

For the theoretical part of work, we start with the study of the blimp robot model in Chapter 2. The commonly used 6-DOF blimp dynamic model is presented in detail. As we observed from the related works on small indoor blimp robots where the 6-DOF model is applied, the obtained results are underwhelming due to the complexity of the model and the difficulties for control laws design raised by the model. Therefore a novel approach to model the motion of blimp robot is proposed. The idea is to use a simplified model complemented with disturbance term which is estimated in real-time to reduce the model complexity while assure the performance of controller and improve its robustness. Based on this idea, the 6-DOF dynamic model is simplified under reasonable assumptions and decoupled into two separate parts: the altitude motion and planar movement model. Tests are carried out to identify the parameters of these nominal models.

Then in Chapter 3, the controller for altitude motion is designed. Since the disturbance term is complemented to the nominal model, it needs to be

estimated and compensated in real-time. Moreover, based on the experiences gained from tests, several hardware defects are concluded and reflected in the changed model. For the purpose of conceiving the output feedback controller, first an observer is designed with the help of differentiators for the state and switching signal estimation. Next, the disturbance term is estimated by proposed method, and compensated in the predictor-based controller. Simulations are accomplished to verify the efficiency and performance of designed controller.

Next, we focus on the blimp movement in horizontal plane, and conceive a disturbance compensation based controller in Chapter 4. Due to the fact that the model of blimp planar movement is under-actuated, two different approaches are discussed to transform the system to a simpler one for the ease of conceiving a feedback stabilizing controller. Then similarly as for the altitude motion controller design, a method to estimate the disturbance terms is proposed, which helps to improve the robustness of conceived controller against perturbations. Trajectory tracking and point stabilization simulations are made to confirm the performance of planar movement disturbance compensation based controller.

Finally, the presentation of practical part of this work comes at Chapter 5. The robot system function requirements to achieve desired indoor applications are analyzed. Based on the analysis, a modular designing idea is proposed for the blimp robot system hardware creation. The NON-A blimp V2 robot is designed and developed from the beginning with the experiences obtained during tests on the prototype robot. In addition, the camera motion capturing system is implemented to enhance the blimp robot system and to provide high-accuracy localization information in the testing environment. Control laws are programmed then real tests are achieved on the developed blimp robot to validate the designed motion control laws.

The results show that our robot implemented with conceived control laws is able to accomplish several indoor missions such as point stabilization and path following, and it is capable to be utilized in applications such as long-term surveillance and monitoring, or just serves as a scientific research and education platform.

Despite the encouraging results obtained, at the end of this thesis, there still remain some issues unsolved, and can be studied in the future work. Several

simplifications are made during the model parameter identification process in Chapter 2, due to the lack of measurement ability during tests. If in the future, we have access to more powerful and accurate testing devices, the model parameters can be re-identified with higher accuracy, and the added-mass terms along X_b- and Y_b-axis direction can be studied and considered when designing the controllers.

In addition, in the future we can study the use of flatness theory to design better trajectory for the blimp to track, and realize exact position and pose control of the robot instead of the practical one.

Furthermore, in the end an external camera motion capturing system is added to the blimp robot system to provide localization information, which prevents the use of blimp robot in a more general indoor environment. In order to make the robot completely autonomous and realize complex tasks such as unknown environment exploration and mapping, it should be able to localize itself only with sensors mounted on-board, which is a challenging task for all the autonomous mobile robots. But unlike ground vehicles or multi-rotors aerial robots which have a much bigger payload capacity, the indoor blimp robot faces the dilemma such that in order to carry more weight and implement accurate sensors in the system, the balloon has to be bigger; but on the other hand, for the robot to move freely in indoor cluttered environment, it is preferable that it has smaller size. According to the analysis, it seems that the on-board camera is the optimal choice for the autonomous navigation and localization of indoor blimp robot, and there already exist lots of works on the use of camera in the mobile robot system. Moreover the blimp robot is a perfect platform for the camera since it has the ability for VTOL, stationary and low-speed flight. Therefore in the future, it might be interesting to implement the camera to blimp robot system and achieve fully autonomous navigation of the robot in any accessible indoor environment.

Although the real tests carried out on the blimp robot at this moment only show that the robot is capable of achieving indoor application such as long-term surveillance, it is worth to mention that the designed NON-A blimp V2 hardware is reserved with plenty of interfaces for devices connection and it has the ability for potential functionalities extension and accomplishing different missions. For

instance, if the blimp is equipped with lightweight thin-film display device on the surface of balloon, it can be used for advertising and entertainment purpose, or applied for interesting human-robot interaction; if the blimp is installed with a RFID scanner, it can be utilized in the warehouse for goods inventory; finally, if the blimp is installed with camera, it can achieve fully autonomous motion in unknown environment and realize exploration or mapping tasks.

In general, there are still a lot of interesting and challenging topics to be studied in the future about the indoor blimp robot. We hope that this work can serve as an introduction to this attractive topic, make some contributions to the robotics field, and inspire the following researchers.

Bibliography

Aguiar, A. P. and J. P. Hespanha (2007). "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty". In: *IEEE Transactions on Automatic Control* 52.8, pp. 1362–1379 (cit. on p. 20).

Al-Jarrah, R. and H. Roth (2013a). "Design blimp robot based on embedded system and software architecture with high level communication and fuzzy logic". In: *Mechatronics and its Applications (ISMA), 2013 9th International Symposium on*. IEEE, pp. 1–6 (cit. on pp. 8, 18, 24).

— (2013b). "Developed blimp robot based on ultrasonic sensors using possibilities distribution and fuzzy logic". In: *Journal of Automation and Control Engineering* 1.2 (cit. on pp. 8, 24).

Aliveal (Dec. 2018). *ALM-2451 2.4GHz Wireless Mini Camera System Datasheet*. https://www.robotshop.com/media/files/PDF/datasheet-alm-2451g.pdf (cit. on p. 56).

Allen, H. J. and E. W. Perkins (1951). *Characteristics of flow over inclined bodies of revolution*. National Advisory Committee for Aeronautics (cit. on p. 11).

Arduino (Dec. 2018). *Arduino Fio Overview*. https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardFio (cit. on p. 128).

Artstein, Z. (1983). "Stabilization with relaxed controls". In: *Nonlinear Analysis, TMA* 7.11, pp. 1163–1173 (cit. on p. 21).

Azinheira, J. R. and A. Moutinho (2008). "Hover control of an UAV with backstepping design including input saturations". In: *IEEE Transactions on Control Systems Technology* 16.3, pp. 517–526 (cit. on p. 21).

Azinheira, J. R., A. Moutinho, and E. C. De Paiva (2006). "Airship hover stabilization using a backstepping control approach". In: *Journal of Guidance, Control, and Dynamics* 29.4, pp. 903–914 (cit. on p. 21).

— (2009). "A backstepping controller for path-tracking of an underactuated autonomous airship". In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 19.4, pp. 418–441 (cit. on p. 21).

Badia, S. B. i, P. Pyk, and P. F. Verschure (2005). "A biologically based flight control system for a blimp-based UAV". In: *Robotics and Automation (ICRA),*

2005 IEEE International Conference on. IEEE, pp. 3053–3059 (cit. on pp. 7, 19, 23, 196).

Barfoot, T. D. (2017). *State Estimation for Robotics*. Cambridge University Press (cit. on p. 34).

Beji, L. and A. Abichou (2005). "Tracking control of trim trajectories of a blimp for ascent and descent flight manoeuvres". In: *International Journal of Control* 78.10, pp. 706–719 (cit. on pp. 7, 21).

Bernuau, E., D. Efimov, W. Perruquetti, and A. Polyakov (2014). "On homogeneity and its application in sliding mode control". In: *Journal of the Franklin Institute* 351.4, pp. 1866–1901 (cit. on p. 77).

Bessert, N. and O. Frederich (2005). "Nonlinear airship aeroelasticity". In: *Journal of fluids and structures* 21.8, pp. 731–742 (cit. on pp. 12, 30).

Bestaoui, Y. and S. Hima (2001). "Some insight in path planning of small autonomous blimps". In: *Archives of Control Sciences, Polish Academy of Sciences* 11.3, pp. 21–49 (cit. on pp. 12, 29, 35).

Britting, K. (1971). *Inertial navigation systems analysis*. John Wiley & Sons Canada, Limited (cit. on p. 15).

Brockett, R. W. (1983). "Asymptotic stability and feedback stabilization". In: *Differential geometric control theory* 27.1, pp. 181–191 (cit. on p. 99).

Burgess, C. P. and A. Starchild (1927). *Airship design*. Vol. 192. Ronald Press Company (cit. on p. 12).

Burri, M., L Gasser, M Kach, M Krebs, S Laube, A Ledergerber, D. Meier, R Michaud, L. Mosimann, L Muri, et al. (2013). "Design and control of a spherical omnidirectional blimp". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, pp. 1873–1879 (cit. on pp. 8, 16, 19, 196).

Carullo, A. and M. Parvis (2001). "An ultrasonic sensor for distance measurement in automotive applications". In: *IEEE Sensors journal* 1.2, p. 143 (cit. on p. 17).

Cheres, E., Z. Palmor, and S. Gutman (1990). "Min-max predictor control for uncertain systems with input delays". In: *IEEE Transactions on Automatic Control* 35.2, pp. 210–214 (cit. on p. 83).

Cook, M., J. Lipscombe, and F. Goineau (2000). "Analysis of the stability modes of the non-rigid airship". In: *The Aeronautical Journal* 104.1036, pp. 279–290 (cit. on p. 11).

Crouch, T. D. (2009). *Lighter than air: an illustrated history of balloons and airships*. Johns Hopkins University Press Baltimore, MD (cit. on p. 4).

Curtiss, H., D. Hazen, and W. Putman (1976). "LTA aerodynamic data revisited". In: *Journal of Aircraft* 13.11. cited By 6, pp. 835–844 (cit. on p. 38).

Dabroom, A. M. and H. K. Khalil (1997). "Numerical differentiation using high-gain observers". In: *Decision and Control (CDC), 1997. Proceedings of the 36th IEEE Conference on*. Vol. 5. IEEE, pp. 4790–4795 (cit. on p. 74).

— (1999). "Discrete-time implementation of high-gain observers for numerical differentiation". In: *International Journal of Control* 72.17, pp. 1523–1537 (cit. on p. 75).

D'Andréa-Novel, B. and S. Thorel (2016). "Control of non holonomic or underactuated mechanical systems: The examples of the unicycle robot and the slider". In: *Esaim Control Optimisation & Calculus of Variations* 22.4 (cit. on p. 51).

D'Andrea, R. (2012). "Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead". In: *IEEE Transactions on Automation Science and Engineering* 9.4, pp. 638–639 (cit. on p. 3).

Digi (Jan. 2019). *Digi XBee Ecosystem*. `https://www.digi.com/xbee` (cit. on p. 186).

Dolce, J. L. and A. Collozza (2005). "High-altitude, long-endurance airships for coastal surveillance". In: *National Aeronautics and Space airship interior temperature* (cit. on p. 5).

Dorrington, G. (2005). "Development of an airship for tropical rain forest canopy exploration". In: *The Aeronautical Journal* 109.1098, pp. 361–372 (cit. on p. 6).

Dudek, G. and M. Jenkin (2008). "Inertial sensors, GPS, and odometry". In: *Springer Handbook of Robotics*. Springer, pp. 477–490 (cit. on p. 15).

Elfes, A., S. S. Bueno, M. Bergerman, E. C. De Paiva, J. G. Ramos, and J. R. Azinheira (2003). "Robotic airships for exploration of planetary bodies with an atmosphere: Autonomy challenges". In: *Autonomous Robots* 14.2-3, pp. 147–164 (cit. on p. 6).

Esfandiari, F. and H. K. Khalil (1992). "Output feedback stabilization of fully linearizable systems". In: *International Journal of Control* 56.5, pp. 1007–1037 (cit. on p. 75).

Etkin, B. (1981). "Turbulent wind and its effect on flight". In: *Journal of Aircraft* 18.5, pp. 327–345 (cit. on p. 12).

Euler, L. (1773). "Exercitationes analyticae". In: *Novi commentarii academiae scientiarum Petropolitanae*, pp. 173–204 (cit. on p. 171).

Everett, H. (1995). *Sensors for mobile robots*. AK Peters/CRC Press (cit. on p. 12).

Fantoni, I., R. Lozano, F. Mazenc, and K. Pettersen (1999). "Stabilization of a nonlinear underactuated hovercraft". In: *Decision and Control (CDC), 1999. Proceedings of the 38th IEEE Conference on*. Vol. 3. IEEE, pp. 2533–2538 (cit. on p. 51).

Finkenzeller, K. (2010). *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons (cit. on p. 17).

Fossen, T. I. et al. (1994). *Guidance and control of ocean vehicles*. Vol. 199. 4. Wiley New York (cit. on pp. 11, 29, 36, 38–41, 171, 172, 178).

Fukao, T., K. Fujitani, and T. Kanade (2003a). "An autonomous blimp for a surveillance system". In: *Intelligent Robots and Systems (IROS), 2003 IEEE/RSJ International Conference on*. Vol. 2. IEEE, pp. 1820–1825 (cit. on pp. 7, 22, 196).

— (2003b). "Image-based tracking control of a blimp". In: *Decision and Control (CDC), 2003. Proceedings. 42nd IEEE Conference on*. Vol. 5. IEEE, pp. 5414–5419 (cit. on pp. 7, 19, 21, 196).

Fukao, T., T. Kanzawa, and K. Osuka (2007). "Tracking control of an aerial blimp robot based on image information". In: *Control Applications, 2007. CCA 2007. IEEE International Conference on*. IEEE, pp. 874–879 (cit. on pp. 7, 21).

Fukao, T., A. Yuzuriha, T. Suzuki, T. Kanzawa, T. Oshibuchi, K. Osuka, T. Kohno, M. Okuyama, Y. Tomoi, and M. Nakadate (2008). "Inverse Optimal Velocity Field Control of an Outdoor Blimp Robot–Blimp Surveillance Systems for Rescue–". In: *IFAC Proceedings Volumes* 41.2, pp. 4374–4379 (cit. on pp. 7, 21).

Fukushima, H., R. Saito, F. Matsuno, Y. Hada, K. Kawabata, and H. Asama (2006). "Model predictive control of an autonomous blimp with input and output constraints". In: *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. IEEE, pp. 2184–2189 (cit. on pp. 7, 16, 22).

Fukushima, H., K. Kon, Y. Hada, F. Matsuno, K. Kawabata, and H. Asama (2007). "State-predictive control of an autonomous blimp in the presence of time delay and disturbance". In: *Control Applications, 2007. CCA 2007. IEEE International Conference on*. IEEE, pp. 188–193 (cit. on pp. 7, 22).

Funk, P., T. Lutz, and S. Wagner (2003). "Experimental investigations on hull-fin interferences of the LOTTE airship". In: *Aerospace Science and Technology* 7.8, pp. 603–610 (cit. on p. 11).

Gardner, J. W. and V. K. Varadan (2001). *Microsensors, MEMS and smart devices*. John Wiley & Sons, Inc. (cit. on p. 6).

Gerken, L. (1990). *Airships: history and technology*. American Scientific Corporation (cit. on p. 5).

Geromel, J., M. C. de Oliveira, and L. Hsu (1998). "LMI characterization of structural and robust stability". In: *Linear Algebra and its Applications* 285.1-3, pp. 69–80 (cit. on p. 82).

Glade, D. (2000). *Unmanned aerial vehicles: Implications for military operations*. Tech. rep. Air Univ Press Maxwell Afb Al (cit. on p. 3).

Gomes, S. (Oct. 1990). "An investigation into the flight dynamics of airships with application to the YEZ-2A". PhD thesis. Cranfield Institute of Technology (cit. on pp. 11, 12, 29, 31, 36, 196).

Green, W. E., K. W. Sevcik, and P. Y. Oh (2005). "A competition to identify key challenges for unmanned aerial robots in near-earth environments". In: *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*. IEEE, pp. 309–315 (cit. on pp. 7, 23).

Guerra, M., C. Vázquez, D. Efimov, G. Zheng, L. Freidovich, and W. Perruquetti (2015). "ε-invariant output stabilization: Homogeneous approach and dead zone compensation". In: *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, pp. 6874–6879 (cit. on pp. 70, 105).

Guerra, M., D. Efimov, G. Zheng, and W. Perruquetti (2016). "Avoiding local minima in the potential field method using input-to-state stability". In: *Control Engineering Practice* 55, pp. 174–184 (cit. on p. 101).

