



**HAL**  
open science

## Quadrotor analysis and model free control with comparisons

Jing Wang

► **To cite this version:**

Jing Wang. Quadrotor analysis and model free control with comparisons. Other [cond-mat.other]. Université Paris Sud - Paris XI, 2013. English. NNT : 2013PA112289 . tel-00952401

**HAL Id: tel-00952401**

**<https://theses.hal.science/tel-00952401>**

Submitted on 29 Apr 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ÉCOLE DOCTORALE: STITS

LABORATOIRE DES SIGNAUX ET SYSTÈMES (L2S)

DISCIPLINE: AUTOMATIQUE

THÈSE DE DOCTORAT

soutenue le 25/11/2013

par

**Jing WANG**

---

## Analyse et Commande Sans Modèle de Quadrotors Avec Comparaisons

---

<b>Directeur de thèse:</b>	Hugues MOUNIER	Professeur (L2S)
<b>Co-directeur de thèse:</b>	Arben CELA	Professeur (ESIEE)
	Silviu-Iulian NICULESCU	Directeur de recherche (L2S)
<b>Jury:</b>		
<b>Rapporteur:</b>	Nicolas MARCHAND	Directeur de recherche (GIPSA)
	Joachim RUDOLPH	Professeur (Saarland Univ.)
<b>Examineur:</b>	Cédric JOIN	Maître de conférences HDR (CRAN)
	Françoise LAMNABHI-LAGARRIGUE	Directeur de recherche (L2S)
	Karim TRABELSI	Enseignant-chercheur (IPSA)
<b>Invité:</b>	Jean-Pierre RIVÈRE	Ingénieur R&D (IPSA)



Gif-sur-Yvette, France



# Acknowledgements

I need hereby to thank many peoples, without whom this dissertation could not have been finished.

Firstly, I would like to thank my supervisors Arben Cela, Hugues Mounier, Silviu-Iulian Niculescu, Karim Trabelsi for their solid scientific support. They have led me in the scientific researches, taught me how to have a comprehensive thinking, influenced me to focus on the details, inspired me to always pursue more thorough results and supported me when there were difficulties. During these three years, what I got is not only the knowledge in control theory, but also the strict and perfectionism attitudes which I think will influence the rest of my life.

Secondly, I want to thank my family, my mother Fengying Zhao, my father Liguang Wang and my sister Yan Wang. They have accompanied and supported me every day during these three years. When I encountered difficulties, they were always there to encourage me and help me to pass through these difficult days. My family is forever a motivation for me to do better each day.

Then, I would like to thank all my friends: my office colleagues Marcel-Stefan Geamanu, Bien Hoang, Sarra Jlassi, my lab colleagues Islam Boussaada, Yijing Chen, Yuling Zhen, Fan Huang, Long Chen, Ning Chu, Jing Dai, Ali Elati, and other friends Peiqing Yu, Shengchun Li, Yida Wen, Binsong Ma, etc. They have built a pleasing circumstance for my research during three years, which helped me to be more efficient on my work.

Finally and most importantly, I would like to thank the engineering school IPSA who financially supported my thesis. Their support gives me a priceless opportunity to finish my thesis. Without them, I definitely could not learn so many things during my thesis. Also, I would like to thank Igor Ciril from IPSA for his help during my thesis.



# Abstract

Inspired by the limitations of traditional PID controllers and the different performance in ideal and realistic cases, the existing quadrotors, their applications and control methods have been intensively studied in this dissertation. Many challenges are shown: embedded quadrotor systems have limit computational resources and energy; the aerodynamic dynamics is rather complex and poorly known; environment has many disturbances and uncertainties; many control methods have been proposed in ideal scenarios in literature without comparison. Therefore, this dissertation focuses on these main points in the control of quadrotors.

A kinematic model and a dynamic model are proposed, including all the important aerodynamic forces and moments. A simplified dynamic model is also given based on some applications. Then, the dynamics of quadrotor is analyzed. Using the normal form theory, the model of quadrotor is simplified to a simplest form named the normal form, which exhibits all possible dynamic properties of the original system. The bifurcations of its normal form are then studied, and the system can be further simplified at its bifurcation point using the center manifold theory.

Based on the research of the applications in the first chapter, five typical realistic scenarios are proposed: an ideal case, the cases with wind disturbance, parameter uncertainties, sensor noises and actuator faults. These realistic cases can show comprehensively the performance of control methods respect to the ideal cases. An event triggered scheme is also proposed with the time triggered scheme in order to further save computational resources. Then, a newly proposed method the model free control is presented. It is a simple but efficient technique for the nonlinear, unknown or partially known dynamics. A backstepping control and a sliding mode control are also proposed for the sake of comparison.

All the control methods are implemented in the time and event triggered schemes in five different scenarios. In order to keep closer to realistic situations, the control gains of each methods are not changed in different scenarios. Based on the study in the first chapter, ten criteria are chosen for measuring the performance of control methods, such as the maximum

---

absolute tracking error, the error variance, the actuation steps, the energy consumption, etc.

# Résumé

Cette thèse traite de la commande de quadrotors, utilisant des techniques comme la commande sans modèle et la commande déclenchée par événements. Divers problèmes sont abordés: les systèmes embarqués ont des limites des ressources de calcul et d'énergie; la dynamique est assez complexe et souvent mal connue; l'environnement induit beaucoup de perturbations et d'incertitudes; de nombreuses méthodes de contrôle ont été proposées pour des scénarios idéaux dans la littérature sans comparaison entre elles. Cette thèse porte sur ces principaux points en commande de quadrotors.

Des modèles cinématiques et dynamiques sont tout d'abord proposés, y compris toutes les forces et tous les couples aérodynamiques jugés significatifs. Un modèle dynamique simplifié est également proposé en vue de certaines applications. Une dynamique de quadrotor est analysée en utilisant la théorie des formes normales. Le modèle est ainsi transformé en une forme plus simple présentant tous les comportements dynamiques possibles du système d'origine. Les bifurcations de cette forme normale sont étudiées, et le système est simplifié à son point de bifurcation en utilisant la théorie de la variété du centre.

Cinq scénarios réalistes sont proposés: un cas idéal, les cas avec perturbation de vent, avec incertitudes paramétrique, avec bruits de capteurs et avec panne d'actionneur. Ces cas réalistes mettent en exergue les performances des méthodes de contrôle par rapport aux cas idéaux. Un schéma déclenché par événements est également proposé. La technique dite de commande sans modèle est ensuite présentée ; Il s'agit d'une technique aussi simple qu'efficace pour commande de dynamiques non-linéaires, inconnues ou partiellement connues. La commande par backstepping et la commande par mode glissant sont également proposées à des fins de comparaison.

Toutes les méthodes de contrôle sont mises en oeuvre en échantillonnage à pas constant et en schéma déclenché par événements selon cinq scénarios différents. Dix critères sont choisis pour évaluer les performances des méthodes de contrôle, incluant l'erreur maximale absolue de suivi, la variance de l'erreur, le nombre d'actionnement et la consommation d'énergie.

---

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Résumé</b>	<b>v</b>
<b>List of Symbols</b>	<b>xi</b>
<b>List of Acronyms</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xx</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Two Inspiring Examples . . . . .	2
1.2.1 Limitation of the Traditional PID Control . . . . .	3
1.2.2 Difference Between Ideal and Realistic Cases . . . . .	3
1.3 Motivations . . . . .	5
1.4 Existing Quadrotors . . . . .	7
1.5 Quadrotor Applications . . . . .	13
1.6 Existing Quadrotor Models . . . . .	15
1.7 Existing Quadrotor Control Methods . . . . .	16
1.8 Contributions . . . . .	19
1.9 Outline . . . . .	21
<b>2 System Modeling</b>	<b>23</b>
2.1 Introduction . . . . .	23

---

2.2	Preliminary Knowledge . . . . .	23
2.3	Kinematic Model . . . . .	25
2.3.1	Position, Linear Velocity and Acceleration . . . . .	25
2.3.2	Angular Velocity and Acceleration . . . . .	27
2.4	Dynamic Model . . . . .	28
2.4.1	Newton Euler Formalism . . . . .	28
2.4.2	Forces . . . . .	29
2.4.3	Moments . . . . .	33
2.5	Quadrotor Model . . . . .	36
2.5.1	Complete Model . . . . .	36
2.5.2	Simplified Simulation Model . . . . .	38
2.6	Constraints . . . . .	40
2.7	Control Oriented Models . . . . .	41
<b>3</b>	<b>Dynamic Analysis</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	Quadrotor Model . . . . .	44
3.3	Normal Form . . . . .	45
3.4	Linear Analysis . . . . .	49
3.4.1	Controllability . . . . .	49
3.4.2	Stability . . . . .	50
3.5	Nonlinear Analysis . . . . .	52
3.5.1	Bifurcations . . . . .	52
3.5.2	Center Manifold . . . . .	54
<b>4</b>	<b>System Control</b>	<b>57</b>
4.1	Scenarios . . . . .	57
4.1.1	Basic Scenario . . . . .	58
4.1.2	Wind Disturbance . . . . .	59
4.1.3	Model Parameter Uncertainties . . . . .	60
4.1.4	Sensor Noises . . . . .	61
4.1.5	Actuator Faults . . . . .	62
4.1.6	Sensor Faults . . . . .	63
4.2	Control Schemes . . . . .	63
4.2.1	Time Triggered Scheme . . . . .	64

---

4.2.2	Event Triggered Scheme . . . . .	64
4.3	Control Methods . . . . .	65
4.3.1	Model Free Control . . . . .	66
4.3.2	Backstepping Control . . . . .	71
4.3.3	Sliding Mode Control . . . . .	74
4.3.4	PID Control . . . . .	75
<b>5</b>	<b>Simulation Results</b>	<b>77</b>
5.1	Time triggered Control . . . . .	77
5.1.1	Model Free Control . . . . .	78
5.1.2	Backstepping Control . . . . .	82
5.1.3	Sliding Mode Control . . . . .	86
5.1.4	Discussion . . . . .	90
5.2	Event triggered Control . . . . .	94
5.2.1	Model Free Control . . . . .	95
5.2.2	Backstepping Control . . . . .	99
5.2.3	Sliding Mode Control . . . . .	103
5.2.4	Discussion . . . . .	108
5.3	Discussion . . . . .	112
<b>6</b>	<b>Conclusions</b>	<b>117</b>
6.1	Conclusion . . . . .	117
6.2	Future Work . . . . .	119
<b>A</b>	<b>MAPLE Code</b>	<b>121</b>
<b>B</b>	<b>Papers</b>	<b>125</b>
	<b>Bibliography</b>	<b>147</b>

---

# List of Symbols

$\mathbf{P}_w, \mathbf{V}_w, \mathbf{a}_w$	Position, velocity and acceleration in the world frame
$\mathbf{P}_b, \mathbf{V}_b, \mathbf{a}_b$	Position, velocity and acceleration in the body frame
$\phi, \theta, \psi$	Tait-Bryan angles (Euler angles) in the world frame
$p, q, r$	Angular velocities in the body frame
$T_i$	Thrust
$H_i$	Hub force
$M_{roll}$	Rolling moment
$M_{pitch}$	Pitching moment
$Q_i$	Rotor torque (yawing moment)
$R_i$	Blade flapping moment
$M_{gx}, M_{gy}$	Rotor gyroscopic effect moments
$G$	Gravity
$F_A$	Archimedes' force
$V_H$	the sideways velocity of the rotor
$V_1$	the inflow velocity
$\lambda$	the inflow ratio
$\mu$	the advance ratio
$W$	the quadrotor weight
$\rho$	the air density
$V_H$	the sideways velocity of the rotor
$A$	the propeller disk area
$\Omega$	the rotational velocity of the rotor
$R_{rad}$	the propeller radius
$a$	the lift slope
$\theta_0, \theta_{tw}$	the linear lift parameters
$\sigma$	the solidity ratio

---

$N$	the number of blades
$\bar{c}$	the blade chord
$\bar{C}_d$	the drag coefficient
$m$	the mass of the quadrotor
$g$	the gravity coefficient
$l$	the distance from the CoG to the rotor
$V$	the volume of the quadrotor
$J_r$	the moments of inertia of the rotors

# List of Acronyms

A/D, D/A	analog-to-digital, digital-to-analog
CoG	center of gravity
CPU	center processing unit
DC	direct current
GPS	global positioning system
GUI	graphical user interface
LiPo	Lithium polymer battery
LQ	linear quadratic
MAV	micro aerial vehicle
MEMS	micro-electro-mechanical system
NMPC	nonlinear model predictive control
OSP	open source project
PID	proportional-integral-derivative
SAV	small aerial vehicle
UAV	unmanned aerial vehicles
UGV	unmanned ground vehicles
VTOL	vertical take-off and landing
ZOH	zero order hold



# List of Figures

1.1	A PID control of a quadrotor. The trajectory along the $x$ axis (a), along the $y$ axis (b). $x_r, y_r$ the references. (c) The tracking errors along the $x$ and $y$ axis.	3
1.2	A backstepping control with the control gains in Table 1.2. (a) in the ideal case. (b) with wind disturbance.	4
1.3	Gyroplane No.1. (The source: <a href="http://www.terpconnect.umd.edu">www.terpconnect.umd.edu</a> .)	8
1.4	Oehmichen No.2. (The source: <a href="http://www.heli4.com">www.heli4.com</a> .)	8
1.5	Convertawings Model A. (The source: <a href="http://www.aviastar.org">www.aviastar.org</a> .)	8
1.6	Curtiss-Wright VZ-7 (left) and X-19 (right). (The source: <a href="http://aviastar.org">aviastar.org</a> .)	8
1.7	Draganflyer X4-P. (The source: <a href="http://www.draganfly.com">www.draganfly.com</a> .)	9
1.8	4 cell 2700mAh Lithium Polymer battery. (The source: <a href="http://draganfly.com">draganfly.com</a> .)	9
1.9	AscTec (a) Falcon. (b) Pelican. (c) Firefly. (d) Hummingbird. (The source: <a href="http://www.asctec.de">www.asctec.de</a> .)	10
1.10	AR. Drone. (The source: <a href="http://www.ardrone2.parrot.com">www.ardrone2.parrot.com</a> .)	11
1.11	Phantom. (The source: <a href="http://www.dji-innovations.com">www.dji-innovations.com</a> .)	11
1.12	Gauai 330X (left) and 500X (right). (The source: <a href="http://www.gauai.com.tw">www.gauai.com.tw</a> .)	11
1.13	ANU x4 [83].	12
1.14	EPFL OS4 [20].	12
2.1	The Cartesian coordinate system of a quadrotor.	24
2.2	The thrust and hub forces on a rotor.	31
2.3	Rotor gyroscopic effect.	35
3.1	(a) Reference trajectory for the quadrotor. (b) Wind disturbance.	51
3.2	The simulation without wind disturbance: (a) the linear static feedback control. (b) a standard PID control.	51
3.3	The simulation with wind disturbance: (a) the linear static feedback control. (b) a standard PID control.	52

---

3.4	The eigenvalues when $K_{12}$ changes from -150 to 150: (a) the real parts. (b) the imaginary parts. . . . .	53
3.5	The phase portrait of the reduced system: (a) $b=-0.5$ . (b) $b=0$ . (c) $b=0.5$ . .	56
4.1	(a) 3D reference trajectory for the quadrotor. (b) Reference trajectory along the x axis (red, line) and y axis (blue, dashed line). . . . .	58
4.2	Wind disturbance w.r.t time. . . . .	60
4.3	The plot of the function $\text{ssign}(x)$ . . . . .	75
5.1	The model free control in the basic scenario with the gains in Table 5.1. . . .	79
5.2	The model free control with wind disturbance and the gains in Table 5.1. . .	80
5.3	The model free control with parameter uncertainties and the gains in Table 5.1.	80
5.4	The model free control with sensor noises and the gains in Table 5.1. . . . .	81
5.5	The model free control with actuator faults and the gains in Table 5.1. . . .	81
5.6	The Backstepping control in the basic scenario with the gains in Table 5.2. .	83
5.7	The backstepping control with wind disturbance and the gains in Table 5.2. .	84
5.8	The backstepping control with parameter uncertainties and the gains in Table 5.2. . . . .	84
5.9	The backstepping control with sensor noises and the gains in Table 5.2. . . .	85
5.10	The backstepping control with actuator faults and the gains in Table 5.2. . .	85
5.11	The sliding mode control in the basic scenario with the gains in Table 5.3. .	87
5.12	The sliding mode control with wind disturbance and the gains in Table 5.3. .	88
5.13	The sliding mode control with parameter uncertainties and the gains in Table 5.3. . . . .	88
5.14	The sliding mode control with sensor noises and the gains in Table 5.3. . . .	89
5.15	The sliding mode control with actuator faults and the gains in Table 5.3. . .	89
5.16	Time triggered three methods in all scenarios: (a) the maximum absolute tracking error. (b) the error variance. (c) the energy consumption $\times 10^{-7}$ . . .	91
5.17	The model free control in the basic scenario with the gains in Table 5.1. . . .	97
5.18	The model free control with wind disturbance and the gains in Table 5.1. . .	97
5.19	The model free control with parameter uncertainties and the gains in Table 5.1.	98
5.20	The model free control with sensor noises and the gains in Table 5.1. . . . .	98
5.21	The model free control with actuator faults and the gains in Table 5.1. . . .	99
5.22	The backstepping control in the basic scenario with the gains in Table 5.2. .	101
5.23	The backstepping control with wind disturbance and the gains in Table 5.2. .	101

---

5.24	The backstepping control with parameter uncertainties and the gains in Table 5.2. . . . .	102
5.25	The backstepping control with sensor noises and the gains in Table 5.2. . . . .	102
5.26	The backstepping control with actuator faults and the gains in Table 5.2. . . . .	103
5.27	The sliding mode control in the basic scenario with the gains in Table 5.3. . . . .	105
5.28	The sliding mode control with wind disturbance and the gains in Table 5.3. . . . .	106
5.29	The sliding mode control with parameter uncertainties and the gains in Table 5.3. . . . .	106
5.30	The sliding mode control with sensor noises and the gains in Table 5.3. . . . .	107
5.31	The sliding mode control with actuator faults and the gains in Table 5.3. . . . .	107
5.32	Event triggered three methods in all scenarios: (a) the maximum absolute tracking error. (b) the error variance. (c) the actuation steps. (d) the energy consumption $\times 10^{-7}$ . . . . .	109
5.33	The comparison of the control methods in the time triggered scheme. . . . .	113
5.34	The comparison of the control methods in the event triggered scheme. . . . .	113
5.35	The maximum absolute tracking error of all the control methods in all scenarios. . . . .	114
5.36	The sum of the error variances of all the control methods in all scenarios. . . . .	115
5.37	The actuation steps of all the control methods in all scenarios. . . . .	115
5.38	The energy consumption $\times 10^{-7}$ of all the control methods in all scenarios. . . . .	116



# List of Tables

1.1	The PID control gains . . . . .	3
1.2	The backstepping control gains in the ideal case . . . . .	4
2.1	The notations used in the kinematic and dynamic models . . . . .	25
2.2	The external forces applied on a quadrotor. . . . .	29
2.3	The variables in the forces and moments. . . . .	30
2.4	The external moments applied on a quadrotor. . . . .	33
5.1	The model free control gains . . . . .	78
5.2	The backstepping control gains . . . . .	82
5.3	The sliding mode control gains . . . . .	86
5.4	The comparisons of the time triggered control methods in the basic scenario.	92
5.5	The comparisons of the time triggered control methods with wind disturbance. n is n-digit number. . . . .	92
5.6	The comparisons of the time triggered control methods with parameter un- certainties. n is n-digit number. . . . .	93
5.7	The comparisons of the time triggered control methods with sensor noises. n is n-digit number. . . . .	93
5.8	The comparisons of the time triggered control methods with actuator faults. n is n-digit number. . . . .	93
5.9	The event triggered model free control gains . . . . .	95
5.10	The backstepping control gains . . . . .	100
5.11	The sliding mode control gains . . . . .	104
5.12	The comparisons of the event triggered control methods in the basic scenario. n is n-digit number. . . . .	110
5.13	The comparisons of the event triggered control methods with wind distur- bance. n is n-digit number. . . . .	110

---

5.14	The comparisons of the event triggered control methods with parameter uncertainties. $n$ is $n$ -digit number. . . . .	111
5.15	The comparisons of the event triggered control methods with sensor noises. $n$ is $n$ -digit number. . . . .	111
5.16	The comparisons of the event triggered control methods with actuator faults. $n$ is $n$ -digit number. . . . .	111
5.17	The maximum absolute tracking errors of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control. . . . .	114
5.18	The sum of the error variances of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control. . . .	114
5.19	The actuation steps of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control. . . . .	115
5.20	The energy consumption of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control. . . .	116

# Chapter 1

## Introduction

### 1.1 Background

Nowadays, unmanned aerial vehicles (UAVs) are intensively studied in military and civilian applications [87]. An unmanned aerial vehicle is an aircraft without a human pilot, which is autonomous or piloted remotely. Depending on the applications, UAVs have various shapes, sizes and configurations. Their sizes range from full-scale aircraft to Micro-Electro-Mechanical System (MEMS). The number of the motors varies from two to four or even more. The configurations are from simple microprocessors to systems with embedded vision, GPS navigation, wireless communication, etc. Due to the limitations of unmanned ground vehicles (UGVs), UAVs draw a great attention in research and industry. UGVs highly depend on surface conditions. If the ground is grass or water, the reliability of UGVs is reduced. UAVs fly in the air and have no contact with ground. This gives UAVs a larger mobility especially to the places where UGVs are not accessible. As conventional aircrafts and pilots are usually expensive and the flight area is often limited, UAVs are a better alternative to manned aerial vehicles.

UAVs have many applications, such as border patrol, surveillance, search and rescue, mapping, communication network, aerial photography, etc. Without human pilots, UAVs are especially suited for dangerous or monotonous missions. In military applications, UAVs are used as reconnaissance and surveillance aircrafts. The development of the high density power storage, especially electrical batteries, makes the civilian application of UAVs realistic. The size of UAVs is reduced, which means less production materials, cost and energy consumption. It makes UAVs accessible for civilian applications. Small aerial Vehicles (SAVs) and micro aerial vehicles (MAVs) are the principal focus in UAV studies, particularly in

civilian applications.

In the last decades, quadrotors became popular in UAV researches due to their simplicity in design and maneuver. A quadrotor, also called a quadrotor helicopter or quadcopter, is a vertical take-off and landing (VTOL) aircraft lifted and propelled by four rotors. It has a cross structure with one rotor at each arm (namely front, back, left and right). The front and back rotors rotate in the other direction with respect to the left and right rotors. A quadrotor is lifted by the sum of the thrusts of each rotor, and propelled by the differences of the thrusts. The major advantage of quadrotors is its simplicity in design and its feasibility in maneuver. Unlike flapping wing aircrafts, quadrotors have rather simple structures. The production cost is cheap, which makes them available in civilian applications and scientific researches. Unlike fixed wing vehicles need minimum velocity to fly, quadrotors are capable to hover, which enlarges its applications. Quadrotors have relative simple dynamics, and is controlled only by the rotational speeds of the four rotors. This makes quadrotors a good workbench for further researches, such as vision, mapping, wireless communication, sensor networks, etc.

## 1.2 Two Inspiring Examples

In this section, two examples are exposed. In the first example, the limitation of the traditional PID control is presented. As widely known, classical PID controllers have simple control laws and do not necessarily need specific knowledge of the system. This is one of the reasons of its popularity in process control [1, 78]. Although traditional PID controllers are widely used in industrial control, they show poor performance for several classes of nonlinear systems, such as the quadrotor system. Therefore, the studies on superior control methods for the quadrotor system are needed.

In the second example, the different performances of a control system in an ideal case and in a realistic case are pointed out. Usually, when a new control method is proposed, some simulations are made in an ideal case to prove the good performance of this method. The control gains are usually high to get a better performance. In ideal cases, the simulation results are good with small tracking errors. However, the close loop systems with high gains lose its stability quite quickly in realistic cases with disturbances. Therefore, it appears that the comparisons of different control methods in ideal cases are not enough. We can not say one control method has a better performance than another simply according to the performances in ideal cases.

### 1.2.1 Limitation of the Traditional PID Control

Traditional PID controllers are widely used in industrial controls. However, when the system is highly nonlinear, time varying with multiple states, PID controllers show poor performances. Here, we implement a traditional PID control on the quadrotor model which is proposed in Chapter 2 (see Equation (3.2)). The control laws are described in Subsection 4.3.4. The selected control gains are shown in Table 1.1.

Table 1.1: The PID control gains

$k_p^z$	1	$k_d^z$	0.4	$k_i^z$	0.05	$k_p^x$	0.2	$k_d^x$	0.25	$k_i^x$	0.001
$k_p^y$	0.2	$k_d^y$	0.25	$k_i^y$	0.001	$k_p^\phi$	1.2	$k_d^\phi$	0.5	$k_i^\phi$	0.5
$k_p^\theta$	1.2	$k_d^\theta$	0.5	$k_i^\theta$	0.5	$k_p^\psi$	1.2	$k_d^\psi$	0.5	$k_i^\psi$	0.5

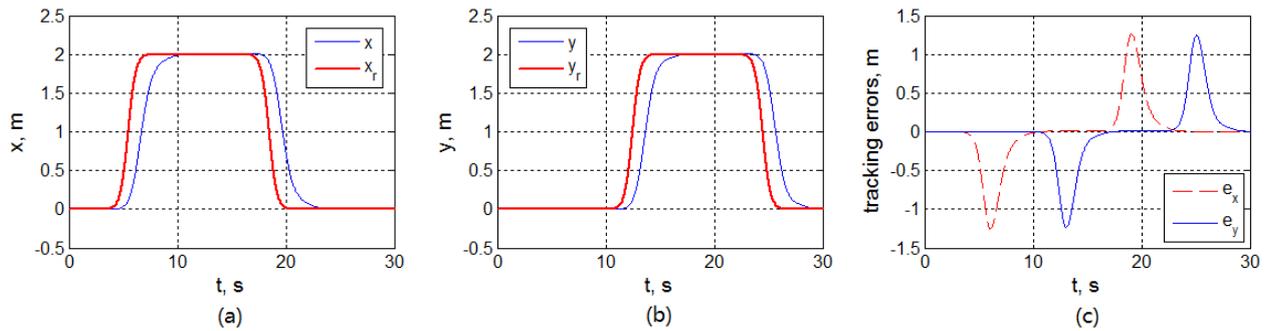


Figure 1.1: A PID control of a quadrotor. The trajectory along the  $x$  axis (a), along the  $y$  axis (b).  $x_r$ ,  $y_r$  the references. (c) The tracking errors along the  $x$  and  $y$  axis.

The simulation results along the axis  $x$  and  $y$  are shown in Figure 1.1 (a) and (b). The values  $x_r$  and  $y_r$  are the references. We observe that the traditional PID control has poor performance on this nonlinear model. It has a slow response time with respect to the desired performance, which gives noticeable tracking errors as shown in Figure 1.1 (c). The maximum absolute tracking error is 1.5m, which is 75% of the fly distance 2m. We can see this traditional PID controller does not suffice in terms of response time for this class of nonlinear system. Therefore, the studies on the control methods for the quadrotor system are necessary.

### 1.2.2 Difference Between Ideal and Realistic Cases

In the literature, control methods are usually simulated or tested in ideal cases without any disturbance. In order to get a better result, high control gains are usually chosen. In realistic

cases, there are many disturbances and uncertainties. A control system with high control gains, which is more vulnerable to changes, may lose its stability in realistic cases. Therefore, the performance in ideal cases can not fully show its features of a control method. It should be tested in different realistic cases to show its advantages and disadvantages.

Here, a backstepping control proposed in Chapter 4 is firstly simulated in an ideal case without disturbances and uncertainties. The chosen control gains are in Table 1.2, which are selected to obtain the smallest maximum absolute tracking error in the ideal case. The tracking errors along the  $x$ ,  $y$  and  $z$  axis are shown in Figure 1.2 (a). In the ideal case, the system shows its stability during the simulation. Used these control gains, the backstepping control has the smallest maximum absolute tracking error 0.009m, which is only 0.45% of the fly distance 2m.

Table 1.2: The backstepping control gains in the ideal case

$\alpha_1$	4	$\alpha_2$	10	$\alpha_3$	4	$\alpha_4$	10	$\alpha_5$	1	$\alpha_6$	5
$\alpha_7$	10	$\alpha_8$	60	$\alpha_9$	4	$\alpha_{10}$	10	$\alpha_{11}$	4	$\alpha_{12}$	10

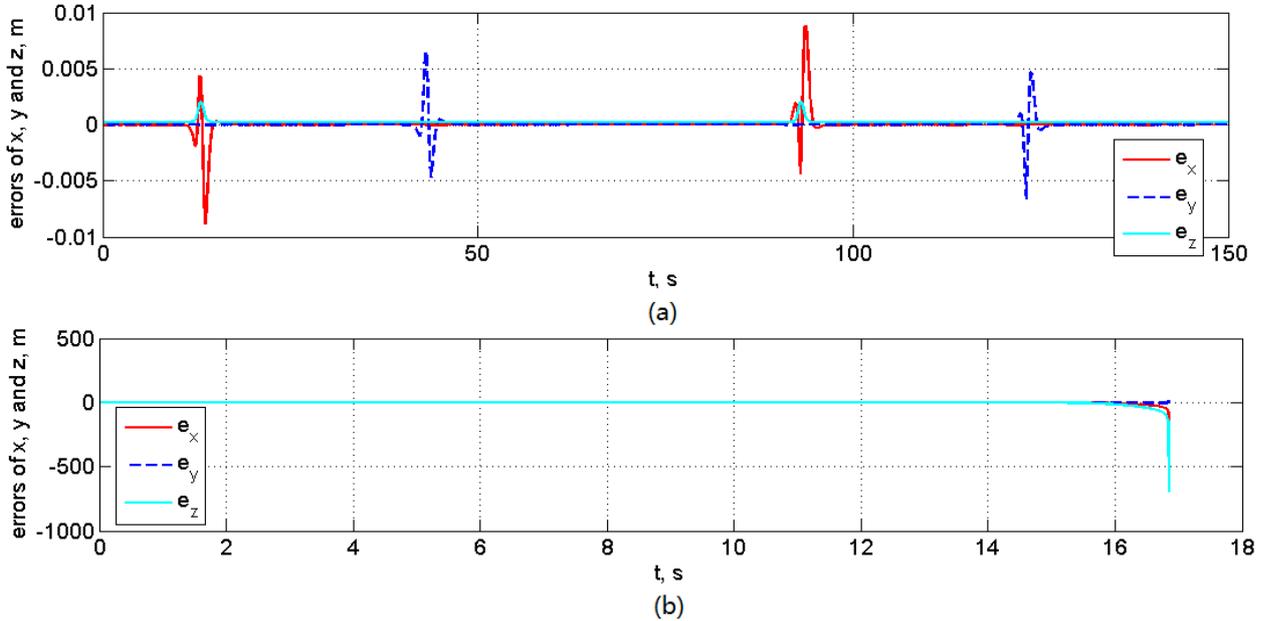


Figure 1.2: A backstepping control with the control gains in Table 1.2. (a) in the ideal case. (b) with wind disturbance.

When wind disturbances occur, such as the one proposed in Chapter 4, the system loses its stability quickly using the same gains in Table 1.2, as shown in Figure 1.2 (b). The results

in the ideal case are not enough to show comprehensively the performance of the control method. Therefore, control methods should be tested in realistic cases with disturbances and uncertainties to show their features.

## 1.3 Motivations

In the last decades, quadrotors have been intensively studied. However, there are still many challenges in the control of a quadrotor.

### **(1) Modeling: simple for computation and close to the real dynamics of quadrotors.**

The nonlinear aerodynamics of quadrotors is rather complex. The aerodynamic forces, the thrust and drag force, are usually modeled as a function of the square of the rotor speed, and the coefficients in these functions are changing while the movements of the quadrotor. The drag forces can be classified as induced drag, translational drag, profile drag and parasitic drag, which are changing with the change of the airflow velocity. The changes of the airflow cause many aerodynamic effects, such as the balding flapping, ground effect and vertical descent effect [16]. These make the mathematical modeling imprecise under certain situations. The aerodynamics is difficult to be modeled in mathematical expressions.

Several dynamic models have been proposed in the literature. Altug et al. have proposed a simple dynamic model in 2002 [4]. In this simple model, only the important aerodynamic thrust force and yawing moment are considered. All other aerodynamic effects are neglected. This model is simple for computation, however it has a bigger difference to the real dynamics than other models. Then, a more realistic model is proposed by Bouabdallah in 2007 [20]. The aerodynamic forces and moments are modeled as functions of the square of the rotor speeds. The parameters are evaluated in static tests, and considered to be constants during the flights. Actually, the parameters are variables and changing during the applications. In this model, some secondary aerodynamic effects are neglected, such as the blade flapping, induced drag, ground effect, etc. Purwin et al. have proposed an iterative learning technique for compensating the errors of current dynamic models in 2009 [86]. However, this may take too many computational resources, as quadrotor systems are usually embedded systems.

Therefore, finding a dynamic model which is simple for computation and at the same time close to the real dynamics is always a challenge in the quadrotor researches.

**(2) Control method: low algorithm complexity and strong stability.**

As in the first inspiring example, the traditional PID has poor performance in the control of a quadrotor. Therefore, a control method with more system information is needed for the quadrotor control.

The complexity of the control algorithm should be rather low. The embedded processors in a quadrotor usually have limited processing speed and computational resources. In the applications, quadrotors need to execute many other tasks: photo shooting, target detection, object delivery, etc. There are other systems which need many computational time and resources. Therefore, the motion control system should be rather simple and use only a small portion of the computational resources.

The control method should also be capable to compensate the model errors. As mentioned above, the aerodynamic effects are hard to be expressed precisely in the control model. The modeling errors always exist. Moreover, due to the computational limits of the embedded processors, the control model sometime is further simplified in order to get a simpler control law. This will further increase the modeling errors. In addition, the parameters of the quadrotor system are measured by static tests or some softwares. There may be parameter uncertainties too. Therefore, the control method should be rather stable to compensate these model and parameter errors.

In realistic scenarios, there are many disturbances. Low cost and small sized Inertial Measurement Units (IMUs) are less accurate and have noises and drifts. The environment produces unexpected disturbances, such as wind disturbance. The control method should be able to compensate these disturbances, and keep the stability of the system with less tracking errors and vibrations possible.

Therefore, finding a control method which has a low algorithm complexity and which is capable to keep a good performance during disturbances is always a challenge in the quadrotor researches.

**(3) Comparison of different control methods: in realistic scenarios with disturbances and uncertainties.**

Many control methods have been proposed for the quadrotor system in the literature. The controls using Lyapunov theory have been proposed by Castillo et al., Bouabdallah et al.. The Backstepping controls have been used by Altuğ et al., Bouabdallah et al., Castillo et al., Madani et al.. The Sliding mode controls have been proposed by Hoffmann et al., Waslander et al., Bouabdallah et al.. A dynamic feedback technique to linearize the system

has been proposed by Mistler et al.. A mixed robust feedback linearization with linear  $\text{GH}_\infty$  controller has been applied by Mokhtari et al.. An adaptive control has been proposed by Guenard et al.. Further detail about these control methods can be found in Section 1.7 of Chapter 1.

However, these methods were usually implemented in ideal scenarios. As shown in the second inspiring example, the good performance in ideal scenarios can not ensure the same good performance in realistic scenarios. In the literature, control methods are usually tested in ideal scenarios. These results can not comprehensively show the features of a control method. The performance of these control methods in realistic scenarios are rarely proposed. Quadrotors have many applications (see Section 1.5 of Chapter 1 for further details). In these applications, there are a lot of disturbances and uncertainties. Without knowing the performance of a control method in realistic scenarios, it is hard to foretell its performance in these applications.

The comparisons among several control methods were rarely proposed in the literature. Many control methods have been proposed in different ideal scenarios. It is difficult to compare their performance and further know their advantages and disadvantages. In applications, different tasks have different system demands: the search & rescue task demands a fast system response time; the aerial photography task demands less system vibration, etc. Without knowing the advantage of each control method, it is hard to choose an appropriate method for a specific application.

Therefore, the comparisons of different control methods in realistic scenarios are needed to choose an appropriate method for a certain application.

## 1.4 Existing Quadrotors

The ‘Gyroplane No.1’, one of the earliest manned quadrotors, is built by Louis and Jacques Breguet with the help of Professor Charles Richet in 1907 in France (see Figure 1.3). It was not a free helicopter, neither controllable nor steerable. But it was the first time a rotary-wing device had lifted itself and a pilot into the air [14].