Hansen, Z. C. (1979). *The Goodyear Airships*. Airship International Press (cit. on p. 6).

Hartley, R. and A. Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press (cit. on pp. 18, 19).

Hoerner, S. F. (1958). *Fluid-dynamic Drag: practical information on aerodynamic drag and hydrodynamic resistance*. Hoerner Fluid Dynamics (cit. on pp. 11, 38).

Hofmann-Wellenhof, B., H. Lichtenegger, and E. Wasle (2007). *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media (cit. on p. 16).

Huang, C. J. and C. Dalton (1976). *Cargo transportation by airships: a systems study*. Tech. rep. NASA (cit. on p. 6).

Hygounenc, E., I.-K. Jung, P. Soueres, and S. Lacroix (2004). "The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping". In: *The International Journal of Robotics Research* 23.4-5, pp. 473–511 (cit. on pp. 6, 21, 30).

Industrial Internet Consortium (2014). "Introductory white paper". In: *IIC, White paper, March* (cit. on p. 2).

InfineonTechnologies (Dec. 2018). *BTS7960B Datasheet*. http://www.robotpower.com/downloads/BTS7960_v1.1_2004-12-07.pdf (cit. on pp. 133, 184).

International Federation of Robotics (2018). *World Robotics Report 2018* (cit. on pp. 1–3).

International Standardization Organization (ISO) (2012). "8373: 2012. Robots and robotic devices–Vocabulary". In: *International Standardization Organization (ISO)* (cit. on p. 2).

InvenSense (Dec. 2018). *MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking Devices*. https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/ (cit. on p. 54).

Isidori, A. (2013). *Nonlinear control systems*. Springer Science & Business Media (cit. on p. 74).

Jones, S. and J. DeLaurier (1983). "Aerodynamic estimation techniques for aerostats and airships". In: *Journal of Aircraft* 20.2, pp. 120–126 (cit. on p. 11).

José, R. Azinheira, E. C. de Paiva, and S. S. Bueno (2002). "Influence of wind speed on airship dynamics". In: *Journal of Guidance, Control, and Dynamics* 25.6, pp. 1116–1124 (cit. on p. 12).

Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996). "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4, pp. 237–285 (cit. on p. 24).

Kale, S., P. Joshi, and R. Pant (2005). "A generic methodology for determination of drag coefficient of an aerostat envelope using CFD". In: *AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*, p. 7442 (cit. on p. 11).

Kantor, G., D. Wettergreen, J. P. Ostrowski, and S. Singh (2001). "Collection of environmental data from an airship platform". In: *Intelligent Systems and Advanced Manufacturing*. International Society for Optics and Photonics, pp. 76–83 (cit. on p. 6).

Kaplan, E. and C. Hegarty (2005). *Understanding GPS: principles and applications*. Artech house (cit. on pp. 16, 17).

Kennedy, S. (2015). "Made in China 2025". In: *Center for Strategic and International Studies* (cit. on p. 2).

Khalil, H. K. (1996). "Noninear systems". In: *Prentice-Hall, New Jersey* 2.5 (cit. on pp. 20–22, 73, 82).

Khoury, G. A. (2012). *Airship technology*. Vol. 10. Cambridge university press (cit. on pp. 5, 12).

Kleeman, L. (1992). "Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning". In: *Robotics and Automation (ICRA), 1992 IEEE International Conference on*. IEEE, pp. 2582–2587 (cit. on p. 17).

Ko, J., D. J. Klein, D. Fox, and D. Haehnel (2007). "Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp".

In: *Robotics and Automation (ICRA), 2007 IEEE International Conference on*, pp. 742–747 (cit. on pp. 7, 24, 196).

Kojima, A., K. Uchida, E. Shimemura, and S. Ishijima (1994). "Robust stabilization of a system with delays in control". In: *IEEE Transactions on Automatic Control* 39.8, pp. 1694–1698 (cit. on p. 83).

Kornienko, A. (Aug. 2006). "System identification approach for determining flight dynamical characteristics of an airship from flight data". PhD thesis. University of Stuttgart (cit. on p. 11).

Kulczycki, E., S. Joshi, R. Hess, and A. Elfes (2006). "Towards controller design for autonomous airships using SLC and LQR methods". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 6778 (cit. on p. 6).

Kusagaya, T., H. Kojima, and H. A. Fujii (2006). "Estimation of flyable regions for planetary airships". In: *Journal of Aircraft* 43.4, pp. 1177–1181 (cit. on p. 6).

Kwon, W. and A. Pearson (1980). "Feedback stabilization of linear systems with delayed control". In: *IEEE Transactions on Automatic Control* 25.2, pp. 266–269 (cit. on p. 83).

Lamb, H. (1932). *Hydrodynamics 6th ed.*, *738* (cit. on pp. 11, 41).

Lasi, H., P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann (2014). "Industry 4.0". In: *Business & Information Systems Engineering* 6.4, pp. 239–242 (cit. on p. 2).

Lee, C.-C. (1990). "Fuzzy logic in control systems: fuzzy logic controller. I". In: *IEEE Transactions on systems, man, and cybernetics* 20.2, pp. 404–418 (cit. on p. 24).

Lee, K. N. (1994). *Compass and gyroscope: integrating science and politics for the environment*. Island Press (cit. on p. 15).

Lee, S. and H. Bang (2007). "Three-dimensional ascent trajectory optimization for stratospheric airship platforms in the jet stream". In: *Journal of Guidance, Control, and Dynamics* 30.5, pp. 1341–1351 (cit. on p. 6).

Lee, Y.-G., D.-M. Kim, and C.-H. Yeom (2006). "Development of Korean high altitude platform systems". In: *International Journal of Wireless Information Networks* 13.1, pp. 31–42 (cit. on p. 6).

Lenz, J. and S Edelstein (2006). "Magnetic sensors and their applications". In: *IEEE Sensors journal* 6.3, pp. 631–649 (cit. on p. 13).

Levant, A. (2003). "Higher-order sliding modes, differentiation and output-feedback control". In: *International Journal of Control* 76.9-10, pp. 924–941 (cit. on pp. 75, 76).

Li, Y., M. Nahon, and I. Sharf (2008). "Dynamics modeling and simulation of flexible airships". PhD thesis. McGill University (cit. on p. 11).

— (2011). "Airship dynamics modeling: A literature review". In: *Progress in Aerospace Sciences* 47.3, pp. 217–239 (cit. on pp. 5, 10–12, 44).

Liberzon, D. and A. S. Morse (1999). "Basic problems in stability and design of switched systems". In: *IEEE Control systems* 19.5, pp. 59–70 (cit. on p. 84).

LinearTechnology (Jan. 2019). *LT1963 Series Datasheet*. https://www.analog.com/media/en/technical-documentation/data-sheets/1963fc.pdf (cit. on p. 182).

Löfberg, J. (2004). "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". In: *Proceedings of the CACSD Conference*. Taipei, Taiwan (cit. on pp. 82, 85).

Lowrie, W. (2007). *Fundamentals of geophysics*. Cambridge university press (cit. on p. 13).

Luo, R. C. and L.-Y. Chung (2002). "Stabilization for linear uncertain system with time latency". In: *IEEE Transactions on Industrial Electronics* 49.4, pp. 905–910 (cit. on p. 83).

Lutz, T., P. Funk, A. Jakobi, and S. Wagner (2002). "Summary of aerodynamic studies on the lotte airship". In: *4th International airship Convention and Exhibition*. July, pp. 1–12 (cit. on p. 11).

Ma, Y., S. Soatto, J. Kosecka, and S. S. Sastry (2012). *An invitation to 3-d vision: from images to geometric models*. Vol. 26. Springer Science & Business Media (cit. on p. 18).

MaxBotix (Dec. 2018). *MB1010 LV-MaxSonar-EZ1*. https://www.maxbotix.com/Ultrasonic_Sensors/MB1010.htm (cit. on p. 56).

Meusnier, J. B. M. (1784). "Mémoire sur l'équilibre des machines aérostatiques, sur les différens moyens de les faire monter & descendre, & spécialement sur celui d'exécuter ces manoeuvres, sans jeter de lest". In: *Journal de Physique* (cit. on p. 5).

Moon, Y. S., P. Park, and W. H. Kwon (2001). "Robust stabilization of uncertain input-delayed systems using reduction method". In: *Automatica* 37.2, pp. 307–312 (cit. on pp. 82, 83).

Moulay, E. and W. Perruquetti (2006). "Finite time stability and stabilization of a class of continuous systems". In: *Journal of Mathematical analysis and applications* 323.2, pp. 1430–1443 (cit. on p. 21).

Moutinho, A. and J. R. Azinheira (2005). "Stability and robustness analysis of the AURORA airship control system using dynamic inversion". In: *Robotics and Automation (ICRA), 2005 IEEE International Conference on*. IEEE, pp. 2265–2270 (cit. on p. 21).

Müller, J., A. Rottmann, L. M. Reindl, and W. Burgard (2009). "A probabilistic sonar sensor model for robust localization of a small-size blimp in indoor environments using a particle filter". In: *Robotics and Automation (ICRA), 2009 IEEE International Conference on*. IEEE, pp. 3589–3594 (cit. on pp. 8, 16, 18).

Müller, J., N. Kohler, and W. Burgard (2011). "Autonomous miniature blimp navigation with online motion planning and re-planning". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, pp. 4941–4946 (cit. on pp. 8, 19, 22).

Munk, M. M. (1924). *The aerodynamic forces on airship hulls*. Tech. rep. National Advisory Committee for Aeronautics (cit. on pp. 11, 12).

— (1934). "Fluid mechanics, part II". In: *Aerodynamic Theory*. Springer, pp. 224–304 (cit. on p. 41).

— (1936). "Aerodynamics of airships". In: *Aerodynamic Theory*. Springer, pp. 32–48 (cit. on pp. 41, 44).

Nakadate, M. (2005). "Development and flight test of SPF-2 low altitude stationary flight test vehicle". In: *AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*, p. 7408 (cit. on p. 12).

NaturalPoint (Oct. 2018). *OptiTrack for robotics*. https://optitrack.com/motion-capture-robotics/ (cit. on pp. 56, 57, 138).

Newman, J. N. (2018). *Marine hydrodynamics*. MIT press (cit. on p. 11).

Newton, I. (1833). *Philosophiae naturalis principia mathematica*. Vol. 1. G. Brookman (cit. on p. 171).

Nexperia (Jan. 2019). *74HC04 Datasheet*. https://assets.nexperia.com/documents/data-sheet/74HC_HCT04.pdf (cit. on p. 185).

Niculescu, S.-I. (2001). *Delay effects on stability: a robust control approach*. Vol. 269. Springer Science & Business Media (cit. on p. 83).

NON-A (Oct. 2018). *Presentation of NON-A research team*. https://www.inria.fr/equipes/non-a (cit. on p. 29).

ONSemiconductor (Dec. 2018). *LB1930MC Application Note*. https://www.onsemi.cn/PowerSolutions/document/ANDLB1930MC-D.PDF (cit. on p. 130).

Paiva, E. C. de, J. R. Azinheira, J. G. Ramos Jr, A. Moutinho, and S. S. Bueno (2006). "Project AURORA: Infrastructure and flight control experiments for a robotic airship". In: *Journal of Field Robotics* 23.3-4, pp. 201–222 (cit. on p. 21).

Pan, W., Y. Dou, G. Wang, M. Wu, R. Ren, and X Xu (2015). "Development and Test of Blimp-Based Compact LIDAR Powewr-Line Inspection System". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40.3, p. 155 (cit. on p. 6).

Perruquetti, W. and T. Floquet (2007). "Homogeneous finite time observer for nonlinear systems with linearizable error dynamics". In: *Decision and Control (CDC), 2007. Proceedings of the 46th IEEE Conference on*, pp. 390–395 (cit. on p. 78).

Perruquetti, W., T. Floquet, and E. Moulay (2008). "Finite-time observers: application to secure communication". In: *IEEE Transactions on Automatic Control* 53.1, pp. 356–360 (cit. on pp. 76, 78, 80).

Post, E. J. (1967). "Sagnac effect". In: *Reviews of Modern Physics* 39.2, p. 475 (cit. on p. 15).

Psaltis, D., A. Sideris, and A. A. Yamamura (1988). "A multilayered neural network controller". In: *IEEE control systems magazine* 8.2, pp. 17–21 (cit. on p. 23).

Rao, J., Z. Gong, J. Luo, and S. Xie (2005). "A flight control and navigation system of a small size unmanned airship". In: *Mechatronics and Automation, 2005 IEEE International Conference*. Vol. 3. IEEE, pp. 1491–1496 (cit. on p. 24).

Rao, J., Z. Gong, J. Luo, Z. Jiang, S. Xie, and W. Liu (2007). "Robotic airship mission path-following control based on ANN and human operator's skill". In: *Transactions of the Institute of Measurement and Control* 29.1, pp. 5–15 (cit. on p. 23).

Reyhanoglu, M. (1997). "Exponential stabilization of an underactuated autonomous surface vessel". In: *Automatica* 33.12. cited By 153, pp. 2249–2254 (cit. on p. 51).

Rioux, M. (1984). "Laser range finder based on synchronized scanners". In: *Applied optics* 23.21, pp. 3837–3844 (cit. on p. 17).

Roh, Y.-H. and J.-H. Oh (1999). "Robust stabilization of uncertain input-delay systems by sliding mode control with delay compensation". In: *Automatica* 35.11, pp. 1861–1865 (cit. on p. 83).

Rottmann, A., C. Plagemann, P. Hilgers, and W. Burgard (2007a). "Autonomous blimp control using model-free reinforcement learning in a continuous state and action space". In: *Intelligent Robots and Systems (IROS), 2007 IEEE/RSJ International Conference on*. IEEE, pp. 1895–1900 (cit. on pp. 8, 18, 24, 196).

Rottmann, A., T. Zitterell, W. Burgard, L. Reindl, and C. Scholl (2007b). "Towards an Experimental Autonomous Blimp Platform". In: *Proc. of the Europ. Conf. on Mobile Robots (ECMR)* (cit. on pp. 8, 16, 24).

Sagatun, S. I. and T. I. Fossen (1991). "Lagrangian formulation of underwater vehicles' dynamics". In: *Systems, Man, and Cybernetics, 1991.'Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on*. IEEE, pp. 1029–1034 (cit. on pp. 36, 43).

Saiki, H., T. Fukao, T. Urakubo, and T. Kohno (2010). "Hovering control of outdoor blimp robots based on path following". In: *Control Applications (CCA), 2010 IEEE International Conference on*. IEEE, pp. 2124–2129 (cit. on pp. 7, 16, 19).

Schmidt, D. K. (2007). "Modeling and near-space stationkeeping control of a large high-altitude airship". In: *Journal of Guidance, Control, and Dynamics* 30.2, pp. 540–547 (cit. on pp. 6, 11).

Seber, G. A. F. and C. J. Wild (2003). *Nonlinear Regression*. Ed. by Hoboken. Wiley-Interscience (cit. on p. 61).

Siegwart, R., I. R. Nourbakhsh, D. Scaramuzza, and R. C. Arkin (2011). *Introduction to autonomous mobile robots*. MIT press (cit. on pp. 3, 4, 9, 10, 13, 14, 16–18).

Smith, O. J. (1959). "A controller to overcome dead time". In: *iSA journal* 6.2, pp. 28–33 (cit. on p. 83).

Sontag, E. D. (1989). "A" universal" construction of Artstein's theorem on nonlinear stabilization." In: *Systems & control letters* 13.2, pp. 117–123 (cit. on p. 21).

Stephenson, C. (2012). *Zeppelins: German Airships 1900–40*. Bloomsbury Publishing (cit. on p. 5).

Stevens, B. L., F. L. Lewis, and E. N. Johnson (2015). *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons (cit. on pp. 4, 20, 22).

STMicroelectronics (Dec. 2018a). *L298 Dual Full Bridge Driver*. https://www.st.com/en/motor-drivers/l298.html (cit. on pp. 134, 184).

— (Dec. 2018b). *STM32F103VE Product*. https://www.st.com/en/microcontrollers/stm32f103ve.html (cit. on pp. 131, 183).

— (Jan. 2019). *ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32*. https://www.st.com/en/development-tools/st-link-v2.html (cit. on p. 183).

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media (cit. on p. 18).

Takaya, T., H. Kawamura, Y. Minagawa, M. Yamamoto, and A. Ohuchi (2006). "PID landing orbit motion controller for an indoor blimp robot". In: *Artificial Life and Robotics* 10.2, pp. 177–184 (cit. on p. 23).