In 1920s in France, Étienne Oehmichen has designed several quadrotors [79]. The ‘Oehmichen No.2’ is probably the first reliable manned helicopter (see Figure 1.4).

The american George de Bothezat has built ‘Convertawings Model A’ (see Figure 1.5), and the first flight took place in 1956. It was also the first four-rotor helicopter which demonstrated successful forward flight [29].

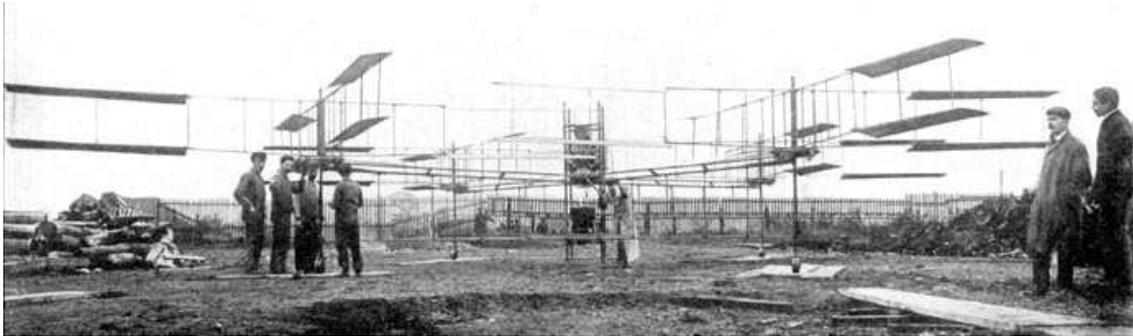


Figure 1.3: Gyroplane No.1. (The source: [www.terpconnect.umd.edu](http://www.terpconnect.umd.edu).)



Figure 1.4: Oehmichen No.2. (The source: [www.heli4.com](http://www.heli4.com).)



Figure 1.5: Convertawings Model A. (The source: [www.aviastar.org](http://www.aviastar.org).)

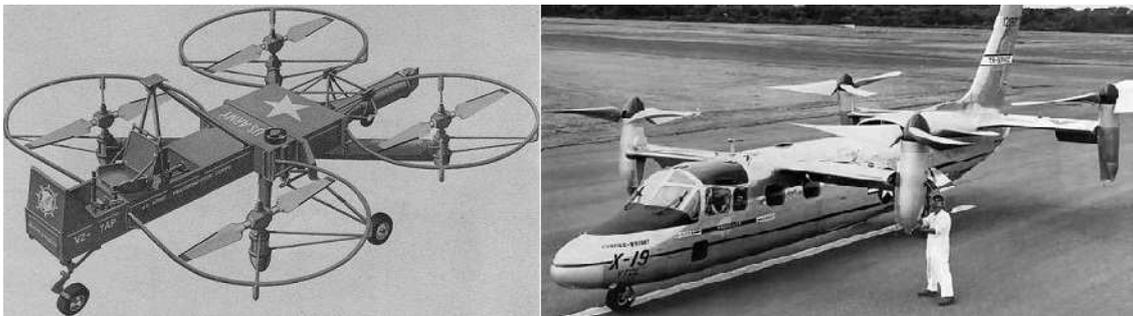


Figure 1.6: Curtiss-Wright VZ-7 (left) and X-19 (right). (The source: [aviastar.org](http://aviastar.org).)

The ‘Curtiss-Wright VZ-7’ was designed by the company Curtiss-Wright for the US Army in 1958 (see Figure 1.6). It was controlled by changing the thrust of the four propellers [30]. In 1963, Curtiss-Wright Corporation developed the ‘X-19’ for the United States Air Force (see Figure 1.6). However it was destroyed in a crash in 1965, and the program of X-19 was subsequently cancelled [31].

In the last decades, various micro unmanned aerial vehicles were designed for scientific researches and civilian applications. There are many successfully commercialized quadrotors,



Figure 1.7: Draganflyer X4-P.  
(The source: [www.draganfly.com](http://www.draganfly.com).)



Figure 1.8: 4 cell 2700mAh Lithium Polymer battery. (The source: [draganfly.com](http://draganfly.com).)

such as Draganflyer, AscTec, AR.Drone, DJI Wookong, Gaii Quad flyer, etc.

Draganflyer Innovations Inc has designed several ‘Draganflyer’ quadrotors since 1998 (see Figure 1.7). There are several types of quadrotor: Draganflyer X4, X6 and X8. They are built using rugged carbon fiber, glass filled injected nylon parts and brushless electric DC motors. The dimension of the quadrotors is about 70cm width, 70cm length and 25cm height. The weight with battery varies from 680g to 1700g, and the payload capability is from 250g to 800g. The onboard battery is usually the Lithium Polymer (LiPo) battery with a maximum voltage 14.8V (see Figure 1.8). The capacity of the batteries varies from 2700mAh to 5400mAh, which enables a quadrotor fly approximately 20 to 30 minutes on a charge. The Draganflyer X4 is equipped seven sensors: three gyroscopes, three accelerometers and one barometric pressure sensor. The Draganflyer X8 is equipped eleven sensors: three gyroscopes, three accelerometers, three magnetometers, one barometric pressure sensor and one GPS receiver. All of these sensors are constantly feeding data to the onboard flight computer. An onboard camera and its corresponding software ‘DraganView’ are also provided. Using the data from the sensors, an onboard software ‘SteadyFlight’ can automatically adjust the flight control and maintain the stability. The control method used in ‘SteadyFlight’ is not available to public. The Draganflyer quadrotors can also be controlled by a handheld flight controller. The Draganflyer quadrotors are used in many applications, such as public safety services, industrial maintenance and inspection, photography and videography, etc. Moreover, they are used in scientific researches in many universities, such as MIT, University of Maryland, Vanderbilt University [39].

Ascending Technologies also proposes several quadrotors and multicopters since 1998 [2]. The AscTec Flacon is designed for professional image and video recordings. The AscTec Firefly, Pelican and Hummingbird are designed for different scientific researches (see Figure 1.9). The Pelican is designed with a maximized power and plenty of space for further devel-

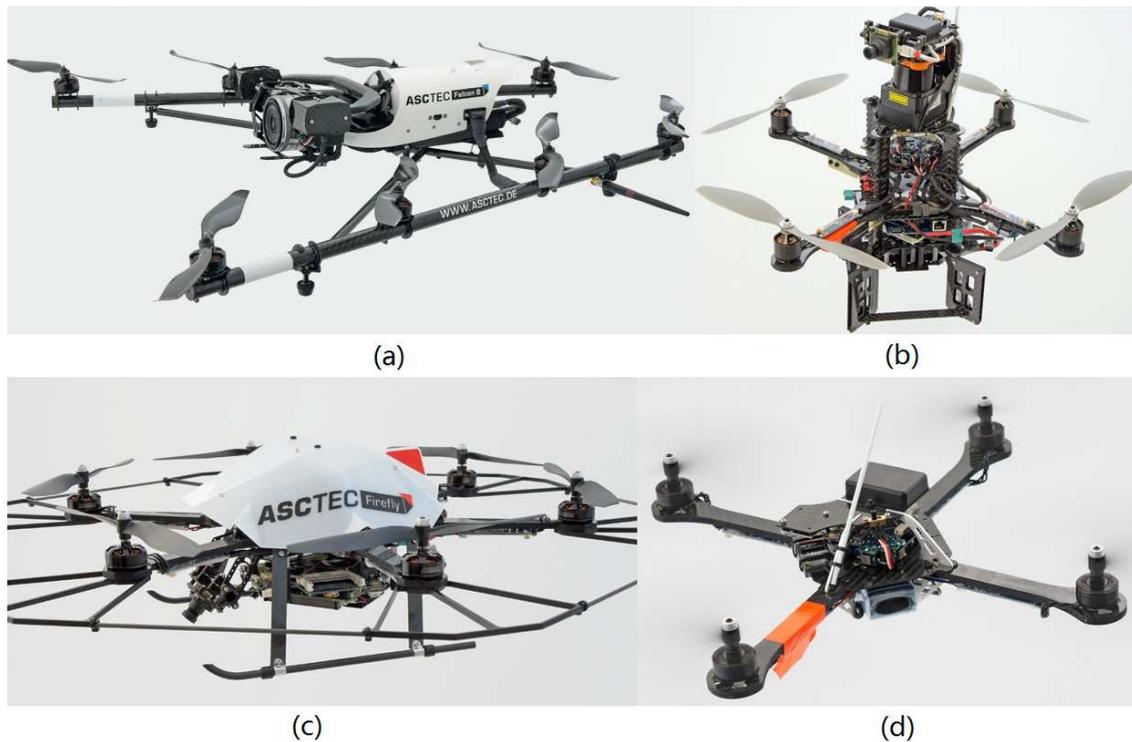


Figure 1.9: AscTec (a) Falcon. (b) Pelican. (c) Firefly. (d) Hummingbird. (The source: [www.ascotec.de](http://www.ascotec.de).)

opment. The Hummingbird is designed for aggressive and fast flight maneuvers. Its robust frame tolerates hard landings and the simple structure makes it easy to repair. The weights of Pelican and Hummingbird with battery are 510g to 630g respectively, and the maximum payloads are 200g and 650g. The Hummingbird has four HACKER X-BL 52s brushless DC motors and a 3 cells 11.1V 2100mAh Lithium Polymer (LiPo) battery which enables the quadrotor fly about 20 minutes. The Hummingbird has two microcontrollers ARM 9 and is equipped nine sensors: three gyroscopes, three accelerometers, one 3D magnetometer, one barometric pressure sensor and one GPS receiver. The quadrotors can be controlled by a handheld flight controller. The AscTec quadrotors are widely used in scientific researches, such as the University of Pennsylvania. They provide an user program which runs at 1kHz to let users to build their own motion control algorithm.

The AR. Drone is produced by Parrot SA since 2010 (see Figure 1.10). It equips two onboard HD cameras, and sends real-time images to the controller via Wi-Fi. It is a well developed quadrotor toy which is remotely controlled through a user-friendly graphical interface application on iphone, iTouch or ipad [3]. AR. Drone, full name Augmented Reality Drone, is designed for video games and home entertainment. The price is less than 300 euros. It equips a 3 cells 11.1V 1000mAh Lithium Polymer (LiPo) battery which enables a



Figure 1.10: AR. Drone.  
(The source: [www.ar-drone2.parrot.com](http://www.ar-drone2.parrot.com).)



Figure 1.11: Phantom.  
(The source: [www.dji-innovations.com](http://www.dji-innovations.com).)



Figure 1.12: Gaudi 330X (left) and 500X (right). (The source: [www.gaudi.com.tw](http://www.gaudi.com.tw).)

quadrotor fly about 10 to 15 minutes. The system has two cameras, three gyroscopes, three accelerometers, one sonar and two microcontrollers. In 2011, Bristeau et al. have exposed the inside navigation and control system in their paper [15]. The system is controlled by several PID controllers.

DJI Innovations also produces quadrotors, multicopters and autopilot systems for commercial and recreational use. The quadrotor Phantom has a dimension 35cm width, 35cm length and 19cm height (see Figure 1.11). It equips a 3 cells 2200mAh Lithium Polymer (LiPo) battery which enables a quadrotor fly approximately 10 to 15 minutes. The maximum tilt angle of the Phantom is 45 degrees. The maximum ascent or descent speed is 6m/s, and the maximum flight velocity is 10m/s. The Phantom has an integrated flight dynamics control system, Naza-M+GPS autopilot system, to ensure the stability of the system. The control method is not available to public. It can also be controlled by a remote controller.

TSH Gaudi Corporation is dedicated to produce helicopters and multicopters since 1996. It has released the quadrotors 330X and 500X in 2010 (see Figure 1.12). The quadrotors have a dimension 33 cm width and length, and the maximum payload is from 700g to 2200g.

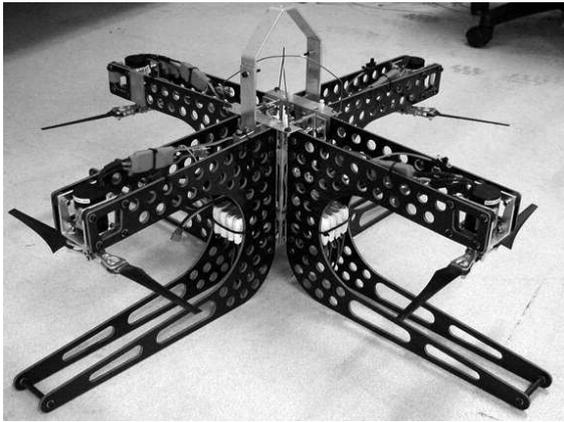


Figure 1.13: ANU x4 [83].



Figure 1.14: EPFL OS4 [20].

It has a 2 cells 2000mAh Lithium Polymer (LiPo) battery, and the flight duration is about 12 minutes. It has a three axis stabilizing system GU-344 to ensure the stability during the flight. The control method is not available to public.

Many open-source projects (OSPs) are also developed on quadrotors, such as Arducopter [12], Aeroquad [13], Openpilot [80], Paparazzi [81], Pixhawk [82], Mikrokopter [75], KK-multicopter [65], Multiwii [76], etc. OSPs use community-hosting sites to generate code or schematics freely. Arducopter provides a graphical-user-interface (GUI) based software to tune control gains and display flight information. Openpilot has a real-time operating system modified from FreeRTOS and a GUI software which is similar to Arducopter. Paparazzi provides nine different autopilot hardware systems which are developed by ENAC university and a GUI based GCS with flight path scripting. Pixhawk project has computer vision equipment developed by ETHZ computer vision group. Mikrokopter is provided by HiSystem GmbH in 2006. It has a GUI based software for gain tuning and health monitoring.

Many universities have also developed their own quadrotors. The Australian National University has built ‘X-4’ since 2004 (see Figure 1.13). The X-4 is much heavier than other quadrotors. It weighs 4.34kg with battery and the maximum payload is 1kg. It comprises a chassis made from aluminium and carbon-fibre and JETI Phasor 30/3 brushless motors. It has a 4 cells 14.8V 2000mAh Lithium Polymer (LiPo) battery which enables the quadrotor fly about 11 minutes. Further details about X-4 can be found in Pounds’s thesis [83]. The control methods used on ‘X-4’ can be found in Section 1.7.

EPFL (Ecole Polytechnique Fédérale de Lausanne) has designed ‘OS4’ since 2004 (see Figure 1.14) . Its weight is 650g. The OS4 has an onboard mini PC for obstacles avoidance, communication and a DSP for attitude and altitude control. It has three gyroscopes, three

accelerometers, five range sensors and one camera. Further details about OS4 can be found in Bouabdallah's thesis [20]. The control methods used on 'OS4' can be found in Section 1.7.

Other quadrotors built by universities can be found in Section 1.7 Existing Quadrotor Control Methods.

## 1.5 Quadrotor Applications

Due to the simplicity in design and maneuver, quadrotors have many applications. As conventional aircrafts and pilots are usually expensive and the flight area is often limited, a quadrotor can be a good alternative. A quadrotor is easy to be transported, and can be set up, controlled simply by one person. The price of a quadrotor is much cheaper than renting a conventional aircraft. A normal commercialized quadrotor costs less than 8000 euros.

In public safety, quadrotors are used for border patrol, surveillance, search & rescue, etc. The commercialized quadrotors Draganfly [39] have been used in many police agencies, such as Prince Albert police service, Royal Canadian mounted police.

In these applications, quadrotors need a rather long flight duration to finish the tasks. The flight duration time depends on two factors: the capacity of the onboard battery and the energy consumption of the quadrotor system. The commercialized quadrotors usually use Lithium Polymer (LiPo) batteries, which have a significantly lighter and higher capacity than an equivalently shaped conventional rechargeable battery. The 4 cell 12V 2700mAh battery is the most common configuration used in many quadrotor systems, such as Draganfly X4 [39] and AscTec Hummingbird [2]. It permits a quadrotor fly about 15 to 20 minutes. The Draganfly X8 has an enhanced power, a 5400mAh lithium polymer battery, to ensure a longer flight duration. However, a bigger battery capacity means a heavier load and a more expensive price. In order to extend the flight duration, we can also reduce the consumption of the quadrotor system. Using a less powerful CPU (center processing unit) can save unwanted energy consumption. However, less powerful CPU means less processing capacity. Other systems, such as vision system, need many processing resources. Therefore, basic motion control systems should have rather simple algorithm, and at the same time should have rather high precise performance. In this situation, a control method with a simple algorithm and high performance is needed.

Moreover, especially in the search & rescue tasks, there are many unexpected obstacles. Quadrotors should have fast response to avoid collisions. Therefore, control methods should

have fast system adjustment to sudden changes.

Quadrotors are also used for aerial photography and video. The AscTec Falcon is designed especially for this application [2]. A quadrotor can be used for commercial purposes, such as the shooting for a wildlife documentary, the aerial view for an advertisement of a real estate, the aerial video for a sport event. It can be also used for personal purpose to record some pictures and videos for souvenir. For the commercial aerial photography and video, the high quality of the photos and videos should be ensured. The pictures should not be blur, and the videos should be taken while moving smoothly. Obtaining high quality aerial pictures and videos can be a big challenge. It depends on the configuration of the onboard camera, and most important the stability of a quadrotor when hovering or moving. Quadrotors need to hover or move with smallest vibrations to ensure the quality of the image. Therefore, the motion control system should be capable to tolerate signal noises from sensors, such as accelerometers and gyroscopes, and arrives a rather stable performance.

Moreover, as the shooting are taken in different circumstances and under different weather conditions, the stability of the quadrotor system is very important. There may be some weather disturbances, such as wind, temperature and humidity. The motion control system should be capable to tolerate these disturbances with smallest tracking errors and vibrations possible.

As discussed above, here we list some criteria that a control system should have :

- Complexity of the control algorithm: the number of the addition and multiplication operations in the control law.
- Maximum absolute tracking error.
- Variance of the tracking error.
- Robustness to model uncertainties: the change in the tracking error.
- Robustness to disturbances: the change in the tracking error.
- Adjusting time to disturbances: the time needed to return to the stable state.
- Energy consumption: the energy spent on motion control.

## 1.6 Existing Quadrotor Models

Several quadrotor dynamic models have been proposed in the literature. However, there is no standard model.

Altug et al. have proposed a simple dynamic model in 2002 [4, 5]. This simple dynamic model used Euler angles with ZYX rotational sequence and considered only the thrust force and yawing moment. All the other aerodynamic forces and moments are neglected, such as drag forces. The thrust is the propulsion which makes quadrotors fly and move forward. The yawing moment is the moment which makes the quadrotor rotate about its hub.

Fay has deduced mathematically the aerodynamic forces and moments of a small quadrotor named mesicopter in 2001 [42]. Based on the blade element theory, he has obtained explicit expressions of important aerodynamic forces and moments. There are two major aerodynamic forces: the thrust and the hub force. The thrust is the propulsion. The hub force is the drag force caused by airflow. There are also two crucial aerodynamic moments: the rotor torque (yawing moment) and the moment caused by the blade flapping effect. The rotor torque makes quadrotor turn about its hub, and the moment caused by the blade flapping effect makes quadrotor turn about its main plane which is perpendicular to the hub.

Hoffmann et al. have analyzed the aerodynamic effects on a quadrotor in 2007 [55]. The first effect is that the total thrust varies not only with the power input, but also with the free stream velocity and the angle of attack with respect to the free stream. The second effect is the blade flapping, that is, the advancing and retreating blades cause different inflow velocities. This induces the roll and pitch moments on the rotor hub as well as a deflection of the thrust vector. This moment is the rolling moment mentioned by Fay in the previous paragraph. The third effect is the interference caused by the vehicle body in the slip stream of the rotor, which results an unsteady thrust behavior.

Bouabdallah has studied the kinematic and dynamic models in 2007 [20]. He considered the aerodynamic forces and moments which were deduced by Fay [42]. He had also taken the gyroscopic effect into account. When a rotating rotor rotates about a perpendicular axis to its hub, it also causes a rotation about a third axis which is perpendicular to the former two. This effect is called the gyroscopic effect. Moreover, he noticed that the center of mass may be not on the same plane as the rotors, which causes extra moments about the roll and pitch axis. The propeller frictions are also considered in the dynamic model.

Purwin et al. have proposed an iterative learning technique for compensating the errors of the current dynamic model in 2009 [86]. The reference trajectory was computed as the solu-

tion of an optimal control problem. Based on a lifted domain description of that same model, an iterative learning controller was synthesized by solving a linear least-squares problem. The non-causality of the approach makes it possible to anticipate recurring disturbances.

Bangura et al. have pointed out that existent dynamic quadrotor models usually had constant thrust and moment coefficients which were derived from static thrust tests in 2012 [16]. These models are no longer valid when quadrotors have dynamic manoeuvres with significant displacements. Therefore, they have proposed an implicit thrust model that incorporates the induced momentum effects associated with the change of the airflow through the rotor. The proposed model used power as input to the system.

## 1.7 Existing Quadrotor Control Methods

In the last decades, many control methods have been proposed for quadrotors.

At the GRASP laboratory in the University of Pennsylvania, Altuğ et al. have proposed a controller using visual feedback as the primary sensor in 2002 [4, 5]. Two control methods are studied. In the first method, the states  $x$  and  $z$  are controlled using a feedback linearizing controller, and the state  $y$  and angle yaw are controlled using a PD controller. In the second method, a backstepping control is used for the states  $x$  and  $y$ , and a PD controller is used for the altitude  $z$  and angle yaw. These methods are proved to be able to stabilize a quadrotor. The problem is the switching between controllers in the control system. In 2007, Gurdan has presented a control system for a quadrotor AscTec Hummingbird [48]. The control system is divided into two subsystems: angular and position subsystems. Each subsystem is controlled by PD controllers. In 2010, the GRASP laboratory has developed a testbed for multi-quadrotors using AscTec Hummingbird [66]. In 2012, Mellinger et al. have developed iterative learning controllers refined through successive experimental trials automatically to compensate the errors in the dynamic model and the noises in the actuators and sensors [67, 68].

The Stanford University has designed the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) which is based on the early design X4 flyer of Draganflyer in 2004 [52–54]. In the testbed, Hoffmann et al. have proposed a sliding mode controller for the altitude loop control and an inner loop standard linear quadratic regulator (LQR) controller for the simplified second order rotational dynamic model [52]. In 2005, Waslander et al. have developed two control methods: an integral sliding mode control and a reinforced learning control to accommodate the combination of noises and disturbances on a quadrotor.

These methods have an improved performance over classical control techniques [98]. In 2007, Hoffmann et al. have extended their study on the aerodynamic modeling of a moving quadrotor, and proposed a PD and a PID controller for each subsystem [55].

The Australian National University has built ‘X-4’ since 2004 (see Figure 1.13). More details about X-4 can be found in Pounds’s thesis and papers [83,84]. Pounds has derived decoupled dynamics in longitudinal (pitch/roll) and azimuthal modes, and implemented the linear SISO control on the decoupled dynamics [85]. In 2012, Bangura et Mahony have pointed out the usual dynamic quadrotor model is based on the constant thrust and torque which are no longer valid when a quadrotor undertakes dynamic manoeuvres with significant displacement velocities. They have thus proposed an implicit thrust model with the induced momentum effects [16]. Then based on the model proposed, a Nonlinear Model Predictive Control (NMPC) is implemented on a quadrotor.

EPFL (Ecole Polytechnique Fédérale de Lausanne) has designed ‘OS4’ since 2004 (see Figure 1.14) [17–19]. The parameters of the OS4 and the comparisons with other models can be found in Bouabdallah’s dissertation [20]. Bouabdallah has presented a dynamic model of quadrotor and proposed many control methods on OS4, such as a control using Lyapunov theory, a PID control, an optimal control, a Linear Quadratic (LQ) control, a backstepping control, a sliding mode control [20]. These methods are dedicated to the stabilization of the three Euler angles. However, there was no trajectory tracking test and further tests in realistic scenarios. In 2004, Bouabdallah et al. have shown the PID control has a better performance than LQ control under model imperfections [21]. In 2005, Bouabdallah et al. have shown the backstepping control has a better result than the sliding mode control due to the switching nature of the latter controller [22].

At the University of Technology of Compiègne (UTC France), Castillo et al. have obtained a dynamic model via a Lagrange approach, and proposed a controller based on the Lyapunov analysis using a nested saturation algorithm on a Draganflyer in 2004 [32,33]. In 2006, Castillo et al. have proposed a controller using the backstepping technique and saturation functions [34]. The comparison with a classical PD controller was also proposed. Some aggressive perturbation forces were manually added to the system during the experiments. The control algorithm had shown to be robust to these disturbances. However, there was no comparison with other control methods. A book on modeling and control of mini-flying machines is also written by Castillo et al. [35].

At the Robotics Laboratory of Versailles (France), Mistler et al. have shown that the nonlinear dynamics in the quadrotor model can not be linearized or decoupled into a non-

linear SISO system by using a static state feedback. They have used a dynamic feedback technique to linearize the system, which makes the closed loop system linear and controllable in 2001 [69]. The stability and the robustness of the proposed control under wind disturbance and parametric uncertainties were studied. However, there was no comparison with other control methods. Mokhtari et al. have applied a dynamic feedback controller of Euler angles in 2004 [70], a mixed robust feedback linearization with linear  $\text{GH}_\infty$  controller in 2005 [71] and with a sliding mode observer in 2006 [72]. In 2006, Madani et al. have proposed a full state backstepping technique on quadrotor by separating the system into three interconnected subsystems [74]. In 2006, Benallegue et al. have presented a feedback linearization-based controller with a high-order sliding mode observer. The sliding mode observer worked as an observer and estimator of external disturbances such as wind and noises [23, 24].

At CEA France, Guenard et al. have proposed an adaptive control method on a quadrotor X4-Flyer in 2006 [49]. An attitude control based on the model of the rotor-craft was designed for stationary and quasi stationary flights. An adaptive controller was developed based on ultrasonic measurements in order to limit the ground effects for the altitude stabilization. Guenard et al. have proposed a practical visual servo control for an unmanned aerial vehicle in 2008 [50]. Bertrand et al. have proposed a hierarchical controller for miniature VTOL UAVs using singular perturbation theory in 2011 [28].

At the University of Alabama (USA), Besnard et al. have developed a sliding mode control driven by a sliding mode disturbance observer in 2007 [25]. The proposed method was robust to external disturbance (including wind, collision, actuator failure) and model uncertainties without using high control gains or extensive computational power [26]. However, there was no comparison with other control methods.

At the Cranfield University (UK), Cowling et al. have proposed a LQR controller for path following in 2007 [36]. They also presented a trajectory planning method using a differential flatness property as a constrained optimization problem in the output space (as opposed to the control space). The control method has been simulated with noise and with wind disturbance modeled as a drift in the translational position [37]. However, there was no comparison with other control methods.

At the Lakehead University (Canada), Tayebi et al. have proposed a  $\text{PD}^2$  controller, where the proportional action is in terms of the vector quaternion and the two derivative actions are in terms of the airframe angular velocity and the vector quaternion velocity since 2004 [92–94]. The method have only tested in ideal cases.

At the University of Sevilla (Spain), Raffo et al. have proposed an integral predictive and

nonlinear  $H_\infty$  controller in 2010 [88,89]. It was a hierarchical scheme consisting of a model predictive controller (MPC) to track the reference trajectory together with a nonlinear  $H_\infty$  controller to stabilize the rotational movements. The method was simulated in the scenarios with parametric uncertainties in comparison with the backstepping control proposed by Bouabdallah [20].

At the University of Patras (Greece), Alexis et al. have presented a switching model predictive attitude controller based on a piecewise affine (PWA) model in 2011 [6]. The control method is tested in the case with wind disturbance.

## 1.8 Contributions

This dissertation is motivated by the challenges in the modeling and control of a quadrotor mentioned in Section 1.3. Therefore, the contributions of this thesis can be summarized as follows:

- A new dynamic model is proposed and a dynamic analysis based on the normal form theory is given.
- A new control method, the event triggered model free control, is proposed for the quadrotor system.
- Comparisons of several control methods in five realistic scenarios are given.

### (1) Modeling.

In this thesis, the relationship of the kinematic variables between the body and world frames are firstly shown. Then, the aerodynamic forces and moments are presented. The world and body frames and the Euler angles rotational sequence are selected based on the most generally used references. The modeling is based on the Bouabsallah's modeling. The Bouabsallah's model is one of the most complete modeling in the literature. However, there are some errors in his model. The hub force is calculated using the linear velocities along the  $x_w$  and  $y_w$  in the world frame. These are not exact. The hub force should be calculated using the linear velocities along the  $x_b$  and  $y_b$  in the body frame on the quadrotor plane, as the aerodynamic force is caused by the airflow speed related to the quadrotor which is changed due to the quadrotor plane. Moreover, some directions of the forces and moment are not correct, such as the direction of the rolling moment, etc. The frames used in Bouabsallah's thesis are not the same as in his program, which is a little bit confusing. Therefore in this

thesis, the chosen world and body frames, the relationship of the linear and angular velocities, the aerodynamic forces and moments are clearly presented. The errors in the Bouabsallah's thesis are corrected. Some unnecessary aerodynamic forces and moments are neglected.

Then based on the normal form theory, the dynamics of the quadrotor is analyzed. The dynamics of the quadrotor has never been studied intensively in the literature. The model of a quadrotor is simplified into a simplest form which exhibits all possible properties of the original system. In the original model, the control inputs are coupled with the states, which makes the analysis of the system difficult. In the normal form, the system states are no longer coupled with the system inputs, and we can see the pitch angle  $\theta$  has more influence on the  $x$  axis and the roll angle  $\phi$  has more influence on the  $y$  axis. The bifurcations of the simplified system are studied. Using the center manifold theory, the system can be further simplified at its bifurcation point. The twelve dimensional system is simplified into a two dimensional system which exhibits all possible properties of the original system.

## (2) Control methods.

In the first inspiring example, the traditional PID controller has poor performance in the control of a quadrotor. As discussed in Section 1.3, quadrotor systems bring out many challenges: the complexity of the control algorithm should be simple; the control method should be stable for model and parameter errors; the control method should be able to compensate disturbances and keep the stability of the system with less tracking errors and vibrations possible; etc.

Based on these needs, a newly proposed control method the model free control is implemented on the quadrotor system. It is a simple but efficient technique for nonlinear, unknown or partially known dynamics. Instead of computing the dynamics of a system, the dynamics is evaluated as a variable  $F$  in real time by the system inputs and outputs. Therefore, the model free control can compensate well the modeling errors and real-time disturbances. The algorithm of this control method is rather simple as it does not fully compute the dynamic model. At the same time, the stability of this method is high as it compensates well the disturbances. To further save the computational resources, the event triggered scheme is proposed on this control method. The model free control adapts well the event triggered scheme. While saving the computational resources, the performance of the event triggered model free control is not heavily changed in comparison with the time triggered model free control.

In order to show its advantages and disadvantages, several other control methods are also proposed, such as a backstepping control and a sliding mode control. Five different realistic scenarios are proposed to get a comprehensive evaluation.

### **(3) Comparison of the control methods in different realistic scenarios.**

As shown in the second inspiring example, the good performance of a control method in an ideal case can not ensure the same good performance in a realistic case. Therefore in this thesis, the applications of quadrotors have been carefully studied, and five typical realistic scenarios have been proposed: a basic ideal case, a case with wind disturbance, a case with parameter uncertainties, a case with sensor noises and a case with actuator faults. In order to close to the realistic cases, the control gains of each control method are the same in five different scenarios. The control gains are usually chosen beforehand and will not be changed during the applications when the disturbances occur. From the results in these realistic scenarios, we can get a comprehensive evaluation of the control methods, which is more objective than using just the result in ideal cases.

In the literature, the comparison of the control methods are rarely presented. Lack of a standard scenario, we can hardly say which method has a better performance. For this purpose, several control methods are also implemented in these five scenarios. From the results in different cases, the advantages and disadvantages of each method can be easily observed. Therefore, an appropriate control method can be selected based on the needs of a certain applications.

## **1.9 Outline**

This dissertation is organized as follows: In Chapter 2, the kinematic and dynamic models of quadrotor are presented. The aerodynamic forces and moments are studied. The full model is proposed at the end of the chapter, and a simplified model is also presented based on some applications. In Chapter 3, the normal form theory is implemented on the quadrotor model, and a simplified model is obtained which exhibits all possible properties of the original system. Then the bifurcations based on the simplified model are studied. Using the center manifold theory, the model is further simplified at its bifurcation point. In Chapter 4, the applications of quadrotors are studied, and five typical realistic scenarios are proposed. Then, the time triggered and event triggered schemes are proposed. At the end, the model free control and other control methods are presented. In Chapter 5, the time triggered and

event triggered control methods implemented in five scenarios are proposed in Chapter 4. In Chapter 6, the conclusion is given and future work is discussed.

# Chapter 2

## System Modeling

### 2.1 Introduction

Many dynamic models have been proposed in the literature. In Section 1.6, these models were discussed in detail. However, there is no standard model. In this chapter, a kinematic model and a dynamic model of quadrotor will be constructed. According to the literature, the most common Euler angle sequence ZYX is selected. In the dynamic modeling, the aerodynamic forces and moments deduced by Fay are used [42]. The dynamic model is based on constant thrust and moment coefficients. Other important moments like the gyroscopic effect moments are also considered. A simplified model is also proposed based on some scenarios. At the end, considering the limits of the motor and the safety of the flight, some constraints are given for the quadrotor system.

### 2.2 Preliminary Knowledge

Before analyze the dynamics of a quadrotor, several assumptions are made:

- The quadrotor is a rigid body.
- The propellers of the motors are also rigid.
- The quadrotor is symmetrical along the  $x$  and  $y$  axis in the body frame. The center of the body frame coincides with the center of gravity. Therefore,  $I_{xx} = I_{yy}$ ,  $I_{xy} = I_{yz} = I_{zx} = 0$ .
- The motor inertia is negligible.

As the quadrotor is a rigid body, we can use the Newton-Euler formalism for the dynamic modeling.

In the analysis, the Cartesian coordinate system is selected as in Figure 2.1. The frame  $\mathcal{W} = \{x_w, y_w, z_w\}$  is the world frame. The axis  $z_w$  is vertical to the ground, and the axis  $x_w$  and  $y_w$  locate in the ground plane. The angles  $\phi, \theta$  and  $\psi$  are Tait-Bryan angles (also called Cardan or Euler angles), which rotate about the  $x_w, y_w$  and  $z_w$  axis respectively. The frame  $\mathcal{B} = \{x_b, y_b, z_b\}$  is the body frame. To simplify the mathematical expressions of physical forces and moments, the center of the body frame coincides with the center of gravity (CoG). The axis  $x_b$  and  $y_b$  are along the two arms of a quadrotor, and the axis  $z_b$  is vertical to the arm plane. The body angular velocities are presented as  $p, q$  and  $r$ , which are physically measured by three gyroscopes sensors. The four rotors are numbered 1, 2, 3 and 4. The first one is along the positive direction of the  $x_b$  axis, and the second one is along the positive direction of the  $y_b$  axis. The third one is along the negative direction of the  $x_b$  axis, and the fourth one is along the negative direction of the  $y_b$  axis. The rotor 1 and 3 rotate clockwise (see from the top of the quadrotor). The rotor 2 and 4 rotate counterclockwise. In the notation, the subscripts 1 to 4 of the forces and moments mean the number of the rotor. For example, the propulsion forces, the thrusts, of each motor are defined  $T_1, T_2, T_3$  and  $T_4$  which are along the directions shown in Figure 2.1.

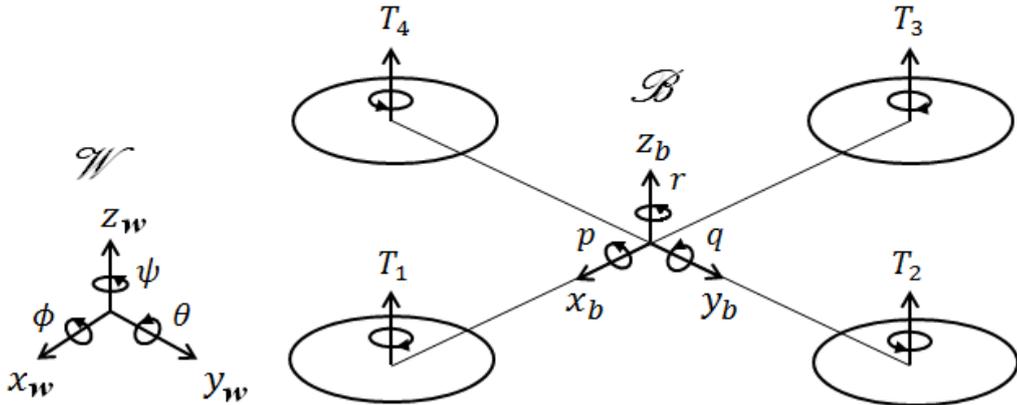


Figure 2.1: The Cartesian coordinate system of a quadrotor.