TexasInstruments (Dec. 2018). *L293x Quadruple Half-H Drivers*. http://www.ti.com/lit/ds/symlink/l293.pdf (cit. on p. 130).

— (Jan. 2019a). *LM2596 SIMPLE SWITCHER Power Converter 150-kHz Manual*. http://www.ti.com/lit/ds/symlink/lm2596.pdf (cit. on p. 182).

— (Jan. 2019b). *MAX3232 3-V to 5.5-V Multichannel RS-232 Line Driver/Receiver Manual*. http://www.ti.com/lit/ds/symlink/max3232.pdf (cit. on p. 186).

Tischler, M., R. Ringland, and H. Jex (1983). "Heavy-lift airship dynamics". In: *Journal of Aircraft* 20.5, pp. 425–433 (cit. on p. 12).

Toshiba (Jan. 2019). *TLP521 Photocoupler*. https://toshiba.semicon-storage.com/us/product/opto/photocoupler/detail.TLP521-1.html (cit. on p. 191).

Trucco, E. and A. Verri (1998). *Introductory techniques for 3-D computer vision*. Vol. 201. Prentice Hall Englewood Cliffs (cit. on p. 18).

Wang, Y., G. Zheng, D. Efimov, and W. Perruquetti (2018). "Differentiator application in altitude control for an indoor blimp robot". In: *International Journal of Control* 91.9, pp. 2121–2130 (cit. on p. 106).

Wilson, J. N. and G. X. Ritter (2000). *Handbook of computer vision algorithms in image algebra*. CRC press (cit. on p. 19).

Wilson, J. (2004). "A new era for airships". In: *Aerospace America* 42.5, pp. 27–31 (cit. on p. 5).

Wong, K., L. Zhiyung, and J. DeLaurier (1985). "An application of source-panel and vortex methods for aerodynamic solutions of airship configurations". In: *6th Lighter-Than-Air Systems Conference*, p. 874 (cit. on p. 11).

Wyeth, G. and I. Barron (1998). "An autonomous blimp". In: *Field and service robotics*. Springer, pp. 464–470 (cit. on pp. 7, 18, 22).

Yamasaki, T. and N. Goto (2003). "Identification of blimp dynamics via flight tests". In: *Transactions of the Japan Society for Aeronautical and Space Sciences* 46.153, pp. 195–205 (cit. on pp. 12, 29).

Yang, Y. (2016). "Progress, contribution and challenges of Compass/Beidou satellite navigation system". In: *Acta Geodaetica et Cartographica Sinica* 39.1, pp. 1–6 (cit. on p. 16).

Yue, D. and Q.-L. Han (2005). "Delayed feedback control of uncertain systems with time-varying input delay". In: *Automatica* 41.2, pp. 233–240 (cit. on p. 83).

Zhang, H. and J. P. Ostrowski (1999). "Visual servoing with dynamics: Control of an unmanned blimp". In: *Robotics and Automation (ICRA), 1999 IEEE International Conference on*. Vol. 1. IEEE, pp. 618–623 (cit. on pp. 7, 19, 20, 196).

Zheng, Z. and L. Sun (2018). "Adaptive sliding mode trajectory tracking control of robotic airships with parametric uncertainty and wind disturbance". In: *Journal of the Franklin Institute* 355.1, pp. 106–122 (cit. on p. 20).

Zheng, Z., W. Huo, and Z. Wu (2013). "Autonomous airship path following control: Theory and experiments". In: *Control Engineering Practice* 21.6, pp. 769–788 (cit. on p. 20).

Zufferey, J.-C., A. Guanella, A. Beyeler, and D. Floreano (2006). "Flying over the reality gap: From simulated to real indoor airships". In: *Autonomous Robots* 21.3, pp. 243–254 (cit. on pp. 7, 12, 19, 23, 29, 36, 39, 41, 43, 44, 64, 196).

Zwaan, S. van der, A. Bernardino, and J. Santos-Victor (2000). "Vision based station keeping and docking for an aerial blimp". In: *Intelligent Robots and Systems (IROS), 2000 IEEE/RSJ International Conference on*. Vol. 1. IEEE, pp. 614–619 (cit. on pp. 7, 19, 22, 29).

Appendix A

Blimp Dynamic Model

The blimp in this work is considered as a rigid body, thus its dynamics can be analyzed by commonly used methods. Moreover, as the blimp motion has similar characteristics compared to the underwater vehicles, the aerodynamic forces and moments also have to be considered, especially the added mass effects. In this appendix, the same analyzing procedure of [Fossen, 1994] is followed to derive the 6-DOF motion model of the blimp as shown in equation (2.3.2).

A.1 Rigid-Body dynamics

Consider a rigid body as shown in Figure A.1, the body-fixed frame is $\mathcal{F}_b$ locates its origin at $\boldsymbol{O}$, the inertial frame is $\mathcal{F}_i$. The rigid body center of gravity is at CG. Usually, the researchers use the Newton-Euler formulation or the Lagrangian formulation to analyze the rigid body dynamics.

The Newton-Euler formulation is based on Newton's Second Law which relates the mass of a rigid body m, acceleration $\dot{\boldsymbol{v}}_c$ and the force $\boldsymbol{f}_c$ in the inertial frame [Newton, 1833]:

$$m\dot{\boldsymbol{v}}_c = \boldsymbol{f}_c \tag{A.1}$$

Later, Euler proposed to express Newton's Second Law in terms of conservation of linear and angular momentum $\boldsymbol{p}_c$ and $\boldsymbol{h}_c$ in the inertial frame, the results are known as Euler's First and Second Axioms, respectively [Euler, 1773]:

$$\dot{\boldsymbol{p}}_c = \boldsymbol{f}_c \quad \boldsymbol{p}_c = m\boldsymbol{v}_c \tag{A.2}$$

$$\dot{\boldsymbol{h}}_c = \boldsymbol{m}_c \quad \boldsymbol{h}_c = \boldsymbol{I}_c\boldsymbol{\omega} \tag{A.3}$$

where $\boldsymbol{f}_c$ and $\boldsymbol{m}_c$ are the forces and moments with respect to the rigid body CG, $\boldsymbol{\omega}$ is the angular velocity vector, and $\boldsymbol{I}_c$ is the inertia tensor about the body CG.

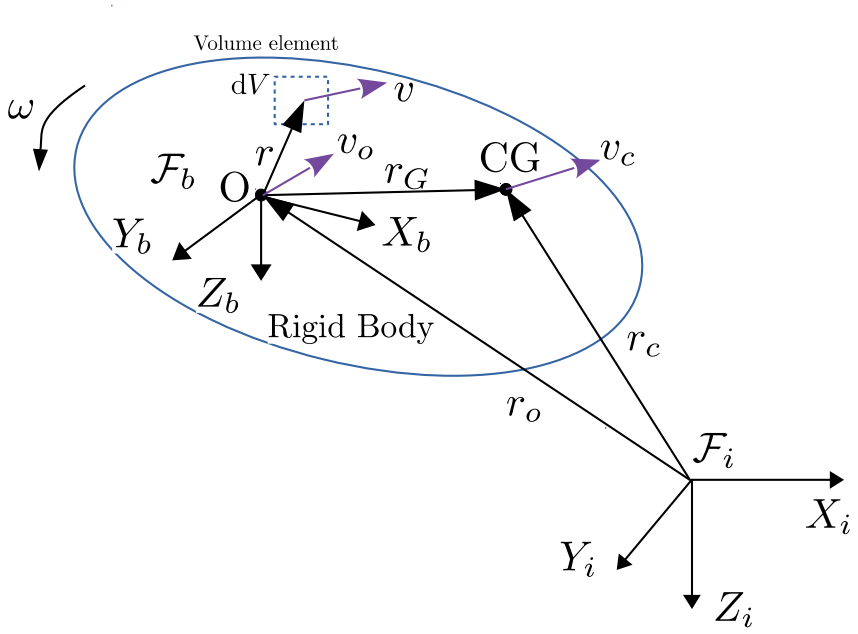

Figure A.1 – Inertial frame $\boldsymbol{X_i Y_i Z_i}$ and body-fixed frame $\boldsymbol{X_b Y_b Z_b}$ for the rigid body

For a rigid body as shown in Figure A.1, its inertia tensor $\boldsymbol{I_o}$ referred to the body-fixed frame origin O, is defined as

$$\boldsymbol{I_o} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{A.4}$$

which is a symmetric matrix, and the I_x, I_y and I_z are the moments of inertia about the $\boldsymbol{X}_b$, $\boldsymbol{Y}_b$, $\boldsymbol{Z}_b$ axis respectively. The terms of $\boldsymbol{I_o}$ are defined as

$$\begin{aligned} I_x &= \int_V (y^2 + z^2)\rho \mathrm{d}V \quad I_{xy} = I_{yx} = \int_V xy\rho \mathrm{d}V \\ I_y &= \int_V (x^2 + z^2)\rho \mathrm{d}V \quad I_{xz} = I_{zx} = \int_V xz\rho \mathrm{d}V \\ I_z &= \int_V (x^2 + y^2)\rho \mathrm{d}V \quad I_{yz} = I_{zy} = \int_V yz\rho \mathrm{d}V \end{aligned}$$

where ρ is the mass density of rigid body, $\mathrm{d}V$ is a volume element as shown in Figure A.1. Therefore, the inertia tensor $\boldsymbol{I_o}$ can be represented in vector form as [Fossen, 1994]

$$\boldsymbol{I_o \omega} = \int_V \boldsymbol{r} \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r}) \rho \mathrm{d}V \tag{A.5}$$

The mass of the rigid body is defined as

$$m = \int_V \rho \mathrm{d}V$$

Assume the mass is time-invariant, for a rigid body, the vector from the origin of $\mathcal{F}_b$ to the CG can be obtained by

$$\boldsymbol{r_G} = \frac{1}{m} \int_V \boldsymbol{r} \rho \mathrm{d}V \tag{A.6}$$

Notice that it is desirable to express the blimp equation of motion in the body-fixed frame, because the aerodynamic and kinematic forces and moments can be easily described in $\mathcal{F}_b$. To derive the equation of motion in the body-fixed frame, one important formula is needed:

For a vector $\boldsymbol{u}$, the change of basis equation

$$\boldsymbol{u}^i = \boldsymbol{R}_i^b \boldsymbol{u}^b$$

where $\boldsymbol{R}_i^b$ is the rotation matrix from $\mathcal{F}_i$ to $\mathcal{F}_b$, taking time derivatives on both sides:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}^i = \boldsymbol{R}_i^b (\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}^b) + \dot{\boldsymbol{R}}_i^b \boldsymbol{u}^b$$

Multiplying both sides with $(\boldsymbol{R}_i^b)^{\mathrm{T}}$ and with the equation (2.5), there is

$$(\boldsymbol{R}_i^b)^{\mathrm{T}} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}^i = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}^b + S(\boldsymbol{\omega}^b)\boldsymbol{u}^b$$

rewrite the equation in short form, we obtain the relation between the time-derivative of a vector seen in inertial frame $\mathcal{F}_i$ and seen in moving frame $\mathcal{F}_b$

$$(\dot{\boldsymbol{u}})^i = (\dot{\boldsymbol{u}})^b + \boldsymbol{\omega}^b \wedge \boldsymbol{u}^b \tag{A.7}$$

where the left-hand side is the time-derivative of $\boldsymbol{u}$ as seen in the inertial frame $\mathcal{F}_i$ but expressed in body-fixed frame $\mathcal{F}_b$, the first term on the right-hand side is the time-derivative of $\boldsymbol{u}$ with respect to moving body-fixed frame $\mathcal{F}_b$ expressed also in $\mathcal{F}_b$, $\boldsymbol{\omega}^b$ is the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$, expressed in $\mathcal{F}_b$.

In the remaining of this Appendix, for the simplicity of expression, the superscripts for vectors are omitted except for the time-derivatives of the vectors.

A.1.1 Translational motion

From Figure A.1, it is clear that:

$$\boldsymbol{r_c} = \boldsymbol{r_o} + \boldsymbol{r_G}$$

Therefore the velocity of the CG is

$$\boldsymbol{v_c} = (\dot{\boldsymbol{r}}_c)^i = (\dot{\boldsymbol{r}}_o)^i + (\dot{\boldsymbol{r}}_G)^i$$

It is reasonable to denote $\boldsymbol{v_o} = (\dot{\boldsymbol{r}}_o)^i$, the velocity of point $\boldsymbol{O}$ seen in inertial frame $\mathcal{F}_i$ and expressed in $\mathcal{F}_b$. In addition, for a rigid body

$$(\dot{\boldsymbol{r}}_G)^b = 0$$

hence from (A.7), there is

$$(\dot{\boldsymbol{r}}_G)^i = (\dot{\boldsymbol{r}}_G)^b + \boldsymbol{\omega} \wedge \boldsymbol{r_G} = \boldsymbol{\omega} \wedge \boldsymbol{r_G}$$

The velocity of the CG becomes

$$\boldsymbol{v_c} = \boldsymbol{v_o} + \boldsymbol{\omega} \wedge \boldsymbol{r_G} \tag{A.8}$$

Then, taking time-derivative on both sides, the acceleration of the CG is obtained

$$(\dot{\boldsymbol{v}}_c)^{\boldsymbol{i}} = (\dot{\boldsymbol{v}}_o)^{\boldsymbol{i}} + (\dot{\boldsymbol{\omega}})^{\boldsymbol{i}} \wedge \boldsymbol{r_G} + \boldsymbol{\omega} \wedge (\dot{\boldsymbol{r}}_G)^{\boldsymbol{i}}$$

Express $(\dot{\boldsymbol{v}}_o)^i$, $(\dot{\boldsymbol{\omega}})^i$ and $(\dot{\boldsymbol{r}}_G)^i$ in the body-fixed frame, there is

$$(\dot{\boldsymbol{v}}_c)^{\boldsymbol{i}} = (\dot{\boldsymbol{v}}_o)^{\boldsymbol{b}} + \boldsymbol{\omega} \wedge \boldsymbol{v_o} + (\dot{\boldsymbol{\omega}})^{\boldsymbol{b}} \wedge \boldsymbol{r_G} + \boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r_G})$$

Notice that there is $(\dot{\boldsymbol{\omega}})^{\boldsymbol{i}} = (\dot{\boldsymbol{\omega}})^{\boldsymbol{b}}$, which means the angular acceleration is equal in the inertial frame and body-fixed frame. Substituting this equation into the Euler's First Axiom (A.2), we finally get

$$m\Big((\dot{\boldsymbol{v}}_o)^{\boldsymbol{b}} + \boldsymbol{\omega} \wedge \boldsymbol{v_o} + (\dot{\boldsymbol{\omega}})^{\boldsymbol{b}} \wedge \boldsymbol{r_G} + \boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r_G})\Big) = \boldsymbol{f}_o \tag{A.9}$$

Note that in the equation all the vectors are expressed in the body-fixed frame $\mathcal{F}_b$.