For the simplicity of reading, the notations used in the kinematic and dynamic models are listed in Table 2.1.

Table 2.1: The notations used in the kinematic and dynamic models

Position, velocity and acceleration in the world frame		$\mathbf{P}_w, \mathbf{V}_w, \mathbf{a}_w$	
Position, velocity and acceleration in the body frame		$\mathbf{P}_b, \mathbf{V}_b, \mathbf{a}_b$	
Tait-Bryan angles (Euler angles) in the world frame		$\phi, \theta, \psi$	
Angular velocities in the body frame		$p, q, r$	
Thrust	$T_i$	Hub force	$H_i$
Rolling moment	$M_{roll}$	Pitching moment	$M_{pitch}$
Rotor torque (yawing moment)	$Q_i$	Blade flapping moment	$R_i$
Rotor gyroscopic effect moments	$M_{gx}, M_{gy}$	Gravity	$G$

## 2.3 Kinematic Model

In the kinematic analysis, the relations of the positions, velocities and accelerations between the world frame and the body frame are studied. Based on the frames defined in the previous section, the translation between the world frame and the body frame can be expressed explicitly.

### 2.3.1 Position, Linear Velocity and Acceleration

The Euler angles  $\phi, \theta, \psi$  are three angles introduced by Leonhard Euler to describe the orientation of a rigid body. They represent a sequence of three elemental rotations about the axes of a coordinate system. Any orientation can be achieved by composing three elemental rotations. There are many rotational sequences, such as  $XYX, XYZ, YZY$ . For example, a sequence  $XYX$  means a body firstly rotates about the  $x$  axis, then rotates about the  $y$  axis, and at last rotates about the  $x$  axis again. Euler angles usually represent rotational sequences about two axis, such as  $ZXZ, XYX, YZY$ . This means there are two rotations about the same axis. The Tait-Bryan angles represent rotational sequences along three different axis, such as  $XYZ, YZX, ZYX$ . Generally we call Tait-Bryan angles also Euler angles. Here we use a common rotational sequence  $ZYX$  for the Euler angles. It means a body firstly rotates about the  $z$  axis, then rotates about the  $y$  axis, and at last rotates about the  $x$  axis. This rotational sequence  $ZYX$  is used in many quadrotor modelings.

Therefore, the rotation matrix  $\mathcal{R}_w^b$  from the world frame to the body frame is:

$$\begin{aligned}
\mathcal{R}_w^b &= \mathcal{R}(x, \phi)\mathcal{R}(y, \theta)\mathcal{R}(z, \psi) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta \cos\psi & \cos\theta \sin\psi & -\sin\theta \\ \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \sin\phi \cos\theta \\ \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi & \cos\phi \cos\theta \end{bmatrix}
\end{aligned} \tag{2.1}$$

As the center of gravity coincides with the origin of the body frame, the origin  $\mathbf{P}_{cb}=(0,0,0)$ . The distance between the origin of the body frame and the origin of the world frame is defined as  $\mathbf{r} = \mathbf{P}_{cb} - \mathbf{P}_{cw}$ . Therefore, an arbitrary point in the body frame  $\mathbf{P}_b = (x_b, y_b, z_b)$  can be expressed in the world frame as:

$$\mathbf{P}_w = \mathbf{P}_b + \mathcal{R}_w^b \mathbf{r} \tag{2.2}$$

The relation between the linear velocity in the world frame and the linear velocity in the body frame can be obtained:

$$\begin{aligned}
\mathbf{V}_w &= \mathcal{R}_b^w \mathbf{V}_b = \mathcal{R}_w^b{}^T \mathbf{V}_b \\
&= \begin{bmatrix} \cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ \cos\theta \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\ -\sin\theta & \sin\phi \cos\theta & \cos\phi \cos\theta \end{bmatrix} \mathbf{V}_b
\end{aligned} \tag{2.3}$$

or expressed in each coordinate as:

$$\begin{cases} V_x^w = (\cos\theta \cos\psi)V_x^b + (\sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi)V_y^b + (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)V_z^b \\ V_y^w = (\cos\theta \sin\psi)V_x^b + (\sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi)V_y^b + (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)V_z^b \\ V_z^w = (-\sin\theta)V_x^b + (\sin\phi \cos\theta)V_y^b + (\cos\phi \cos\theta)V_z^b \end{cases} \tag{2.4}$$

Then, the relation between the linear acceleration in the world frame and in the body frame can be obtained:

$$\mathbf{a}_w = \mathcal{R}_b^w \dot{\mathbf{V}}_w \tag{2.5}$$

### 2.3.2 Angular Velocity and Acceleration

A rotational sequence ZYX is chosen for the Euler angles. It means the world frame firstly rotates about its  $z$  axis, then rotates about its  $y$  axis and at last rotates about its  $x$  axis to get the body frame. Therefore, the angular velocity  $\dot{\phi}$  is the same with the angular velocity  $p$  about the  $x$  axis in the body frame, as the  $x$  axis is not rotated during the last step of the rotation. The angular velocity  $\dot{\theta}$  along the  $y$  axis is influenced only by the last step rotation about the  $x$  axis. Therefore, this velocity only needs to multiply the rotational matrix about the  $x$  axis. As the  $z$  axis is firstly rotated, its angular velocity  $\dot{\psi}$  needs to multiply two rotational matrices about the  $y$  axis firstly then about the  $x$  axis.

Therefore, the relation between the angular velocities  $p, q, r$  in the body frame and the angular velocities  $\dot{\phi}, \dot{\theta}, \dot{\psi}$  in the world frame is presented:

$$\begin{aligned}
 \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \mathcal{R}_{ang} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\
 &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathcal{R}^T(x, \phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathcal{R}^T(x, \phi)\mathcal{R}^T(y, \theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\theta & \sin\phi \cos\theta \\ 0 & -\sin\phi & \cos\phi \cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}
 \end{aligned} \tag{2.6}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.7}$$

The relation between the angular accelerations in the body frame and in the world frame can be obtained by differentiating Equation (2.6).

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \dot{\mathcal{R}}_{ang} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \mathcal{R}_{ang} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{2.8}$$

## 2.4 Dynamic Model

In the dynamic analysis, we first write the Newton Euler equations for a quadrotor. Then, the aerodynamic forces and moments applied on the quadrotor are studied. At last, a simplified dynamic model is further deduced.

### 2.4.1 Newton Euler Formalism

In classical mechanics, the Newton Euler equations show the translational and rotational dynamics of a rigid body. The Newton Euler formalism is the grouping of Euler's two motion laws for a rigid body into a single equation with six components. These laws relate the motion of the center of gravity of a rigid body with the sum of external forces and moments applied on the rigid body. Without loss of generality, we define the origin of the body frame coincides with the center of gravity. Supposing there is a six degrees of freedom (d.o.f.) rigid body which has a mass  $m$  and an inertia matrix  $\mathbf{I}$  about the center of gravity,  $\mathbf{I} \in \mathbf{R}^{3 \times 3}$ . The linear velocity of the center of gravity is  $\mathbf{V}_b$ , and the body angular velocity is  $\boldsymbol{\omega}_b$  in the body frame,  $\mathbf{V}_b, \boldsymbol{\omega}_b \in \mathbf{R}^{3 \times 1}$ . The external force and the moment are  $\mathbf{F}_b$  and  $\mathbf{M}_b$  in the body frame,  $\mathbf{F}_b, \mathbf{M}_b \in \mathbf{R}^{3 \times 1}$ .

Therefore, the relation between the velocities and the external forces and moments in the body frame is:

$$\begin{bmatrix} m\mathbf{I} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_b \\ \dot{\boldsymbol{\omega}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}_b \times (m\mathbf{V}_b) \\ \boldsymbol{\omega}_b \times (\mathbf{I}\boldsymbol{\omega}_b) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \mathbf{M}_b \end{bmatrix} \quad (2.9)$$

The subscript  $3 \times 3$  shows the dimension of the matrix is 3 by 3. The symbol  $\mathbf{0}_{3 \times 3}$  means a  $3 \times 3$  dimensional zero matrix.

In a quadrotor system, the body frame is chosen as in Figure 2.1. Applying the Newton Euler formalism to the quadrotor system, the relation between the forces and the linear accelerations is:

$$\begin{bmatrix} \ddot{x}_b \\ \ddot{y}_b \\ \ddot{z}_b \end{bmatrix} = \begin{bmatrix} r\dot{y}_b - q\dot{z}_b \\ p\dot{z}_b - r\dot{x}_b \\ q\dot{x}_b - p\dot{y}_b \end{bmatrix} + \frac{1}{m} \begin{bmatrix} F_x^b \\ F_y^b \\ F_z^b \end{bmatrix} \quad (2.10)$$

where  $\dot{x}_b$ ,  $\dot{y}_b$  and  $\dot{z}_b$  are the linear velocities along the  $x$ ,  $y$  and  $z$  axis in the body frame, and  $p$ ,  $q$  and  $r$  are the angular velocities around the  $x$ ,  $y$  and  $z$  axis in the body frame.

As the quadrotor is symmetric along the  $x$ ,  $y$  and  $z$  axis, the cross components in the inertia matrix are equal to zero,  $I_{xy} = I_{yz} = I_{zx} = 0$ . Therefore, the inertia matrix can be written:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.11)$$

Thus, the relation between the moments and the angular accelerations in the body frame is:

$$\begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})qr \\ (I_{zz} - I_{xx})pr \\ (I_{xx} - I_{yy})pq \end{bmatrix} + \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \quad (2.12)$$

Now, we need to find the external forces  $\mathbf{F}_b$  and the moment  $\mathbf{M}_b$  applied on the quadrotor.

## 2.4.2 Forces

The aerodynamic forces and moments applied on a quadrotor have been studied by Fay [42]. Using the blade element theory, he has deduced explicitly the expressions for aerodynamic forces and moments. These expressions are used in the modeling. The summary of the external forces mentioned in this subsection and their directions are listed in Table 2.2. All the parameters used in this subsection are listed in Table 2.3. All the forces are expressed in the body frame.

Table 2.2: The external forces applied on a quadrotor.

Force		Direction
Thrusts	$T_i$	along the $z_b$ axis
Hub forces	$H_i$	in the $x_b, y_b$ plane, along the direction of the linear velocity
Gravity	$G$	along the $z_w$ axis

### 1. Thrust & hub forces

The propulsion of a quadrotor is induced by the rotations of the four propellers. The inflow velocity (also called induced velocity) is the velocity of the air when it crosses the

Table 2.3: The variables in the forces and moments.

$V_H$	the sideways velocity of the rotor	$V_1$	the inflow velocity
$\lambda$	the inflow ratio	$\mu$	the advance ratio
$W$	the quadrotor weight	$\rho$	the air density
$J_r$	the moments of inertia of the rotors	$A$	the propeller disk area
$\Omega$	the rotational velocity of the rotor	$R_{rad}$	the propeller radius
$a$	the lift slope	$\theta_0, \theta_{tw}$	the linear lift parameters
$\sigma$	the solidity ratio	$N$	the number of blades
$\bar{c}$	the blade chord	$\bar{C}_d$	the drag coefficient
$m$	the mass of the quadrotor	$g$	the gravity coefficient
$l$	the distance from the CoG to the rotor	$V$	the volume of the quadrotor

propeller blade. It is assumed to be uniform and without discontinuity. The inflow velocity can be modeled as:

$$V_1 = \sqrt{-\frac{V_H^2}{2} + \sqrt{\left(\frac{V_H^2}{2}\right)^2 + \left(\frac{W}{2\rho A}\right)^2}} \quad (2.13)$$

where  $W$  is the quadrotor weight,  $\rho$  is the air density,  $A$  is the propeller disk area,  $V_H$  is the sideways velocity of the rotor, which is the quadrotor linear velocity in the  $x_b, y_b$  plane of the body frame:

$$V_H = \sqrt{\dot{x}_b^2 + \dot{y}_b^2} \quad (2.14)$$

The inflow ratio  $\lambda$  is a dimensionless quantity used in the helicopter literature to relate the inflow velocity to the rotor tip velocity. The inflow ratio is defined as:

$$\lambda = \frac{V_1 - \dot{z}_b}{\Omega R_{rad}} \quad (2.15)$$

where  $\Omega$  is the rotational velocity of the rotor,  $R_{rad}$  is the propeller radius.

Another dimensionless quantity is the advance ratio. It relates the horizontal velocity with the tip velocity of a rotor. The advance ratio  $\mu$  is:

$$\mu = \frac{V_H}{\Omega R_{rad}} \quad (2.16)$$

When the air crosses the propellers, it induces two aerodynamic forces: the lift force and the drag force. The lift force is perpendicular to the local velocity, while the drag force is parallel to it. We can project these two forces onto the thrust force and the hub force.

The thrust force  $T$  is along the  $z_b$  axis in the body frame. The rest of the force which is in the plane  $x_b, y_b$  of the body frame can be separated in two directions: the hub force  $H$  which is in the direction of the sideways velocity  $V_H$ , the projection of the forwarding velocity  $\mathbf{V}$  in the  $x_b, y_b$  plane; the other one is the side force  $Y$  which is in the perpendicular direction of the sideways velocity in the same plane. These forces are shown in Figure 2.2. The ellipse is the rotor plate.

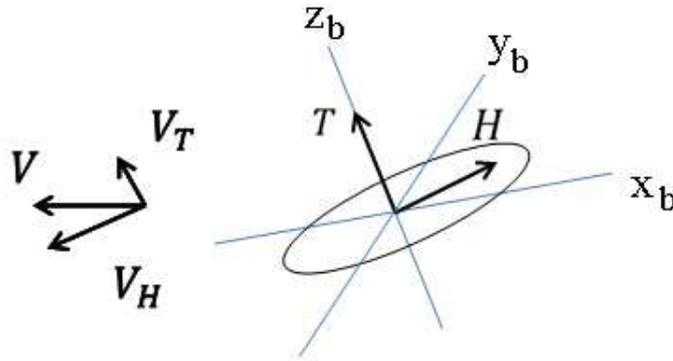


Figure 2.2: The thrust and hub forces on a rotor.

Using the blade element theory, Fay has deduced the expressions of the aerodynamic forces [42]. He has proved the side force  $Y$  is always zero. This means the drag force is only along the direction of the linear velocity of the quadrotor, which is named hub force. The thrust  $T$  and the hub force  $H$  on a rotor are modeled as:

$$\begin{aligned} T &= C_T \rho A (\Omega R_{rad})^2 \\ H &= C_H \rho A (\Omega R_{rad})^2 \end{aligned} \quad (2.17)$$

The coefficients of the thrust and hub force are:

$$\begin{aligned} C_T &= \sigma a \left[ \left( \frac{1}{6} + \frac{1}{4} \mu^2 \right) \theta_0 - (1 + \mu^2) \frac{\theta_{tw}}{8} - \frac{1}{4} \lambda \right] \\ C_H &= \sigma a \left[ \frac{1}{4a} \mu \bar{C}_d + \frac{1}{4} \lambda \mu \left( \theta_0 - \frac{\theta_{tw}}{2} \right) \right] \end{aligned} \quad (2.18)$$

where  $a$  is the lift slope,  $\theta_0$  and  $\theta_{tw}$  are the linear lift parameters,  $\bar{C}_d$  is the drag coefficient,  $\sigma$  is the solidity ratio which defines the ratio of the blade area to the disk area. The solidity

ratio is defined as:

$$\sigma = \frac{N\bar{c}}{\pi R} \quad (2.19)$$

where  $N$  is the number of blades,  $\bar{c}$  is the blade chord. Further details about the parameters can be found in [27, 42].

The expressions above are the thrust and hub force on a rotor. As a quadrotor has four rotors, the thrusts and hub forces are defined as  $T_i$ ,  $H_i$ ,  $i = 1, \dots, 4$ , which are related to each rotor respectively. The total thrust and hub force on a quadrotor are:

$$\begin{aligned} -\sum_{i=1}^4 H_{ix} &= -H_{1x} - H_{2x} - H_{3x} - H_{4x} \\ -\sum_{i=1}^4 H_{iy} &= -H_{1y} - H_{2y} - H_{3y} - H_{4y} \\ \sum_{i=1}^4 T_i &= T_1 + T_2 + T_3 + T_4 \end{aligned} \quad (2.20)$$

As the hub force  $H_i$  is along the negative direction of the sideways velocity, it can be separated into two forces  $H_{ix}$  and  $H_{iy}$ .  $H_{ix}$  is along the  $x_b$  axis and  $H_{iy}$  is along the  $y_b$  axis in the body frame. The thrust  $T_i$  is always along the  $z_b$  axis in the body frame.

## 2. Gravity

The gravity is along the  $z_w$  axis of the world frame:

$$G = -mg \quad (2.21)$$

where  $m$  is the mass of a quadrotor,  $g$  is the gravity coefficient.

## 3. Total external forces

The thrust and hub forces are described in the body frame. The forces in the world frame can be translated into the body frame using the following translation:

$$\mathbf{F}_b = \mathcal{R}_w^b \mathbf{F}_w \quad (2.22)$$

Therefore, the total external force vector on a quadrotor expressed in the body frame is:

$$\mathbf{F}_b = \begin{bmatrix} -\sum_{i=1}^4 H_{ix} \\ -\sum_{i=1}^4 H_{iy} \\ \sum_{i=1}^4 T_i \end{bmatrix} + \mathcal{R}_w^b \begin{bmatrix} 0 \\ 0 \\ -G \end{bmatrix} \quad (2.23)$$

### 2.4.3 Moments

Using the blade element theory, Fay has also deduced explicitly the expressions for the aerodynamic moments [42]. These moments are used in the modeling. The summary of the external moments mentioned in this subsection and their directions are listed in Table 2.4. All the variables used in this subsection are listed in Table 2.3. All the moments are expressed in the body frame.

Table 2.4: The external moments applied on a quadrotor.

Moment	Notation	Direction
Rolling moment	$M_{roll}$	about the $x_b$ axis in the body frame.
Pitching moment	$M_{pitch}$	about the $y_b$ axis in the body frame.
Yawing moments	$Q_i$	about the $z_b$ axis in the body frame.
Blade flapping moments	$R_{ix}, R_{iy}$	about the $x_b, y_b$ axis in the body frame.
Gyroscopic effect moments	$M_{gx}, M_{gy}$	about the $x_b, y_b$ axis in the body frame.

#### 1. Rolling moment & Pitching moment

As each rotor generates a thrust, the difference between each pair of rotors causes the moments around the  $x_b$  and  $y_b$  axis in the body frame. The difference of the thrusts between the rotor 2 and 4 causes the rolling moment about the  $x_b$  axis, and the difference of the thrusts between the rotor 1 and 3 causes the pitching moment about the  $y_b$  axis. The direction of the moments is decided according to the right hand rule. Therefore, the rolling moment  $M_{roll}$  and the pitching moment  $M_{pitch}$  are defined as:

$$\begin{aligned} M_{roll} &= l(T_2 - T_4) \\ M_{pitch} &= l(-T_1 + T_3) \end{aligned} \quad (2.24)$$

where  $l$  is the distance from the center of gravity (CoG) to the rotor,  $T_i$ ,  $i = 1, \dots, 4$  are the thrusts of the four rotor. The numbers of the rotors are defined in Figure 2.1.

## 2. Yawing moment

The aerodynamic forces also cause moments about the shaft and hub. The moment about the shaft  $Q$  makes the quadrotor turn about the  $z_b$  axis of the body frame. It determines the power required for the rotor to keep the rotor spinning. The yawing moment  $Q$  is modeled as:

$$Q = C_Q \rho A (\Omega R_{rad})^2 R_{rad} \quad (2.25)$$

The coefficient of the yawing moment is:

$$C_Q = \sigma a \left[ \frac{1}{8a} (1 + \mu^2) \bar{C}_d + \lambda \left( \frac{1}{6} \theta_0 - \frac{1}{8} \theta_{tw} - \frac{1}{4} \lambda \right) \right] \quad (2.26)$$

All the parameters in the coefficient  $C_Q$  can be found in Subsection 2.4.2. Further details about the parameters can be found in [27, 42].

There are four rotors in a quadrotor. Therefore each rotor has a yawing moment  $H_i$ ,  $i = 1, \dots, 4$ . The four rotors are separated into two pairs with different rotational directions. The rotor 1 and 3 rotate clockwise (see from the top of the quadrotor). The rotor 2 and 4 rotate counterclockwise. These two pairs of rotors induce moments in different direction. Thus, the total yawing moment is:

$$\sum_{i=1}^4 Q_i = Q_1 - Q_2 + Q_3 - Q_4 \quad (2.27)$$

where the subscripts 1, ..., 4 are the numbers of the rotors defined in Figure 2.1.

## 3. Blade flapping moment

During the translational flight, the advancing blade experiences a higher velocity relative to the air than the retreating blade. This effect is called the blade flapping. This results more lift on the advancing blade than the retreating blade. The difference of the lifts applies a moment to the rotor disk and leads to a gyroscope response that tilts the rotor disk back with a flapping angle  $\beta$ . The blade flapping moment  $R$  is perpendicular to the rotor shaft

and the linear velocity of the quadrotor, which is modeled as:

$$R = C_R \rho A (\Omega R_{rad})^2 R_{rad} \quad (2.28)$$

with the coefficient as:

$$C_R = -\sigma a \mu \left( \frac{1}{6} \theta_0 - \frac{1}{8} \theta_{tw} - \frac{1}{8} \lambda \right) \quad (2.29)$$

The parameters of  $C_R$  are mentioned in Subsection 2.4.2. Further details about the parameters can be found in [27, 42].

The blade flapping moment of each rotor  $R_i$ ,  $i = 1, \dots, 4$  can be separated into two moments  $R_{ix}$  and  $R_{iy}$ . The total blade flapping moments of a quadrotor are:

$$\begin{aligned} \sum_{i=1}^4 R_{ix} &= R_{1x} - R_{2x} + R_{3x} - R_{4x} \\ \sum_{i=1}^4 R_{iy} &= R_{1y} - R_{2y} + R_{3y} - R_{4y} \end{aligned} \quad (2.30)$$

These two moments result the turning about the  $x_b$  and  $y_b$  axis in the body frame.

#### 4. Rotor gyroscopic effect

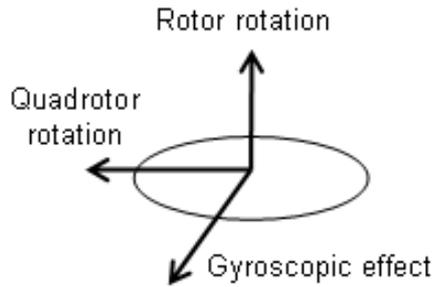


Figure 2.3: Rotor gyroscopic effect.

When the rotor rotates around its own axis and a second axis, it also brings into rotation around a third axis which is perpendicular to the former two axis. This effect is called the gyroscopic effect. Therefore, when a quadrotor rotates along the  $x_b$  axis in the body frame, each rotor also has a moment  $M_{gy}$  along the  $y_b$  axis induced by the rotor gyroscopic effect. When a quadrotor rotates along the  $y_b$  axis in the body frame, each rotor also has a moment

$M_{gx}$  along the  $x_b$  axis. The direction of the moment is according to the right hand rule (see Figure 2.3).

The gyroscopic effect moments are modeled as:

$$\begin{aligned} \sum_{i=1}^4 M_{gix} &= J_r q (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) \\ \sum_{i=1}^4 M_{giy} &= -J_r p (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) \end{aligned} \quad (2.31)$$

where  $J_r$  is the moments of inertia of the rotors,  $\Omega_i$ ,  $i = 1, \dots, 4$ , are the rotational speeds of four rotors.

## 5. Total moments

Therefore, the total external moment vector on a quadrotor expressed in the body frame is:

$$\mathbf{M}_b = \begin{bmatrix} \sum_{i=1}^4 R_{ix} + M_{roll} + \sum_{i=1}^4 M_{gix} \\ \sum_{i=1}^4 R_{iy} + M_{pitch} + \sum_{i=1}^4 M_{giy} \\ \sum_{i=1}^4 Q_i \end{bmatrix} \quad (2.32)$$

## 2.5 Quadrotor Model

In the previous sections, we have deduced the Newton-Euler equations for a quadrotor system. Then the important external forces and moments are analyzed. In this section, the full model of a quadrotor system will be firstly proposed. Based on some applications, this model can be further simplified.

### 2.5.1 Complete Model

In the previous sections, the aerodynamic forces and moments on a quadrotor have been deduced. From Equations (2.3), (2.7), (2.10) and (2.12), a full model for a quadrotor is shown in Equation (2.33). The definitions of the symbols can be found in Table 2.1.

$$\begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{z}_w \end{bmatrix} = \begin{bmatrix} \cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ \cos\theta \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\ -\sin\theta & \sin\phi \cos\theta & \cos\phi \cos\theta \end{bmatrix} \left( \begin{bmatrix} r\dot{y}_b - q\dot{z}_b \\ p\dot{z}_b - r\dot{x}_b \\ q\dot{x}_b - p\dot{y}_b \end{bmatrix} + \frac{1}{m} \mathbf{F}_b \right) \quad (2.33a)$$

$$\mathbf{F}_b = \begin{bmatrix} -\sum_{i=1}^4 H_{ix} \\ -\sum_{i=1}^4 H_{iy} \\ \sum_{i=1}^4 T_i \end{bmatrix} + \mathcal{R}_w^b \begin{bmatrix} 0 \\ 0 \\ -G \end{bmatrix} \quad (2.33b)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.33c)$$

$$\begin{bmatrix} I_x \dot{p} \\ I_y \dot{q} \\ I_z \dot{r} \end{bmatrix} = \begin{bmatrix} (I_y - I_z)qr \\ (I_z - I_x)pr \\ (I_x - I_y)pq \end{bmatrix} + \mathbf{M}_b, \quad \mathbf{M}_b = \begin{bmatrix} M_x^b \\ M_y^b \\ M_z^b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^4 R_{ix} + M_{roll} + \sum_{i=1}^4 M_{gix} \\ \sum_{i=1}^4 R_{iy} + M_{pitch} + \sum_{i=1}^4 M_{giy} \\ \sum_{i=1}^4 Q_i \end{bmatrix} \quad (2.33d)$$

where

$$\begin{aligned} T &= C_T \rho A (\Omega R_{rad})^2, & H &= C_H \rho A (\Omega R_{rad})^2 \\ Q &= C_Q \rho A (\Omega R_{rad})^2 R_{rad}, & R &= C_R \rho A (\Omega R_{rad})^2 R_{rad} \\ M_{roll} &= l(T_2 - T_4), & M_{pitch} &= l(-T_1 + T_3) \end{aligned} \quad (2.34)$$

$$\sum_{i=1}^4 M_{gix} = J_r q (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4), \quad \sum_{i=1}^4 M_{giy} = -J_r p (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)$$

In Equation (2.33), the first two equations are the position dynamics, and the other three equations are the angle dynamics. The angle dynamics is only related on the angular information. While in the potion dynamics, the positions are affected by the changes of the angles.

This model can be a good workbench for testing control methods, and especially for comparing different methods. In this model, a rotational sequence ZYX for Tait-Bryan angles is chosen, which is generally used in the literature. Most of the control methods proposed in the literature can be easily implemented on this model without many modifications. Also, important aerodynamic forces and moments are considered in the modeling, while in the literature usually the thrust and the yawing moment are included in the models. The important aerodynamic forces, the thrusts and hub forces, are modeled based on the blade element theory. The sideways velocity related to the quadrotor body frame is used in the modeling of the forces, since the aerodynamic forces are induced by the local airflow around the rotors. The modeling is more precise than the model proposed by Bouabdallah [20]. In his thesis, he has used the sideways velocity in the world frame. As quadrotors tilt, the sideways velocity in the body frame is not always equal to the one in the world frame. Moreover, the reference coordinate system is clearly proposed in this thesis. The directions of the rotors' rotation, the aerodynamic forces and moments are clearly given. In Bouabdallah's thesis, the coordinate system is different from the one used in his program, and this caused some direction errors in his modeling. In the workbench proposed in this thesis, all the previously mentioned errors have been corrected. The aerodynamic forces are modeled more precisely, and some forces and moments are neglected base on the aerodynamic effect.

## 2.5.2 Simplified Simulation Model

As discussed in Chapter 1.5, quadrotors have many applications, such as surveillance, border patrol, aerial photograph and video, etc. In these applications, quadrotors usually need to follow a trajectory or hover around a point. The movement of quadrotors is rather slow and without violent displacement. The Phantom quadrotor (see Figure 1.11 in Section 1.4) has a maximum tilt angle 45 degrees and a maximum flight velocity 10m/s. However in applications, the tilt angle is usually inferior to 14 degrees (0.25rad), the angular velocity is inferior to 0.2rad/s, and the flight velocity is inferior to 2m/s.

$$\begin{aligned} \phi, \theta &\leq 0.25 \text{ rad}, & \dot{\phi}, \dot{\theta}, \dot{\psi} &\leq 0.2 \text{ rad/s} \\ p, q, r &\leq 0.2 \text{ rad/s}, & v_b &\leq 2 \text{ m/s} \end{aligned} \tag{2.35}$$

The relation between the angular velocities  $\dot{\phi}, \dot{\theta}, \dot{\psi}$  in the world frame and the angular velocities  $p, q, r$  in the body frame is:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\theta & \sin\phi \cos\theta \\ 0 & -\sin\phi & \cos\phi \cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.36)$$

where the transformation matrix is limited:

$$\begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\theta & \sin\phi \cos\theta \\ 0 & -\sin\phi & \cos\phi \cos\theta \end{bmatrix} \preceq \begin{bmatrix} 1 & 0 & -0.247 \\ 0 & 0.9689 & 0.239 \\ 0 & -0.247 & 0.939 \end{bmatrix} \quad (2.37)$$

In applications, in order to save energy and simplify control laws, one of the arms of the quadrotor is set along the direction of the forward velocity to reduce the rotational movement. Thus, every time quadrotors tilt only around one axis  $x_b$  or  $y_b$  in the body frame. Therefore, we can consider the angular velocities  $\dot{\phi}$ ,  $\dot{\theta}$  and  $\dot{\psi}$  in the world frame are equal to the angular velocities  $p$ ,  $q$  and  $r$  in the body frame respectively:

$$\dot{\phi} = p, \quad \dot{\theta} = q, \quad \dot{\psi} = r \quad (2.38)$$

In the control system, a quadrotor is controlled by the rotational speeds of four rotors  $\Omega_i$ . As the aerodynamic forces and moments are proportional to the squared rotational speeds of a rotor, the thrust and yawing moment are chosen as the control inputs in the control system:

$$\begin{aligned} u_1 &= \sum_{i=1}^4 T_i, & u_2 &= l(T_2 - T_4), \\ u_3 &= l(-T_1 + T_3), & u_4 &= (-1)^{i+1} \sum_{i=1}^4 Q_i. \end{aligned} \quad (2.39)$$

Then, the rotational speeds  $\Omega_i$  can be computed using  $u_i$  through:

$$u_1 = b \sum_{i=1}^4 \Omega_i^2, \quad u_2 = bl(\Omega_2^2 - \Omega_4^2), \quad (2.40)$$

$$u_3 = bl(-\Omega_1^2 + \Omega_3^2), \quad u_4 = (-1)^{i+1} d \sum_{i=1}^4 \Omega_i^2. \quad (2.41)$$

where  $b$  and  $d$  are the coefficients of the thrust and the yawing moment,  $l$  is the distance between the center of gravity and the center of the rotor.

Thus, a simplified system can be obtained from Equation (2.33):

$$I_{xx}\ddot{\phi} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + u_2 + (-1)^{i+1} \sum_{i=1}^4 R_{mxi}, \quad (2.42a)$$

$$I_{yy}\ddot{\theta} = \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + u_3 + (-1)^{i+1} \sum_{i=1}^4 R_{myi}, \quad (2.42b)$$

$$I_{zz}\ddot{\psi} = \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + u_4, \quad (2.42c)$$

$$m\ddot{z}_w = \sin\theta \sum_{i=1}^4 H_{xi} - (\sin\phi \cos\theta) \sum_{i=1}^4 H_{yi} + (\cos\phi \cos\theta)u_1 - mg + \rho g V_{vol} \quad (2.42d)$$

$$m\ddot{x}_w = -(\cos\theta \cos\psi) \sum_{i=1}^4 H_{xi} - (\sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi) \sum_{i=1}^4 H_{yi} \\ + (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)u_1 \quad (2.42e)$$

$$m\ddot{y}_w = -(\cos\theta \sin\psi) \sum_{i=1}^4 H_{xi} - (\sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi) \sum_{i=1}^4 H_{yi} \\ + (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)u_1 \quad (2.42f)$$

$$T = C_T \rho A (\Omega R_{rad})^2, \quad H = C_H \rho A (\Omega R_{rad})^2 \quad (2.42g)$$

$$Q = C_Q \rho A (\Omega R_{rad})^2 R_{rad}, \quad R = C_R \rho A (\Omega R_{rad})^2 R_{rad} \quad (2.42h)$$

## 2.6 Constraints

Quadrotors have various constraints in applications. The mechanical structure has its limits. Quadrotors usually use brushless electric DC rotors which have limits in their rotational speeds. The Hummingbird quadrotor made by Ascending Technologies (see Figure 1.9 in Section 1.4) is equipped with four brushless DC motor, type HACKER x-BL 52s. The rated motor speed without load is 8330RPM (Revolutions Per Minute), that is 872 rad/s. The rotational speed decreases to two third with load. Here, we define a maximum rotational speed during flights as:

$$10 \text{ rad/s} \leq \Omega \leq 300 \text{ rad/s} \quad (2.43)$$

For the sake of safety, we should also impose some constraints on the speed. The Phantom quadrotor (see Figure 1.11 in Section 1.4) has a maximum tilt angle 45 degrees (0.78 rad), a maximum ascent or descent speed of 6m/s and a maximum flight velocity of 10m/s. We

set a maximum ascent or descending acceleration of 5g ( $g=9.8 \text{ m/s}^2$ ) and a maximum tilt acceleration of 48 rad/s<sup>2</sup>. Therefore, the limits of the control inputs are:

$$\begin{aligned} m \cdot g \leq u_1 \leq 6 m \cdot g = 31.2 \text{ N}, \quad |u_2|, |u_3| \leq 48 I_{xx} = 0.3 \text{ N} \cdot \text{m} \\ |u_4| \leq 48 I_{zz} = 0.5 \text{ N} \cdot \text{m} \end{aligned} \quad (2.44)$$

where  $m$  is the total mass which is set as 0.53 kg in the simulation, and  $I_{xx}$ ,  $I_{yy}$ ,  $I_{zz}$  are the parameters of the moment of inertia,  $I_{xx} = I_{yy} = 6.22 \times 10^{-3} \text{ kg} \cdot \text{m}^2$ ,  $I_{zz} = 1.12 \times 10^{-2} \text{ kg} \cdot \text{m}^2$ .

## 2.7 Control Oriented Models

In Section 1.4, many existing quadrotors are presented. These quadrotors are built using microcontrollers, such as ARM 9. Embedded systems have computational and energy limits. Some parts are neglected to simplify the control laws.

A control oriented model for the backstepping control and sliding mode control is:

$$\begin{aligned} I_{xx} \ddot{\phi} &= u_2 + \dot{\theta} \dot{\psi} (I_{yy} - I_{zz}) + J_r \dot{\theta} \Omega_r, \\ I_{yy} \ddot{\theta} &= u_3 + \dot{\phi} \dot{\psi} (I_{zz} - I_{xx}) - J_r \dot{\phi} \Omega_r, \\ I_{zz} \ddot{\psi} &= u_4 + \dot{\theta} \dot{\phi} (I_{xx} - I_{yy}), \\ m \ddot{x} &= (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) u_1, \\ m \ddot{y} &= (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) u_1, \\ m \ddot{z} &= -mg + (\cos \theta \cos \phi) u_1. \end{aligned} \quad (2.45)$$

A control oriented model for the model free control is:

$$\begin{aligned} I_{xx} \ddot{\phi} &= u_2 + F_\phi, \\ I_{yy} \ddot{\theta} &= u_3 + F_\theta, \\ I_{zz} \ddot{\psi} &= u_4 + F_\psi, \\ m \ddot{x} &= (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) u_1 + F_x, \\ m \ddot{y} &= (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) u_1 + F_y, \\ m \ddot{z} &= (\cos \theta \cos \phi) u_1 + F_z. \end{aligned} \quad (2.46)$$

A partial knowledge model is used in the model free control. The coefficients of the input  $u_1$  are kept in the control oriented model to keep the control precision. Other parts in the

system are included into the unknown part  $F$ . Based on the type of a system, the control oriented model for the model free control can be fully model-free or with partial knowledge. If we set the coefficients of the input  $u_1$  as constants, then a fully model free control is used.

The PID control does not need particularly the knowledge of system, and it does not need a control oriented model.

# Chapter 3

## Dynamic Analysis

### 3.1 Introduction

In this chapter, we will use the normal form theory for the dynamic analysis. The normal form theory is a useful approach in studying the dynamic system properties [58]. By employing successive coordinate transformations, a system is changed into its simplest form, which exhibits all possible properties of the original system. This makes the analysis of the dynamic system easier.

In linear systems, different kinds of the normal forms were deduced and the Brunovsky form is used for the design of the control laws. Linear systems have four normal forms: controllable, observable, controller and observer form. The normal forms for nonlinear systems have been addressed firstly by Poincaré and then have been studied during the last century. His technique has been used in the area of nonlinear vector fields, Hamilton dynamic systems, nonlinear mappings and bifurcation phenomenon, etc. In the last decades, the transformation of a nonlinear system into a Brunovsky using successive changes of coordinates and state feedback was studied by many authors. The normal forms for linear and nonlinear systems have been proposed by Krener in 1987 [59]. However, most of the nonlinear systems do not admit a Brunovsky form or the transformation of a system into the Brunovsky form is numerically quite difficult which needs to solve the first order linear partial differential equations. The approximate versions of the nonlinear controller and observer normal forms were proposed by Krener et al. in 1987 [60]. A set of extended quadratic controller normal forms of the linearly controllable systems with single input was given by Kang and Krener in 1992 [62]. They have also proved there exists a dynamic state feedback so that the extended system has a linear approximation which is accurate at least second degree. In 2006, Kang

et al. have proposed a method to deduce the normal forms of any degree for the systems with a single input by using successive changes of coordinates and feedback [63]. For multi-input systems, Tall et al. have deduced the normal forms for the systems with two inputs in 2002 [95].

Based on the normal forms, the bifurcations of a system have been studied by several authors. A local bifurcation of a parameterized system occurs at an equilibrium where there is a change in the stability. For the differential equations depending on a single parameter, the typical bifurcations are fully understood: the fold (or saddle node), the transcritical and the Hopf bifurcation [58]. Krener et al. have presented the quadratic and cubic normal forms of a scalar input nonlinear control system around an equilibrium point. Some important control bifurcations, the analogues of the classical fold, transcritical and Hopf bifurcations were studied [61]. Center manifold is usually applied with the normal forms theory. It reduces the system to a center manifold associated with parts of the system whose eigenvalues have zero real parts at a bifurcation point [38].

To the best of our knowledge, the normal forms and center manifold theories have never been used in the analysis of a quadrotor. In this chapter, the normal form of the quadrotor system is firstly calculated. By using such a methodology, the highly coupled parts in the quadrotor system are eliminated. Under certain control laws, the normal form is reduced into a two dimensional system at the bifurcation point by using the center manifold theory. A former research can be found in [101].

## 3.2 Quadrotor Model

The kinematic and dynamic models are presented in Chapter 2. All the definitions of the variables can be found in Chapter 2. A rotational sequence ZYX is chosen for the Tait-Bryan angles. Thus, the rotational matrix from the body frame to the world frame  $\mathcal{R}_b^w$  is presented:

$$\mathcal{R}_b^w = \begin{bmatrix} \cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ \cos\theta \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\ -\sin\theta & \sin\phi \cos\theta & \cos\phi \cos\theta \end{bmatrix} \quad (3.1)$$

In the dynamic analysis, only the thrust  $T$  and the yawing moment are considered and all other aerodynamical forces and moments are neglected. As the thrusts are along the  $z_b$

axis in the body frame, the system can be expressed as:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \mathcal{R}_b^w \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} = \begin{bmatrix} (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) T \\ (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) T \\ \cos\phi \cos\theta T - g \end{bmatrix} \quad (3.2)$$

In Equation (3.1), the third row is the simplest than other rows and columns. Therefore, we can choose the rotational sequence XYZ for the Tait-Bryan angles to simplify the system. The rotational sequence is inverse, and the matrix  $\mathcal{R}_b^w$  will be the inverse matrix. The third row is now changed to the third column, and will be multiplied with the thrust. Note that the rotational sequence will not change the dynamics of the system. Thus, the system is further simplified, and the model of a quadrotor is shown in Equation (3.3).

$$\begin{aligned} \ddot{x} &= -w_1 \sin\theta, & \ddot{\phi} &= w_2, \\ \ddot{y} &= w_1 \cos\theta \sin\phi, & \ddot{\theta} &= w_3, \\ \ddot{z} &= w_1 \cos\theta \cos\phi - g, & \ddot{\psi} &= w_4. \end{aligned} \quad (3.3)$$

The variables  $w_i (i = 1..4)$  are the linear and angular accelerations induced by four rotors, which are the inputs of the system. The rotation angles  $\phi$ ,  $\theta$  and  $\psi$  are along the world axis  $x$ ,  $y$  and  $z$  respectively, namely roll, pitch and yaw.  $g = 9.8\text{m/s}^2$  is the gravity coefficient.

We introduce the variables as  $x_1 = x$ ,  $x_2 = \dot{x}$ ,  $x_3 = y$ ,  $x_4 = \dot{y}$ ,  $x_5 = z$ ,  $x_6 = \dot{z}$ ,  $x_7 = \phi$ ,  $x_8 = \dot{\phi}$ ,  $x_9 = \theta$ ,  $x_{10} = \dot{\theta}$ ,  $x_{11} = \psi$ ,  $x_{12} = \dot{\psi}$ . Therefore, the system is rewritten as:

$$\begin{aligned} \dot{x}_1 &= x_2, & \dot{x}_7 &= x_8, \\ \dot{x}_2 &= -w_1 \sin(x_9), & \dot{x}_8 &= w_2, \\ \dot{x}_3 &= x_4, & \dot{x}_9 &= x_{10}, \\ \dot{x}_4 &= w_1 \cos(x_9) \sin(x_7), & \dot{x}_{10} &= w_3, \\ \dot{x}_5 &= x_6, & \dot{x}_{11} &= x_{12}, \\ \dot{x}_6 &= w_1 \cos(x_9) \cos(x_7) - g, & \dot{x}_{12} &= w_4. \end{aligned} \quad (3.4)$$

### 3.3 Normal Form

In this section, some nonlinear changes of coordinates and inputs for the quadrotor system will be deduced. The normal form of the original system of any degree will be found. A

normal form is a simplest form of the original system, which exhibits all possible properties of the original system.

The equilibria of the system (3.4) are:

$$\mathbf{x}_e = (c_1, 0, c_2, 0, c_3, 0, k\pi, 0, k\pi, 0, c_4, 0), \quad \mathbf{w} = (g, 0, 0, 0),$$

where  $c_i (i = 1, \dots, 4) \in \mathbb{R}$  are constants,  $k = 0, \pm 1, \pm 2, \dots$  is integer and  $g$  is the gravity.

Note that the angles are limited in the real control system, the tilt angle of a quadrotor is limited from -90 degrees to 90 degrees,  $\phi, \theta \in (-\pi/2, \pi/2)$ . The yaw angle is limited to 360 degrees,  $\psi \in [0, \pi)$ . Therefore, the variable  $k$  here is equal to 0. Without losing generality, the constants  $c_i (i = 1, \dots, 4)$  can be set as 0. Therefore, only one equilibrium is considered:

$$\mathbf{x}_0 = (\mathbf{x}, \mathbf{w}) = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, g, 0, 0, 0)$$

We move  $\mathbf{x}_0$  to the origin by changing the coordinates of the inputs:

$$\begin{aligned} w_1 &= u_1 + g, & w_2 &= u_2 \\ w_3 &= u_3, & w_4 &= u_4 \end{aligned} \tag{3.5}$$

Then, using the Taylor series of the functions  $\sin(x)$  and  $\cos(x)$  at  $x = 0$ , the system can be further written in polynomial form as follows:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -gx_9 - u_1x_9 + \frac{gx_9^3}{6} + \frac{u_1x_9^3}{6} + O^5, \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= gx_7 + u_1x_7 - \frac{gx_7^3}{6} - \frac{gx_9^2x_7}{2} - \frac{u_1x_7^3}{6} - \frac{u_1x_9^2x_7}{2} + O^5, \\ \dot{x}_5 &= x_6, \\ \dot{x}_6 &= u_1 - \frac{gx_7^2}{2} - \frac{gx_9^2}{2} - \frac{u_1x_7^2}{2} - \frac{u_1x_9^2}{2} + \frac{gx_7^4}{24} + \frac{gx_9^2x_7^2}{4} + \frac{gx_9^4}{24} + O^5, \\ \dot{x}_7 &= x_8, & \dot{x}_8 &= u_2, & \dot{x}_9 &= x_{10}, \\ \dot{x}_{10} &= u_3, & \dot{x}_{11} &= x_{12}, & \dot{x}_{12} &= u_4. \end{aligned} \tag{3.6}$$

Here,  $O^5$  are the polynomials with 5th and higher degree.

Using the following state and input transformation:

$$\begin{aligned}
y_1 &= x_1, & y_2 &= x_2, & y_3 &= x_3, & y_4 &= x_4 \\
y_5 &= x_5, & y_6 &= x_6, & y_7 &= gx_7, & y_8 &= gx_8 \\
y_9 &= gx_9, & y_{10} &= gx_{10}, & y_{11} &= x_{11}, & y_{12} &= x_{12} \\
v_1 &= u_1, & v_2 &= gu_2, & v_3 &= gu_3, & v_4 &= u_4
\end{aligned} \tag{3.7}$$

The system (3.6) is then changed into the Brunovsky form:

$$\begin{aligned}
\dot{y}_1 &= y_2, \\
\dot{y}_2 &= -y_9 - \frac{v_1 y_9}{g} + \frac{y_9^3}{6g^2} + \frac{v_1 y_9^3}{6g^3} + O^5, \\
\dot{y}_3 &= y_4, \\
\dot{y}_4 &= y_7 + \frac{v_1 y_7}{g} - \frac{y_7 y_9^2}{2g^2} - \frac{y_7^3}{6g^2} - \frac{v_1 y_7^3}{6g^3} - \frac{v_1 y_7 y_9^2}{2g^3} + O^5, \\
\dot{y}_5 &= y_6, \\
\dot{y}_6 &= v_1 - \frac{y_7^2}{2g} - \frac{y_9^2}{2g} - \frac{v_1 y_7^2}{2g^2} - \frac{v_1 y_9^2}{2g^2} + \frac{y_7^4}{24g^3} + \frac{y_7^2 y_9^2}{4g^3} + \frac{y_9^4}{24g^3} + O^5, \\
\dot{y}_7 &= y_8, & \dot{y}_8 &= v_2, & \dot{y}_9 &= y_{10}, \\
\dot{y}_{10} &= v_3, & \dot{y}_{11} &= y_{12}, & \dot{y}_{12} &= v_4.
\end{aligned} \tag{3.8}$$

The system (3.8) can also be written as follows:

$$\begin{aligned}
\dot{y} &= f(y) + g(y)v \\
&= Ay + f^{(2)}(y) + f^{(3)}(y) + Bv + g^{(1)}(y)v + g^{(2)}(y)v + O^4
\end{aligned} \tag{3.9}$$

where  $A$ ,  $B$  are the coefficients of the linear parts,  $f^{(2)}(y)$ ,  $g^{(1)}(y)v$  are the second degree homogeneous polynomials of the system,  $f^{(3)}(y)$ ,  $g^{(2)}(y)v$  are the third degree homogeneous polynomials,  $O^4$  are the polynomials with 4th and higher degree.

Now we are going to obtain a third degree normal form of the quadrotor system. Firstly, we take a third-degree homogeneous transformation for example [60]:

$$y = z + \phi^{(2)}(z) + \phi^{(3)}(z) \tag{3.10}$$

which  $z$  are the new states of the system.  $\phi^{(2)}(z)$  is a second degree homogeneous polynomial and  $\phi^{(3)}(z)$  is a third degree homogeneous polynomial of the states  $z$ , whose coefficients will be defined later.

We get the derivative of Equation (3.10). Therefore, the derivative of the new states  $z$  is:

$$\dot{z} = \left(I + \frac{d\phi^{(2)}}{dz} + \frac{d\phi^{(3)}}{dz}\right)^{-1} \dot{y} \quad (3.11)$$

where

$$\left(I + \frac{d\phi^{(2)}}{dz} + \frac{d\phi^{(3)}}{dz}\right)^{-1} = I - \frac{d\phi^{(2)}}{dz} - \frac{d\phi^{(3)}}{dz} + \left(\frac{d\phi^{(2)}}{dz}\right)^2 + \left(\frac{d\phi^{(3)}}{dz}\right)^2 + 2\frac{d\phi^{(2)}}{dz} \frac{d\phi^{(3)}}{dz} \dots$$

In Equation (3.9), we rewrite the  $f(y)$  and  $g(y)$  using the new states  $z$ .

$$\begin{aligned} f(y) &= A(y) + f^{(2)}(y) + f^{(3)}(y) + O^4 \\ &= Az + A\phi^{(2)}(z) + f^{(2)}(z) + A\phi^{(3)}(z) + f^{(3)}(z) \dots \\ g(y) &= B + g^{(1)}(y) + g^{(2)}(y) + O^3 = B + g^{(1)}(z) + g^{(1)}(\phi^{(2)}(z)) + g^{(2)}(z) \dots \end{aligned} \quad (3.12)$$

Therefore, with the help of Equations (3.9), (3.11), we have the new system:

$$\begin{aligned} \dot{z} &= Az + Bv + A\phi^{(2)}(z) + f^{(2)}(z) + g^{(1)}(z)v - \frac{d\phi^{(2)}}{dz}Az - \frac{d\phi^{(2)}}{dz}Bv + A\phi^{(3)}(z) \\ &\quad + f^{(3)}(z) + g^{(2)}(z)v + g^{(1)}(\phi^{(2)}(z))v - \frac{d\phi^{(2)}}{dz}(A\phi^{(2)}(z) + f^{(2)}(z) + g^{(1)}(z)v) \\ &\quad - \frac{d\phi^{(3)}}{dz}(Az + Bv) + \left(\frac{d\phi^{(2)}}{dz}\right)^2(Az + Bv) + O^4 \end{aligned} \quad (3.13)$$

To simplify of the system, the states  $z$  and the inputs  $v$  should be separated, which means the polynomial  $g^{(1)}(z)v$ ,  $g^{(2)}(z)v$  should be canceled in the third degree normal form. Thus we have:

$$\begin{aligned} g^{(1)}(z) - \frac{d\phi^{(2)}}{dz}B &= 0 \\ g^{(2)}(z) + g^{(1)}(\phi^{(2)}(z)) - \frac{d\phi^{(2)}}{dz}g^{(1)}(z) - \frac{d\phi^{(3)}}{dz}B + \left(\frac{d\phi^{(2)}}{dz}\right)^2B &= 0 \end{aligned}$$

Therefore, the transformation in Equation (3.10) can be obtained:

$$\begin{aligned} \phi^{(2)}(z) &= \left(0, -\frac{z_6 z_9}{g}, 0, \frac{z_6 z_7}{g}, 0, 0, 0, 0, 0, 0, 0\right), \\ \phi^{(3)}(z) &= \left(0, 0, 0, 0, 0, -\frac{z_6 z_7^2}{2g^2} - \frac{z_6 z_9^2}{2g^2}, 0, 0, 0, 0, 0\right). \end{aligned}$$

Using the same method, we can calculate the normal form of any degree:

$$\begin{aligned}
\dot{z}_1 &= z_2 - \frac{z_6 z_9}{g} + \frac{2z_6 z_9^3}{3g^3} + \frac{z_6 z_7^2 z_9}{2g^3} + O^5, \\
\dot{z}_2 &= -z_9 + \frac{z_6 z_{10}}{g} - \frac{z_9^3}{3g^2} - \frac{z_7^2 z_9}{2g^2} - \frac{z_6 z_7^2 z_{10}}{2g^3} + \frac{z_6 z_7 z_8 z_9}{g^3} + O^5, \\
\dot{z}_3 &= z_4 + \frac{z_6 z_7}{g} - \frac{z_6 z_7 z_9^2}{g^3} - \frac{2z_6 z_7^3}{3g^3} + O^5, \\
\dot{z}_4 &= z_7 - \frac{z_6 z_8}{g} + \frac{z_7^3}{3g^2} + \frac{z_6 z_8 z_9^2}{g^3} + O^5, \\
\dot{z}_5 &= z_6 - \frac{z_6 z_7^2}{2g^2} - \frac{z_6 z_9^2}{2g^2} + O^5, \\
\dot{z}_6 &= v_1 - \frac{z_7^2}{2g} - \frac{z_9^2}{2g} + \frac{z_6 z_7 z_8}{g^2} + \frac{z_6 z_9 z_{10}}{g^2} + \frac{5z_7^4}{24g^3} - \frac{z_7^2 z_9^2}{4g^3} - \frac{5z_9^4}{24g^3} + O^5, \\
\dot{z}_7 &= z_8, & \dot{z}_8 &= v_2, & \dot{z}_9 &= z_{10}, \\
\dot{z}_{10} &= v_3, & \dot{z}_{11} &= z_{12}, & \dot{z}_{12} &= v_4.
\end{aligned} \tag{3.14}$$

Here,  $O^5$  are the polynomials with 5th and higher degree.

A Maple package ‘QualitativeODE’ has been made for calculating a normal form of the quadrotor. The programs can be found in Appendix B.

## 3.4 Linear Analysis

From the normal form, we can easily get the linearized quadrotor system. In this section, we will analyze the controllability and the stability of the linearized system. A linear static feedback control is also proposed to show its influence on the system’s stability.

### 3.4.1 Controllability

The linearized quadrotor system is presented in following form:

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{v} \tag{3.15}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, the rank of the controllability matrix  $\mathcal{C}$  is:

$$\mathcal{R}ank(\mathcal{C}) = \mathcal{R}ank\left(\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}\right) = 12 \quad (3.16)$$

Therefore, the linearized system is fully controllable at its equilibrium, the origin of the normal form. If the linearized nonlinear system is controllable at its equilibrium, then the nonlinear system is accessible at its equilibrium.

### 3.4.2 Stability

To check the stability of the linearized system, we calculate the eigenvalues of the matrix  $A$  in Equation (3.15). They turn out to be all 0. The linearized system is marginally stable. Therefore, we can use linear static feedback to control the stability of the system.

If the system is time invariant, the indirect method of Lyapunov says that if the eigenvalues of Jacobian matrix of the system at the origin are in the open left half complex plane, then the origin is asymptotically stable. Therefore, we can define the state feedback as follows to move all the eigenvalues of the system to the open left half plane.  $x_r, y_r, z_r, \psi_r$  are the references.

$$\begin{aligned} v_1 &= -256(z_5 - z_r) - 32z_6, \\ v_2 &= -1700(z_3 - y_r) - 1000z_4 - 256z_7 - 32z_8 \\ v_4 &= -256(z_{11} - \psi_r) - 32z_{12}, \\ v_3 &= 1700(z_1 - x_r) + 1000z_2 - 256z_9 - 32z_{10} \end{aligned} \quad (3.17)$$

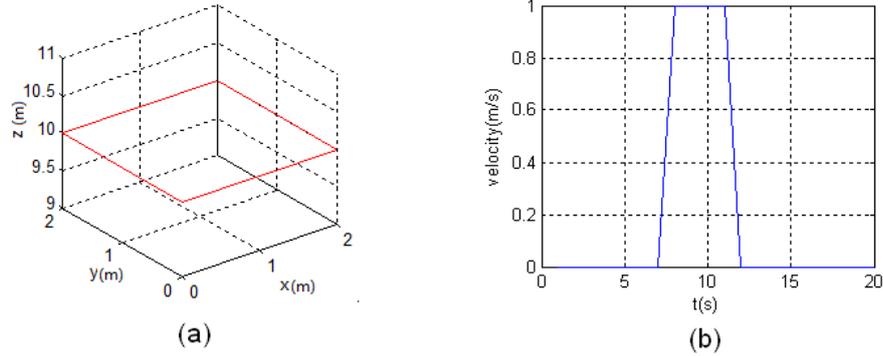


Figure 3.1: (a) Reference trajectory for the quadrotor. (b) Wind disturbance.

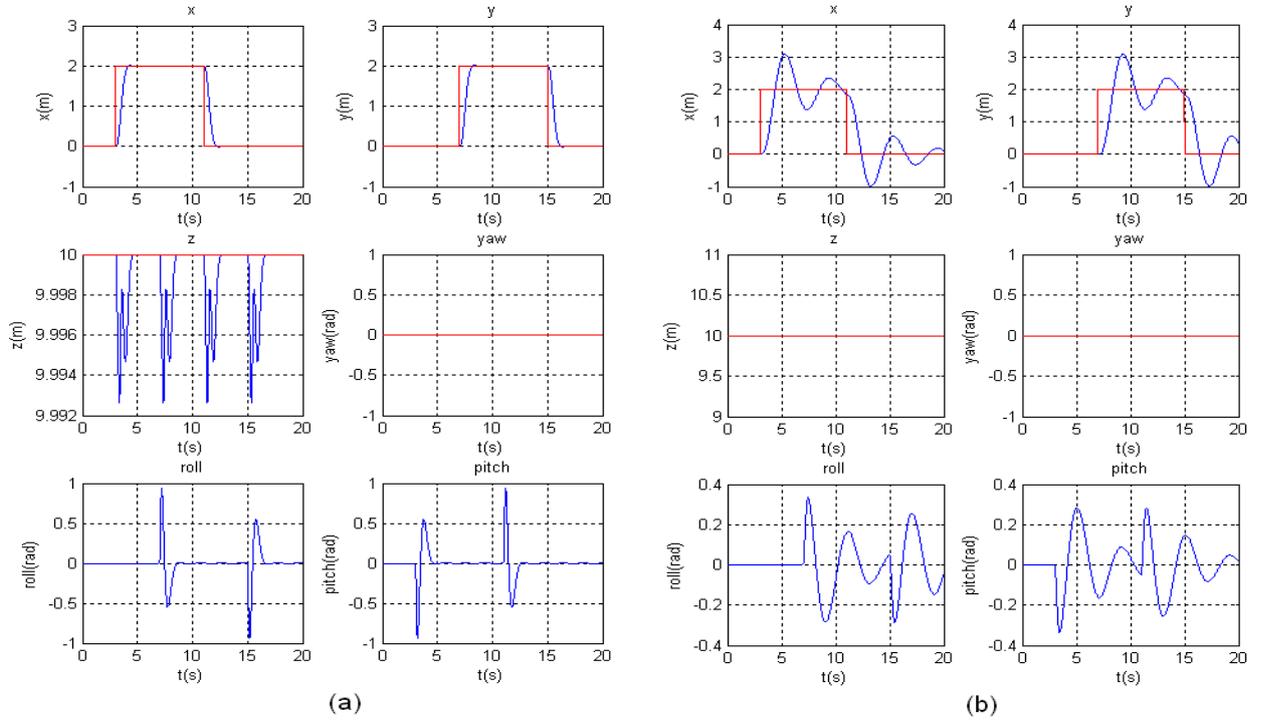


Figure 3.2: The simulation without wind disturbance: (a) the linear static feedback control. (b) a standard PID control.

In order to verify the stability of the control system, some simulations are made in the condition without and with wind disturbance. The simulation task is to let a quadrotor follow a square path with a length of 2m while hovering at an altitude of 10m, which is given in Figure 3.1. The desired response time is 1s, which means the quadrotor should fly 2m within 1 second. The totally sample time is 20s. For comparison, the simulations using a standard PID control are also given.

The simulation results without wind disturbance are given in Figure 3.2. We can see

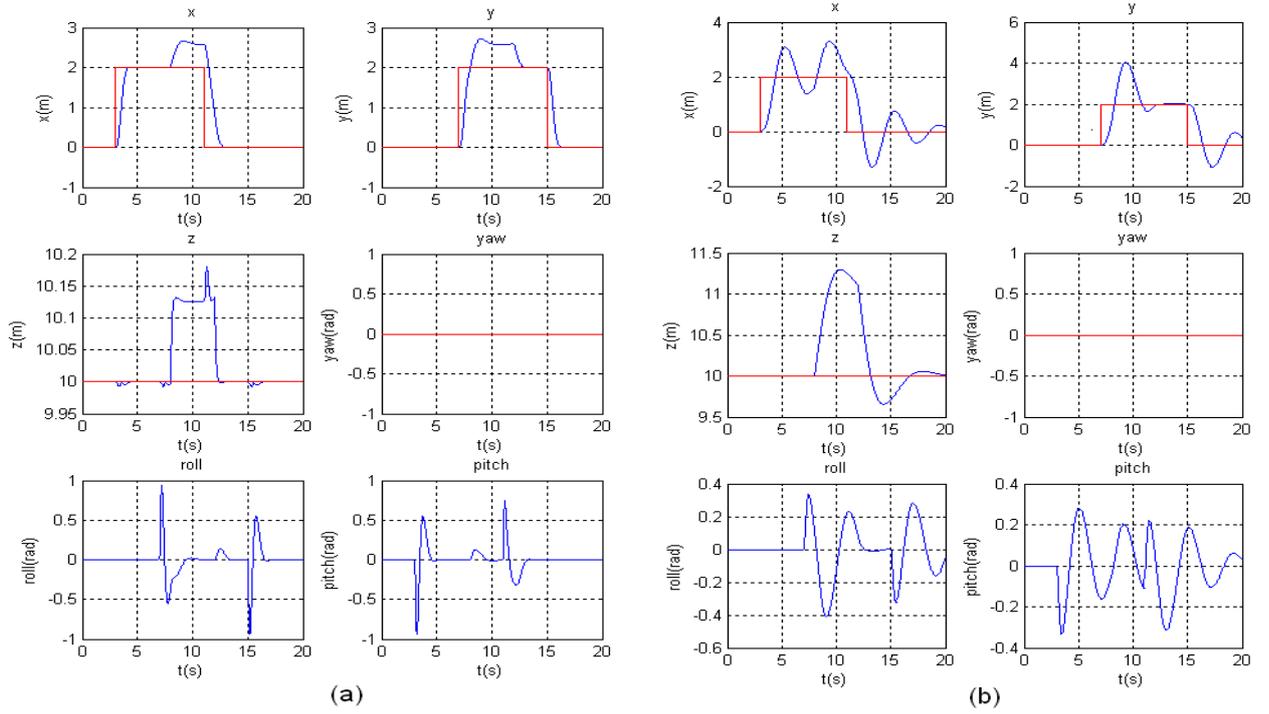


Figure 3.3: The simulation with wind disturbance: (a) the linear static feedback control. (b) a standard PID control.

that the linear static feedback control has better performance than a standard PID control. During the trajectory, there may be wind disturbance with velocity 1m/s as in Figure 3.1, which occurs in all  $x$ ,  $y$  and  $z$  axis. The simulation results are given in Figure 3.3. The desired response time is 1s. The linear static feedback can keep the stability during the wind disturbance, and has better performance than the standard PID control.

## 3.5 Nonlinear Analysis

In this section, the bifurcations of the quadrotor system is analyzed, and the system can be further simplified using the center manifold theory.

### 3.5.1 Bifurcations

It is easy to see that in the linear part of the equation (3.14),  $z_1$  is related only to  $z_2, z_9, z_{10}, v_3$ ;  $z_3$  is related to  $z_4, z_7, z_8, v_2$ ;  $z_5$  is related to  $z_6, v_1$ ;  $z_{11}$  is related to  $z_{12}, v_4$ . Therefore, the

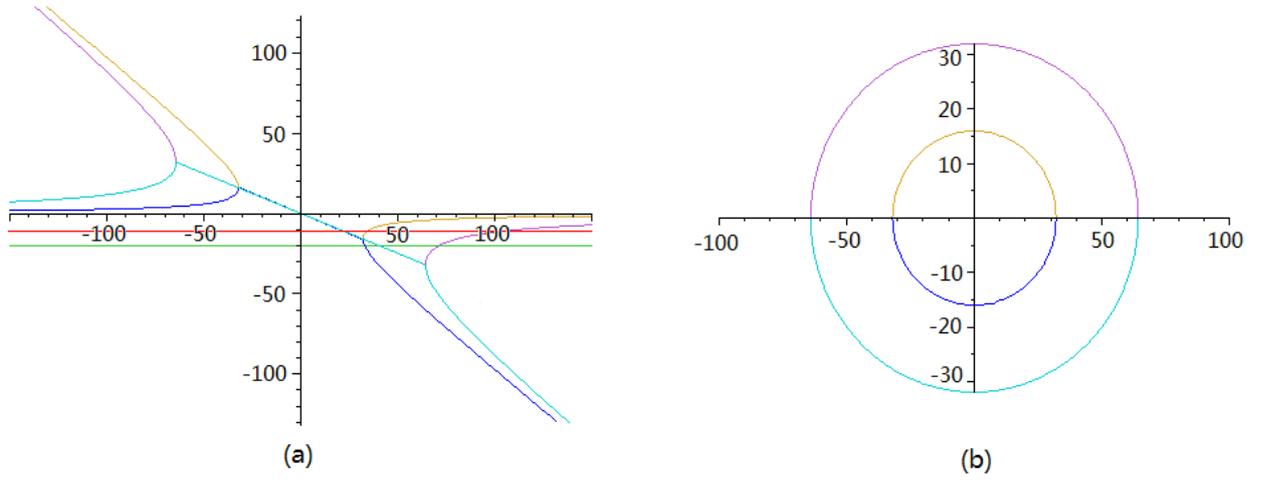


Figure 3.4: The eigenvalues when  $K_{12}$  changes from -150 to 150: (a) the real parts. (b) the imaginary parts.

control laws can be defined as:

$$\begin{aligned}
 v_1 &= K_{11}z_5 + K_{12}z_6, \\
 v_2 &= K_{21}z_3 + K_{22}z_4 + K_{23}z_7 + K_{24}z_8, \\
 v_4 &= K_{41}z_{11} + K_{42}z_{12}, \\
 v_3 &= K_{31}z_1 + K_{32}z_2 + K_{33}z_9 + K_{34}z_{10}.
 \end{aligned} \tag{3.18}$$

In this way, we can move the related eigenvalues in each group separately without changing the eigenvalues in other groups. Here, we define  $v_i (i = 1, \dots, 4)$  as:

$$\begin{aligned}
 v_1 &= -256z_5 + K_{12}z_6, \\
 v_2 &= -100z_3 - 308z_4 - 256z_7 - 32z_8, \\
 v_4 &= -1024z_{11} + K_{42}z_{12}, \\
 v_3 &= 100z_1 + 308z_2 - 256z_9 - 32z_{10}.
 \end{aligned} \tag{3.19}$$

The system has three equilibria:

$$\begin{aligned}
 \mathbf{P}_1^e &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\
 \mathbf{P}_2^e &= (0, 0, 43.45, 0, -0.057, 0, -16.97, 0, 0, 0, 0, 0) \\
 \mathbf{P}_3^e &= (0, 0, -43.45, 0, -0.057, 0, 16.97, 0, 0, 0, 0, 0)
 \end{aligned} \tag{3.20}$$

However, only the origin  $P_1^e$  can be stable when  $K_{12}, K_{42}$  change.

At the equilibrium  $P_1^e$ , for simplicity  $K_{12} = K_{42}$ , when  $K_{12}$  changes, the real and imaginary parts of the eigenvalues are in Figure 3.4. When  $K_{12} < 0$ , the system has four eigenvalues with positive real parts, and the system becomes unstable. When  $K_{12} > 0$ , the system has all eigenvalues with negative real parts, and the system is asymptotically stable. When  $K_{12} = 0$ , the system has two pairs of pure imaginary eigenvalues  $\pm 16i$  and  $\pm 32i$ , and all other eigenvalues have negative real parts, which is a four dimensional center manifold. The stability cannot be determined by the linear part of the system. It depends on the nonlinearity of the system.

### 3.5.2 Center Manifold

The aim of this part is to get the reduced system which can determine the stability and possible local bifurcations of the system at one bifurcation point. A system can be written as:

$$\dot{x} = A(b)x + F(x), \quad x \in R^n$$

where  $b$  is a free parameter,  $b \in R$ .

At its origin  $x = [0, \dots, 0]$ ,  $J(b)$  is the Jordan form of the matrix  $A(b)$  and  $Q$  is a matrix which enables  $Q(b)J(b)Q^{-1}(b) = A(b)$ . Therefore, we have:

$$\begin{aligned} \dot{x} &= Q(b)J(b)Q^{-1}(b)x + F(x) \\ Q^{-1}(b)\dot{x} &= J(b)Q^{-1}(b)x + Q^{-1}(b)F(x) \end{aligned}$$

we define  $y = Q^{-1}(b)x$ , then

$$\dot{y} = J(b)y + Q^{-1}(b)F(Q(b)y) = J(b)y + \tilde{F}(y) \quad (3.21)$$

we can separate the Jordan matrix  $J$  as matrices  $B$  and  $C$  whose eigenvalues have zero real parts and negative real parts respectively. Therefore, we can rewrite the system (3.21) at the origin with  $x = [0, \dots, 0]$ .

$$\dot{y}_0 = By_0 + f(y_0, y_-), \quad \dot{y}_- = Cy_- + g(y_0, y_-).$$

Since the center manifold is tangent to  $E^c$  (the  $y_- = 0$  space), we define

$$y_- = h(y_0, b), \quad h(0, 0) = Dh(0, 0) = 0, \quad \dot{b} = 0. \quad (3.22)$$

We can calculate the function  $h(y_0, b)$  by using

$$\dot{y}_- = Dh(y_0, b)\dot{y}_0 = Dh(y_0, b)[By_0 + f(y_0, h(y_0, b))] = Cy_0 + g(y_0, h(y_0, b))$$

Therefore, we can get the local evolution equations of  $y_0$  which can determine the stability of the original system.

In quadrotor center manifold analysis, the control laws are defined as:

$$\begin{aligned} v_1 &= -256z_5 - bz_6 - z_5^3, \\ v_2 &= -100z_3 - 308z_4 - 256z_7 - 32z_8, \\ v_4 &= -10z_{11} - 24z_{12}, \\ v_3 &= 100z_1 + 308z_2 - 256z_9 - 32z_{10}. \end{aligned}$$

The bifurcation of the system is like in previous subsection. When  $b < 0$ , the system has two eigenvalues with positive real parts. When  $b > 0$ , the system has all eigenvalues with negative real parts. When  $b = 0$ , the system has two pure imaginary eigenvalues  $\pm 16i$ , and all other eigenvalues have negative real parts. The stability depends on the nonlinear parts of the system. We can use the center manifold theory to simplify the system, and further simplify the study of the bifurcation of the system. In this control system,  $y_0 = [y_1, y_2]^T = [z_5, z_6]^T$  and  $y_- = [y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}]^T = [z_1, z_2, z_3, z_4, z_7, z_8, z_9, z_{10}, z_{11}, z_{12}]^T$ .

We seek a quadratic center manifold ( $a$  are parameters to be defined later):

$$y_i = a_{i200}y_1^2 + a_{i020}y_2^2 + a_{i002}b^2 + a_{i110}y_1y_2 + a_{i101}y_1b + a_{i011}y_2b, \quad i = 3..12$$

Using the method mentioned before, we get:

$$h(y_0, b) = [-0.62b^2, -0.62b^2, -0.76b^2, -0.76b^2, 0, 0, 0, 0, -0.42b^2, -23.58b^2] \quad (3.23)$$

Therefore, the reduced system on the center manifold can be written:

$$\begin{aligned} \dot{y}_1 &= 16y_2 - 0.41b^4 - 0.011b^8 - (b + 0.057b^4)y_1 + 0.00024y_2^3 \\ \dot{y}_2 &= -16y_1 + 0.67b^4y_1 \end{aligned} \quad (3.24)$$

In the reduced system, when  $b$  is positive (negative), the origin is a stable (unstable) focus. When  $b = 0$ , the origin is a center. The phase portrait of Equation (3.24) when  $b = -0.5$ ,  $b = 0$  and  $b = 0.5$  are depicted in Figure 3.5.