A.1.2 Rotational motion

The angular momentum about $\boldsymbol{O}$ is defined as

$$\boldsymbol{h_o} = \int_V \boldsymbol{r} \wedge \boldsymbol{v} \rho \mathrm{d}V \tag{A.10}$$

Taking the time-derivative of the equation yields

$$(\dot{\boldsymbol{h}}_o)^{\boldsymbol{i}} = \int_V \boldsymbol{r} \wedge (\dot{\boldsymbol{v}})^{\boldsymbol{i}} \rho \mathrm{d}V + \int_V (\dot{\boldsymbol{r}})^{\boldsymbol{i}} \wedge \boldsymbol{v} \rho \mathrm{d}V \tag{A.11}$$

The first term on right-hand side is the moment vector

$$\boldsymbol{m}_o = \int_V \boldsymbol{r} \wedge (\dot{\boldsymbol{v}})^{\boldsymbol{i}} \rho \mathrm{d}V \tag{A.12}$$

From Figure A.1, it can be seen that

$$\boldsymbol{v} = (\dot{\boldsymbol{r}}_{\boldsymbol{o}})^{\boldsymbol{i}} + (\dot{\boldsymbol{r}})^{\boldsymbol{i}} \tag{A.13}$$

Substituting (A.13) and (A.12) into (A.11) yields

$$(\dot{\boldsymbol{h}}_o)^{\boldsymbol{i}} = \boldsymbol{m}_o - \boldsymbol{v}_o \wedge \int_V (\dot{\boldsymbol{r}})^{\boldsymbol{i}} \rho \mathrm{d}V \tag{A.14}$$

The expression can be rewritten by differentiating (A.6) with respect to time

$$m(\dot{\boldsymbol{r}}_G)^{\boldsymbol{i}} = \int_V (\dot{\boldsymbol{r}})^{\boldsymbol{i}} \rho \mathrm{d}V$$

Since $(\dot{\boldsymbol{r}}_G)^{\boldsymbol{i}} = \boldsymbol{\omega} \wedge \boldsymbol{r_G}$, there is

$$\int_V (\dot{\boldsymbol{r}})^{\boldsymbol{i}} \rho \mathrm{d}V = m(\boldsymbol{\omega} \wedge \boldsymbol{r_G})$$

Then (A.14) is rewritten as

$$(\dot{\boldsymbol{h}}_o)^{\boldsymbol{i}} = \boldsymbol{m}_o - m\boldsymbol{v}_o \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r_G}) \tag{A.15}$$

The next step is to rewrite the angular momentum (A.10) in another way, similar to (A.8), for any volume element $\mathrm{d}V$, there is $\boldsymbol{v} = \boldsymbol{v_o} + \boldsymbol{\omega} \wedge \boldsymbol{r}$, then substituting

$\boldsymbol{v}$ in (A.10) there is

$$\boldsymbol{h}_o = \int_V \boldsymbol{r} \wedge \boldsymbol{v} \rho \mathrm{d}V = \int_V \boldsymbol{r} \wedge \boldsymbol{v}_o \rho \mathrm{d}V + \int_V \boldsymbol{r} \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r}) \rho \mathrm{d}V \tag{A.16}$$

The first term on the right-hand side can be rewritten by using the definition of the CG (A.6), that is

$$\int_V \boldsymbol{r} \wedge \boldsymbol{v}_o \rho \mathrm{d}V = \left(\int_V \boldsymbol{r} \rho \mathrm{d}V \right) \wedge \boldsymbol{v}_o = m \boldsymbol{r}_G \wedge \boldsymbol{v}_o$$

The second term can be replaced by the definition of the inertia tensor (A.5), hence (A.16) becomes

$$\boldsymbol{h}_o = \boldsymbol{I}_o \boldsymbol{\omega} + m \boldsymbol{r}_G \wedge \boldsymbol{v}_o$$

Taking the derivative on both sides of this equation and get

$$(\dot{\boldsymbol{h}}_o)^i = \boldsymbol{I}_o (\dot{\boldsymbol{\omega}})^b + \boldsymbol{\omega} \wedge (\boldsymbol{I}_o \boldsymbol{\omega}) + m (\dot{\boldsymbol{r}}_G)^i \wedge \boldsymbol{v}_o + m \boldsymbol{r}_G \wedge (\dot{\boldsymbol{v}}_o)^i$$

Replacing $(\dot{\boldsymbol{r}}_G)^i$, $(\dot{\boldsymbol{v}}_o)^i$ with their expression in body-fixed frame, get

$$(\dot{\boldsymbol{h}}_o)^i = \boldsymbol{I}_o (\dot{\boldsymbol{\omega}})^b + \boldsymbol{\omega} \wedge (\boldsymbol{I}_o \boldsymbol{\omega}) + m (\boldsymbol{\omega} \wedge \boldsymbol{r}_G) \wedge \boldsymbol{v}_o + m \boldsymbol{r}_G \wedge ((\dot{\boldsymbol{v}}_o)^b + \boldsymbol{\omega} \wedge \boldsymbol{v}_o) \tag{A.17}$$

The equations (A.15) and (A.17) are equal, notice that

$$(\boldsymbol{\omega} \wedge \boldsymbol{r}_G) \wedge \boldsymbol{v}_o = -\boldsymbol{v}_o \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r}_G)$$

We finally get

$$\boldsymbol{I}_o (\dot{\boldsymbol{\omega}})^b + \boldsymbol{\omega} \wedge (\boldsymbol{I}_o \boldsymbol{\omega}) + m \boldsymbol{r}_G \wedge ((\dot{\boldsymbol{v}}_o)^b + \boldsymbol{\omega} \wedge \boldsymbol{v}_o) = \boldsymbol{m}_o \tag{A.18}$$

The rotational equations of motion are often referred to as the Euler equations. Note that all the vectors in the equation are expressed in the body-fixed frame $\mathcal{F}_b$.

A.1.3 6-DOF rigid-body dynamic model

From the equations (A.9) and (A.18), the 6-DOF rigid-body dynamic model can be obtained. The following notations are used

$$
\begin{aligned}
&\boldsymbol{f_o} = \boldsymbol{\tau_1} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^{\mathrm{T}} && \text{external forces} \\
&\boldsymbol{m_o} = \boldsymbol{\tau_2} = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^{\mathrm{T}} && \text{moment of external forces about } O \\
&\boldsymbol{v_o} = \boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^{\mathrm{T}} && \text{linear velocity of rigid body expressed in } \mathcal{F}_b \\
&\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathrm{T}} && \text{angular velocity of rigid body expressed in } \mathcal{F}_b \\
&\boldsymbol{r_G} = \begin{bmatrix} x_G & y_G & z_G \end{bmatrix}^{\mathrm{T}} && \text{coordinate of CG in } \mathcal{F}_b
\end{aligned}
$$

and $\boldsymbol{I_o}$ is defined by (A.4). Then (A.9) and (A.18) become:

$$
\begin{aligned}
m[\dot{v}_x - v_y\omega_z + v_z\omega_y - x_G(\omega_y^2 + \omega_z^2) + y_G(\omega_x\omega_y - \dot{\omega}_z) + z_G(\omega_x\omega_z + \dot{\omega}_y)] &= f_x \\
m[\dot{v}_y - v_z\omega_x + v_x\omega_z - y_G(\omega_z^2 + \omega_x^2) + z_G(\omega_y\omega_z - \dot{\omega}_x) + x_G(\omega_y\omega_x + \dot{\omega}_z)] &= f_y \\
m[\dot{v}_z - v_x\omega_y + v_y\omega_x - z_G(\omega_x^2 + \omega_y^2) + x_G(\omega_z\omega_x - \dot{\omega}_y) + y_G(\omega_z\omega_y + \dot{\omega}_x)] &= f_z \\
I_x\dot{\omega}_x + (I_z - I_y)\omega_y\omega_z - (\dot{\omega}_z + \omega_x\omega_y)I_{xz} + (\omega_z^2 - \omega_y^2)I_{yz} + (\omega_x\omega_z - \dot{\omega}_y)I_{xy} & \\
+m[y_G(\dot{v}_z - v_x\omega_y + v_y\omega_x) - z_G(\dot{v}_y - v_z\omega_x + v_x\omega_z)] &= \tau_x \\
I_y\dot{\omega}_y + (I_x - I_z)\omega_z\omega_x - (\dot{\omega}_x + \omega_y\omega_z)I_{xy} + (\omega_x^2 - \omega_z^2)I_{zx} + (\omega_y\omega_x - \dot{\omega}_z)I_{yz} & \\
+m[z_G(\dot{v}_x - v_y\omega_z + v_z\omega_y) - x_G(\dot{v}_z - v_x\omega_y + v_y\omega_x)] &= \tau_y \\
I_z\dot{\omega}_z + (I_y - I_x)\omega_x\omega_y - (\dot{\omega}_y + \omega_z\omega_x)I_{yz} + (\omega_y^2 - \omega_x^2)I_{xy} + (\omega_z\omega_y - \dot{\omega}_x)I_{zx} & \\
+m[x_G(\dot{v}_y - v_z\omega_x + v_x\omega_z) - y_G(\dot{v}_x - v_y\omega_z + v_z\omega_y)] &= \tau_z
\end{aligned}
\tag{A.19}
$$

It can be written in a more compact form as

$$
\boldsymbol{M}_{\mathrm{RB}}\dot{\boldsymbol{\xi}} + \boldsymbol{C}_{\mathrm{RB}}(\boldsymbol{\xi})\boldsymbol{\xi} = \boldsymbol{\tau}_{\mathrm{RB}} \tag{A.20}
$$

where

$$
\begin{aligned}
\boldsymbol{\xi} &= \begin{bmatrix} (\boldsymbol{v})^{\mathrm{T}} & (\boldsymbol{\omega})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathrm{T}} \\
\boldsymbol{\tau}_{\mathrm{RB}} &= \begin{bmatrix} (\boldsymbol{\tau_1})^{\mathrm{T}} & (\boldsymbol{\tau_2})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} f_x & f_y & f_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^{\mathrm{T}}
\end{aligned}
$$

Note that the vectors $\boldsymbol{\xi}$ and $\boldsymbol{\tau}_{\mathrm{RB}}$ are both expressed in the body-fixed frame.

The inertia matrix of rigid-body $\boldsymbol{M}_{\mathrm{RB}}$ is written as

$$
\boldsymbol{M}_{\mathrm{RB}} = \begin{bmatrix} m\mathrm{I}_{3\times 3} & -mS(\boldsymbol{r_G}) \\ mS(\boldsymbol{r_G}) & \boldsymbol{I_o} \end{bmatrix}
$$

where $\mathrm{I}_{3\times 3}$ is the identity matrix of dimension 3×3, $\boldsymbol{I_o}$ is the inertia tensor with respect to O, and $S(\boldsymbol{r_G})$ is the skew-symmetric matrix of $\boldsymbol{r_G}$ as defined in

Definition 2.2.

The matrix $\boldsymbol{C}_{\mathrm{RB}}$ consists of the Coriolis term $\boldsymbol{\omega} \wedge \boldsymbol{v}$ and the centripetal vector term $\boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r_G})$, a skew-symmetric representation of $\boldsymbol{C}_{\mathrm{RB}}$ can be derived by using Kirchhoff's equations, we will not give the proofs, readers can read the book [Fossen, 1994] for more details. In general the matrix can be defined by

$$\boldsymbol{C}_{\mathrm{RB}}(\xi) = \begin{bmatrix} \mathbf{0}_{3\times 3} & -S(\boldsymbol{M}_{11}\boldsymbol{v} + \boldsymbol{M}_{12}\boldsymbol{\omega}) \\ -S(\boldsymbol{M}_{11}\boldsymbol{v} + \boldsymbol{M}_{12}\boldsymbol{\omega}) & -S(\boldsymbol{M}_{21}\boldsymbol{v} + \boldsymbol{M}_{22}\boldsymbol{\omega}) \end{bmatrix}$$

where $\boldsymbol{M}_{ij}(i, j = 1, 2)$ are the four 3×3 sub-matrices of the inertia matrix $\boldsymbol{M}_{\mathrm{RB}}$.

A.2 Aerodynamic forces and moments

The blimp filled with helium has similar density of the ambient air, thus when it moves in the air, it suffers similar aerodynamic (hydrodynamic) forces and moments compared to underwater vehicles. Therefore they can be studied in the same way [Fossen, 1994].

Since the research of hydrodynamic forces and moments on a rigid body is beyond the scope of the thesis, only some conclusions are given regarding this topic. The hydrodynamic forces and moments $\boldsymbol{\tau_H}$ can be written as

$$\boldsymbol{\tau_H} = -\boldsymbol{M}_{\mathrm{Added}}\dot{\xi} - \boldsymbol{C}_{\mathrm{Added}}(\xi)\xi - \boldsymbol{D}(\xi)\xi - \boldsymbol{g}(\eta) \tag{A.21}$$

where the first two terms on the right-hand side $-\boldsymbol{M}_{\mathrm{Added}}\dot{\xi} - \boldsymbol{C}_{\mathrm{Added}}(\xi)\xi$ are related to the added mass due to the inertia of the surrounding fluid, $\boldsymbol{D}(\xi)$ is the total hydrodynamic damping matrix, including potential damping effects, skin friction, wave drift damping and damping due to vortex shedding etc., $\boldsymbol{g}(\eta)$ is the restoring forces vector due to Archimedes (weight and buoyancy).

The right-hand side of the equations (A.19) and (A.20) represent the external forces and moments acting on the rigid body, which include the hydrodynamic forces and moments term $\boldsymbol{\tau_H}$, environmental forces $\boldsymbol{\tau_E}$ (can be considered as external disturbances) and propulsion forces vector $\boldsymbol{\tau}$ generated by actuators (for the blimp robot, it is the forces and moments of the motors with propellers):

$$\boldsymbol{\tau}_{\mathrm{RB}} = \boldsymbol{\tau_H} + \boldsymbol{\tau_E} + \boldsymbol{\tau}$$

Then, from equations (A.20) and (A.21), the 6-DOF dynamic model of the blimp robot is

$$\boldsymbol{M}\dot{\xi} + \boldsymbol{C}(\xi)\xi + \boldsymbol{D}(\xi)\xi + \boldsymbol{g}(\eta) = \boldsymbol{\tau_E} + \boldsymbol{\tau}$$

where

$$\boldsymbol{M} = \boldsymbol{M}_{\mathrm{RB}} + \boldsymbol{M}_{\mathrm{Added}}$$
$$\boldsymbol{C}(\xi) = \boldsymbol{C}_{\mathrm{RB}}(\xi) + \boldsymbol{C}_{\mathrm{Added}}(\xi)$$

If the environmental forces $\boldsymbol{\tau_E}$ is considered as disturbances and ignored from the model, we finally get the dynamic model as

$$\boldsymbol{M}\dot{\xi} + \boldsymbol{C}(\xi)\xi + \boldsymbol{D}(\xi)\xi + \boldsymbol{g}(\eta) = \tau \tag{A.22}$$

Note that this 6-DOF dynamic model is expressed in the blimp body-fixed frame $\mathcal{F}_b$. If we add the superscripts for the vectors to mention which frame they are with respect to, it becomes

$$\boldsymbol{M}\dot{\xi}^b + \boldsymbol{C}(\xi^b)\xi^b + \boldsymbol{D}(\xi^b)\xi^b + \boldsymbol{g}(\eta^n) = \tau^b$$

which is the same as (2.10).

Appendix B

Blimp V2 Electric Circuit Design

The electric circuit design for NON-A blimp V2 robot has two parts, that is the main control board and motor driver board. The required functionalities and demanded interfaces are analyzed as shown in Figure 5.3. The electric circuit schematics for the two boards are conceived independently with the connection interface reserved, then the PCB are designed via Altium Designer.

In this appendix, only the final version of electric circuit schematics and PCB are presented, the process of "design, test, debug, modify" developing loop is not presented here for clarity of writing.

B.1 Main Control Board

The main control board includes the following components:

- Power conversion module
- Micro-controller and accessories
- Motor driver board interface
- Communication module interface
- Sensors interfaces and reserved ports

B.1.1 Power conversion module

The robot utilizes 7.4V LiPo (Lithium Polymer) battery as its external power source, the motors are supplied by 7.4V voltage, but for the micro-controller

STM32 and sensors, the voltage needs to be transformed to lower level, therefore a power conversion module is conceived to achieve the task.

The schematic for power conversion module is shown in Figure B.1.

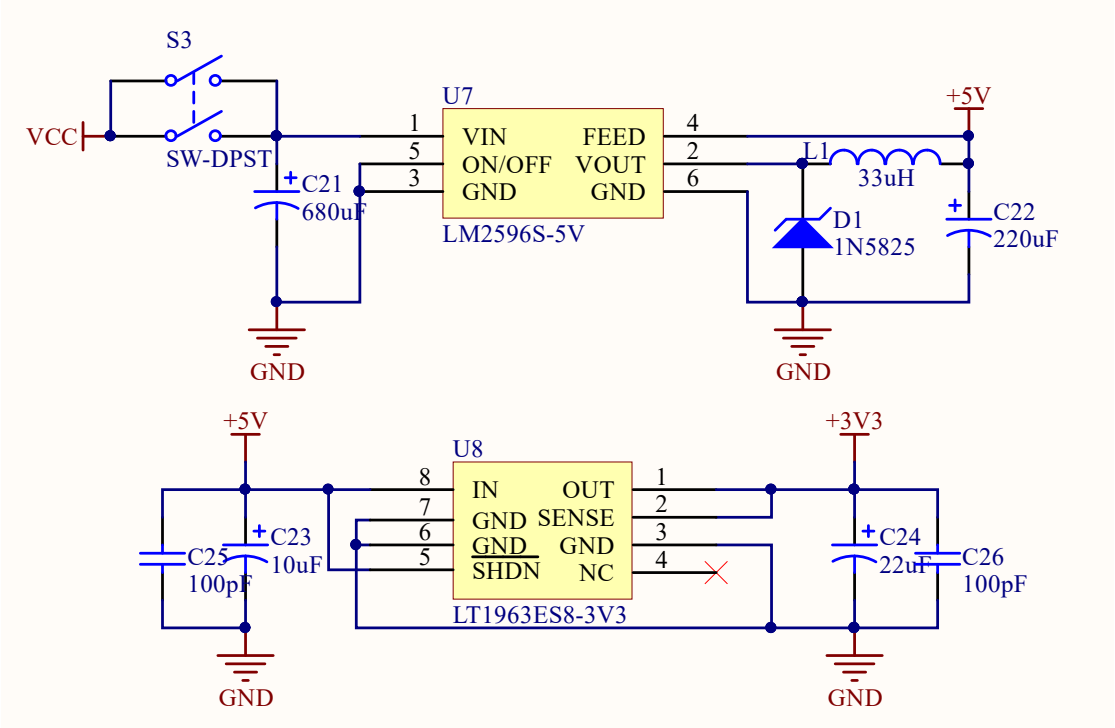

Figure B.1 – Power conversion module schematic

We use a LM2596 power converter chip to convert the voltage from 7.4V to 5V, and the 5V output can be used as a sub-power source for sensors such as US range finder and wireless camera [TexasInstruments, 2019a]. Then the voltage is converted again from 5V to 3.3V by the LT1963 regulator chip, the 3.3V output can be used to supply the micro-controller STM32 and other sensors and chips [LinearTechnology, 2019].

B.1.2 Micro-controller and accessories

For the micro-controller STM32F103VET6, we want to exploit the potential of its resources to a maximum, thus we design a schematic such that all of its GPIO pins are led out to headers. This circuit is shown in Figure B.2.

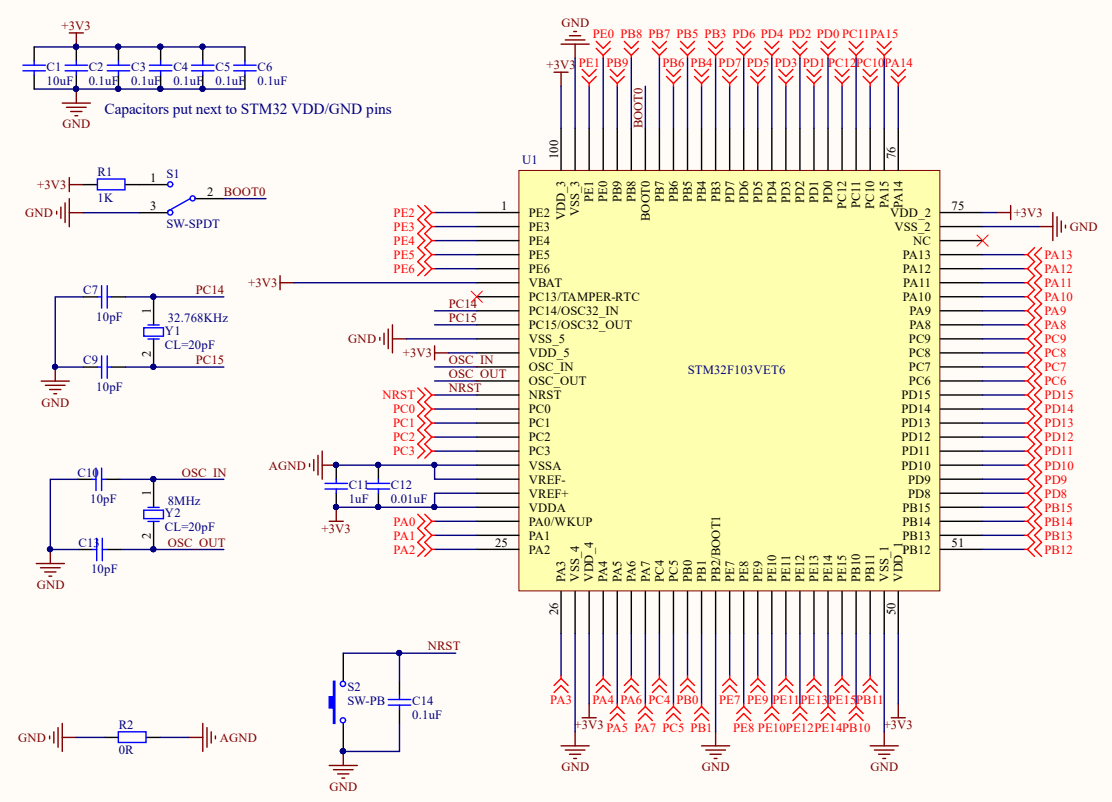

Figure B.2 – Micro-controller and its accessories schematic

The 8MHz clock provided by external crystal oscillator (Y2 in Figure B.2) is multiplicated 9 times by STM32 internal PLL (Phase-Locked Loop), hence the micro-controller can work at a frequency of 72MHz.

Some of the led out GPIO pins are grouped together to form interfaces such as SPI, I2C for sensors, while the others are put in regular headers for possible extension [STMicroelectronics, 2018b].

The micro-controller STM32 supports online debugging function with JTAG (Joint Test Action Group) standard, which is very useful when developing the program for robot [STMicroelectronics, 2019]. Thus an interface with JTAG is designed as shown in Figure B.3.

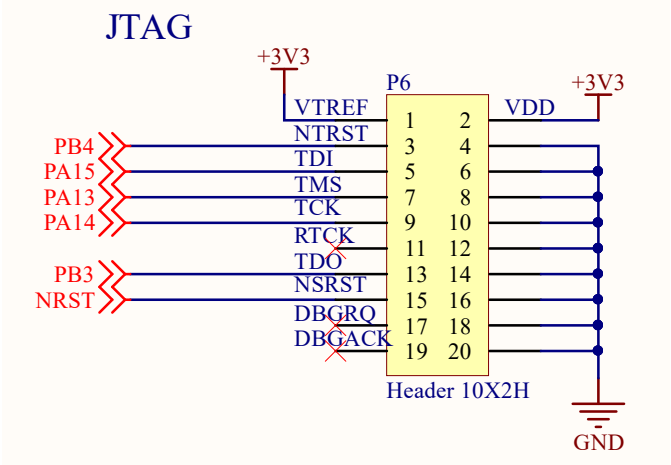

Figure B.3 – JTAG interface for micro-controller schematic

B.1.3 Motor driver board interface

The schematic of the interface on main control board for the connection with motor driver board is shown in Figure B.4.

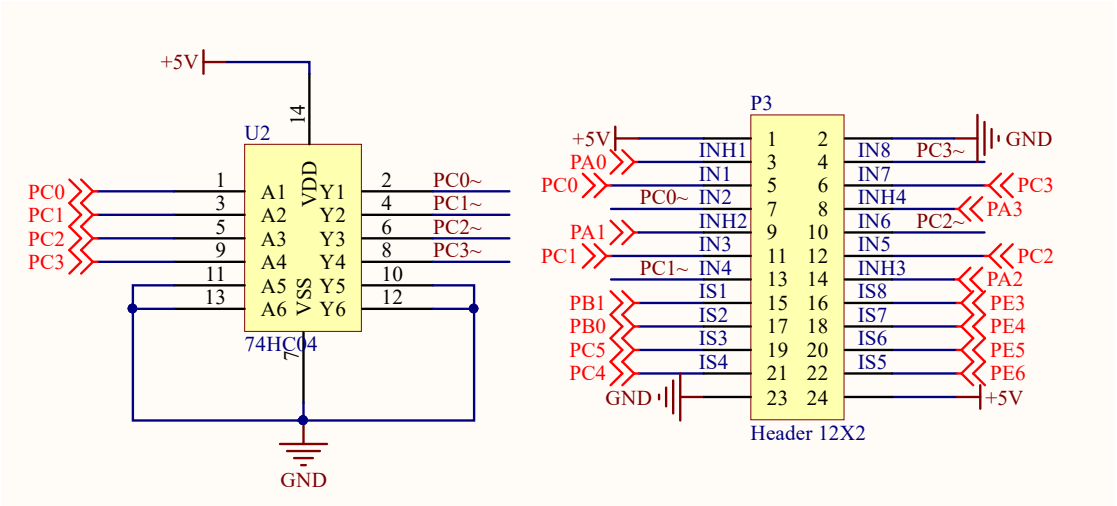

Figure B.4 – Motor driver board interface schematic

Since for both the BTS7960B chip used in the first version of motor driver board and the L298N chip used in the second version of motor driver board, it need two I/O port for the control of DC motor rotating direction and one PWM signal for the control of DC motor rotating speed [InfineonTechnologies, 2018;

STMicroelectronics, 2018a]. Therefore, for the four motors mounted on NON-A blimp V2 robot, it needs 12 GPIO ports of the micro-controller STM32.

In order to save some of the ports resource, it can be observed that the two ports for motor rotating direction control always have opposite signals for rotating forward and backward, thus we can use this property to save some ports resources, we decided to use a 74HC04 Hex inverter chip to invert four of the I/O signals (PC0, PC1, PC2, PC3 in Figure B.4) and get four pair of opposite signals (PC0~, PC1~, PC2~, PC3~in Figure B.4) (Note that here PC signifies "Port C" of the micro-controller) [Nexperia, 2019].

Then the four PWM signals (PA0, PA1, PA2, PA3 in Figure B.4) are also led to the header (P3 in Figure B.4). The same header is appeared on the motor driver board to connect these two boards together.

B.1.4 Communication module interface

For the communication module interface, we allocated one USART for serial communication with PC and one USART to plug the wireless communication module XBee. The schematic is shown in Figure B.5.

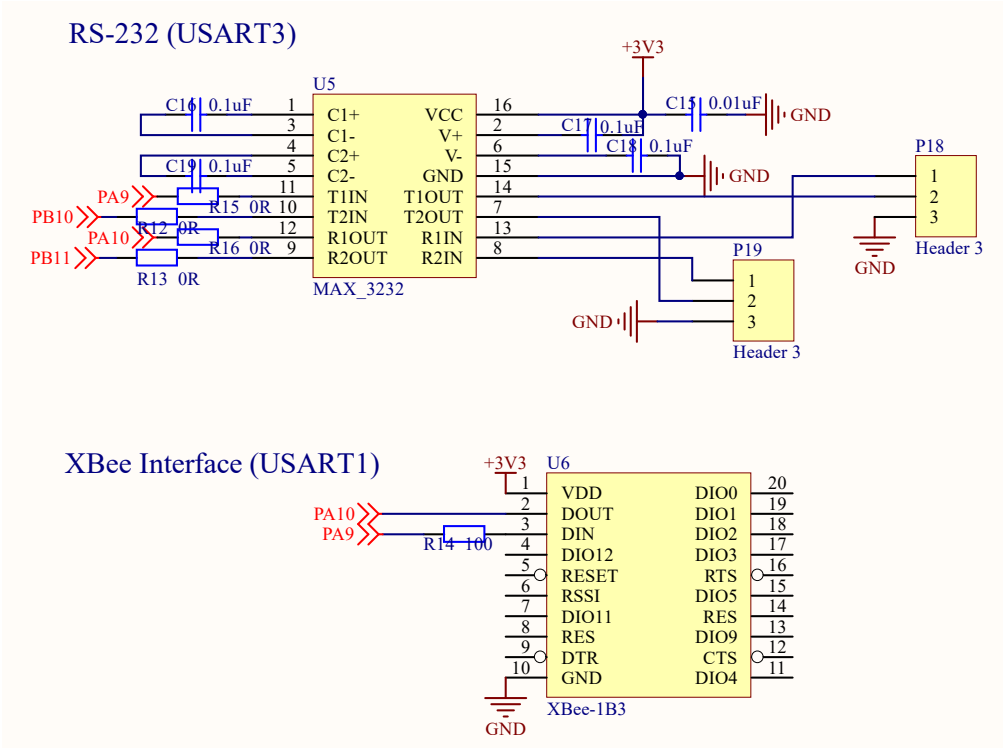

Figure B.5 – Communication module interface schematic

For the serial communication port, since the micro-controller STM32 utilizes TTL (Transistor-Transistor Logic) signal level, while the PC uses RS-232 (Recommended Standard 232) level, thus the level needs to be transformed and made compatible between the two devices. For this purpose, a MAX3232 chip is used to achieve the level transformation [TexasInstruments, 2019b].

Although the serial communication is convenient for program debugging purpose, it is not able to be used when the robot is moving. The XBee is a powerful wireless communication module, which has multiple choices of types with different communication standard such as Wi-Fi, Bluetooth and 802.15.4 etc [Digi, 2019]. Since we only utilizes the basic data transmission function of the XBee module, we only have to design a interface that connects the data out and data in pins as shown in Figure B.5.

B.1.5 Sensors interfaces and reserved ports

For the chosen sensors such as the IMU MPU-6050, the US range finder LV-MaxSonar-EZ1, we design interfaces for the sensor modules which can be found in market, hence the sensors can be "Plug and Play" easily. Moreover the SPI interface is also widely used for sensors, so the SPI1 and SPI2 of STM32 are led out for potential use. The schematic is shown in Figure B.6.

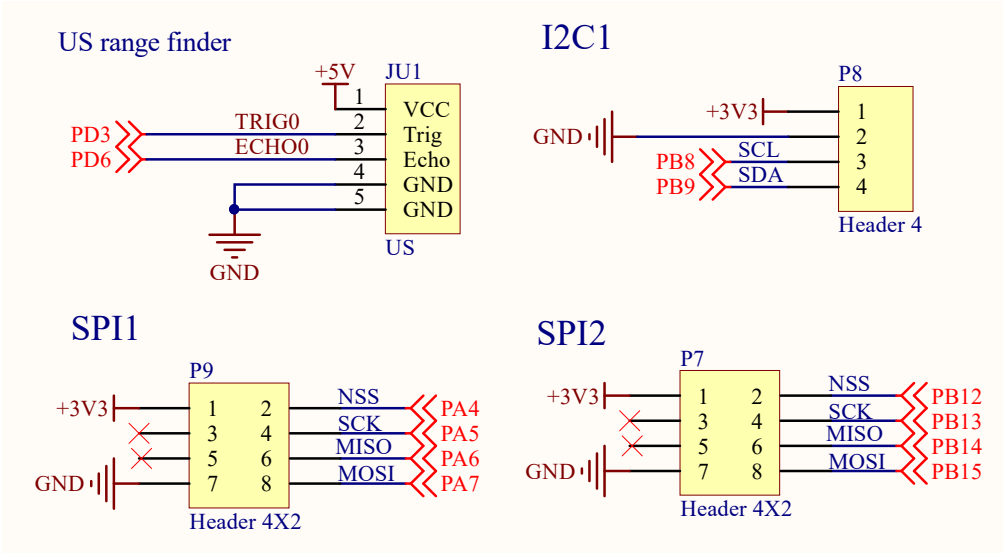

Figure B.6 – Sensors interfaces schematic

Other GPIO ports are reserved and led out to headers, they can be used for sensor connection, external interrupt, power supply or other functions. The schematic is shown in Figure B.7.

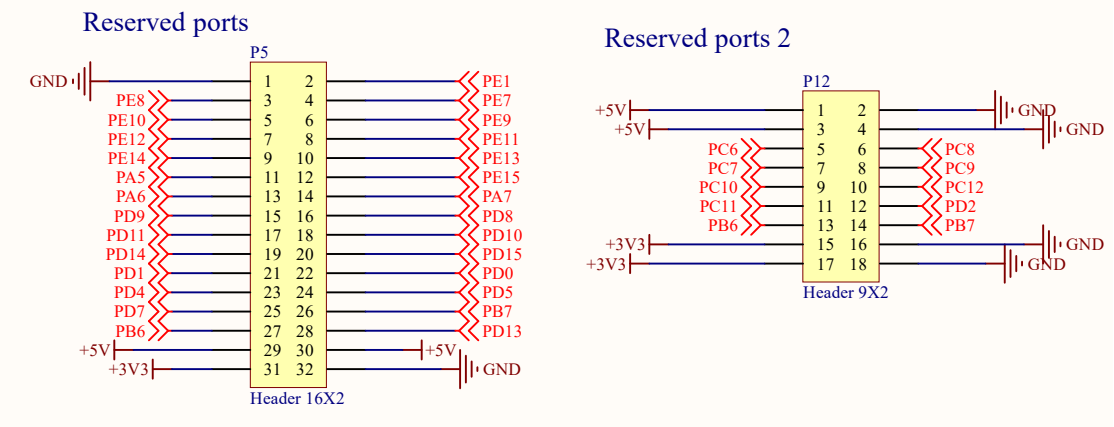

Figure B.7 – Reserved STM32 GPIO ports schematic

B.1.6 PCB design

After the designed schematics are verified and debugged, we can start to create the PCB of the main control board. The electric circuit is a rather simple one, so we only use two layers for the circuit on a board of size 90mm × 80mm, the designed PCB is shown in Figure B.8.