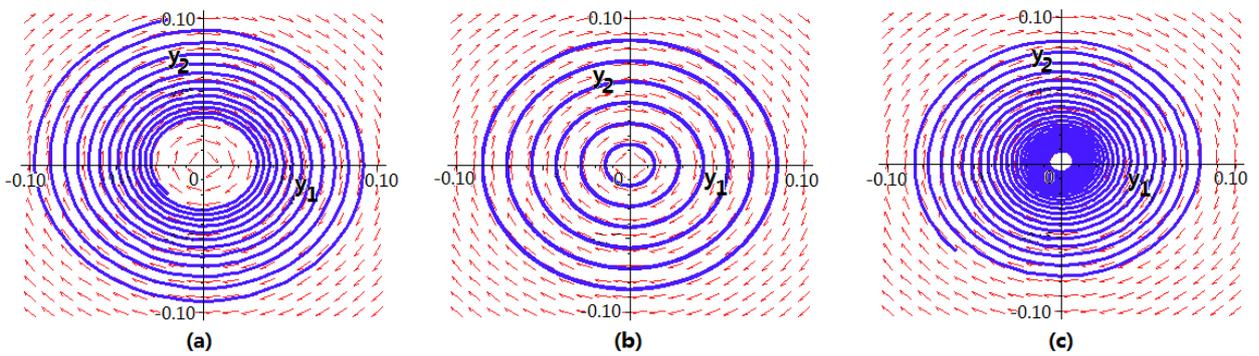


Figure 3.5: The phase portrait of the reduced system: (a)  $b=-0.5$ . (b)  $b=0$ . (c)  $b=0.5$ .

# Chapter 4

## System Control

### 4.1 Scenarios

Due to the simplicity in design and maneuver, quadrotors have many applications, such as border patrol, surveillance, search & rescue, aerial photography and video etc. As discussed intensively during the applications of quadrotors in Chapter 1, we list here the points that a control system should have in applications:

- Complexity of the control algorithm: the number of the addition and multiplication operations in the control law.
- Maximum absolute tracking error.
- Variance of the tracking error.
- Robustness to model uncertainties: the change in the tracking error.
- Robustness to disturbances: the change in the tracking error.
- Adjusting time to disturbances: the time needed to return to the stable state.
- Energy consumption: the energy spent on motion control.

In this thesis, without losing generality, an aerial photography scenario is selected. Firstly, a basic scenario without disturbance is presented. However, in real applications as discussed above, circumstances can be rather turbulent, which results a totally different effect on quadrotor systems comparing to the scenario without disturbance. Usually in scientific researches, control methods are only tested in the basic scenario without disturbance. The

good performance in these conditions can not prove the same good performance in the scenario with disturbance. A simple example is given in Chapter 1. A Backstepping control has small maximum absolute tracking errors in an ideal case. However, it loses its stability very quickly when there is wind disturbance. Therefore, in this thesis, in order to get closer to the realistic applications as mentioned above, several other scenarios with wind disturbance, model parameter uncertainties, sensor noises and actuator faults are also presented.

### 4.1.1 Basic Scenario

In this scenario, a small quadrotor with a 4 cell 2700mAh Lithium Polymer battery and an onboard camera is used. The task is the aerial photo shooting in an open space yard. The quadrotor needs to fly to four points, and at each point the quadrotor needs to take several photos. These four points are the four vertices of a square path with the length of 2m. The quadrotor hovers at the altitude of 10m. The 3D trajectory is given in Figure 4.1.

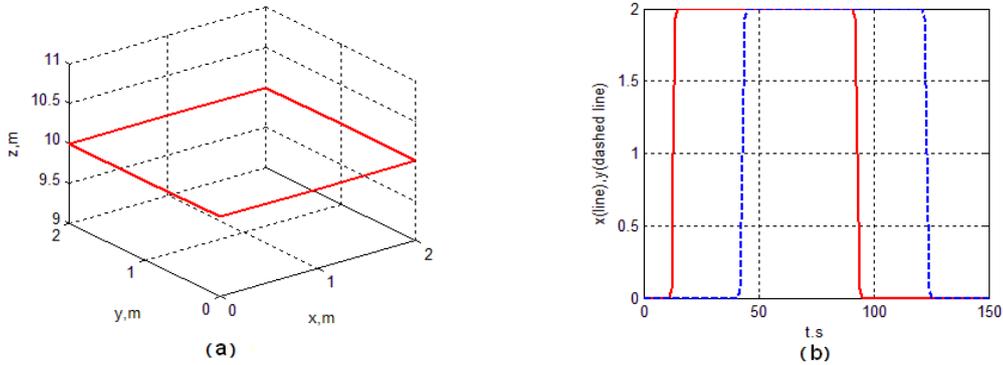


Figure 4.1: (a) 3D reference trajectory for the quadrotor. (b) Reference trajectory along the x axis (red, line) and y axis (blue, dashed line).

When arrived at each corner, the quadrotor hovers for about 15s to take pictures, then flies to the next corner. The reference trajectory is expressed in Equation (4.1). The parameter  $h_d = 2\text{m}$  is the length of the trajectory, and  $T_f = 6\text{s}$  is the time needed to reach the desired corner. The total simulation time is 150s. In the basic scenario, we consider the quadrotor is in an ideal situation without environmental disturbance, model errors and sensor disturbance, etc.

$$\sigma(t) = \begin{cases} 0 & \text{when } 0 \leq t \leq t_1, t_4 < t \leq 150\text{s}, \\ h_d \frac{(t - t_1)^5}{(t - t_1)^5 + (T_f - t + t_1)^5} & \text{when } t_1 < t \leq t_2, \\ 2 & \text{when } t_2 < t \leq t_3, \\ h_d - h_d \frac{(t - t_3)^5}{(t - t_3)^5 + (T_f - t + t_3)^5} & \text{when } t_3 < t \leq t_4. \end{cases} \quad (4.1)$$

$$h_d = 2\text{m}, \quad T_f = 6\text{s}.$$

$$x = \sigma(t), \quad \text{with } t_1 = 10\text{s}, t_2 = 16\text{s}, t_3 = 90\text{s}, t_4 = 96\text{s}.$$

$$y = \sigma(t), \quad \text{with } t_1 = 40\text{s}, t_2 = 46\text{s}, t_3 = 120\text{s}, t_4 = 126\text{s}.$$

$$z = 10\text{m}.$$

### 4.1.2 Wind Disturbance

This scenario is dedicated to test the stability of control systems while encountering environment disturbance, namely wind. Quadrotors are easy to be affected by external disturbances while its flight. As the shooting is taken place at an open space yard, the most general disturbance is the wind.

Several wind models are proposed in literature [41, 51]. The most general model is the wind force model where wind imposes extra forces on the system [51]. The force induced by wind can be expressed as:

$$F_w = k_w A_w v_w^2 \quad (4.2)$$

where  $v_w$  is the wind velocity,  $A_w$  is the area of the surface facing the wind,  $k_w$  is a coefficient,  $k_w = 0.0022 \text{ N}\cdot\text{s}^2/\text{m}^2$ .

A another model is the wind torque model [51]. Wind disturbance can also induce extra torques on the system which is expressed as:

$$T_w = c_w A_w D \alpha_w v_w^2 \quad (4.3)$$

where  $v_w$  is the wind velocity,  $A_w$  is the area of the surface facing the wind,  $D$  is the diameter of the area facing the wind,  $\alpha_w$  is the static air density,  $c_w$  is a torque coefficient.

As the surface and dimension of quadrotors are relatively small. The torque induced by wind can be neglected. In this scenario, only the force induced by wind is considered. Here, we suppose the wind caused a same acceleration intensity on all  $x$ ,  $y$  and  $z$  axis. These

accelerations are perturbations in the equations related to the forces in the quadrotor model Equations (2.42e), (2.42f) and (2.42d). The model is expressed as follows in Equation (4.4).

$$\begin{aligned}\tilde{\ddot{x}}(t) &= \ddot{x}(t) + a(t) \\ \tilde{\ddot{y}}(t) &= \ddot{y}(t) + a(t) \\ \tilde{\ddot{z}}(t) &= \ddot{z}(t) + a(t)\end{aligned}\quad (4.4)$$

This acceleration with respect to time is depicted in Figure 4.2, and expressed explicitly in Equation (4.5). The wind disturbance  $a(t)$  has a maximum acceleration about  $1.7\text{m/s}^2$ .

$$a(t) = \begin{cases} 0 & \text{when } 0\text{s} \leq t \leq 30\text{s}, \\ 0.8 \sin\left(\frac{\pi(t-30)}{31}\right) + 0.4 \sin\left(\frac{\pi(t-30)}{7}\right) \\ \quad + 0.08 \sin\left(\frac{\pi(t-30)}{2}\right) + 0.056 \sin\left(\frac{24\pi(t-30)}{11}\right) & \text{when } 30\text{s} < t \leq 57\text{s}, \\ 0 & \text{when } 57\text{s} < t \leq 70\text{s}, \\ 1.35 \sin\left(\frac{\pi(t-70)}{55}\right) + 0.15 \sin\left(\frac{\pi(t-70)}{2}\right) \\ \quad + 0.225 \sin\left(\frac{\pi(t-70)}{5}\right) + 0.105 \sin\left(\frac{24\pi(t-70)}{11}\right) & \text{when } 70\text{s} < t \leq 124\text{s}, \\ 0 & \text{when } 124\text{s} \leq t \leq 150\text{s}. \end{cases}\quad (4.5)$$

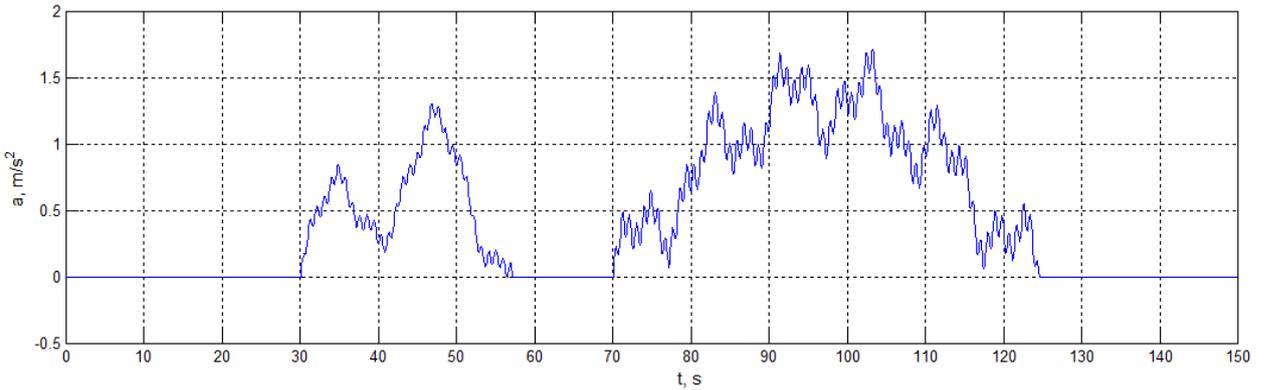


Figure 4.2: Wind disturbance w.r.t time.

### 4.1.3 Model Parameter Uncertainties

This scenario is dedicated to test the tolerance of control systems to modeling errors. Due to the complexity of the aerodynamic effects, the external forces and moments are difficult to be

expressed explicitly in the model. Therefore, there is a difference between the real dynamics and the model used in the control system. Furthermore, considering the complexity of the control algorithm and the limit of the computational resources of the onboard system, some unimportant forces and moments are neglected in the control system, such as frictions. This further increases the difference between the real dynamics and the model.

In Chapter 2, we have deduced a mathematical model of quadrotor (see Equation (3.2)). There are several parameters to be determined according to real quadrotors. The mass  $m$  can be directly evaluated by a balance. The value we get is usually precise. The inertia matrix elements  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$  are often computed based on mechanical structures which are designed on some mechanical design softwares, such as CATIA, Pro/ENGINEER. As the density of the material is not precise and later on motors and other elements are added into the system, the value of the inertia matrix can be influenced and cause some model uncertainties.

In Equation (3.2), the inputs  $u_i$  of the quadrotor model are defined as the sum of the thrusts and the yawing moments. Then, the rotational speeds  $\Omega_i$  of each rotor are calculated using the coefficient parameters  $b$  and  $d$ .

$$u_1 = b \sum_{i=1}^4 \Omega_i^2, \quad u_2 = bl(\Omega_2^2 - \Omega_4^2), \quad u_3 = bl(-\Omega_1^2 + \Omega_3^2), \quad u_4 = (-1)^{i+1} d \sum_{i=1}^4 \Omega_i^2. \quad (4.6)$$

The parameters  $b$  and  $d$  are variables and change during the flights. However, in the control system, they are treated as constants which are computed based on some static tests. These parameters also cause some model uncertainties.

Therefore, control systems should be capable to tolerate model uncertainties or parameter uncertainties in real conditions. The stability of the system should always be ensured.

In this scenario, some random parameter uncertainties are given. Supposing the inertia matrix elements  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$  are underrated 50%. The coefficients  $b$  and  $d$  are also underrated, and the value used in the control system are only 80% of the real values.

$$\begin{aligned} \tilde{I}_{xx} &= 0.5 I_{xx}, & \tilde{I}_{yy} &= 0.5 I_{yy}, & \tilde{I}_{zz} &= 0.5 I_{zz}, \\ \tilde{b} &= 0.8 b, & \tilde{d} &= 0.8 d. \end{aligned} \quad (4.7)$$

#### 4.1.4 Sensor Noises

This scenario is dedicated to test the tolerance of control systems to noises of the system states. As the cost and space are limited for quadrotors, MEMS (Micro-Electro-Mechanical

Systems) sensors are usually used. The precision of these sensor are low and the output signals are always noisy. Moreover, weather condition changes in each application, such as temperature, humidity, wind. This can also cause sensor noises. In many applications, such as aerial photo and video, a limit of vibrations is important for the quality of images.

The position states  $x, y, z$  and angular states  $\phi, \theta, \psi$  of the quadrotor system are measured by sensors. The position states  $x, y, z$  can be estimated by accelerometers. Accelerometers give only the transient linear acceleration. In order to get the linear velocity and position information, we need to integrate the acceleration once and twice. This may cause big errors and drifts, especially when the accelerometer has already initial data noises. The position states can be also measured through vision systems by using triangular position determination or other position determination methods. However, when the quadrotor is hovering or especially moving, the vibration of quadrotors can cause unwanted noises. The position states can be measured through GPS, sonar, pressure sensor or other distance measure sensors, where sensor noises are always a problem. The angular states  $\phi, \theta, \psi$  are usually estimated by gyroscopes. Gyroscopes give the rotational velocities information. In order to get the rotational position, we need the integrate the acceleration once. Depending on the precision of a gyroscope, sensor noises can be small or big enough to influence control systems.

In this scenario, we add the sensor noises on all 12 states of the system, namely the position states  $x, y, z, \dot{x}, \dot{y}, \dot{z}$  and angular states  $\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}$ . The expression of the noise added in the system is as in Equation 4.8. The function `Rand()` is a MATLAB function which generates a random number between 0 to 1. Thus, `Rand()+0.5` generates a random value  $\in [0.5, 1.5]$ . Therefore, all the states have  $\pm 50\%$  noises.

$$\tilde{x} = x \cdot (\text{Rand}() + 0.5) \quad (4.8)$$

### 4.1.5 Actuator Faults

This scenario is dedicated to test the stability of control systems under sudden strong signal disturbances. As motors and sensors are worn out, some severe situations may occur during flights, such as actuator fault, sensor failure, etc. In a very short time interval, a sudden and rather strong abnormal signal occurs. Control systems should not be greatly influenced and loose its stability during the failure.

Suppose the quadrotor used for this shooting is rather old, and has already been used for several years. During the flights, the rotors of the quadrotor have actuator faults. In a

short time period, the rotational speed of one rotor is no longer as what is required by the control system. After this period, the rotors rotate without any problem as wanted. If this fault is not detected during the flight, the performance of quadrotor will be influenced.

In this scenario, we add the actuator faults on two rotors during the flight task, rotor 3 and 4 (see in Figure 2.1 on Page 26). These faults are not detected by the program, and influence directly the control system performance. Two rotors are on two arms of the quadrotor which enables to see the difference between the roll and pitch dynamics.

Tested in simulation, an actuator fault time 0.2s can cause an unrecoverable error for the system to retain its stability. Therefore in this scenario, a transient time of actuator faults 0.1s is selected. Two faults are planned. The first fault happens at 42s during the reference tracking along the  $y$  axis. The other fault happens when the quadrotor is hovering. Therefore, we can check out the stability of the system during tracking and hovering. The program is written as following:

```
if (t>42 && t<42.1 ) Omd(3)=220; end
if (t>90 && t<90.1 ) Omd(4)=220; end
```

where  $Omd()$  is the angular velocity of the rotor.  $Omd(3)=220$  means the angular velocity of the rotor 3 is set to be 220rad/s.

#### 4.1.6 Sensor Faults

As sensors are worn out, they can cause some sensor faults in applications. The sensor data are considered always correct in control systems. Therefore, sensor faults are difficult to be detected by program. Other sensors are needed to detect whether one sensor has faults or not. Thus, sensor faults are not simulated in this thesis.

## 4.2 Control Schemes

Nowadays, the control laws of control systems are mainly implemented in digital platforms since computer and microprocessors offer many advantages with respect to analog platforms. As a digital computer is a discrete system, A/D converters are used to change sensor data into digital inputs for the control system, and D/A converters are used to change digital control signals into continuous control outputs. The D/A converter is usually a zero-order hold (ZOH), which holds each sample value for one sampling interval. There are two main

schemes to schedule the control signals on digital platforms: the time triggered scheme and the event triggered scheme.

### 4.2.1 Time Triggered Scheme

The time triggered scheme is widely used in many control methods. The control task is executed periodically every sampling period  $\mathcal{T}_t$ , which is previously defined in the program before the real-time control. Every  $\mathcal{T}_t$ , the control system computes the new control signals and sends them to the actuators. It is the most common and simplest implementation. The main difficulty of this approach is the selection of a sampling period  $\mathcal{T}_t$  which guarantees a desired level of the control performance. A small sampling period can keep the tracking errors under an acceptable limit during two sampling times. However, a small sampling period means the control system needs more computational time and resources, which needs to take from the upper system. Usually in order to keep the stability of the system, a sampling period  $\mathcal{T}_t$  is chosen based on the worst case scenario. For linear systems, the selection of the sampling period ensuring the stability of the system is studied base on the construction of an equivalent discrete model. For nonlinear systems, Nesic et al. have proposed a common approach to find approximate discrete-time models, and then studied how the sampling period influence the stability of the system [77]. However in real-time control applications, the sampling periods are selected usually by the experience of the engineers and some rough previous simulations.

### 4.2.2 Event Triggered Scheme

Contrarily to the time triggered control scheme in which the control signals are sent to the actuators every fixed sampling time, in the event based scheme, the control signals are sent only upon the triggering of an event. The event driven control was firstly proposed by Årzén [7]. The comparisons of time driven and event driven control scheme for first order stochastic and nonlinear systems are proposed in [8] and [40] respectively. The event triggered control scheme has many advantages: in networked control systems, in order to save the bandwidth and energy, the information is only transmitted when some significant events have occurred [99]; in some embedded applications, reducing the number of the control law computations can save energy (such as in wireless communication, ...) and give more computation time to other sub-systems (localization system, vision system, etc) [9].

A basic Årzén's event based controller consists of two parts: a time triggered event

detector  $\mathcal{C}_t$  and an event triggered PID controller  $\mathcal{C}_e$  [7]. The latter computes the control signal to be delivered to the actuators. The former  $\mathcal{C}_t$  runs at a fixed sampling period  $h_e$ , and upon fulfillment of a certain event triggering law  $\mathbf{L}_e$ , sends events to  $\mathcal{C}_e$ . Upon reception of the event,  $\mathcal{C}_e$  computes the control signal and sends it to the actuators.

Usual event triggering laws  $\mathbf{L}_e$  include:

(1) Error threshold law:

$$|e(t_k)| > e_{lim}, \quad (4.9)$$

where  $e = y_r - y$  is the tracking error,  $t_k$  is the current discrete sensing time by  $\mathcal{C}_e$ , and  $e_{lim}$  is a fixed limit.

(2) Error difference threshold:

$$|e(t_k) - e(t_{k-1})| > e_{lim}. \quad (4.10)$$

(3) ISS based law:

$$e(t_k) = \sigma \frac{a}{b} |y(t_k)|, \quad (4.11)$$

assuming the system can be rendered ISS (Input to State Stable) through static feedback [91]. Here,  $\sigma$  is chosen less than one to ensure an associated Lyapounov function to decrease,  $a$  and  $b$  are chosen according to the Lipschitz constants of  $\mathcal{K}_\infty$  (consisting of all functions  $\gamma: \mathbf{R}^+ \rightarrow \mathbf{R}^+$  which are continuous, strictly increasing, satisfying  $\gamma(0) = 0$  and  $\lim_{\xi \rightarrow \infty} \gamma(\xi) = \infty$ . See [91]).

In the following motion control schemes, we will take the following event triggering schemes:

$$|e(t_k) - e(t_{k-1})| > e_{lim} \quad (4.12)$$

## 4.3 Control Methods

Many control methods have been implemented on quadrotors. In Section 1.7 of Chapter 1, these control methods have been discussed in detail. A linear SISO control is used by Pounds in the decoupled dynamics in longitudinal (pitch/roll) and azimuthal modes [83]. The controls using Lyapunov theory have been proposed by Castillo et al. using a nested

saturation algorithm on Draganflyer in 2004 [32], and also proposed by Bouabdallah et al. using the decoupled dynamics in 2004 [20]. The Backstepping controls have been used by Altuğ et al. using visual feedback as the primary sensor in 2002 [4], Bouabdallah et al. using the decoupled dynamics in 2005 [20], Castillo et al. adding saturation functions in 2006 [34], and Madani et al. by separating the system into three interconnected subsystems in 2006 [74]. The Sliding mode controls have been proposed by Hoffmann et al. for the altitude loop control in 2004 [52], Waslander et al. in 2005 [98], and Bouabdallah et al. using the decoupled dynamics in 2005 [20]. A dynamic feedback technique to linearize the system has been proposed by Mistler et al. in 2001 [69]. A mixed robust feedback linearization with linear  $\text{GH}_\infty$  controller has been applied by Mokhtari et al. in 2005 [71]. An adaptive control has been proposed by Guenard et al. on the quadrotor X4-Flyer in 2006 [49].

In this section, we propose a newly developed control method named the model free control. This method is firstly implemented on the quadrotor model proposed in Chapter 2. At the same time, several other control methods are also proposed, such as a backstepping control, a sliding mode control, etc. The aim of this part is to compare different control methods in various scenarios mentioned in the previous sections. The advantages and disadvantages of each control method will be shown clearly. Therefore, we can choose the most appropriate control method for controlling the quadrotors according to the scenarios.

### 4.3.1 Model Free Control

The recently introduced model free control provides an elegant and efficient solution to the previous challenges [43–46]. It is a simple but efficient technique for nonlinear, unknown or partially known dynamics. This method is also called intelligent PID (or i-PID). While retaining the PID reduced computational cost, it is able to cope with general types of nonlinearities. The comparison between intelligent PID and traditional PID controllers is given in d'Andréa-Novel et al. [10]. The paper emphasizes the easy tuning of i-PID gains and gives a clear-cut explanation of the performance of usual PIDs. A comparison between a high-order sliding mode controllers and intelligent PID controllers is proposed by Riachy et al. in 2011 [90].

Model free control has been implemented in some academic SISO systems [43], joint motion control in humanoid locomotion [96], DC motors in flexible joints [97], non-minimum phase systems [11], dc/dc converters [73], hydroelectric power plants [56], small programmable devices [57], a quadrotor [100,102], etc.

A finite dimensional SISO system can be described implicitly as

$$\mathbf{E}(y, \dot{y}, \dots, y^{(\iota)}, u, \dot{u}, \dots, u^{(\kappa)}) = 0, \quad (4.13)$$

where  $\mathbf{E} : \mathbf{R}^{\iota+1} \times \mathbf{R}^{\kappa+1} \rightarrow \mathbf{R}$  is a smooth function. Assume that for integer  $\nu$ ,  $0 < \nu \leq \iota$ ,  $\partial E / \partial y^{(\nu)} \neq 0$ . The implicit function theorem [64] allows to express  $y^{(\nu)}$  locally:

$$y^{(\nu)} = \mathbf{E}(t, y, \dot{y}, \dots, y^{(\nu-1)}, y^{(\nu+1)}, \dots, y^{(\iota)}, u, \dot{u}, \dots, u^{(\kappa)}), \quad (4.14)$$

In a very short time interval, the system (4.14) can be modeled as:

$$y^{(\nu)} = \alpha u + F, \quad (4.15)$$

The coefficient of the input  $\alpha \in \mathbf{R}$  can be constant or time-varying, while it is considered as a constant in each time interval. Its value can be obtained by the knowledge of models or system identification methods. If  $\alpha$  is chosen as a constant, the control method is a pure model free control without any knowledge of the system. If  $\alpha$  is time-varying, the method is partial model free control with some system information. The time-varying parameter  $F$  represents the rest of the dynamics, and it is considered as a constant in each time interval.

The value  $F$  can be estimated by the real-time value of  $y^{(\nu)}$  and  $u$  in the previous time interval:

$$\hat{F} = \hat{y}^{(\nu)} - \alpha \hat{u}, \quad (4.16)$$

where  $\hat{y}^{(\nu)}$  is the  $\nu^{\text{th}}$  derivative of the measure  $y$  in the previous time interval, and  $\hat{u}$  is the control input  $u$  in the previous time interval.

A model free controller is defined as:

$$u = \frac{1}{\alpha} \left( y_r^{(\nu)} - \hat{F} + \Lambda(e) \right), \quad (4.17)$$

where  $y_r$  is a reference trajectory, and  $e = y_r - y$  is the tracking error.  $\Lambda(e)$  is a function which makes the closed loop error dynamics  $e^{(\nu)} = \Lambda(e)$  asymptotically stable. Here we choose the function  $\Lambda(e)$  as:  $\Lambda(e) = K_{v-1}e^{(v-1)} + K_{v-2}e^{(v-2)} + \dots + K_1\dot{e} + K_0e$ , where  $K_i$ ,  $i = 0, \dots, v-1$  are tuning gains.

Combining Equations (4.16) and (4.17), a model free controller is obtained:

$$u = \hat{u} + \frac{1}{\alpha} (y_r^{(\nu)} - \hat{y}^{(\nu)} + K_{v-1}e^{(v-1)} + K_{v-2}e^{(v-2)} + \dots + K_1\dot{e} + K_0e). \quad (4.18)$$

**(1) Accuracy between the modeled system (4.15) and the original system (4.14):**

At the beginning of one time interval  $t_0$ ,  $y = \tilde{y}$ ,  $u = \tilde{u}$ . The system (4.14) is expressed as:

$$\tilde{y}^{(\nu)} = \mathbf{E}(\tilde{y}, \tilde{u}), \quad (4.19)$$

The modeled system (4.15) at  $t = t_0$  is precise:

$$\tilde{y}^{(\nu)} = \alpha\tilde{u} + \tilde{F}, \quad (4.20)$$

At the end of this time interval  $t_0 + \Delta t$ , the system becomes

$$y = \tilde{y} + \Delta y, \quad u = \tilde{u} + \Delta u, \quad (4.21)$$

Therefore, the output  $y$  can be expanded into a Taylor series about  $(\tilde{y}, \tilde{u})$ :

$$\begin{aligned} y^{(\nu)} &= E(\tilde{y} + \Delta y, \tilde{u} + \Delta u) \\ &= E(\tilde{y}, \tilde{u}) + \frac{\partial E}{\partial y}(\tilde{y}, \tilde{u})\Delta y + \frac{\partial E}{\partial u}(\tilde{y}, \tilde{u})\Delta u \\ &\quad + \frac{1}{2} \left( \frac{\partial^2 E}{\partial^2 y}(\tilde{y}, \tilde{u})\Delta y^2 + \frac{\partial^2 E}{\partial u \partial y}(\tilde{y}, \tilde{u})\Delta u \Delta y + \frac{\partial^2 E}{\partial^2 u}(\tilde{y}, \tilde{u})\Delta u^2 \right) + \dots, \end{aligned} \quad (4.22)$$

And the modeled system's output is:

$$y_{appr}^{(\nu)} = \alpha(\tilde{u} + \Delta u) + \tilde{F} = E(\tilde{y}, \tilde{u}) + \alpha\Delta u = E(\tilde{y}, \tilde{u}) + \frac{\partial E}{\partial u}(\tilde{y}, \tilde{u})\Delta u, \quad (4.23)$$

Therefore, the accuracy error in each time interval is:

$$\begin{aligned} e_{appr} &= y^{(\nu)} - y_{appr}^{(\nu)} \\ &= \frac{\partial E}{\partial y}(\tilde{y}, \tilde{u})\Delta y + \frac{1}{2} \left( \frac{\partial^2 E}{\partial^2 y}(\tilde{y}, \tilde{u})\Delta y^2 + \frac{\partial^2 E}{\partial u \partial y}(\tilde{y}, \tilde{u})\Delta u \Delta y + \frac{\partial^2 E}{\partial^2 u}(\tilde{y}, \tilde{u})\Delta u^2 \right) + \dots, \end{aligned} \quad (4.24)$$

The accuracy error is the high order parts of the Taylor series of the original system. As the short time interval is defined relatively small with respect to system dynamics, we consider this error is negligible.

**(2) Stability of the model free control:**

Combining Equations (4.15) and (4.17), the error dynamics of the system is:

$$y^{(\nu)} = y_r^{(\nu)} + F - \hat{F} + K_{v-1}e^{(v-1)} + K_{v-2}e^{(v-2)} + \dots + K_1\dot{e} + K_0e, \quad (4.25)$$

$$e^{(\nu)} + K_{v-1}e^{(v-1)} + K_{v-2}e^{(v-2)} + \dots + K_1\dot{e} + K_0e + e_F = 0 \quad (4.26)$$

where  $e_F = F - \hat{F} \in \mathbf{R}$  is time-varying. In each short time interval  $h$ ,

$$e_F = |y^{(\nu)} - \hat{y}^{(\nu)} - \alpha(u - \hat{u})| \leq \left| \frac{d\hat{y}^{(\nu)}}{dt} h \right| + \left| \frac{d\hat{u}}{dt} h \right|. \quad (4.27)$$

In physical systems,  $\frac{d\hat{y}^{(\nu)}}{dt}$  and  $\frac{d\hat{u}}{dt}$  are usually bounded and not equal to infinite,  $\left| \frac{d\hat{y}^{(\nu)}}{dt} \right| < \epsilon_{y^{(\nu)}}$ ,  $\left| \frac{d\hat{u}}{dt} \right| < \epsilon_u$ , where the measure  $\hat{y}^{(\nu)}$  is supposed noise free or previously filtered. If a rather small time interval  $h$  with respect to system dynamics is selected,  $e_F$  will be bounded in a small limit  $|e_F| < \epsilon, \epsilon > 0$ . Thus, by choosing appropriate gains  $K_i$ , the tracking error  $e$  in Equation (4.26) will converge to this small limit  $\epsilon$ . Therefore, the system is practically stable. To further prove the stability of the control method, more information about a specific system is needed.

**(3) General setting of the model free control on the model of quadrotor:**

The control oriented model is proposed in Section 2.7. Firstly, we control the altitude  $z$ . We rewrite the vertical dynamics of the quadrotor model in Equation (2.42d) as:

$$m\ddot{z} = (\cos\theta \cos\phi)u_1 + F_z. \quad (4.28)$$

where  $\cos\theta \cos\phi$  is the system information used in partial model free control. The value  $F_z$  does not only represent  $-mg$ , but also includes neglected vertical dynamics. In discrete time, the unknown part  $F_z$  can be expressed as following, where  $\hat{\ddot{z}}(k)$  is an estimate of  $\ddot{z}(k)$ :

$$\hat{F}_z = m\hat{\ddot{z}}(t_k) - (\cos\theta \cos\phi)u_1(t_{k-1}). \quad (4.29)$$

Thus, the chosen control law is:

$$u_1(t_k) = u_1(t_{k-1}) + \frac{m}{\cos\theta \cos\phi} \left( \hat{e}_{2d}^z(t_k) + k_1^z e_d^z(t_k) + k_0^z e^z(t_k) \right), \quad (4.30)$$

with

$$\begin{aligned}
\hat{e}_{2d}^z(t_k) &= \ddot{z}_r(t_k) - \hat{\ddot{z}}(t_k), & e_d^z(t_k) &= \dot{z}_r(t_k) - \hat{\dot{z}}(t_k), \\
e^z(t_k) &= z_r(t_k) - z(t_k), \\
\hat{\dot{z}}(t_k) &= \frac{T_f}{T_f + h} \hat{\dot{z}}(t_{k-1}) + \frac{1}{T_f + h} \left( \dot{z}(t_k) - \dot{z}(t_{k-1}) \right), \\
\hat{\ddot{z}}(t_k) &= \frac{T_f}{T_f + h} \hat{\ddot{z}}(t_{k-1}) + \frac{1}{T_f + h} \left( \ddot{z}(t_k) - \ddot{z}(t_{k-1}) \right),
\end{aligned} \tag{4.31}$$

where  $\ddot{z}_r$ ,  $\dot{z}_r$ ,  $z_r$  are the reference acceleration, velocity and position of  $z$ . The variable  $h$  is the sampling period,  $h = t_k - t_{k-1}$ .

Then we control the position  $x$  and  $y$ . As the input  $u_1$  is already used in the control of the altitude  $z$ , we now use  $u_2$  and  $u_3$  to control the positions  $x$  and  $y$ . Therefore, we need to differentiate twice the equations related to  $x$  and  $y$  in Equations (2.42e) and (2.42f) in order to get the control inputs  $u_2$  and  $u_3$ . Then, we obtain:

$$\begin{aligned}
x^{(4)} &= \frac{u_1}{mI_{xx}} (\sin \psi \cos \phi - \cos \psi \sin \theta \sin \phi) u_2 + \frac{u_1}{mI_{yy}} (\cos \psi \cos \theta \cos \phi) u_3 + F_x, \\
y^{(4)} &= -\frac{u_1}{mI_{xx}} (\cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi) u_2 + \frac{u_1}{mI_{yy}} (\sin \psi \cos \theta \cos \phi) u_3 + F_y,
\end{aligned} \tag{4.32}$$

where the partial model free control is used and  $F_x, F_y$  are the remaining parts of the horizontal and lateral system. To simplify the notations, we define

$$\begin{aligned}
A &= \frac{u_1}{mI_{xx}} (\sin \psi \cos \phi - \cos \psi \sin \theta \sin \phi), \\
B &= \frac{u_1}{mI_{yy}} (\cos \psi \cos \theta \cos \phi), \\
C &= -\frac{u_1}{mI_{xx}} (\cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi), \\
D &= \frac{u_1}{mI_{yy}} (\sin \psi \cos \theta \cos \phi)
\end{aligned} \tag{4.33}$$

We implement the model free control scheme in a similar manner as previous:

$$\begin{pmatrix} u_2(t_k) \\ u_3(t_k) \end{pmatrix} = \begin{pmatrix} u_2(t_{k-1}) \\ u_3(t_{k-1}) \end{pmatrix} + \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} \begin{pmatrix} \hat{e}_{4d}^x + k_3^x e_{3d}^x + k_2^x e_{2d}^x + k_1^x e_d^x + k_0^x e^x \\ \hat{e}_{4d}^y + k_3^y e_{3d}^y + k_2^y e_{2d}^y + k_1^y e_d^y + k_0^y e^y \end{pmatrix}, \tag{4.34}$$

where  $\hat{e}_{4d}^x, \hat{e}_{4d}^y$  are the errors between the references  $x_r^{(4)}, y_r^{(4)}$  and the estimates of  $x^{(4)}, y^{(4)}$ .

For the yaw angle control, we consider the equation related  $\psi$  as:

$$I_{zz}\ddot{\psi} = u_4 + F_\psi. \quad (4.35)$$

Then the control law is:

$$u_4(t_k) = u_4(t_{k-1}) + I_{zz} \left( \hat{e}_{2d}^\psi(t_k) + k_1^\psi e_d^\psi(t_k) + k_0^\psi e^\psi(t_k) \right), \quad (4.36)$$

where  $\hat{e}_{2d}^\psi$  is the error between the reference  $\ddot{\psi}_r$  and the estimate of  $\ddot{\psi}$ .

### 4.3.2 Backstepping Control

For the purpose of comparison, a backstepping control is also applied on the quadrotor system. We choose the method proposed by Bouabdallah [20]. In the control system, we neglect some parts in the model Equations (3.2) to simplify the control laws, such as the hub forces. The control model is as follows:

$$\begin{aligned} I_{xx}\ddot{\phi} &= \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + u_2, \\ I_{yy}\ddot{\theta} &= \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + u_3, \\ I_{zz}\ddot{\psi} &= \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + u_4, \\ m\ddot{x} &= (\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi)u_1, \\ m\ddot{y} &= (-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi)u_1, \\ m\ddot{z} &= -mg + (\cos\theta \cos\phi)u_1. \end{aligned} \quad (4.37)$$

Note that the angular subsystem is in cascade with the position subsystem. Therefore, the system can be separated into two subsystems: the angular and position subsystems. We firstly control the angular subsystem, and then use the angles to control the position subsystem. We rewrite the system into the state space form using the state vector  $(x_1, \dots, x_{12})$  with

$$\begin{aligned} x_1 &= \phi, & x_2 &= \dot{\phi}, & x_3 &= \theta, & x_4 &= \dot{\theta}, \\ x_5 &= \psi, & x_6 &= \dot{\psi}, & x_7 &= z, & x_8 &= \dot{z}, \\ x_9 &= x, & x_{10} &= \dot{x}, & x_{11} &= y, & x_{12} &= \dot{y}. \end{aligned} \quad (4.38)$$

Therefore, the system can be written as:

$$\begin{aligned}
\dot{x}_1 &= x_2, & \dot{x}_2 &= a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 u_2, \\
\dot{x}_3 &= x_4, & \dot{x}_4 &= a_3 x_2 x_6 + a_4 x_2 \Omega + b_2 u_3, \\
\dot{x}_5 &= x_6, & \dot{x}_6 &= a_5 x_2 x_4 + b_3 u_4, \\
\dot{x}_7 &= x_8, & \dot{x}_8 &= -g + \frac{\cos x_1 \cos x_3}{m} u_1, \\
\dot{x}_9 &= x_{10}, & \dot{x}_{10} &= \frac{u_x}{m} u_1, \\
\dot{x}_{11} &= x_{12}, & \dot{x}_{12} &= \frac{u_y}{m} u_1,
\end{aligned} \tag{4.39}$$

where  $u_x = \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi$ ,  $u_y = -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi$ ,

$$\begin{aligned}
a_1 &= \frac{I_{yy} - I_{zz}}{I_{xx}}, & a_2 &= \frac{J_r}{I_{xx}}, & a_3 &= \frac{I_{zz} - I_{xx}}{I_{yy}}, & a_4 &= -\frac{J_r}{I_{yy}}, \\
a_5 &= \frac{I_{xx} - I_{yy}}{I_{zz}}, & b_1 &= \frac{1}{I_{xx}}, & b_2 &= \frac{1}{I_{yy}}, & b_3 &= \frac{1}{I_{zz}}.
\end{aligned} \tag{4.40}$$

Here we define a new state  $z_1$  which equals the tracking error of the state  $x_1$ :

$$z_1 = x_{1d} - x_1, \tag{4.41}$$

where  $x_{1d}$  is the reference  $x$  position.

Using the Lyapunov theorem, we build the Lyapunov function of  $z_1$  as following:

$$\begin{aligned}
V(z_1) &= \frac{1}{2} z_1^2, \\
\dot{V}(z_1) &= z_1 (\dot{x}_{1d} - \dot{x}_2).
\end{aligned} \tag{4.42}$$

The Lyapunov function of  $z_1$  should be positive definite and its derivative should be negative semi-definite. Here we define a virtual control reference for  $x_2$ :

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1, \quad \alpha_1 > 0 \tag{4.43}$$

which ensures the derivative of the Lyapunov function of  $z_1$  negative semi-definite:

$$\dot{V}(z_1) = z_1 (\dot{x}_{1d} - \dot{x}_2) = -\alpha_1 z_1^2 < 0. \tag{4.44}$$

Here we introduce another new state  $z_2$ :

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1. \quad (4.45)$$

We build the Lyapunov function for  $z_2$  as following:

$$V(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2) \quad (4.46)$$

Then, its derivative is:

$$\dot{V}(z_1, z_2) = z_2(a_1 x_4 x_6 + a_2 x_4 \omega + b_1 u_2) - z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) - z_1 z_2 - \alpha_1 z_1^2. \quad (4.47)$$

Therefore, the control input  $u_2$  is defined as following to ensure  $\dot{V}(z_1, z_2) < 0$ :

$$u_2 = \frac{1}{b_1} \left( z_1 - a_1 x_4 x_6 - a_2 x_4 \omega - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2 \right), \quad \alpha_2 > 0. \quad (4.48)$$

In the angular subsystem, the other control laws are defined using the same method:

$$\begin{aligned} u_3 &= \frac{1}{b_2} \left( z_3 - a_3 x_2 x_6 - a_4 x_2 \omega - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4 \right), \\ u_4 &= \frac{1}{b_3} \left( z_5 - a_5 x_2 x_4 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6 \right), \\ z_3 &= x_{3d} - x_3, \quad z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3, \\ z_5 &= x_{5d} - x_5, \quad z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5. \end{aligned} \quad (4.49)$$

All the  $\alpha_i$ ,  $i = 1, \dots, 12$ . are the control gains.

In the position subsystem, the control laws are defined as:

$$\begin{aligned} u_1 &= \frac{m}{\cos x_1 \cos x_3} \left( z_7 + g - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8 \right), \\ u_x &= \frac{m}{u_1} \left( z_9 - \alpha_9(z_{10} + \alpha_9 z_9) - \alpha_{10} z_{10} \right), \\ u_y &= \frac{m}{u_1} \left( z_{11} - \alpha_{11}(z_{12} + \alpha_{11} z_{11}) - \alpha_{12} z_{12} \right), \\ z_7 &= x_{7d} - x_7, \quad z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7, \\ z_9 &= x_{9d} - x_9, \quad z_{10} = x_{10} - \dot{x}_{9d} - \alpha_9 z_9, \\ z_{11} &= x_{11d} - x_{11}, \quad z_{12} = x_{12} - \dot{x}_{11d} - \alpha_{11} z_{11}. \end{aligned} \quad (4.50)$$

From  $u_x$  and  $u_y$ , we can compute the needed angles  $\phi$ ,  $\theta$  and  $\psi$  for the position control,

then use them as the angular references in the angular subsystem control. More details of the control can be found in [22].

### 4.3.3 Sliding Mode Control

For the sake of comparison, a sliding mode control is also proposed on the same scenario. We choose the method proposed by Bouabdallah [20]. The model in Equation (4.37) is also used for the sliding mode control. The definition of all the parameters can be found in the previous Subsection 4.3.2 Backstepping control. The quadrotor system is also divided into two subsystems: the angular and position subsystems. This sliding mode control is based on the backstepping control proposed in the previous subsection. In the control of the angle roll  $\phi$ , the new state  $z_1$  is defined as in the backstepping control. In Equation (4.43), the sliding surface is chosen:

$$\begin{aligned} S_\phi &= z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1, & z_1 &= x_{1d} - x_1. \\ \dot{S}_\phi &= \dot{x}_2 - \ddot{x}_{1d} - \alpha_1 \dot{z}_1 \\ &= a_1 x_4 x_6 + a_2 x_4 \omega + b_1 u_2 - \ddot{\phi}_d + \alpha_1 (z_2 + \alpha_1 z_1). \end{aligned} \quad (4.51)$$

The time derivative of the attractive surface should satisfy  $S\dot{S} < 0$ . Therefore, we define  $\dot{S}_\phi$  as

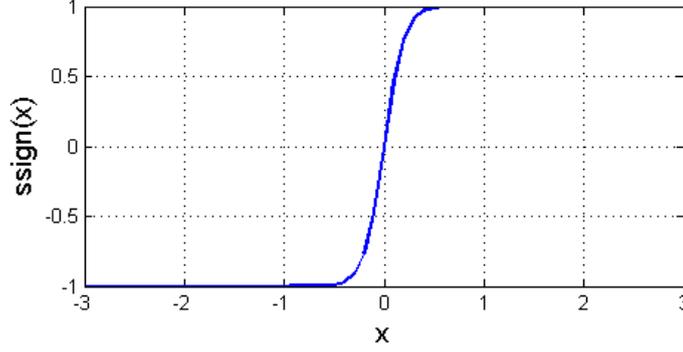
$$\dot{S}_\phi = -k_1 \text{ssign}(S_\phi) - k_2 S_\phi. \quad (4.52)$$

where  $k_1, k_2 > 0$  are the control gains.  $\text{ssign}()$  is an extended sign function which is shown in Equation (4.53) and Figure 4.3:

$$\text{ssign}(x) = \begin{cases} -1 & \text{when } x \leq -1, \\ \frac{2(x+1)^5}{(x+1)^5 + (1-x)^5} - 1 & \text{when } -1 < x < 1, \\ 1 & \text{when } x \geq 1. \end{cases} \quad (4.53)$$

Therefore, the other sliding surfaces in the angular subsystem are chosen as:

$$\begin{aligned} S_\theta &= z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3, & z_3 &= x_{3d} - x_3, \\ S_\psi &= z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5, & z_5 &= x_{5d} - x_5. \end{aligned} \quad (4.54)$$

Figure 4.3: The plot of the function  $\text{ssign}(x)$ .

where  $\dot{x}_{id}$  ( $i = 1, 3, 4, 5, 7, 9, 11$ .) are the references. The control laws are:

$$\begin{aligned} u_3 &= \frac{1}{b_2} \left( -k_3 \text{ssign}(S_\theta) - k_4 S_\theta - a_3 x_2 x_6 - a_4 x_2 \omega + \ddot{\theta}_d - \alpha_2^2 z_3 \right), \\ u_4 &= \frac{1}{b_3} \left( -k_5 \text{ssign}(S_\psi) - k_6 S_\psi - a_5 x_2 x_4 + \ddot{\psi}_d - \alpha_3^2 z_5 \right). \end{aligned} \quad (4.55)$$

In the position subsystem, the sliding surfaces are chosen as:

$$\begin{aligned} S_x &= z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7, & z_7 &= x_{7d} - x_7, \\ S_y &= z_{10} = x_{10} - \dot{x}_{9d} - \alpha_9 z_9, & z_9 &= x_{9d} - x_9, \\ S_z &= z_{12} = x_{11} - \dot{x}_{11d} - \alpha_{11} z_{11}, & z_{11} &= x_{11d} - x_{11}, \end{aligned} \quad (4.56)$$

Therefore, the control laws are defined:

$$\begin{aligned} u_1 &= \frac{m}{\cos x_1 \cos x_3} \left( -k_7 \text{ssign}(S_z) - k_8 S_z + g + \ddot{z}_d - \alpha_7^2 z_7 \right), \\ u_x &= \frac{m}{u_1} \left( -k_9 \text{ssign}(S_z) - k_{10} S_z + \ddot{x}_d - \alpha_9^2 z_9 \right), \\ u_y &= \frac{m}{u_1} \left( -k_{11} \text{ssign}(S_z) - k_{12} S_z + \ddot{y}_d - \alpha_{11}^2 z_{11} \right). \end{aligned} \quad (4.57)$$

#### 4.3.4 PID Control

In the PID control, the quadrotor system is also divided into two subsystems: the angular and position subsystem. A traditional PID control laws are implemented:

$$\begin{aligned}
u_1 &= K_p^z e_z + K_d^z \dot{e}_z + K_i^z I(e_z), & e_z &= z - z_r \\
u_x &= K_p^x e_x + K_d^x \dot{e}_x + K_i^x I(e_x), & e_x &= x - x_r \\
u_y &= K_p^y e_y + K_d^y \dot{e}_y + K_i^y I(e_y), & e_y &= y - y_r \\
u_2 &= K_p^\phi e_\phi + K_d^\phi \dot{e}_\phi + K_i^\phi I(e_\phi), & e_\phi &= \phi - \phi_r \\
u_3 &= K_p^\theta e_\theta + K_d^\theta \dot{e}_\theta + K_i^\theta I(e_\theta), & e_\theta &= \theta - \theta_r \\
u_4 &= K_p^\psi e_\psi + K_d^\psi \dot{e}_\psi + K_i^\psi I(e_\psi), & e_\psi &= \psi - \psi_r
\end{aligned} \tag{4.58}$$

No model information is taken into account in this control laws.

# Chapter 5

## Simulation Results

### 5.1 Time triggered Control

In this section, the control methods proposed in Section 4.3 will be implemented in different scenarios: a basic scenario and the scenarios with wind disturbance, model parameter uncertainties, sensor noises and actuator faults. The descriptions of these scenarios can be found in Section 4.1. The reference trajectory is a square path with a length of 2m. The 3D trajectory is given in Figure 4.1. The expression of the reference trajectory is in Equation (4.1). The total simulation time is 150s.

A sampling period of 10ms is selected in simulation. The commercialized quadrotor AscTec Hummingbird is designed for scientific researches (see Figure 1.9 in Chapter 1). The system proposes a user function which runs at 1kHz [48]. In this function, the user can process the sensor data, compute the reference trajectory and execute further application programs. The basic control stabilization program should not occupy too much processing time in order to allow other programs have more processing resources. Based on the real system, a processing frequency of 100Hz for the control task appears to be reasonable. The sampling period of a basic control should not be too long for the sake of the stability. The linear acceleration of a quadrotor is usually inferior to  $g$  ( $g = 9.8\text{m/s}^2$ ). In one sampling period of 10ms, the maximum changes of the linear velocity and position are 0.098m/s and 0.49mm respectively. Therefore, a sampling period of 10ms is selected which is rather small with respect to system dynamics and at the same time achievable in the real control.

In this chapter, all the control methods are sampled in the time triggered scheme, and the sampling period is set to be 10ms. There are 15000 actuation steps. Every 10ms, the system reads the data from sensors, computes new control signals and sends them to actuators.

### 5.1.1 Model Free Control

In this subsection, we implement the model free control in several different scenarios. The control law are described in Section 4.3 on Page 73. The total simulation time is 150s and the sampling period is 10ms. There are 15000 actuation steps.

#### 1. Basic scenario

This method is firstly simulated in the basic scenario without disturbance. The chosen control gains are in Table 5.1. The simulation results in the basic scenario without disturbance are in Figure 5.1. The first figure is the tracking errors along the  $x$ ,  $y$  and  $z$  axis. The second figure is the angles  $\phi$ ,  $\theta$  and  $\psi$ . The third figure is the angular velocities of the four rotors. The numbers of the rotor are defined in Figure 2.1 on Page 26.

Table 5.1: The model free control gains

$k_3^x$	5	$k_2^x$	18	$k_1^x$	15	$k_0^x$	12	$k_1^z$	2	$k_0^z$	5
$k_3^y$	5	$k_2^y$	18	$k_1^y$	15	$k_0^y$	12	$k_1^\psi$	5	$k_0^\psi$	10

In the basic scenario, the system using the model free control kept the stability during the flight and followed the reference trajectory. The maximum absolute tracking error is 0.042m, which is 2.1% of the total length 2m.

#### 2. Wind disturbance

In this subsection, the model free control is simulated in the scenario with wind disturbance. The wind is depicted in Section 4.1 on Page 65. In the applications, the control gains are pre-selected and usually not changed according to weather conditions and disturbances. In order to simulate the realistic case, the control gains are not changed, which are the same as in the basic scenario in Table 5.1. The simulation results are in Figure 5.2.

The control system shows its stability to wind disturbance. The system kept stable during the disturbance. The maximum absolute tracking error is 0.042m, which is the same with the basic scenario. The model free control compensated well the wind disturbance.

#### 3. Parameter uncertainties

Then the model free control is simulated in the scenario with model parameter uncertainties. The parameter uncertainties are defined in Section 4.1 on Page 66. The parameters are estimated smaller than the real values. Therefore, the system is under-actuated. The

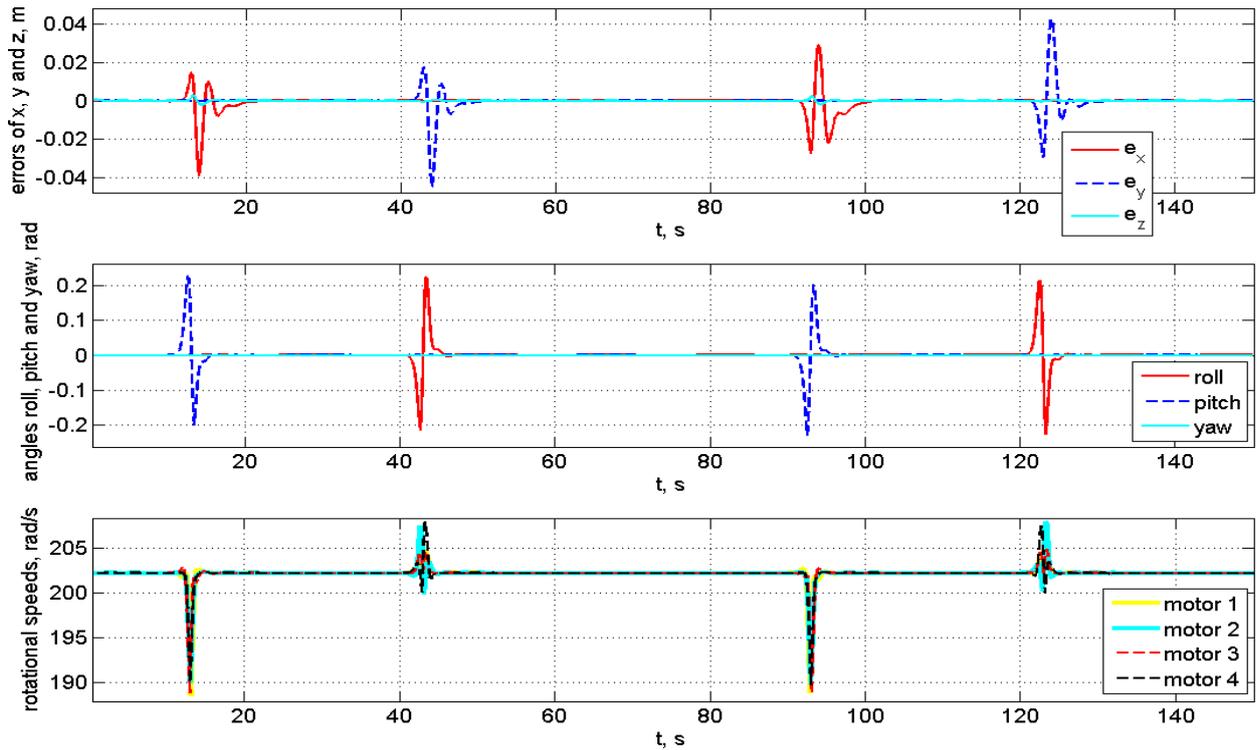


Figure 5.1: The model free control in the basic scenario with the gains in Table 5.1.

control gains are in Table 5.1. The simulation results are in Figure 5.3.

The system stays stable with parameter uncertainties. The maximum absolute tracking error is 0.075m, while 0.042m in the basic scenario.

#### 4. Sensor noises

The model free control is also simulated in the scenario with sensor noises. The sensor noises are defined in Section 4.1 on Page 68. The gains are not changed as in Table 5.1. The simulation results are in Figure 5.4.

The maximum absolute tracking error is 0.08m, while 0.042m in the basic scenario.

#### 5. Actuator faults

The model free control is also simulated in the scenario with actuator faults. The actuator faults are defined in Section 4.1 on Page 69. The gains are not changed as in Table 5.1. The simulation results are in Figure 5.5.

There were two actuator faults: one was during the hovering and the other is during the tracking. The system kept stable during these actuator faults. The model free control

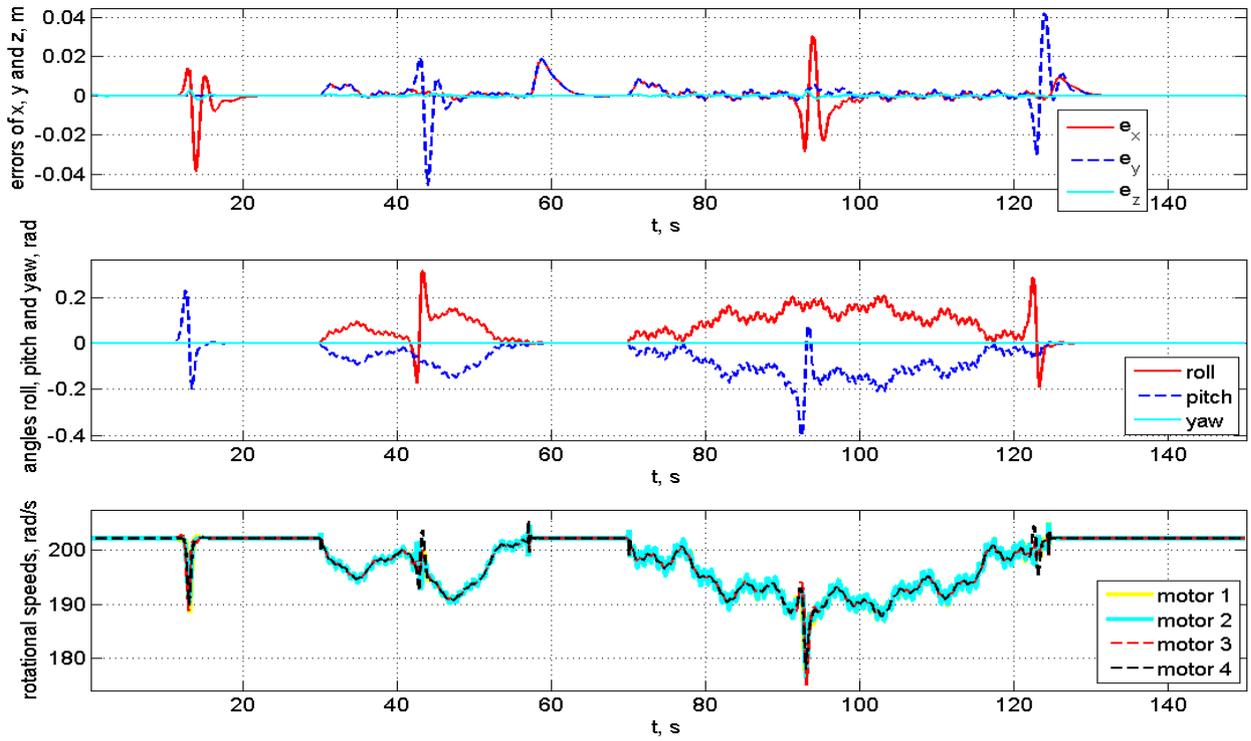


Figure 5.2: The model free control with wind disturbance and the gains in Table 5.1.

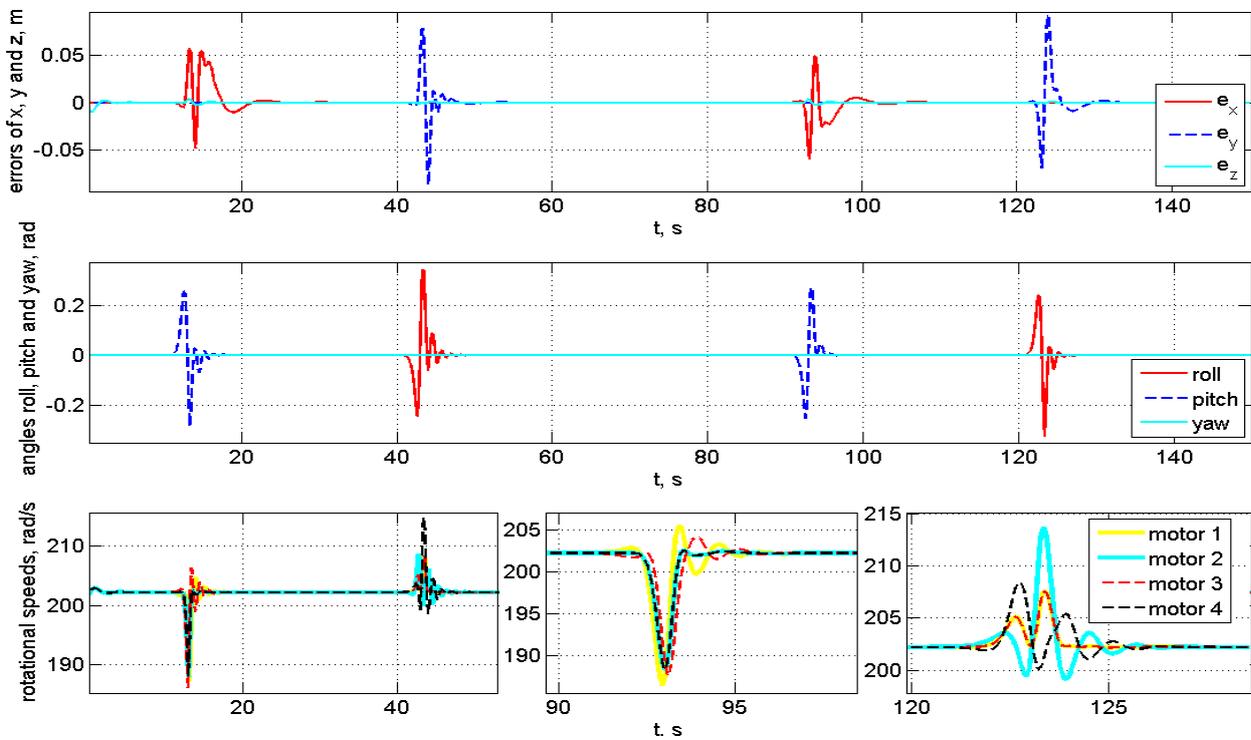


Figure 5.3: The model free control with parameter uncertainties and the gains in Table 5.1.

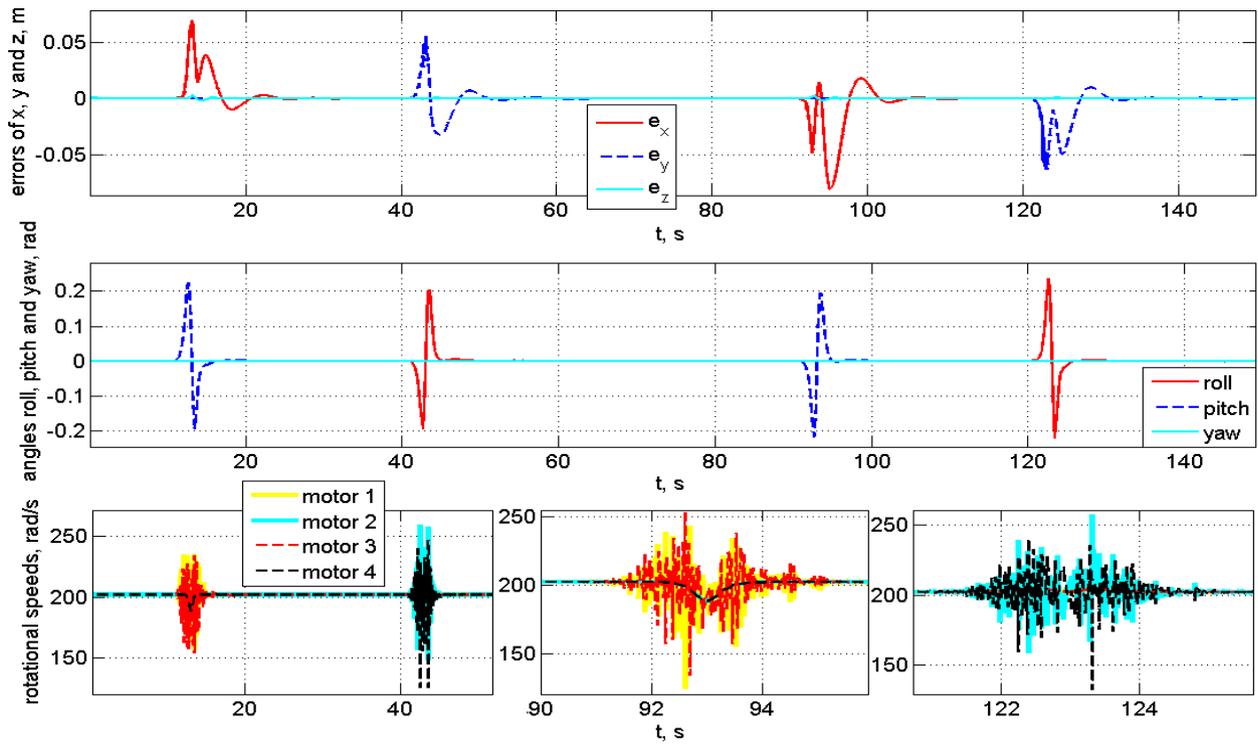


Figure 5.4: The model free control with sensor noises and the gains in Table 5.1.

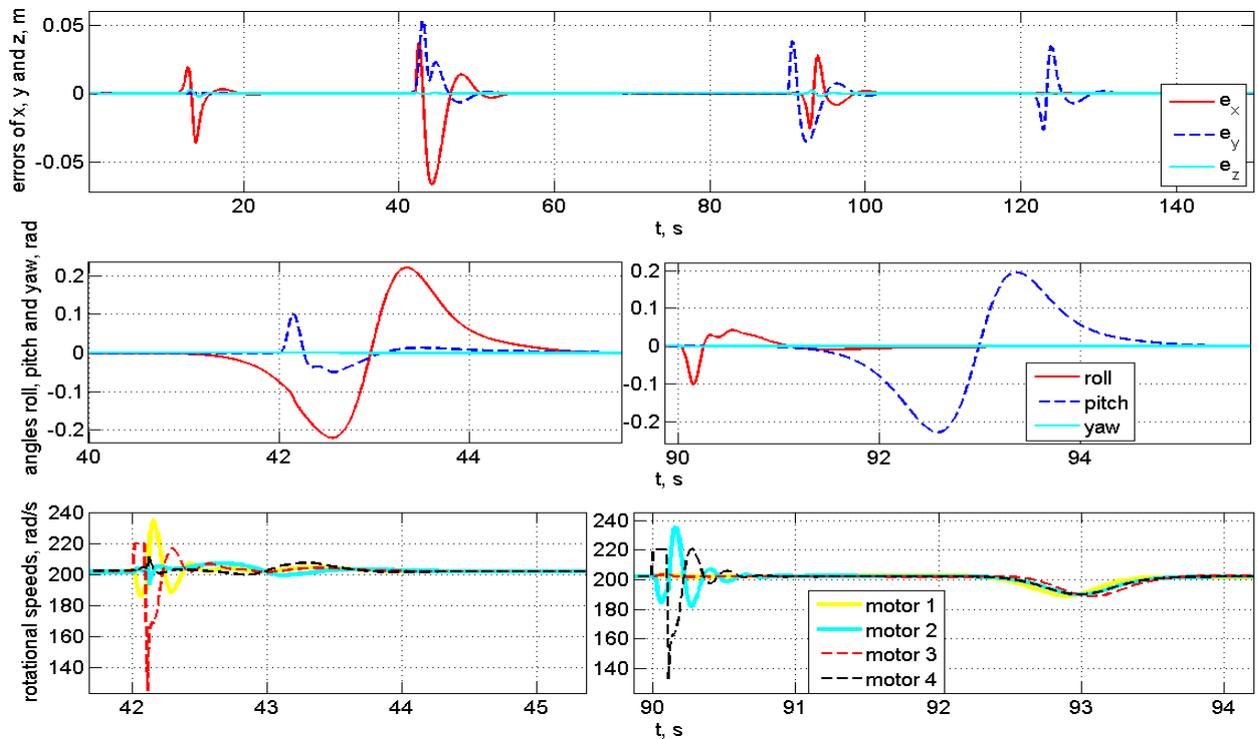


Figure 5.5: The model free control with actuator faults and the gains in Table 5.1.

responds quickly to the system fault, and the system adjusts to its stable states in 1s. The maximum absolute tracking error is 0.07m, which is happened at the actuator fault during tracking. The actuator fault during hovering caused 0.04m maximum absolute tracking error. The actuator fault caused the quadrotor tilt about 0.1 rad during hovering.

## 5.1.2 Backstepping Control

In the subsection, we implement the backstepping control method proposed by Bouabdallah [20]. The control laws are described in Subsection 4.3.2 on Page 78. The simulation time is 150s, and the sampling period is 10ms. There are total 15000 actuation steps.

### 1. Basic scenario

This methods is firstly simulated in the basic scenario without disturbance. The control gains are chosen as in Table 5.2. The simulation results in the basic scenario without disturbance are in Figure 5.6. The first figure is the tracking errors along the  $x$ ,  $y$  and  $z$  axis. The second figure is the angles  $\phi$ ,  $\theta$  and  $\psi$ . The third figure is the angular velocities of the four rotors. The numbers of the rotor are defined in Figure 2.1 in Chapter 2.

Table 5.2: The backstepping control gains

$\alpha_1$	4	$\alpha_2$	10	$\alpha_3$	4	$\alpha_4$	10	$\alpha_5$	1	$\alpha_6$	5
$\alpha_7$	5	$\alpha_8$	30	$\alpha_9$	2	$\alpha_{10}$	3	$\alpha_{11}$	2	$\alpha_{12}$	3

In the basic scenario, the system using the backstepping control kept stable and followed the reference trajectory. The maximum tracking error is 0.09m, which is 4.5% of the total length 2m. It is bigger than 0.042m of the model free control.

### 2. Wind disturbance

In this subsection, the backstepping control is simulated in the scenario with wind disturbance. The wind is depicted in Section 4.1. The control gains are in Table 5.2. The simulation results are in Figure 5.7.

During the wind disturbance, the control system has big tracking errors. The maximum absolute tracking error is 0.23m, which is 11.5% of the total length 2m. In the basic scenario, the maximum error is only 0.09m. In the model free control, the maximum tracking error is 0.042m.

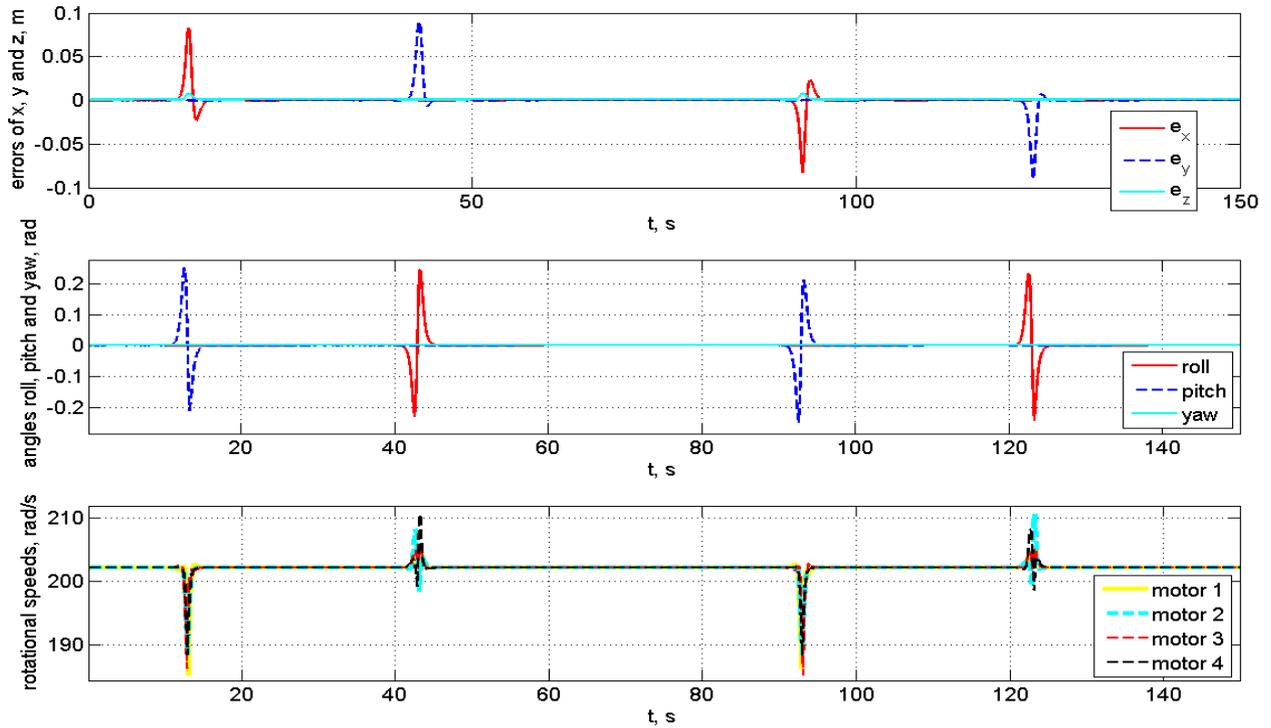


Figure 5.6: The Backstepping control in the basic scenario with the gains in Table 5.2.

### 3. Parameter uncertainties

Then, the backstepping control is simulated in the scenario with model parameter uncertainties. The parameter uncertainties are defined in Section 4.1. The gains are not changed as in Table 5.2. The simulation results are in Figure 5.8. The maximum absolute tracking error is 0.11m.