In the view of the electric circuit layout design, the following terms are taken into consideration:

- The red color signifies electric circuits on the top layer, while the blue color is for the bottom layer, and copper polygon plane is placed for the ground of circuit. The footprints for the chips, headers and electric components are chosen properly.
- Components from the same module are put as close to each other as possible for the shortest leads to connect them.
- On the Figure B.8, we can see that the power conversion module is arranged at the bottom left corner, and the copper polygon ground is largely separated from the copper polygon ground of the signal part on the right except for several parts. This is for avoiding signal perturbations from the power circuit to the signal circuit.
- The micro-controller chip is located on the right half of the board, all of its pins are led out to other chips or headers on the border of the board. The leads are designed to go only vertically and horizontally and make as few corners and via holes as possible to reduce the oscillation of signal during transmission in the leads.

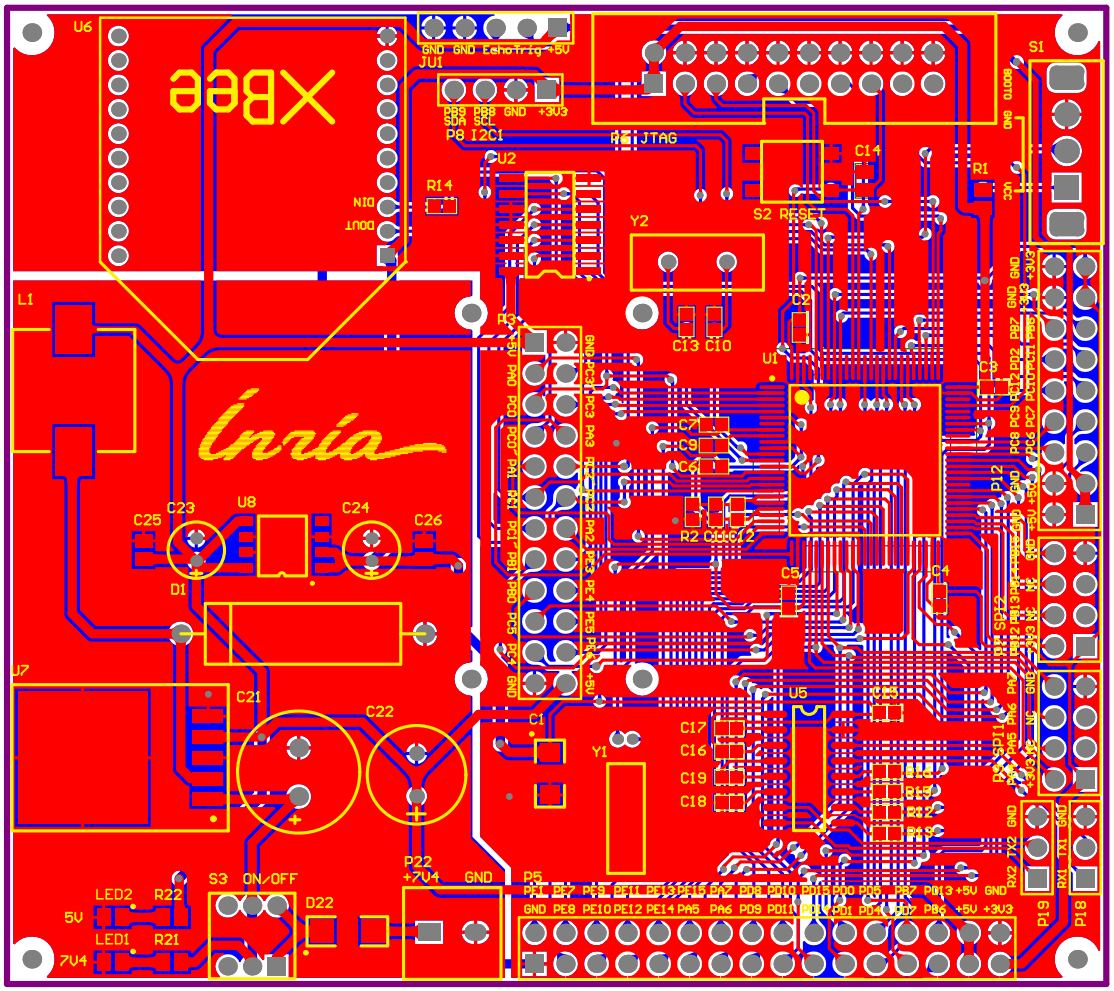

Figure B.8 – Main control board PCB design

- The external crystal oscillators Y1 and Y2 are put far away from the signal leads to avoid perturbation.
- The leads of receiving and transmitting from the communication modules are designed to be parallel one from another and with similar lengths and corners to assure the synchronization of the communication in two directions.

In the view of physical consideration, we have the following designs:

- The weight is balanced on the board, which means there are heavy components on both the left and right, top and bottom of the board, and the mass center of the board with components is approximately located at its geometric center (see also the 3D Figure 5.4).

- On the center of the board, we put the header to connect with motor driver board, and after the two boards are connected, the header can be regarded as a support between the two boards on the center to reinforce the structure strength.
- On the four corners, we designed big through holes for the fixing screw to pass.
- Another four smaller through holes are put around the center header, to pass the fixing screw with the second version of the motor driver board, because it is lighter and smaller than this main control board.
- The headers for sensors, the reserved ports, the XBee module, the JTAG interface and the power plug are located on the borders of the board, for the ease to use and plug.

B.2 Motor Driver Board

As it is mentioned before, the motor driver board also has two versions, from the use of MOSFET chips which support big current output to the use of motor driver chips which is lighter but less efficient.

B.2.1 First version with BTS7960B

The first version of motor driver boards is realized with a high current PN half bridge chip BTS7960B, which means in order to form a full bridge to driver one motor, it needs two of this chip. The designed schema is shown in Figure B.9.

The Figure B.9 only shows the full bridge to drive one DC motor, where the enable port INH1 is connected with the PWM signal, and IN1OUT, IN2OUT are the opposite I/O signal pair from main control board, OUT1 and OUT2 are connected to the two ends of DC motor.

An example of PWM wave motor control with the full bridge is shown in Table B.1.

INH	IN1	IN2	Direction	Speed
60%PWM	1	0	Forward	60%
40%PWM	0	1	Backward	40%

Table B.1 – Example of PWM wave motor control

In fact, the IN1OUT and IN2OUT are not directly from main control board, they are the output of an optoelectronic isolation circuit as shown in Figure B.10.

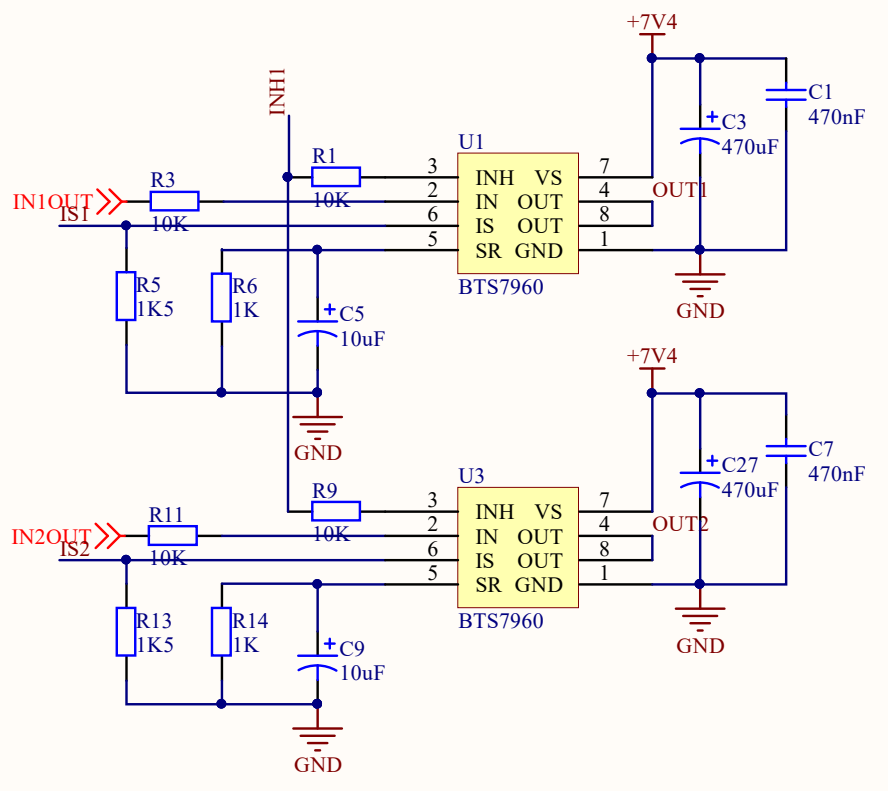

Figure B.9 – Full bridge motor driver circuit with BTS7960B schematic

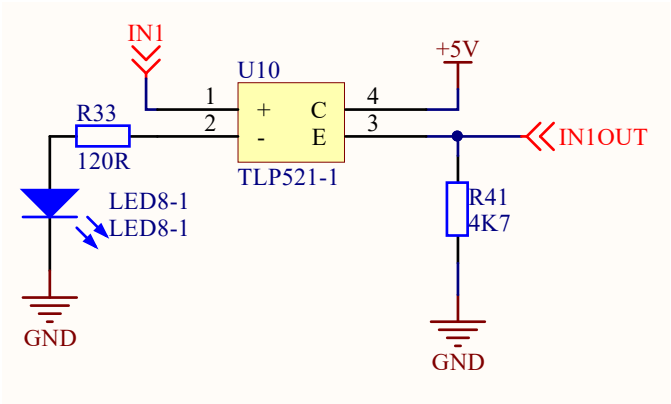

Figure B.10 – Optoelectronic isolation circuit for motor driver board (first version) schematic

Since the signals in motor driver board is stronger than main control board, thus the optoelectronic isolation is added to avoid the influences, it is achieved by TLP521 photocoupler chip, where the IN1 and IN1OUT signal are isolated

by phototransistor [Toshiba, 2019].

The connectors for motor, power and the interface with main control board are shown in Figure B.11.

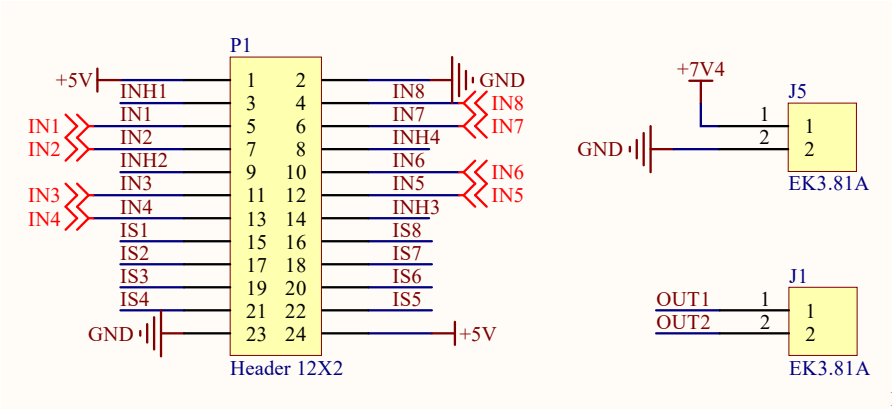

Figure B.11 – Connectors and interface for motor driver board schematic

Note that the Header P1 is same to the one shown in Figure B.4, so these two headers on the two boards can be plugged together. And in B.11, only one of the four motor output header is shown.

The PCB design for the first version of motor driver board with BTS7960B is shown in Figure B.12.

Similar consideration for the electric circuit layout and physical demands are taken into account when design this PCB. In addition, some more points are worth to be noticed:

- All the leads for high current output of motors are set with larger width to assure the heat are not cumulated at the leads.
- The board has same size as the main control board (90mm × 80mm), the four through holes at corners are aligned to pass the fixing screws.
- The header at center of the board is also aligned with the one on main control board (see also the 3D Figure 5.5).
- the four output connector for motors are located on the borders of the board for the ease of plug.

B.2.2 Second version with L298N

As it is mentioned before, although the motor driver board with BTS7960B is powerful and can provide a maximum current output of 43A, it is a little too heavy for the chosen balloon with helium gas.

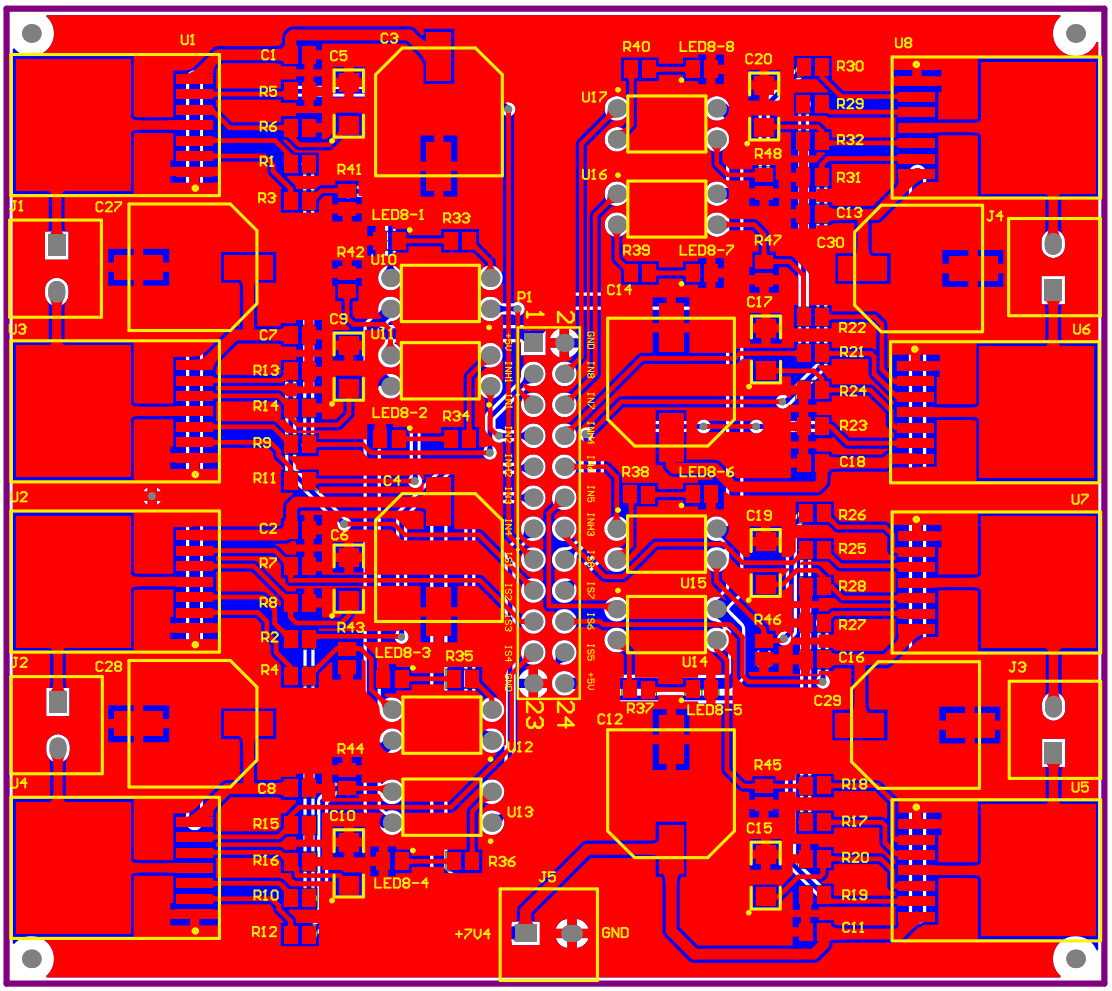

Figure B.12 – Motor driver board (first version) PCB design

That is why we re-design a second version of the motor driver board with a less powerful yet sufficient motor driver chip L298N, and with much smaller size and weight. The L298N integrates two full-bridge in one chip, thus we only need two of such chips to drive the four DC motors of maximum current 2A. The motor driver circuit schematic is shown in Figure B.13.

Note that the 5V sub-power source from the main control board power conversion module is used to supply the L298N chip. Although the L298N generates more heats at work and has limited current output capacity, we believe it is the only solution for a lighter version of the motor driver board, and it is enough for our tests on the robot at present. If we can find a bigger balloon with higher payload, we should come back to the motor driver board with BTS7960B chip for better performance.