### 4. Sensor noises

The backstepping control is also simulated in the scenario with sensor noises. The sensor noises are defined in Section 4.1. The gains are in Table 5.2. The simulation results are in Figure 5.9.

### 5. Actuator faults

The backstepping control is also simulated in the scenario with actuator faults. The actuator faults are defined in Section 4.1. The gains are in Table 5.2. The simulation results are in Figure 5.10.

Two actuator faults happened separately during the tracking and hovering. In both cases, the backstepping control kept its stability. The actuator fault during the moving caused 0.15m

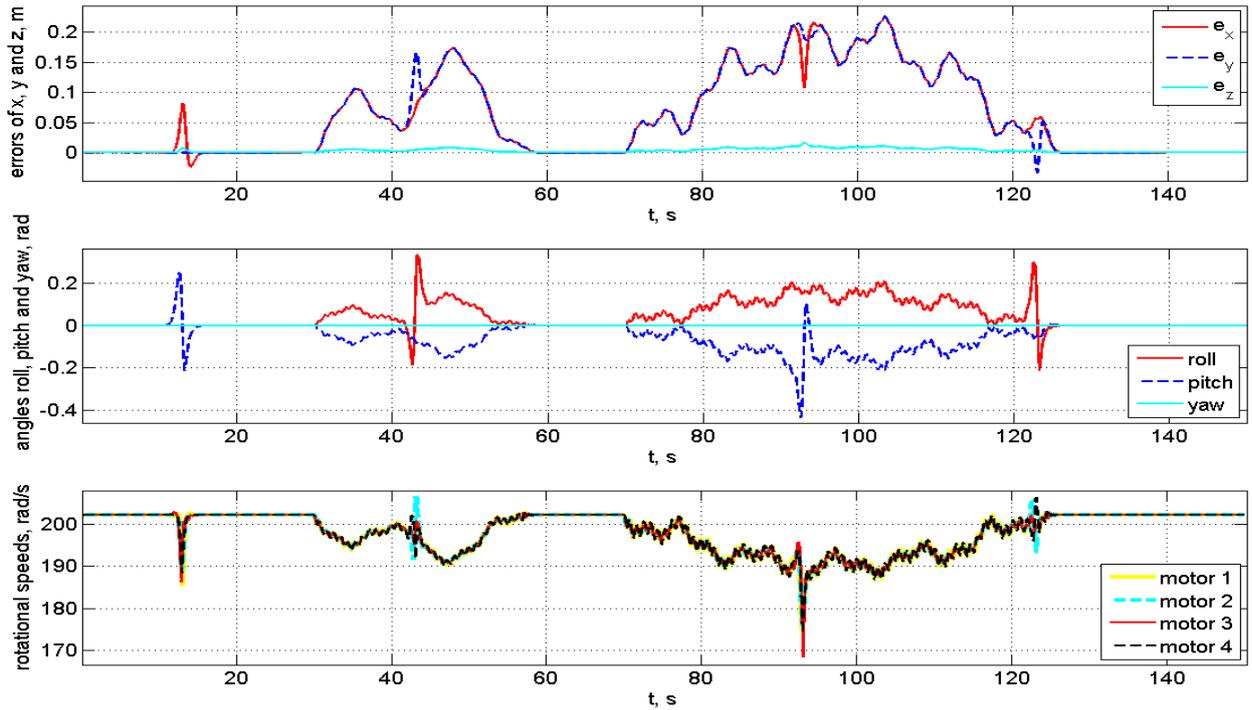


Figure 5.7: The backstepping control with wind disturbance and the gains in Table 5.2.

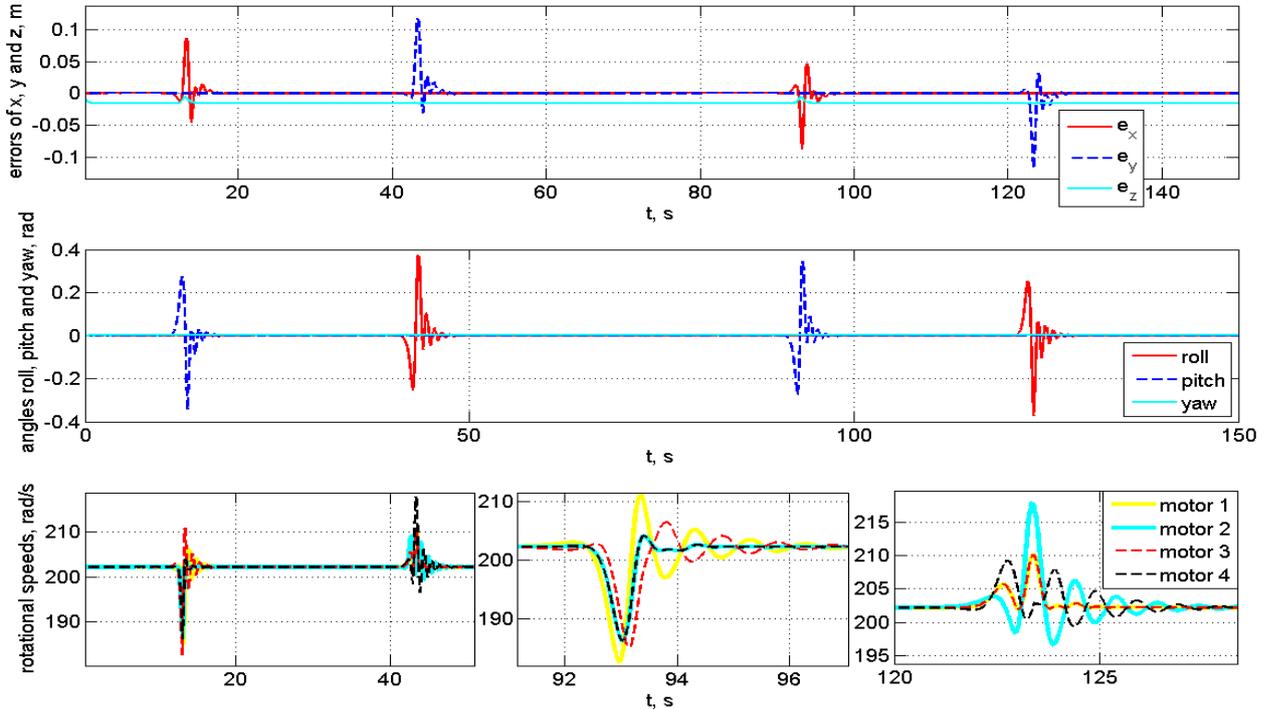


Figure 5.8: The backstepping control with parameter uncertainties and the gains in Table 5.2.

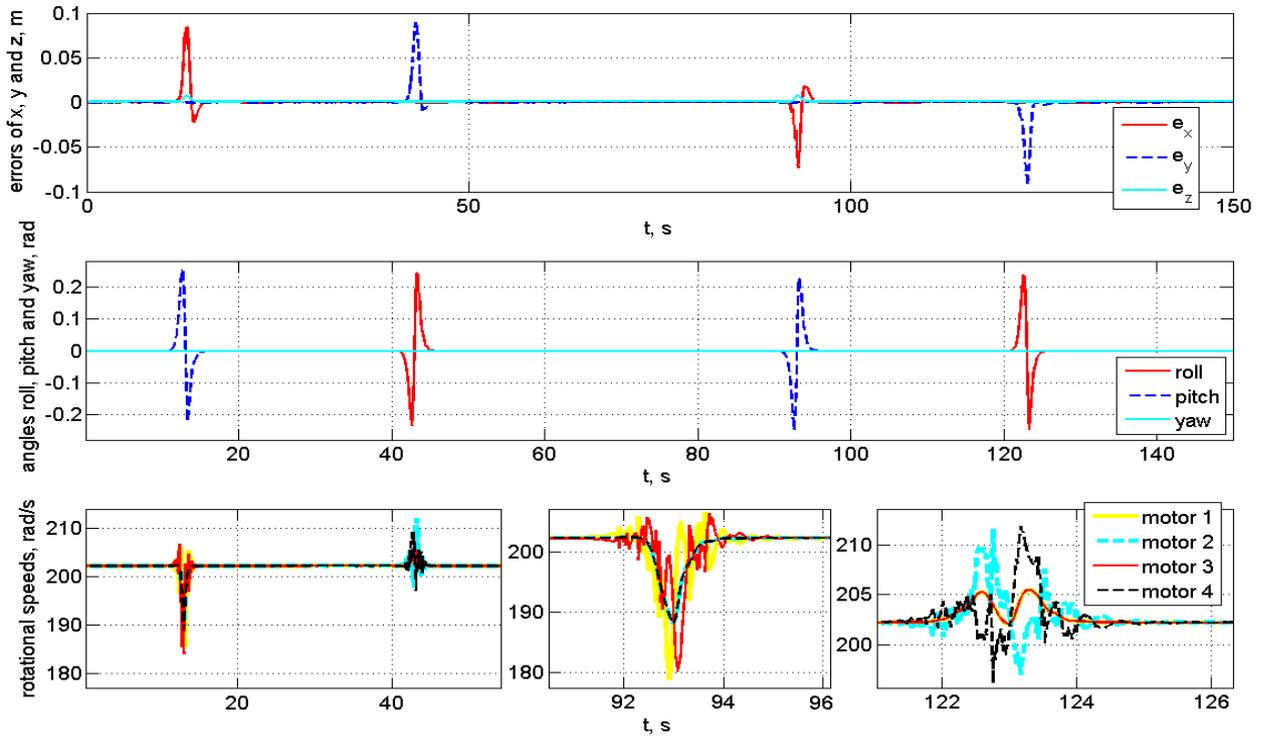


Figure 5.9: The backstepping control with sensor noises and the gains in Table 5.2.

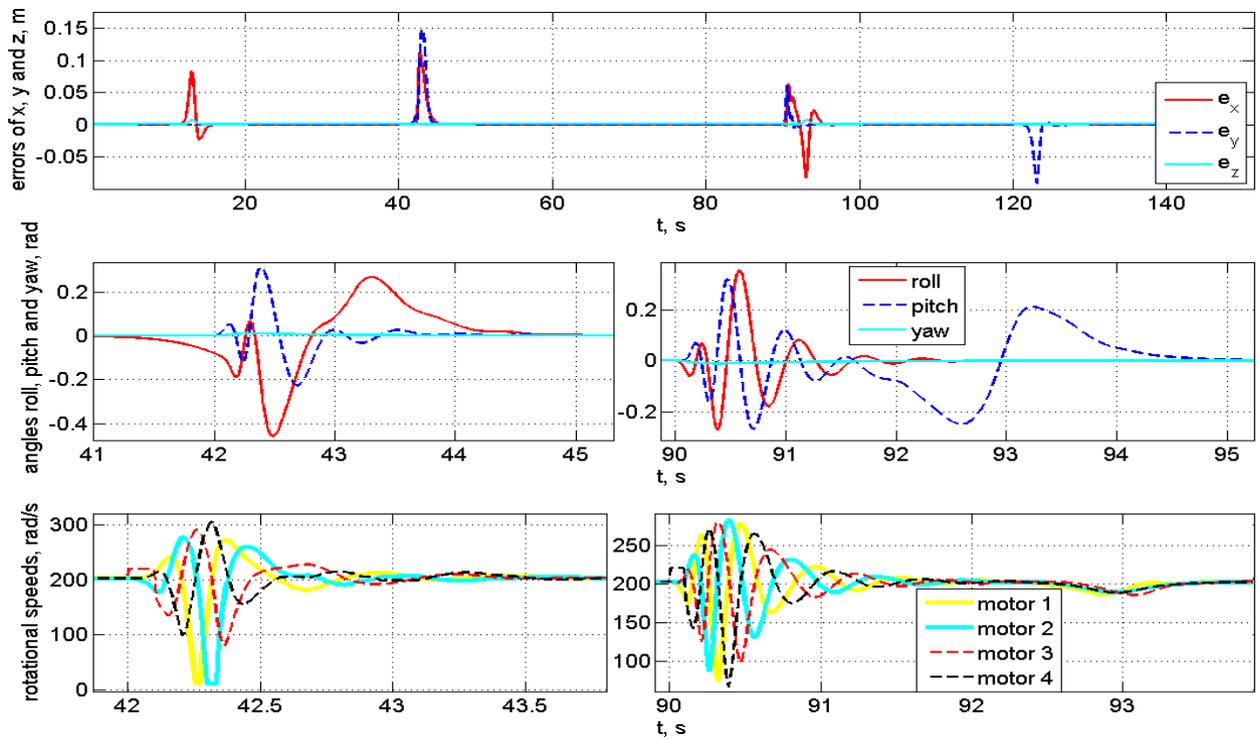


Figure 5.10: The backstepping control with actuator faults and the gains in Table 5.2.

maximum tracking error, comparing to 0.09m in the basic scenario. The actuator fault during hovering caused 0.06m maximum tracking error. While in the model free control, the fault caused 0.04m maximum tracking error during hovering. The backstepping control caused big vibrations after the actuator faults. The actuator faults caused the quadrotor tilt about 0.28rad during hovering using the backstepping control, while in the model free control the tilt angle is only 0.1 rad. After the faults, the system found its stable position in about 1.5s, while 0.5s in the model free control.

### 5.1.3 Sliding Mode Control

In this subsection, we implement the sliding mode control method proposed by Bouabdallah [20]. The control laws are described in Section 4.3.3 on Page 81. The simulation time is 150s, and the sampling period is 10ms. There are total 15000 actuation steps.

#### 1. Basic scenario

The sliding mode control is firstly implemented in the basic scenario without disturbance. The control gains are chosen as in Table 5.3. The simulation results in the basic scenario without disturbance are in Figure 5.11.

Table 5.3: The sliding mode control gains

$k_1$	2	$k_2$	10	$\alpha_1$	6	$k_3$	2	$k_4$	10	$\alpha_3$	6
$k_5$	1	$k_6$	2	$\alpha_5$	2	$k_7$	5	$k_8$	15	$\alpha_7$	8
$k_9$	1	$k_{10}$	4	$\alpha_9$	3	$k_{11}$	1	$k_{12}$	4	$\alpha_{11}$	3

In the basic scenario, the sliding mode control showed a good performance. The system stays stable during the task, and the maximum absolute tracking error is 0.07m which is almost the same with 0.09m of the backstepping control, bigger than 0.042m of the model free control.

#### 2. Wind disturbance

In this subsection, the sliding mode control is simulated in the scenario with wind disturbance. The wind is depicted in Section 4.1. The control gains are always as in Table 5.3. The simulation results are in Figure 5.12.

During the wind disturbance, the control system has big tracking errors. The maximum absolute tracking error is 0.19m, which is 9.5% of the total length 2m, while in the back-

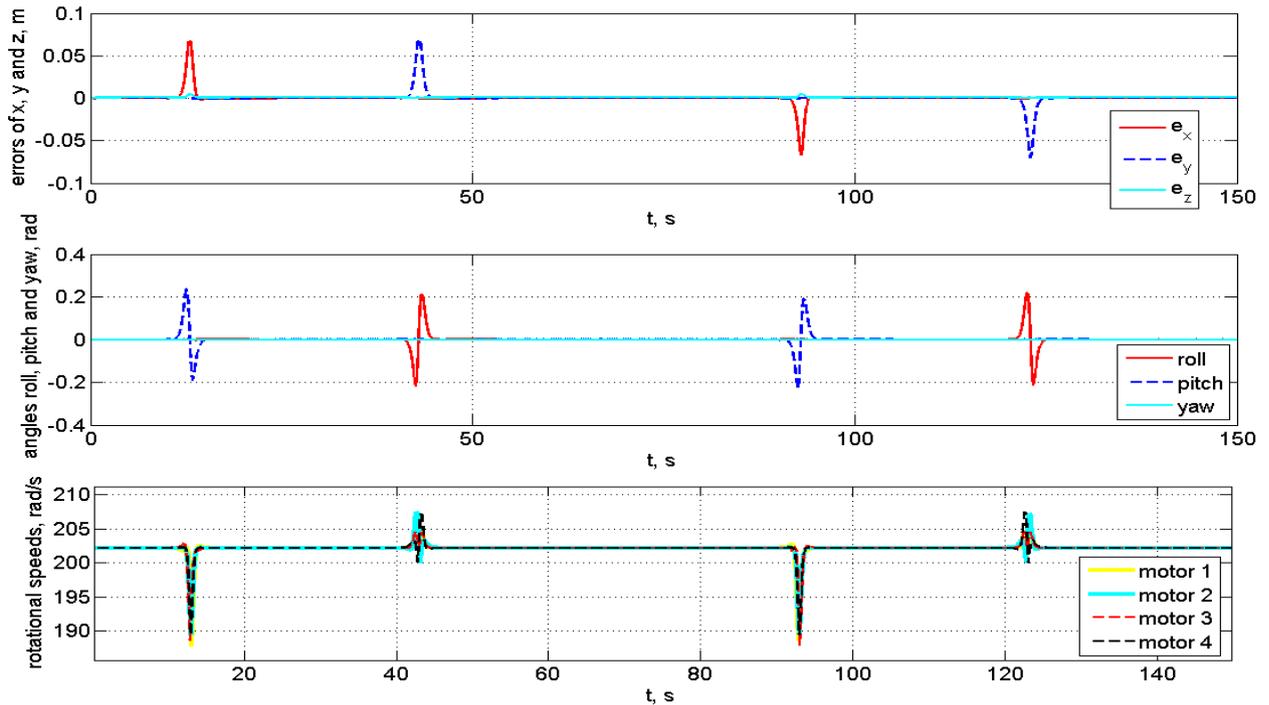


Figure 5.11: The sliding mode control in the basic scenario with the gains in Table 5.3.

stepping control the error is 0.23m. In the model free control, the maximum tracking error is only 0.042m.

### 3. Parameter uncertainties

In the scenario with model parameter uncertainties which are defined in Section 4.1, the simulation results are in Figure 5.13. The gains are in Table 5.3. The maximum absolute tracking error is 0.055m, while 0.08 in the basic scenario.

### 4. Sensor noises

The sliding mode control is also simulated in the scenario with sensor noises. The sensor noises are defined in Section 4.1. The simulation results are in Figure 5.14. The selected gains are in Table 5.3.

### 5. Actuator faults

In the scenario with actuator faults which are defined in Section 4.1. The simulation results are in Figure 5.15. The selected gains are in Table 5.3.

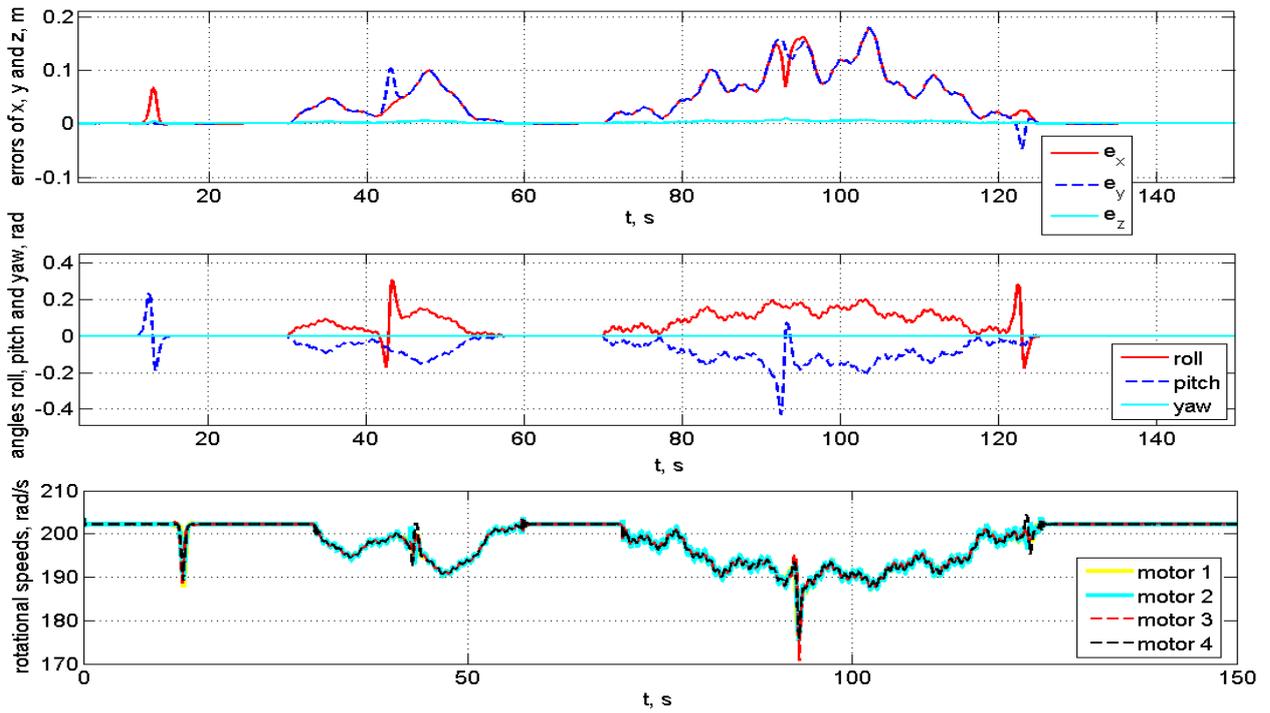


Figure 5.12: The sliding mode control with wind disturbance and the gains in Table 5.3.

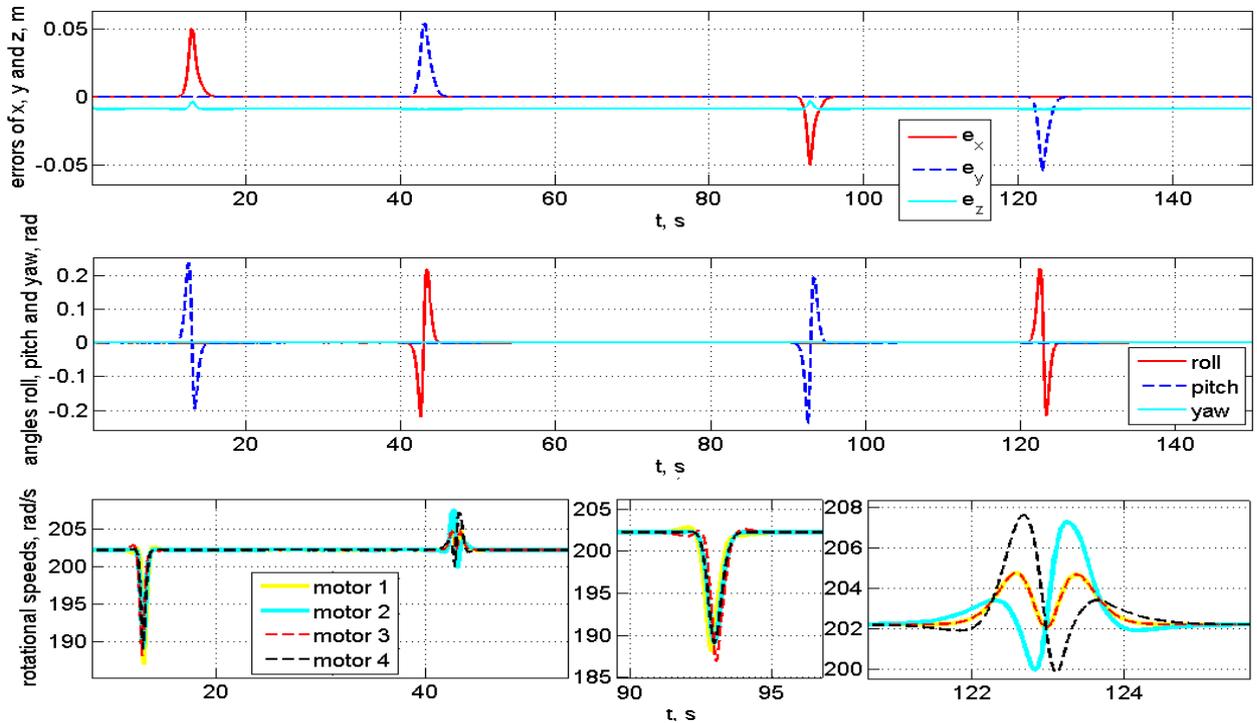


Figure 5.13: The sliding mode control with parameter uncertainties and the gains in Table 5.3.

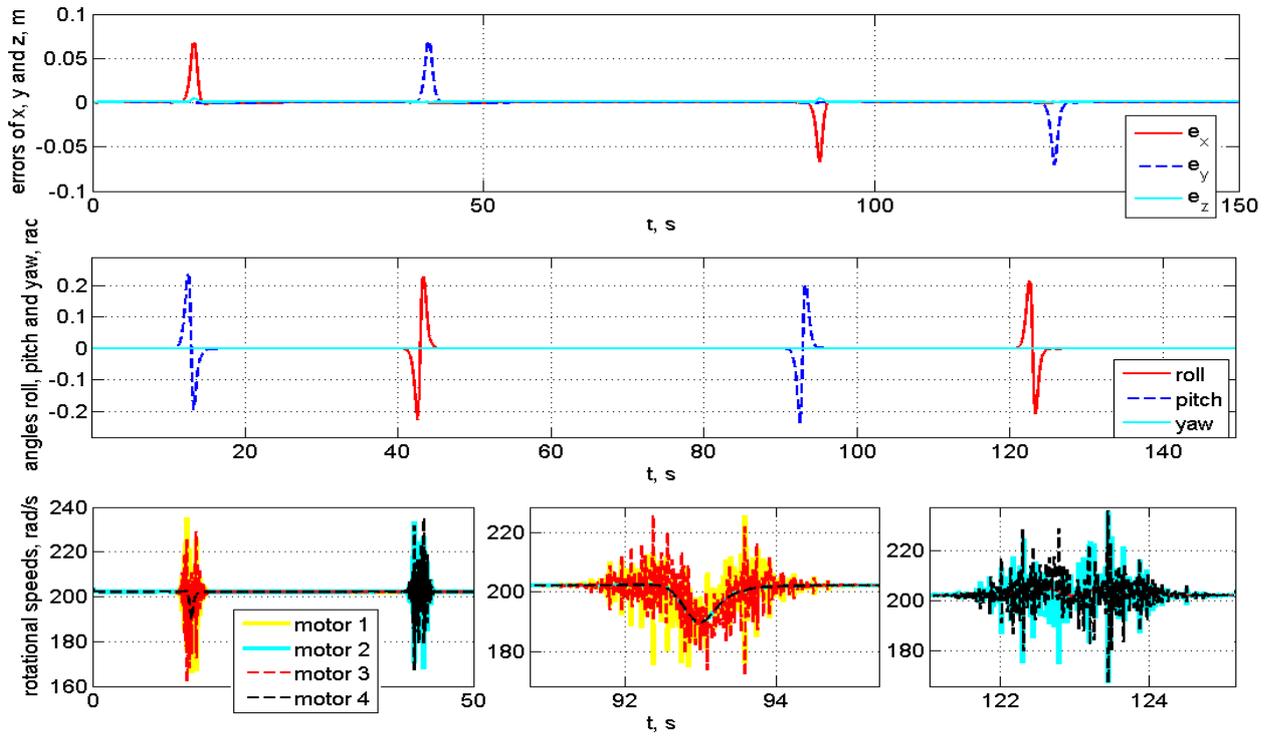


Figure 5.14: The sliding mode control with sensor noises and the gains in Table 5.3.

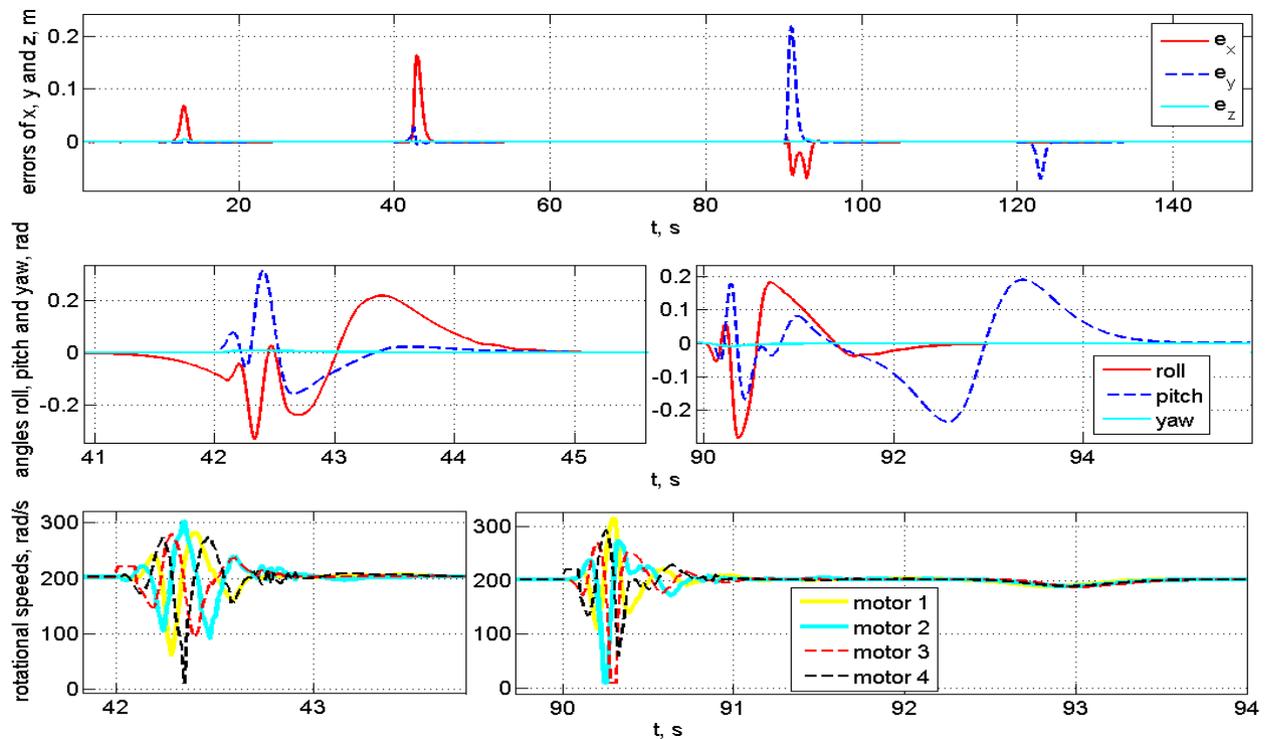


Figure 5.15: The sliding mode control with actuator faults and the gains in Table 5.3.

The fault during the hovering caused 0.21m tracking error. During hovering, the system was back to normal stable state in about 2s, while the responding time is 1s in the model free control and 1.5s in the backstepping control. The actuator fault caused the quadrotor tilt about 0.29rad during hovering, while 0.1rad in the model free control and 0.28rad in the backstepping control.

### 5.1.4 Discussion

In order to show the differences of these control methods explicitly, ten important criteria are listed as follows for the comparison:

- (1) The sum of the variances of the tracking errors in the x, y and z axis in basic scenario without disturbance:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (2) The sum of the variances of the tracking errors with wind disturbance:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (3) The sum of the variances of the tracking errors with parameter uncertainties:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (4) The sum of the variances of the tracking errors with sensor noises:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (5) The sum of the variances of the tracking errors with actuator faults:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (6) Adjusting time:  $t$ .
- (7) Maximum tilt angle:  $\phi$  or  $\theta$ .
- (8) The number of the actuation steps.
- (9) The computational complexity: the number of the addition and multiplication operations in the control law.
- (10) The energy consumption: the energy spent on motion control, the sum of the four motors' squared angular speeds:  $\sum_{i=1}^4 \int_t \omega_i^2$ .

In the time triggered scheme, the maximum absolute tracking error, the sum of the variances of the tracking errors and the energy consumption of three methods in five scenarios are shown in Figure 5.16. From the figure, we can see the time triggered model free control has generally a better performance in these criteria.

The comparison of three control methods in the basic scenario is shown in Table 5.4. From the simulation results, the control methods have almost the same performance in the basic scenario. All the control show a good stability during the flight, and the tracking errors are less than 0.1m, which is 5% of the length of 2m. The model free control has smallest

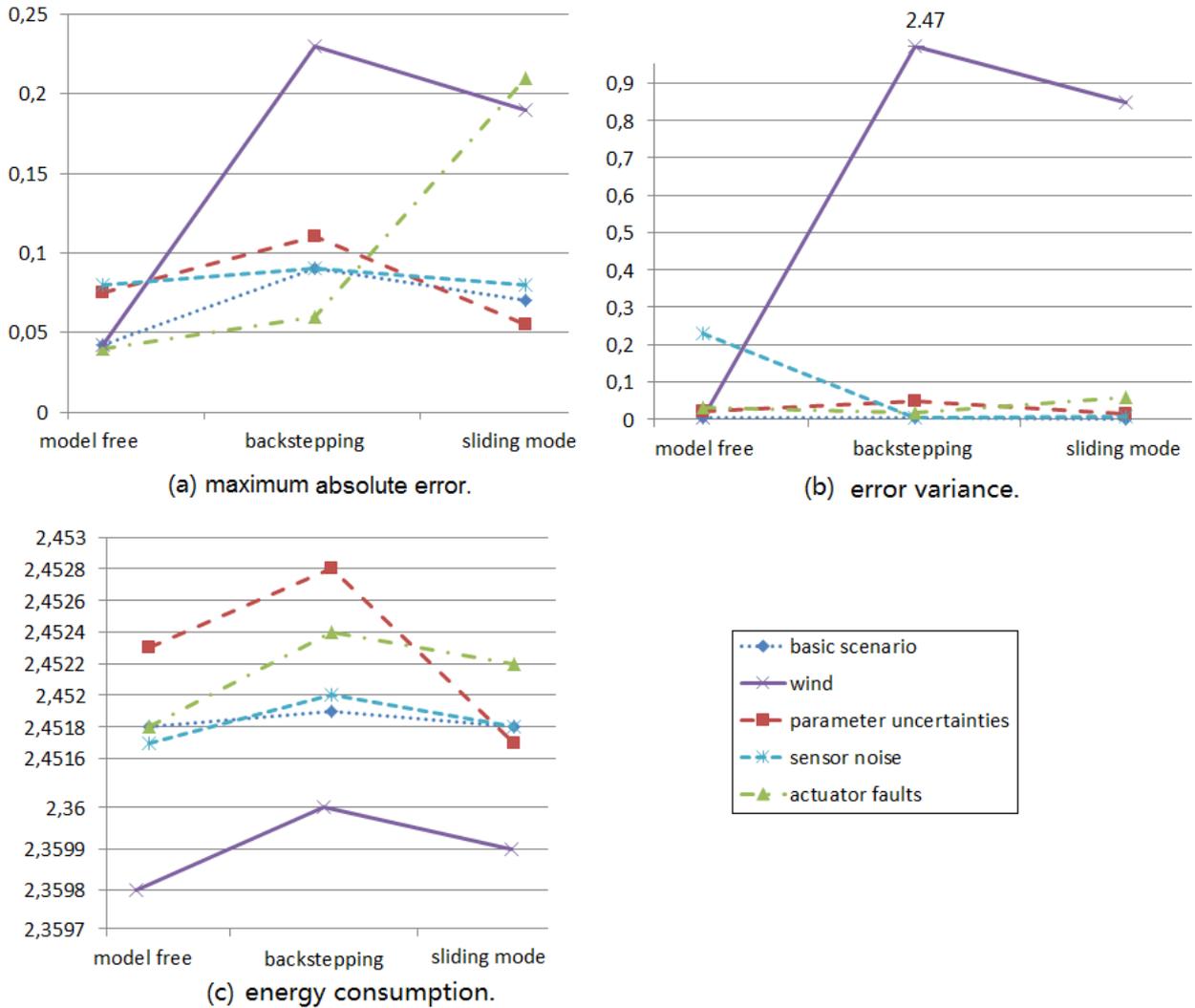


Figure 5.16: Time triggered three methods in all scenarios: (a) the maximum absolute tracking error. (b) the error variance. (c) the energy consumption  $\times 10^{-7}$ .

maximum absolute tracking error, however the sum of the variances of the errors is not the smallest.