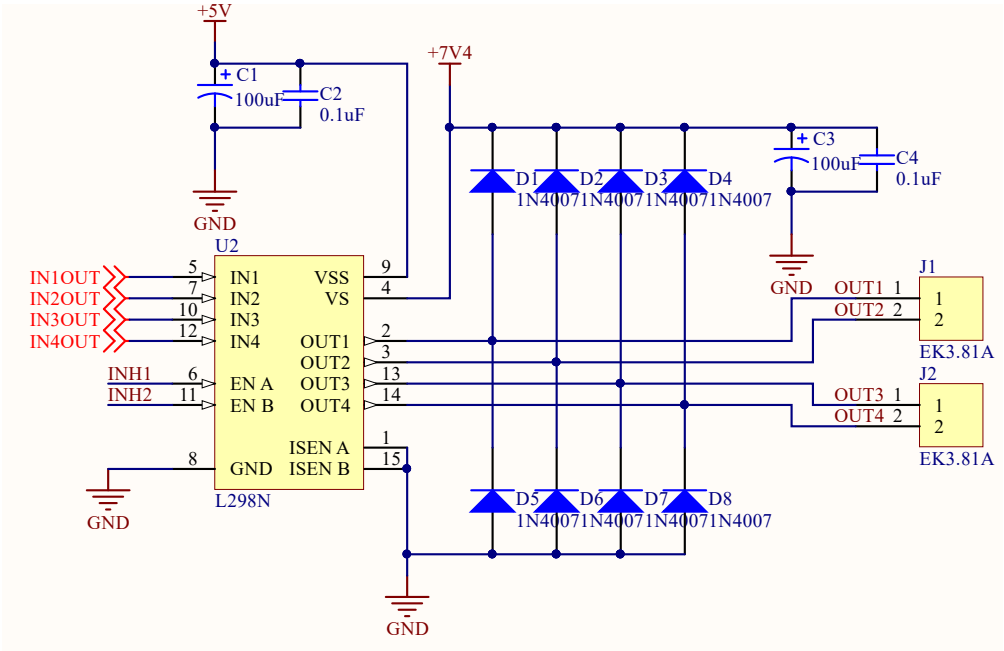

Figure B.13 – Full bridge motor driver circuit with L298N schematic

In the second version, we also used the optoelectronic isolation to avoid the influences, as shown in Figure B.14.

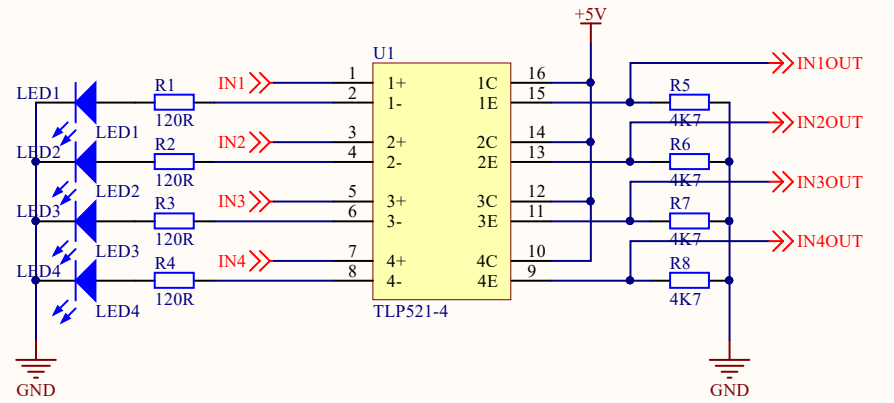

Figure B.14 – Optoelectronic isolation circuit for motor driver board (second version) schematic

The TLP521-4 chip can support four inputs and four outputs, and one of this chip is lighter than four of the TLP521-1 chip used in the first version of motor driver board. Similarly, it separates the signal from main control board to the one runs in the motor driving board, thus improves the system reliability.

The PCB design for the second version of motor driver board with L298N is shown in Figure B.15.

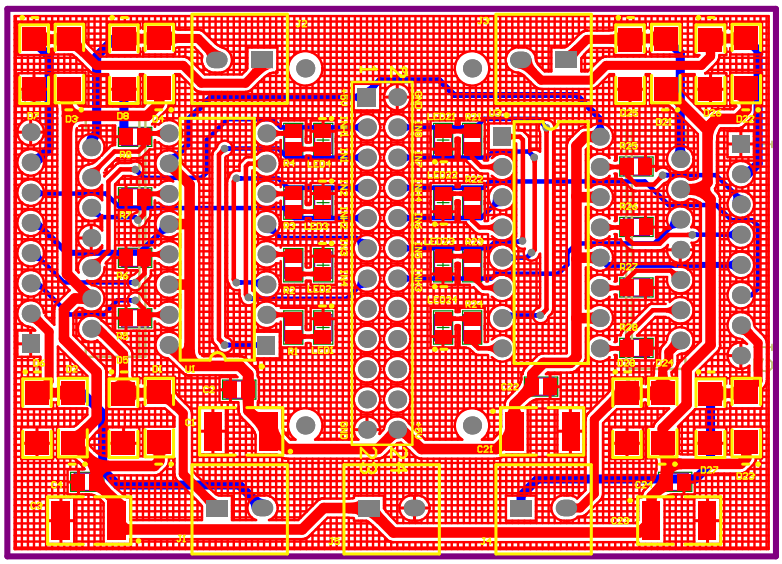

Figure B.15 – Motor driver board (second version) PCB design

The following considerations are taken into account to reduce weight:

- The board size is reduced to 64mm × 46mm.
- Use hatched copper polygon pour instead of solid one for the ground connection, which is a balance between the heat diffusion and weight.
- Use two TLP521-4 chip instead of eight TLP521-1 chip, see also the 3D Figure 5.6.

The second version of motor driver board is one third lighter than the first version and can be carried by the balloon we chose.

There are also four through holes around the center header, which is used for fixing this motor driver board with main control board.

At this point, we finished the presentation of the created electric circuit for our NON-A blimp V2 robot. The main control board, motor driver board together with communication modules, sensors modules and DC motors bought from market are then integrated together to form the blimp V2 robot and tested in indoor environment to validate the conceived motion control laws.

Résumé Substantiel

Introduction

Aujourd'hui, la robotique est un domaine scientifique et technologique en plein essor. Parmi les robots volants, il convient de mentionner les dirigeables qui sont des aéronefs plus légers que l'air (en anglais LTA), pour leurs avantages par rapport à d'autres véhicules aériens. Parmi ces avantages, citons:

- Capacité à décollé et se posé verticalement, le vol stationnaire et à basse vitesse;
- Ratio "charge utile / poids" élevé;
- Grande endurance dans l'air;
- Faible consommation d'énergie;
- Faible niveau de bruit acoustique;
- Interaction Homme-Robot sécurisée.

Par conséquent, le robot dirigeable est idéal pour diverses applications d'intérieur telles que:

- Surveillance de longue durée;
- Publicité et divertissement;
- Exploration et cartographie d'environnement inconnu;
- Inventaire de marchandises en entrepôt;
- Aide à la navigation piétonne dans les grands établissements;
- Plate-forme de recherche scientifique et d'éducation.

Cependant, contrairement aux dirigeables de plus grande taille, le dirigeable destiné à application en intérieur est limité en taille et donc en charge utile. Ainsi seuls des capteurs et des actionneurs de faible poids peuvent être intégrés dans un micro-système intégré, ce qui signifie que la mesure des capteurs embarqués ne peut pas être très précise et la capacité des actionneurs est limité. De plus, bien que la structure du robot dirigeable miniature ne soit pas aussi compliquée que celle de ces dirigeables extérieurs et que l'environnement intérieur soit plus stable et moins perturbé, l'espace de travail du dirigeable intérieur est plus encombré et rempli d'obstacles. Ainsi, il faut une plus grande précision du contrôle du mouvement du dirigeable pour réaliser des opérations en intérieur.

En général, l'étude des véhicules aériens miniatures demeure un défi, c'est pourquoi nos travaux de recherche s'intitulent "Développement d'un robot dirigeable pour opération en intérieur".

Les chercheurs accordent de plus en plus d'attention aux robots dirigeables autonomes ces dernières années. Ils utilisent des robots dirigeables comme plate-formes d'expérimentation dans diverses études, telles que la localisation du robot, l'algorithme d'évitement d'obstacles, la planification de trajectoire et le suivi de trajectoire, etc. Ici, une brève revue des travaux existants est présentée par ordre chronologique.

Dans [Gomes, 1990], les auteurs ont étudié le modèle dynamique et analysé les modes propres du dirigeable. Le travail [Zhang and Ostrowski, 1999] a utilisé des techniques d'asservissement visuel pour contrôler le dirigeable suivant un objet quasi-statique. [Fukao et al., 2003a,b] a utilisé des informations sur l'image pour obtenir la position du dirigeable, ils ont réalisé un contrôle circulaire du robot autour de la cible spécifiée. [Badia et al., 2005] utilisait un contrôleur bio-inspiré pour suivre la trajectoire planifiée du dirigeable et éviter les collisions. Le travail de [Zufferey et al., 2006] a utilisé des contrôleurs neuronaux en simulation pour cartographier les entrées basées vision dans les commandes du moteur afin d'accélérer le mouvement du robot volant tout en évitant les collisions. Les auteurs de [Ko et al., 2007] ont combiné leur modèle amélioré de processus gaussiens à l'apprentissage par renforcement et ont conçu un contrôleur de l'angle de lacet de dirigeable. Le travail de [Rottmann et al., 2007a] a également utilisé l'apprentissage par renforcement pour concevoir un contrôleur pouvant être utilisé pour contrôler l'altitude du dirigeable sans connaître la dynamique du système ou les paramètres d'environnement. Dans l'article de [Burri et al., 2013], un dirigeable sphérique capable de mouvement holonomique est conçu.

Comme on peut le voir d'après les travaux correspondants, certains d'entre eux n'ont pas proposé de modèles pour le robot dirigeable intérieur, tandis que d'autres ont utilisé des modèles complexes, qui nécessitent des données expérimentales précises pour identifier les paramètres, mais il existe encore

des perturbations de l'environnement qui ne peuvent pas être modélisées. De l'analyse ci-dessus, nous savons que la quantité de capteurs montés sur le robot et la complexité de calcul des algorithmes de contrôle et d'estimation sont limitées. C'est pourquoi nous avons l'intention d'utiliser un modèle simple pour représenter le mouvement du dirigeable et de concevoir un contrôleur capable d'estimer et de compenser les perturbations en temps réel. Ainsi, la précision du contrôle est assurée tout en minimisant sa complexité ainsi que le nombre de dispositif embarqués.

Le travail est composé de parties théoriques et pratiques. Pour la partie théorique, il inclut la modélisation, la conception des contrôleurs et les simulations. Pour la partie pratique, il contient la conception et la création du système de robot dirigeable, ainsi que les résultats des tests.

Modélisation

Pour la partie théorique du travail, nous commençons par l'étude du modèle du robot dirigeable au Chapitre 2. Le modèle dynamique à 6 degrés de liberté du dirigeable couramment utilisé est présenté en détail, il est donné par (équation (2.10)):

$$\boldsymbol{M}\dot{\boldsymbol{\xi}}^{\boldsymbol{b}} + \boldsymbol{C}(\boldsymbol{\xi}^b)\boldsymbol{\xi}^b + \boldsymbol{D}(\boldsymbol{\xi}^b)\boldsymbol{\xi}^b + \boldsymbol{g}(\boldsymbol{\eta}^n) = \boldsymbol{\tau}^b$$

De manière explicite, il prend la forme (équation (2.20)):

$$\begin{aligned}
&m'_x\dot{v}^b_x + m\dot{\omega}^b_y z_G - \omega^b_z(m'_y v^b_y - m\omega^b_x z_G) + m'_z\omega^b_y v^b_z - v^b_x(D_{v_x} + D_{v_x^2}|v^b_x|) + (f_G - f_B)\sin\theta = f_{px}\\
&m'_y\dot{v}^b_y - m\dot{\omega}^b_x z_G + \omega^b_z(m'_x v^b_x + m\omega^b_y z_G) - m'_z\omega^b_x v^b_z - v^b_y(D_{v_y} + D_{v_y^2}|v^b_y|) - (f_G - f_B)\cos\theta\sin\phi = f_{py}\\
&m'_z\dot{v}^b_z + \omega^b_x(m'_y v^b_y - m\omega^b_x z_G) - \omega^b_y(m'_x v^b_x + m\omega^b_y z_G) - v^b_z(D_{v_z} + D_{v_z^2}|v^b_z|) - (f_G - f_B)\cos\theta\cos\phi = f_{pz}\\
&I'_x\dot{\omega}^b_x - m\dot{v}^b_y z_G + I'_z\omega^b_y\omega^b_z - v^b_z(m'_y v^b_y - m\omega^b_x z_G) - \omega^b_z(I'_y\omega^b_y + mv^b_x z_G) + m'_z v^b_y v^b_z\\
&\qquad - \omega^b_x(D_{\omega_x} + D_{\omega_x^2}|\omega^b_x|) + z_G f_G\cos\theta\sin\phi = \tau_{px}\\
&I'_y\dot{\omega}^b_y + m\dot{v}^b_x z_G - I'_z\omega^b_x\omega^b_z + v^b_z(m'_x v^b_x + m\omega^b_y z_G) + \omega^b_z(I'_x\omega^b_x - mv^b_y z_G) - m'_z v^b_x v^b_z\\
&\qquad - \omega^b_y(D_{\omega_y} + D_{\omega_y^2}|\omega^b_y|) + z_G f_G\sin\theta = \tau_{py}\\
&I'_z\dot{\omega}^b_z + v^b_x(m'_y v^b_y - m\omega^b_x z_G) - v^b_y(m'_x v^b_x + m\omega^b_y z_G) + \omega^b_x(I'_y\omega^b_y + mv^b_x z_G) - \omega^b_y(I'_x\omega^b_x - mv^b_y z_G)\\
&\qquad - \omega^b_z(D_{\omega_z} + D_{\omega_z^2}|\omega^b_z|) = \tau_{pz}
\end{aligned}$$

Comme nous l'avons observé lors de travaux sur de petits robots dirigeables intérieurs où le modèle à 6 degrés de liberté est appliqué, les résultats obtenus par les précédents chercheurs sont décevants à cause de la complexité du modèle et des difficultés de conception des lois de commande soulevées par le modèle.

Au contraire, dans notre travail, une nouvelle approche pour modéliser le mouvement du robot dirigeable est proposée. L'idée est d'utiliser un modèle simplifié complété par un terme de perturbation estimé en temps réel afin

de réduire la complexité du modèle tout en garantissant les performances du contrôleur et en améliorant sa robustesse.

Partant de cette idée, le modèle dynamique à 6 degrés de liberté est simplifié sous des hypothèses raisonnables (Hypothèses 2.1, 2.2 et 2.3), il devient (équation (2.22)):

$$\begin{aligned} m'_x \dot{v}^b_x - m'_y \omega^b_z v^b_y - D_{v_x} v^b_x &= f_{px} \\ m'_y \dot{v}^b_y + m'_x \omega^b_z v^b_x - D_{v_y} v^b_y &= f_{py} \\ m'_z \dot{v}^b_z - D_{v_z} v^b_z + (f_B - f_G) &= f_{pz} \\ I'_z \dot{\omega}^b_z + (m'_y - m'_x) v^b_x v^b_y - D_{\omega_z} \omega^b_z &= \tau_{pz} \end{aligned}$$

On peut noter que le mouvement dans la direction verticale (mouvement en altitude) et le mouvement dans le plan horizontal du dirigeable étudié dans ce travail peuvent être découplés, ce qui signifie qu'ils sont indépendants les uns des autres. De plus, le contrôle du mouvement du robot dirigeable peut être divisé en deux sous-problèmes: le contrôle du mouvement en altitude et le contrôle du mouvement planaire, ils peuvent être analysés et résolus indépendamment, puis combinés pour obtenir un contrôle complet du mouvement du dirigeable.

Par conséquent, après découplage et transformation, nous obtenons le modèle de mouvement d'altitude (équation (2.25)):

$$\ddot{z} = a_z \dot{z} + b_z u_z + c_z$$

et le modèle de mouvement planaire (équation (2.29)):

$$\begin{cases} \ddot{x} = c_\psi b u + \kappa_1(a_x, a_y, \psi)\dot{x} + \kappa_3(a_x, a_y, \psi)\dot{y} \\ \ddot{y} = s_\psi b u + \kappa_2(a_x, a_y, \psi)\dot{y} + \kappa_3(a_x, a_y, \psi)\dot{x} \\ \ddot{\psi} = b_\psi v + a_\psi \dot{\psi} \end{cases}$$

Les deux modèles simplifiés sont considérés comme des modèles nominaux de mouvements découplés de robot dirigeable, ils sont complétés par des termes de perturbation pour la conception des contrôleurs robustes. Des tests sont effectués pour identifier les paramètres de ces modèles nominaux.

Contrôle d'altitude

Au Chapitre 3, le contrôleur pour le mouvement d'altitude est conçu.

En raison du défaut matériel du robot prototype de dirigeable NON-A observé lors des tests, le modèle d'altitude nominale est modifié pour incorporer les défauts. Ainsi, le modèle d'altitude nominale complété par le terme de

perturbation devient (équation (3.1)):

$$\ddot{z}(t) = a_{z\sigma}\dot{z}(t) + b_{z\sigma}\left(u_z(t-\tau_{\text{nom}}) + d_z(t)\right)$$

Par conséquent, le système de contrôle de l'altitude du dirigeable étudié dans ce travail est considéré comme un *système commuté avec un retard constant complété par des perturbations bornées incertaines*.

L'altitude du dirigeable z et la vitesse sur l'axe vertical $\dot{z}$ sont choisis comme vecteur d'état $X = \begin{bmatrix} z & \dot{z} \end{bmatrix}^{\mathrm{T}}$, le système peut être écrit sous la forme d'espace d'état en combinant le modèle nominal et le terme de perturbation estimé en temps réel (équation (3.2)):

$$\begin{cases} \dot{X}(t) = A_{z\sigma}X(t) + B_{z\sigma}u_z(t-\tau_{\text{nom}}) + B_{z\sigma}d_z(t) \\ y(t) = C_zX(t) \end{cases}$$
$$\sigma \in \mathcal{P} = \{1,2,3,4\}$$

Afin de concevoir un contrôleur basé sur la compensation de perturbation pour le mouvement d'altitude, les étapes suivantes sont proposées:

1) Concevoir un observateur capable d'estimer l'état et le signal de commutation d'un système commuté retardé;

2) Concevoir un estimateur en temps réel du terme de perturbation;

3) Concevoir un contrôleur avec compensation des perturbations qui stabilise le système en boucle fermée.

Pour la conception de l'observateur, nous avons étudié et comparé les différentiateurs HG, HOSM et HOMD et le différenciateur HOMD est finalement choisi pour la tâche d'estimation, il prend la forme (équation (3.23)):

$$\begin{aligned} \dot{x}_1 &= -k_1\lceil x_1 - y\rfloor^{\alpha} + x_2 \\ \dot{x}_2 &= -k_2\lceil x_1 - y\rfloor^{2\alpha-1} + x_3 \\ \dot{x}_3 &= -k_3\lceil x_1 - y\rfloor^{3\alpha-2} \end{aligned}$$

et l'observateur est (équation (3.24)):

$$\hat{X} = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \end{bmatrix}^{\mathrm{T}}$$
$$\hat{\sigma} = \begin{cases} 1, & u_z \geq 0 \text{ and } \hat{x}_2 \geq 0 \\ 2, & u_z \geq 0 \text{ and } \hat{x}_2 < 0 \\ 3, & u_z < 0 \text{ and } \hat{x}_2 \geq 0 \\ 4, & u_z < 0 \text{ and } \hat{x}_2 < 0 \end{cases}$$

Ensuite, afin d'estimer la perturbation $d_z(t)$ en temps réel, un filtre est conçu (équation (3.25)):

$$\begin{cases} \dot{X}_{\text{fil}}(t) = A_{z\hat{\sigma}} X_{\text{fil}}(t) + B_{z\hat{\sigma}} u_z(t - \tau_{\text{nom}}) + L(y(t) - y_{\text{fil}}(t)) \\ y_{\text{fil}}(t) = C_z X_{\text{fil}}(t) \end{cases}$$

Soit $e(t) = X(t) - X_{\text{fil}}(t)$ l'erreur entre le vecteur d'état de système réel et celui de filtre. Nous obtenons enfin l'expression de l'estimation de perturbation (équation (3.27)):

$$\hat{d}_z(t) = \frac{\ddot{e}_1(t) + (l_1 - a_{z\sigma})\dot{e}_1(t) + (l_2 - a_{z\sigma} l_1) e_1(t)}{b_{z\sigma}}$$

Puis, nous avons choisi d'utiliser le contrôleur basé sur le prédicteur de Smith pour compenser le retard nominal fixe, et on a transformé le système en un système à boucle fermée sans retard. Les incertitudes générées par l'approximation du délai sont incluses dans le terme de perturbation et compensées dans le contrôleur. Le prédicteur est (équation (3.31)):

$$\begin{aligned} \hat{X}(t + \tau_{\text{nom}}) = e^{A_{z\hat{\sigma}} \tau_{\text{nom}}} \hat{X}(t) &+ \int_{-\tau_{\text{nom}}}^{0} e^{-A_{z\hat{\sigma}} s} B_{z\hat{\sigma}} u_z(t + s) \mathrm{d}s \\ &+ \int_{-\tau_{\text{nom}}}^{0} e^{-A_{z\hat{\sigma}} s} B_{z\hat{\sigma}} \hat{d}_z(t + \tau_{\text{nom}} + s) \mathrm{d}s \end{aligned}$$

Finalement, le terme de perturbation est compensé dans le contrôleur (équation (3.32)):

$$u_z(t) = -K_{z\hat{\sigma}} \hat{X}(t + \tau_{\text{nom}}) - \hat{d}_z(t + \tau_{\text{nom}})$$

Des simulations sont effectuées pour vérifier l'efficacité et les performances du contrôleur conçu.

Contrôle du mouvement dans le plan horizontal

Dans le Chapitre 4, nous nous concentrons sur le mouvement du dirigeable dans le plan horizontal et concevons un contrôleur basé également sur la compensation de perturbations.

Le modèle nominal de mouvement planaire complété par des termes de

perturbation devient (équation (4.1)):

$$\begin{cases} \ddot{x} = c_\psi bu + \kappa_1(a_x, a_y, \psi)\dot{x} + \kappa_3(a_x, a_y, \psi)\dot{y} + d_x \\ \ddot{y} = s_\psi bu + \kappa_2(a_x, a_y, \psi)\dot{y} + \kappa_3(a_x, a_y, \psi)\dot{x} + d_y \\ \ddot{\psi} = b_\psi v + a_\psi \dot{\psi} + d_\psi \end{cases} \tag{B.1}$$

En raison du fait que le modèle de mouvement plan de dirigeable est sous-actionné, deux approches différentes sont discutées pour transformer le système en une forme plus simple facilitant la conception d'un contrôleur de stabilisation à rétroaction. En fin, nous avons suivi l'approche de transformation de coordonnées pour la simplification de système sous-actionné, par lequel le problème du contrôle de la position exacte (x, y) et de l'orientation ψ du robot est transformé en un problème de contrôle pratique de la position du point $Q(s, r)$ qui est proche de l'origine du repère lié au corps du robot O_b (Voir Figure 4.1). Par conséquent, lorsque la position du point Q est réglée sur la position souhaitée, le centre du robot (x, y) se trouve sur un cercle centré sur (s, r) et de rayon q.

La position de point Q est (équation (4.5)):

$$\begin{bmatrix} s \\ r \end{bmatrix} = \begin{bmatrix} x + q\cos\psi \\ y + q\sin\psi \end{bmatrix}$$

Après la transformation, nous avons (équation (4.7)):

$$\begin{bmatrix} \ddot{s} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} (\kappa_1\dot{s} + \kappa_3\dot{r}) + \left(c_\psi bu - qs_\psi b_\psi v\right) \\ \qquad + \left(\kappa_1 qs_\psi\dot{\psi} - \kappa_3 qc_\psi\dot{\psi} - qc_\psi\dot{\psi}^2 - qs_\psi a_\psi\dot{\psi}\right) + \left(d_x - qs_\psi d_\psi\right) \\ (\kappa_2\dot{r} + \kappa_3\dot{s}) + \left(s_\psi bu + qc_\psi b_\psi v\right) \\ \qquad + \left(-\kappa_2 qc_\psi\dot{\psi} + \kappa_3 qs_\psi\dot{\psi} - qs_\psi\dot{\psi}^2 + qc_\psi a_\psi\dot{\psi}\right) + \left(d_y + qc_\psi d_\psi\right) \end{bmatrix}$$

En choisissant le vecteur d'état comme $X = \begin{bmatrix} s & \dot{s} & r & \dot{r} \end{bmatrix}^{\mathrm{T}}$, le système linéarisé pour le mouvement planaire de dirigeable peut être écrit sous la forme d'espace d'état (équation (4.12)):

$$\begin{cases} \dot{X} = AX + B(U + \Delta + D) \\ y = CX \end{cases}$$

En conséquence, le système de contrôle de mouvement dans le plan horizontal du robot dirigeable V2 étudié est considéré comme un *système non-linéaire complété par des perturbations bornées incertaines*.

Ensuite, comme pour la conception du contrôleur de mouvement d'altitude, une méthode d'estimation des termes de perturbation est proposée, un filtre est

conçu (équation (4.13)):

$$\begin{cases} \dot{X}_{\text{fil}} = AX_{\text{fil}} + B(U + \Delta) + L(y - y_{\text{fil}}) \\ y_{\text{fil}} = CX_{\text{fil}} \end{cases}$$

L'erreur $e_{\text{fil}} = X - X_{\text{fil}}$ est utilisé pour dériver l'expression de l'estimation de perturbation (équation (4.16)):

$$\begin{cases} \hat{d}_s = \ddot{e}_1 + (l_{11} - \kappa_1)\dot{e}_1 + (l_{21} - \kappa_1 l_{11} - \kappa_3 l_{31})e_1 + (l_{12} - \kappa_3)\dot{e}_3 + (l_{22} - \kappa_1 l_{12} - \kappa_3 l_{32})e_3 \\ \hat{d}_r = \ddot{e}_3 + (l_{32} - \kappa_2)\dot{e}_3 + (l_{42} - \kappa_2 l_{32} - \kappa_3 l_{12})e_3 + (l_{31} - \kappa_3)\dot{e}_1 + (l_{41} - \kappa_2 l_{31} - \kappa_3 l_{11})e_1 \end{cases}$$

Comme mentionné précédemment, le problème de contrôle exact de la position du robot (x, y) et de l'orientation ψ est transformé en un contrôle pratique de la position du point $Q(s, r)$ qui se trouve à proximité de la centre du repère lié au corps du robot O_b. Pour que le dirigeable suive une trajectoire de référence $\begin{bmatrix} s_{\text{ref}} \\ r_{\text{ref}} \end{bmatrix}$ dans le plan horizontal, un contrôleur basé sur la compensation de perturbation est conçu (Théorème 4.2):

$$U = \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} -\Delta_{c1} - \hat{d}_s - K_p e_s - K_d \dot{e}_s + \ddot{s}_{\text{ref}} \\ -\Delta_{c2} - \hat{d}_r - K_p e_r - K_d \dot{e}_r + \ddot{r}_{\text{ref}} \end{bmatrix}$$

dont le terme d'erreur est:

$$e = \begin{bmatrix} e_s \\ e_r \end{bmatrix} = \begin{bmatrix} s - s_{\text{ref}} \\ r - r_{\text{ref}} \end{bmatrix}$$

De plus, une méthode pour concevoir la trajectoire de référence en utilisant des polynômes de temps est proposée.

En fin, des simulations de suivi de trajectoire et de stabilisation en un point sont effectuées pour confirmer les performances du contrôleur basé sur la compensation de perturbation de mouvement planaire.

La mise en œuvre

Finalement, la présentation de la partie pratique de ce travail vient au Chapitre 5, nous résumons tous les travaux théoriques précédents sur la modélisation du dirigeable, la conception de contrôleurs pour le mouvement d'altitude et le mouvement planaire du robot, et nous les implémentons sur le robot développé par nos soins.

Premièrement, nous nous concentrons sur une application intérieure possible

qui est la surveillance à long terme, les fonctionnalités nécessaires du système de robot dirigeable sont analysées. En fonction de l'analyse, nous avons proposé une division du système et décidé de suivre une procédure de conception modulaire pour chaque sous-systèmes.

Pour le matériel du robot dirigeable, il a passé avec succès deux générations de conception et de création, nous avons présenté brièvement la première génération - le prototype du robot dirigeable NON-A, suivi par l'introduction détaillée sur le V2 du robot dirigeable NON-A, y compris la conception du circuit électrique de la carte de commande et la conception de la structure du robot (Voir Figure 5.4, 5.5, 5.6, 5.7, 5.8 et 5.9).

Ensuite, basé sur des expériences acquises lors des tests, nous avons décidé de mettre en œuvre le système de caméra de capture de mouvement (OptiTrack) dans notre système de robot dirigeable. Le schéma de l'implémentation d'OptiTrack dans le système de robot est présenté, les avantages et les inconvénients sont analysés.

Après, nous voulons tester les contrôleurs de mouvement conçus dans Chapitre 3 et Chapitre 4. Des tests de stabilisation d'altitude sont effectués à la fois sur le robot prototype et sur le robot V2, les résultats satisfaisants prouvent l'efficacité et la robustesse du contrôleur d'altitude basé sur prédicteur avec compensation des perturbations sur le robot réel.

Enfin, le contrôleur de mouvement planaire est combiné avec le contrôleur d'altitude et ils forment le contrôleur de mouvement complet qui est validé par des tests de stabilisation en un point et de suivi de trajectoire.

Les vidéos sont prises pour montrer plus facilement les résultats obtenus et les performances des contrôleurs, ce qui prouve que le robot dirigeable développé est capable de réaliser des opérations en intérieur, telles que la surveillance à long terme, en se stabilisant à une position fixe ou en suivant une trajectoire prédéfinie.

En outre, le V2 robot dirigeable NON-A créé peut servir en tant que plateforme de recherche scientifique et d'éducation ou être utilisé dans d'autres applications intérieures grâce aux interfaces et aux ressources informatiques réservées sur la carte de contrôle.

Development of a blimp robot for indoor operation

Abstract

Recently, the blimp robot has attracted more and more attentions of the researchers for its advantages compared to other aircrafts, such as ability for VTOL, stationary and low speed flight, long endurance in air and safe Human-Robot interaction, etc. Therefore it is an ideal platform for various indoor applications. In this thesis, we study the modeling and motion control of an indoor blimp robot, and develop a real robot for indoor operations such as the long-term surveillance. The work is composed of both theoretical and practical parts. For the theoretical part, first, under reasonable assumptions, the 6-DOF dynamic model is simplified and divided into two independent parts: the altitude motion and the horizontal plane movement. Then, to ensure the accuracy of modeling and control, the nominal model is complemented with disturbance terms which are estimated in real-time and compensated in the designed controllers. Simulations are carried out to verify the performance and robustness of the controllers. For the practical part of the work, based on the functionality analysis of the robot to achieve desired indoor applications, the hardware of the blimp robot is conceived and created. Finally, real tests are made on the blimp robot platform for the validation of the designed motion control laws, and satisfying results are obtained.

Keywords: blimp robot, navigation, estimation, uncertainty compensation, robust control

---

Développement d'un robot dirigeable pour opération en intérieur

Résumé

Récemment, le robot dirigeable a attiré l'attention de plus en plus des chercheurs grâce à ses avantages par rapport à d'autres aéronefs, tels que la capacité de VTOL, le vol stationnaire et à basse vitesse, une grande autonomie, et une interaction Homme-Robot sûre, etc. Ainsi c'est une plate-forme idéale pour diverses applications d'intérieur. Dans cette thèse, nous étudions la modélisation et le contrôle du mouvement d'un robot dirigeable d'intérieur et développons un prototype pour les opérations intérieures comme la surveillance. Le travail est composé de parties théoriques et pratiques. Concernant la partie théorique, d'abord, sous des hypothèses raisonnables, le modèle dynamique à 6-DOF est simplifié et divisé en deux parties indépendantes : le mouvement de l'altitude et le mouvement dans le plan horizontal. Ensuite, à fin d'assurer la précision de la modélisation et du contrôle, le modèle nominal est complété par des termes de perturbation qui sont estimés en temps réel et compensés dans les contrôleurs conçus. Des simulations sont effectuées pour vérifier les performances et la robustesse des contrôleurs. Pour la partie pratique du travail, basée sur l'analyse des fonctionnalités du robot afin de réaliser les applications intérieures souhaitées, le matériel du robot dirigeable est conçu et créé. Enfin, de vrais tests sont effectués sur la plate-forme de robot dirigeable pour la validation des lois de contrôle de mouvement conçues, et des résultats satisfaisants sont obtenus.

Mots clés : robot dirigeable, navigation, estimation, compensation d'incertitude, contrôle robuste