The comparison of three control methods in the scenario with wind disturbance is shown in Table 5.5. The model free shows its advantage during the wind disturbance. The maximum absolute tracking error is the same as in the basic scenario. The sum of the variances of the tracking errors increased by 6%. The mode free control compensates well the wind disturbance. This is due to the real time updated system dynamics which is computed at each sampling period. The value  $\hat{F}$  is computed using the real time system measurement  $\hat{y}$  and the last time control input  $\tilde{u}$ . The unexpected wind disturbance is considered in  $\hat{F}$  in the real time control. Therefore, the wind disturbance has little influence for the model free

Table 5.4: The comparisons of the time triggered control methods in the basic scenario.

basic scenario	model free	backstepping	sliding mode
maximum absolute error	0.042	0.09	0.07
error variance	0.0049	0.0028	0.0003
maximum tilt angle	0.21	0.22	0.205
energy consumption	2.4518e+7	2.4519e+7	2.4518e+7
actuation steps	15000	15000	15000
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

Table 5.5: The comparisons of the time triggered control methods with wind disturbance.  $n$  is  $n$ -digit number.

wind	model free	backstepping	sliding mode
maximum absolute error	0.042	0.23	0.19
error variance	0.0052	2.4723	0.8476
maximum tilt angle	0.4	0.41	0.41
energy consumption	2.3598e+7	2.3600e+7	2.3598e+7
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

control. As the same time, the backstepping control and sliding mode control have large tracking errors during the wind disturbance. The maximum tracking errors increased by 155.5% and 171.4% in the backstepping control and sliding mode control respectively. The sums of the variances of the tracking errors increased 88196% and 28433%.

The comparison in the scenario with parameter uncertainties is shown in Table 5.6. All the control methods show good performance in this scenario. When the parameters are underestimated and the system is under-actuated, all the control methods keep their stability and the tracking errors are rather small. In the model free control, the maximum tracking error and the sum of the variances increased by 78% and 333% related to the basic scenario. In the backstepping control, the maximum tracking error and the sum of the variances increased by 22% and 1571%. In the sliding mode control, the maximum tracking error and the sum of the variances increased by -21.4% and 4500%.

The comparison in the scenario with sensor noises is shown in Table 5.7. The backstepping control has the smallest error variance. The sensor noises caused oscillations in the other two methods, especially in the sliding mode control. It is due to the switching feature of the sliding mode control.

Table 5.6: The comparisons of the time triggered control methods with parameter uncertainties.  $n$  is  $n$ -digit number.

parameter uncertainties	model free	backstepping	sliding mode
maximum absolute error	0.075	0.11	0.055
error variance	0.0212	0.0468	0.0138
maximum tilt angle	0.35	0.38	0.22
energy consumption	2.4523e+7	2.4528e+7	2.4517e+7
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

Table 5.7: The comparisons of the time triggered control methods with sensor noises.  $n$  is  $n$ -digit number.

sensor noises	model free	backstepping	sliding mode
maximum absolute error	0.08	0.09	0.08
error variance	0.2309	0.0029	0.0088
energy consumption	2.4517e+7	2.4520e+7	2.4518e+7
maximum tilt angle	0.22	0.23	0.21
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

Table 5.8: The comparisons of the time triggered control methods with actuator faults.  $n$  is  $n$ -digit number.

actuator faults	model free	backstepping	sliding mode
maximum error caused by fault	0.04	0.06	0.21
maximum tilt angle caused by fault	0.1	0.28	0.29
adjusting time after the fault	1	1.5	2.0
error variance	0.0327	0.0167	0.0585
energy consumption	2.4518e+7	2.4524e+7	2.4522e+7
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

The comparison in the scenario with actuator faults is shown in Table 5.8. The model free control has the smallest adjusting time after the faults. It responds quickly to the change of the dynamics due to the real time computation of the value  $\hat{F}$ . The sliding mode control has the largest adjusting time, and the actuator faults caused a biggest oscillation in the actuators and therefore the tilt angles. The maximum tilt angle of the backstepping control and the sliding mode control are 0.28rad and 0.29rad, while 0.1 rad in the model free control.

## 5.2 Event triggered Control

Quadrotors have many applications, such as border patrol, surveillance, search & rescue, aerial photography, etc. In these applications, a long flight duration time is needed to finish the tasks. The flight duration time depends on two factors: the capacity of the onboard battery and the energy consumption of the quadrotor system. The commercialized quadrotors usually use Lithium Polymer (LiPo) batteries. However, a bigger battery capacity means a heavier load and a more expensive price. In order to extend the flight duration, we can also reduce energy consumption. As in these applications mentioned above, quadrotors have usually a smooth trajectory and without aggressive movement. When quadrotors follow a trajectory, the tracking errors are often in a limited bound. Therefore, an event triggered control scheme is proposed for these applications. Limit tolerances are set for control systems, and a control system is actuated only when the limits are passed. In this way, unnecessary computations are avoided and the energy consumption is reduced.

In this chapter, all the control methods proposed in Section 4.3 will be implemented in the event triggered scheme. As in the time triggered scheme, the control methods are also implemented in the different scenarios: the basic scenario, the scenarios with wind disturbance, model parameter uncertainties, sensor noises and actuator faults. The descriptions of these scenarios can be found in Section 4.1. The reference trajectory is a square path with a length of 2m. The 3D trajectory is given in Figure 4.1. The expression of the reference trajectory is in Equation (4.1). The total simulation time is 150s.

The thresholds of the event triggered laws are selected based on the tasks of quadrotors. A threshold which keeps the preciseness of tracking and reduces the actuation steps should be used. The selected scenario is reference tracking. The quadrotor needs to track a square trajectory while hovering at a certain altitude. We set that the tracking along the  $x$  and  $y$  axis should be precise and that along the  $z$  axis can have a bigger tracking error. The thresholds set for  $x$ ,  $y$  and  $z$  are 0.001m, 0.001m and 0.02m respectively. As mentioned in the previous paragraph, the maximum changes of the linear velocity and position are 0.098m/s and 0.49mm respectively in one sampling period 10ms. A threshold of 0.001m is reasonable, which keeps the preciseness of tracking and at the same time reduces the actuation steps.

Here, the event triggering laws are chosen to be the error difference threshold as in Equation (4.10). We set the error difference threshold of  $z$  0.02m, yaw angle 0.1rad,  $x$  and

$y$  0.001m.

$$e_z < 0.02\text{m}, \quad e_x, e_y < 0.001\text{m}, \quad e_\psi < 0.1\text{rad}. \quad (5.1)$$

Every sampling period 10ms, the control system verifies the tracking error differences. If one of the error differences is bigger than the limit mentioned above, the system will compute new control signals and send them to the actuator. If all of the error differences are smaller than the limits, the control system will not compute the control signals, and simply keep the old control signals to the next sampling. Therefore, the system can save computational resources when quadrotors have small tracking errors, especially when quadrotor move smoothly or hovering.

### 5.2.1 Model Free Control

In this section, we implement the model free control in several different scenarios. The control laws and the scenarios are described in Chapter 4. The event triggering law is defined at the beginning of this chapter.

#### 1. Basic scenario

The event triggered model free control is firstly simulated in the basic scenario without disturbance. The control gains are chosen as in Table 5.9. The simulation results in the basic scenario without disturbance are in Figure 5.17. The first figure is the tracking errors along the  $x$ ,  $y$  and  $z$  axis. The second figure is the angles  $\phi$ ,  $\theta$  and  $\psi$ . The third figure is the angular velocities of the four rotors. The numbers of the rotor are defined in Figure 2.1 in Chapter 2.

Table 5.9: The event triggered model free control gains

$k_3^x$	3	$k_2^x$	9	$k_1^x$	8	$k_0^x$	2.5	$k_1^z$	2	$k_0^z$	5
$k_3^y$	3	$k_2^y$	8	$k_1^y$	8	$k_0^y$	3	$k_1^\psi$	5	$k_0^\psi$	10

In the basic scenario, the system using the model free control kept the stability during the flight and followed the reference trajectory. The maximum absolute tracking error is 0.06m, while 0.042m in the time triggered model free control. The actuation steps is 9008, which is 60% of 15000 in the time triggered scheme.

## 2. Wind disturbance

The model free control is also simulated in the scenario with wind disturbance. The wind is depicted in Section 4.1. In order to simulate the realistic case, the control gains are not changed, which are the same as in the basic scenario in Table 5.9. The simulation results are in Figure 5.18.

The control system shows its stability in the wind disturbance. The system kept stable during the disturbance. The maximum absolute tracking error is 0.06m, which is the same with the basic scenario. The actuation steps is 12110, while 9008 in the basic scenario.

## 3. Parameter uncertainties

Then the model free control is simulated in the scenario with model parameter uncertainties. The parameter uncertainties are defined in Section 4.1. The parameters are estimated smaller than the real values. Therefore, the system is under-actuated. The gains are not changed as in Table 5.9. The simulation results are in Figure 5.19.

The system stays stable with parameter uncertainties. The maximum absolute tracking error is 0.19m, which increased 217% respecting to 0.06m in the basic scenario. In the event triggered scheme, the parameter uncertainties caused some oscillations in the system. The actuation steps is 10803, while 9008 in the basic scenario.

## 4. Sensor noises

The model free control is also simulated in the scenario with sensor noises. The sensor noises are defined in Section 4.1. The gains are not changed as in Table 5.9. The simulation results are in Figure 5.20.

The actuation steps is 15000. Actually the event triggered scheme is changed into the time triggered scheme, as the actuation laws are triggered at each steps. The maximum absolute tracking error is 0.08m, while 0.06m in the basic scenario.

## 5. Actuator faults

The model free control is also simulated in the scenario with actuator faults. The actuator faults are defined in Section 4.1. The gains are in Table 5.9. The simulation results are in Figure 5.21.

The system kept stable during the actuator faults. The system adjusts to its stable states in 0.5s. The adjusting time is the same as in the time triggered scheme. The maximum absolute tracking error is 0.13m, which is happened at the actuator fault during tracking.

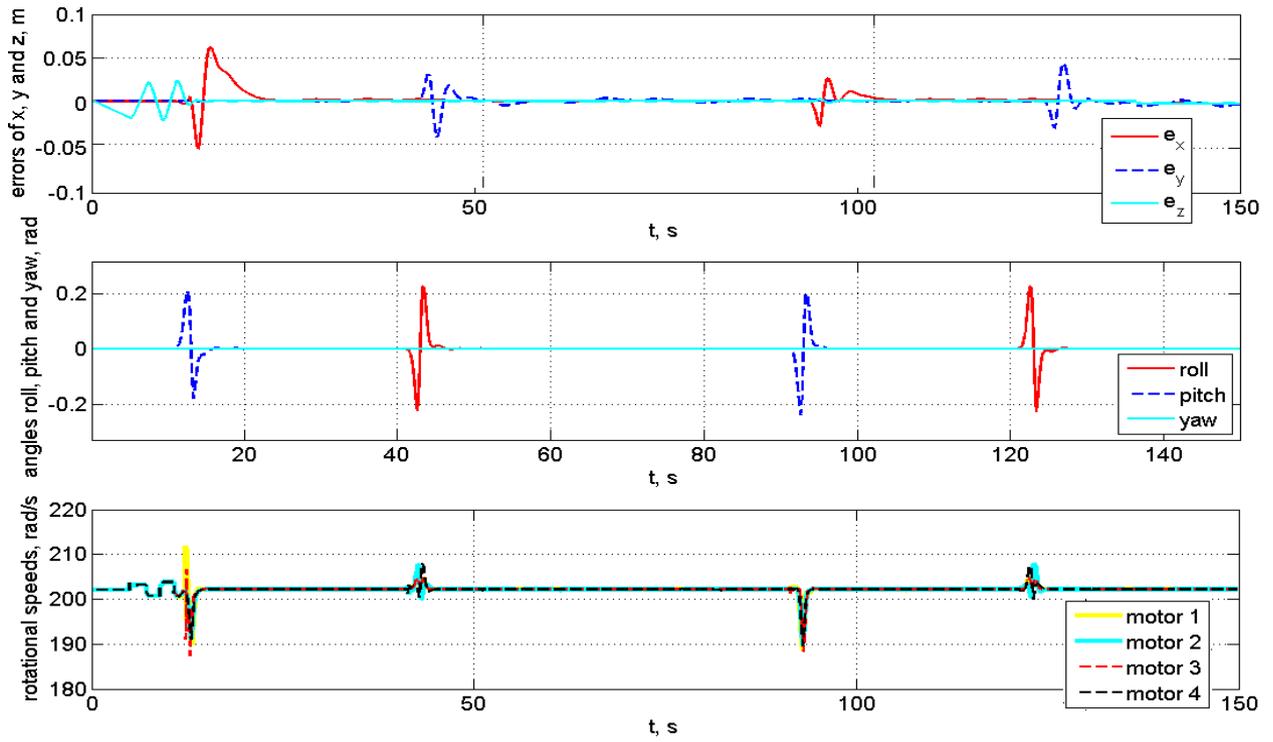


Figure 5.17: The model free control in the basic scenario with the gains in Table 5.1.

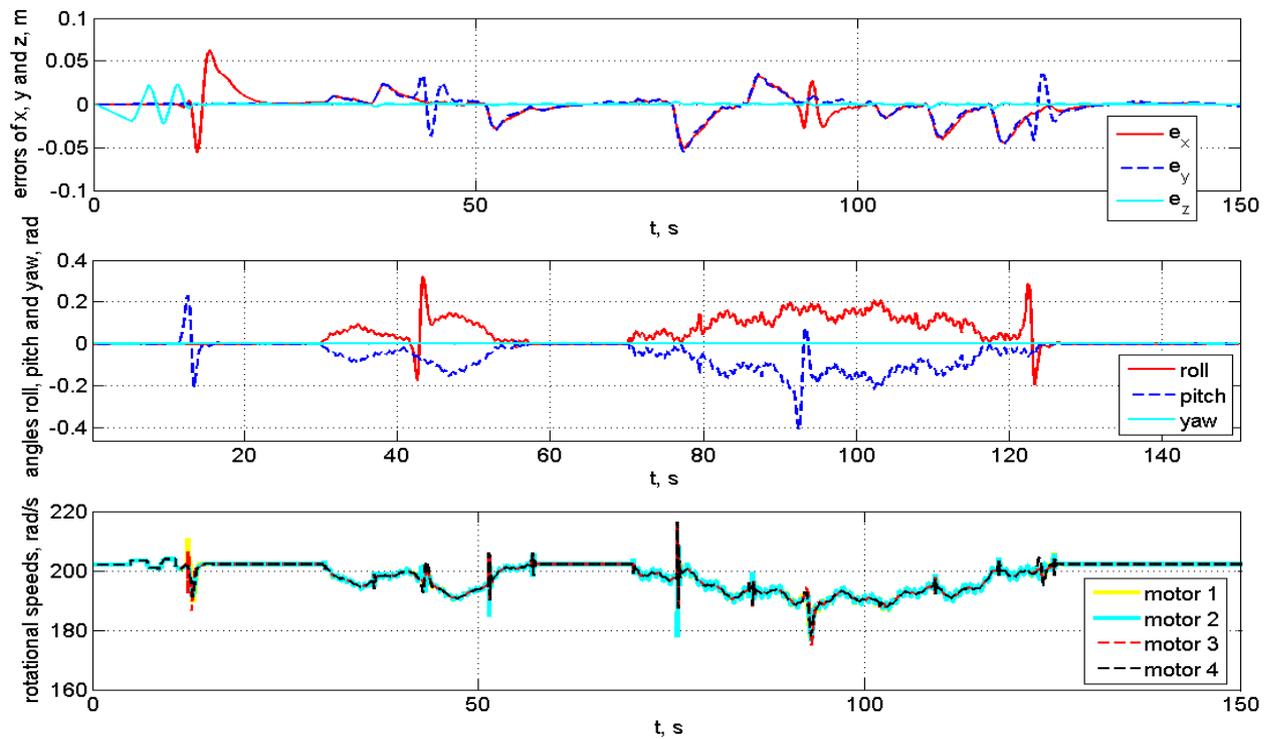


Figure 5.18: The model free control with wind disturbance and the gains in Table 5.1.

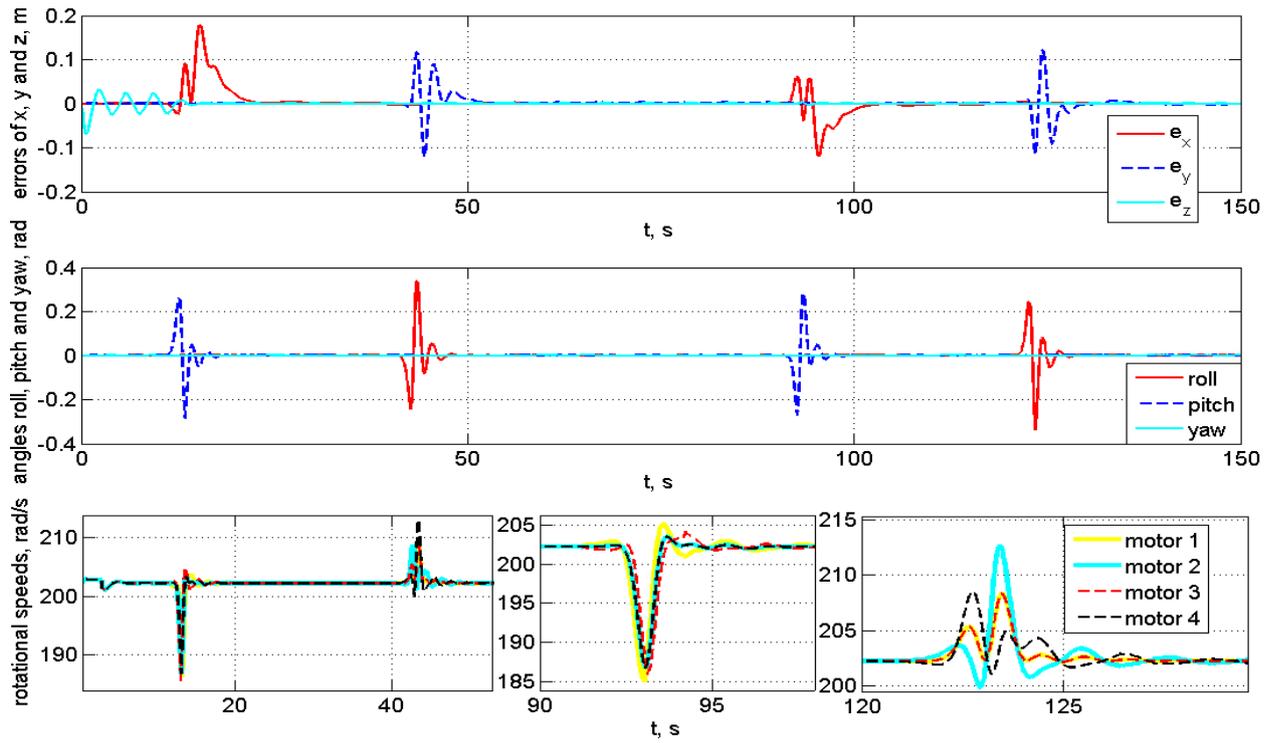


Figure 5.19: The model free control with parameter uncertainties and the gains in Table 5.1.

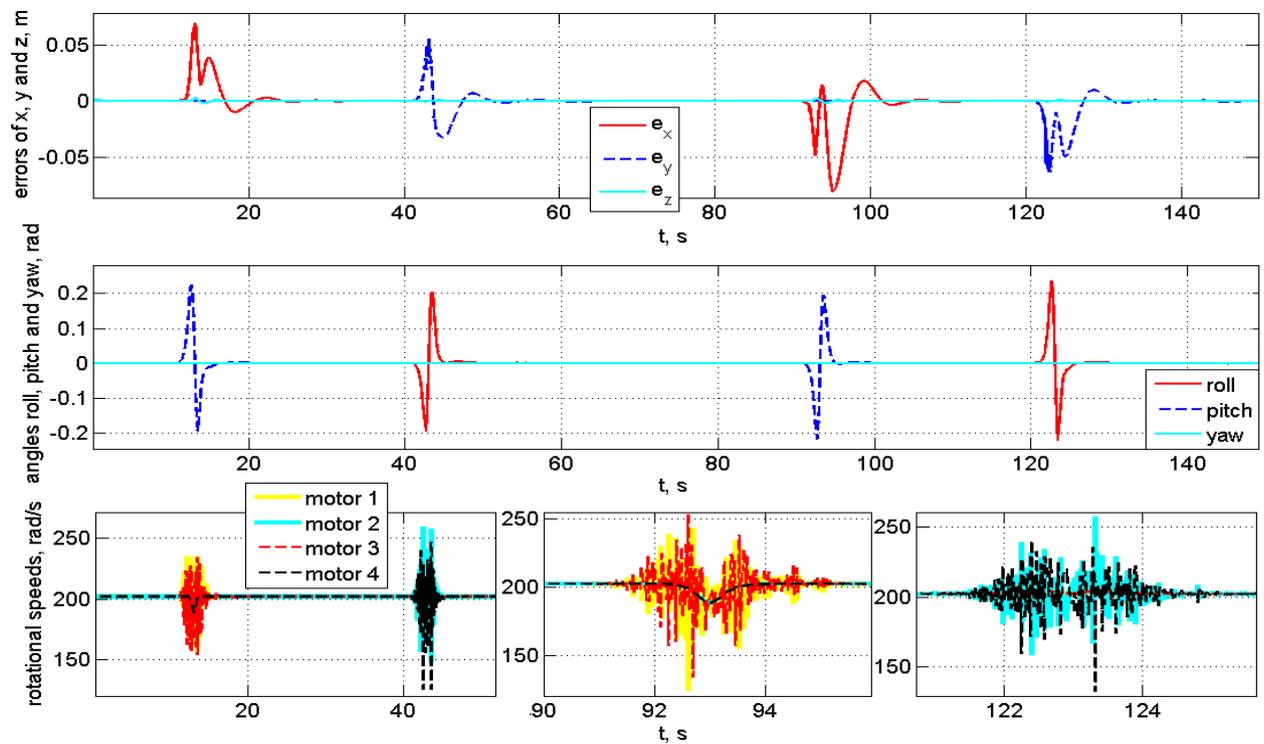


Figure 5.20: The model free control with sensor noises and the gains in Table 5.1.

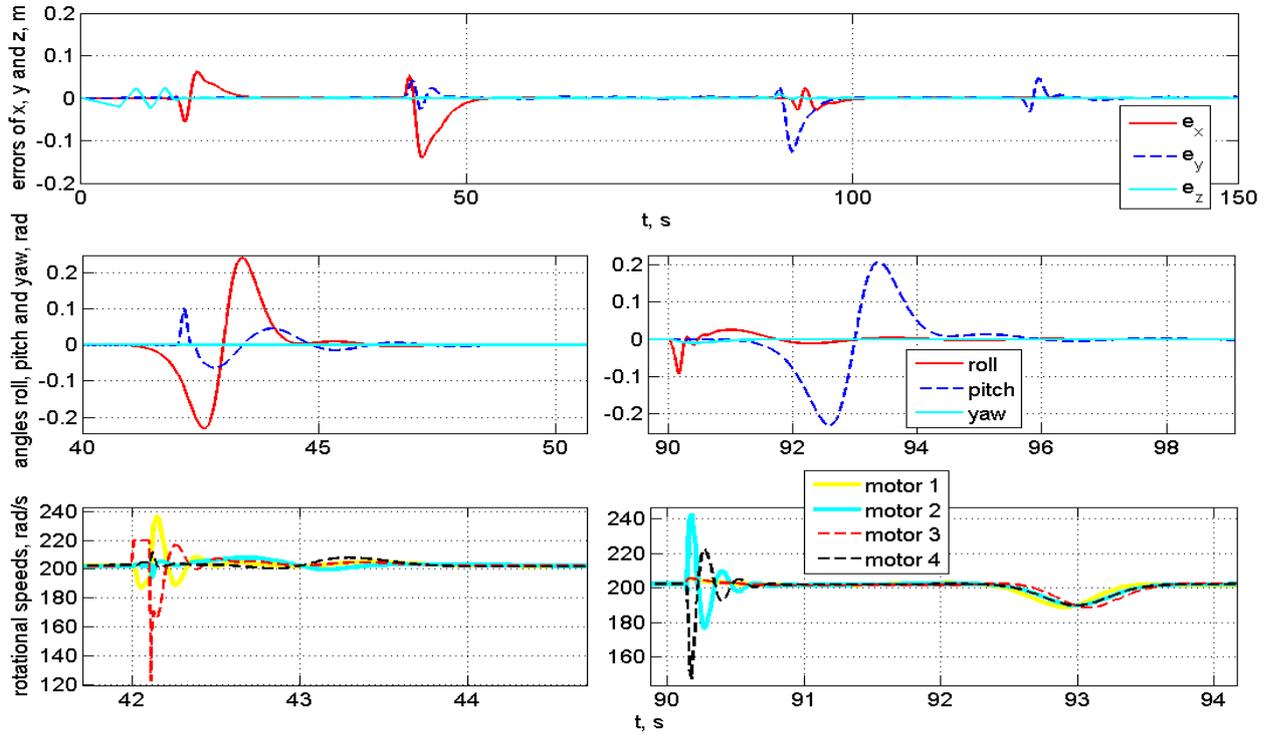


Figure 5.21: The model free control with actuator faults and the gains in Table 5.1.

The actuator fault during hovering caused 0.12m maximum absolute tracking error. The actuator fault caused the quadrotor tilt about 0.1 rad during hovering. The actuation steps is 9298, while 9008 in the basic scenario.

## 5.2.2 Backstepping Control

In this subsection, we implement the backstepping control method proposed by Bouabdallah [20]. The control laws are described in Section 4.3.2. The simulation time is 150s. The event triggering law is defined at the beginning of this chapter.

### 1. Basic scenario

The backstepping control is firstly simulated in the basic scenario without disturbance. The control gains are chosen as in Table 5.10. The simulation results in the basic scenario without disturbance are in Figure 5.22. The first figure is the tracking errors along the  $x$ ,  $y$  and  $z$  axis. The second figure is the angles  $\phi$ ,  $\theta$  and  $\psi$ . The third figure is the angular velocities of the four rotors. The numbers of the rotor are defined in Figure 2.1 in Chapter 2.

In the basic scenario, the backstepping control followed the reference trajectory while

Table 5.10: The backstepping control gains

$\alpha_1$	2	$\alpha_2$	5	$\alpha_3$	2	$\alpha_4$	5	$\alpha_5$	1	$\alpha_6$	1
$\alpha_7$	2	$\alpha_8$	10	$\alpha_9$	2	$\alpha_{10}$	3	$\alpha_{11}$	2	$\alpha_{12}$	3

keeping stable. The maximum tracking error is 0.1m, which is 4.5% of the total length 2m. In the time triggered scheme, the maximum tracking error is 0.09m. The actuation steps is 9987, while 9008 of the model free control in the basic scenario.

## 2. Wind disturbance

In this subsection, the backstepping control is simulated in the scenario with wind disturbance. The wind is depicted in Section 4.1. The control gains are the same as in Table 5.10. The simulation results are in Figure 5.23.

During the wind disturbance, the control system had big tracking errors. The maximum absolute tracking error is 0.22m, which is 11% of the total length 2m. In the basic scenario, the maximum error is only 0.09m. In the model free control, the maximum tracking error is 0.06m. After the wind disturbance, the system followed the reference trajectory, and kept the tracking error in a small limit as in the scenario without wind disturbance. The actuation steps is 12116, while 9987 in the basic scenario.

## 3. Parameter uncertainties

Then, the backstepping control is simulated in the scenario with model parameter uncertainties. The parameter uncertainties are defined in Section 4.1. The gains are in Table 5.10. The simulation results are in Figure 5.24.

The maximum absolute tracking error is 0.22m, while 0.11m in the time triggered scheme. The maximum tilt angle is 0.49rad. The actuation steps is 14976, while 9987 in the basic scenario.

## 4. Sensor noises

The backstepping control is also simulated in the scenario with sensor noises. The sensor noises are defined in Section 4.1. The control gains are in Table 5.10. The simulation results are in Figure 5.25. The actuation steps is 15000.

## 5. Actuator faults

The backstepping control is also simulated in the scenario with actuator faults. The actuator

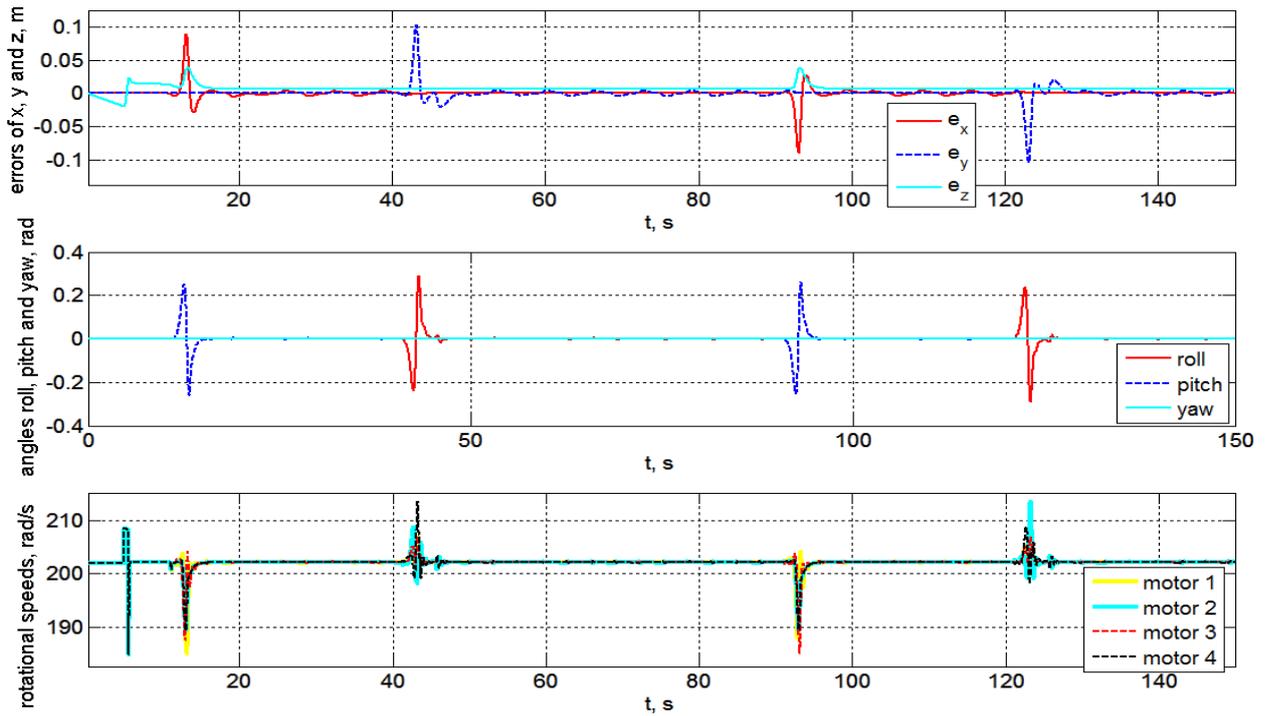


Figure 5.22: The backstepping control in the basic scenario with the gains in Table 5.2.

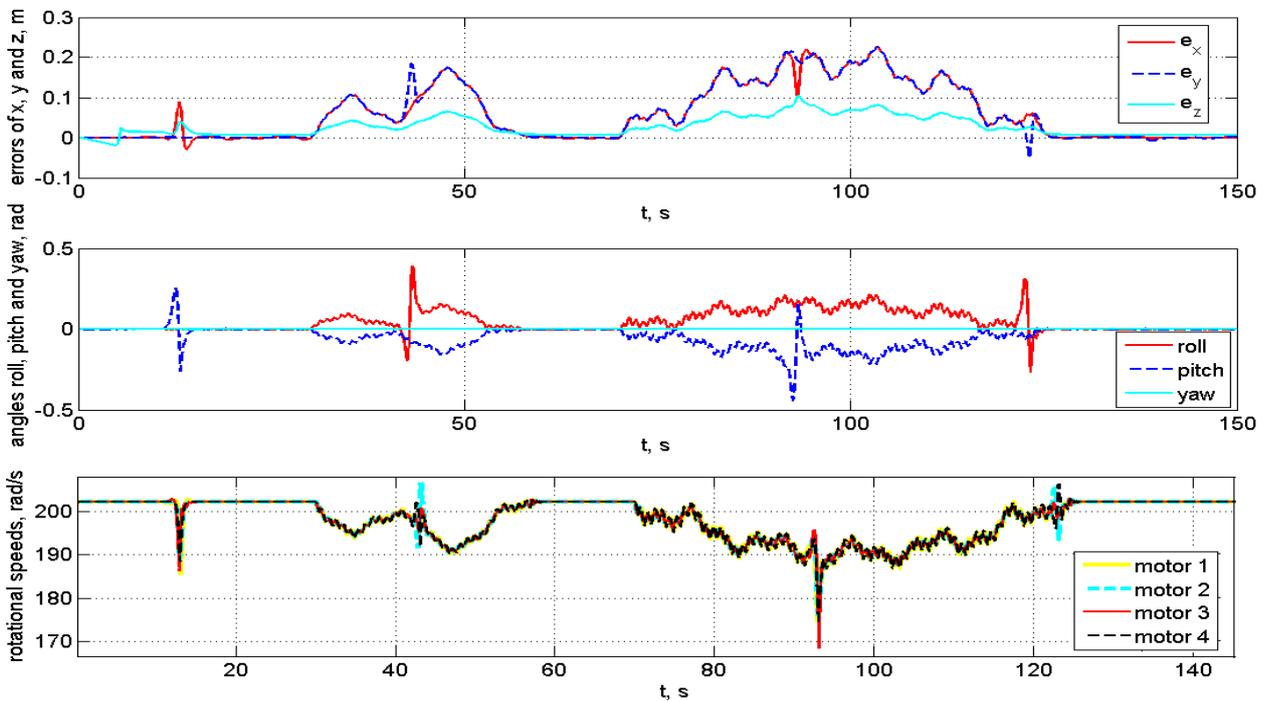


Figure 5.23: The backstepping control with wind disturbance and the gains in Table 5.2.

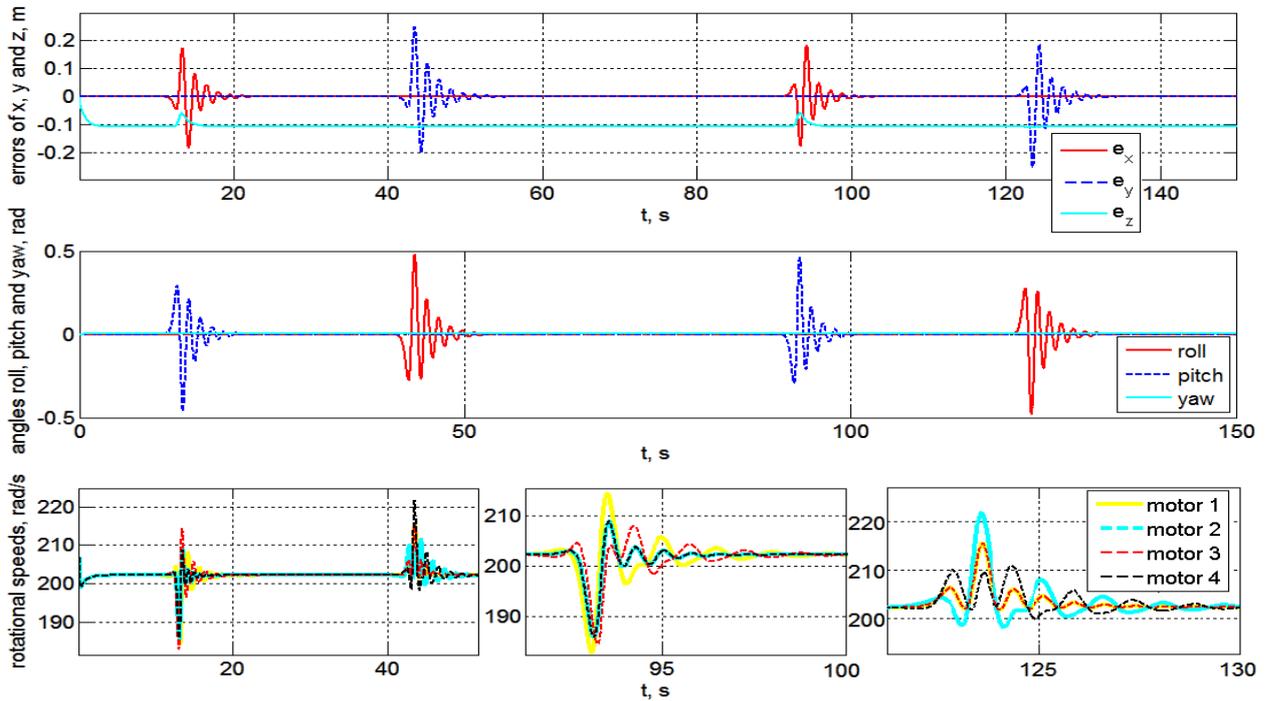


Figure 5.24: The backstepping control with parameter uncertainties and the gains in Table 5.2.

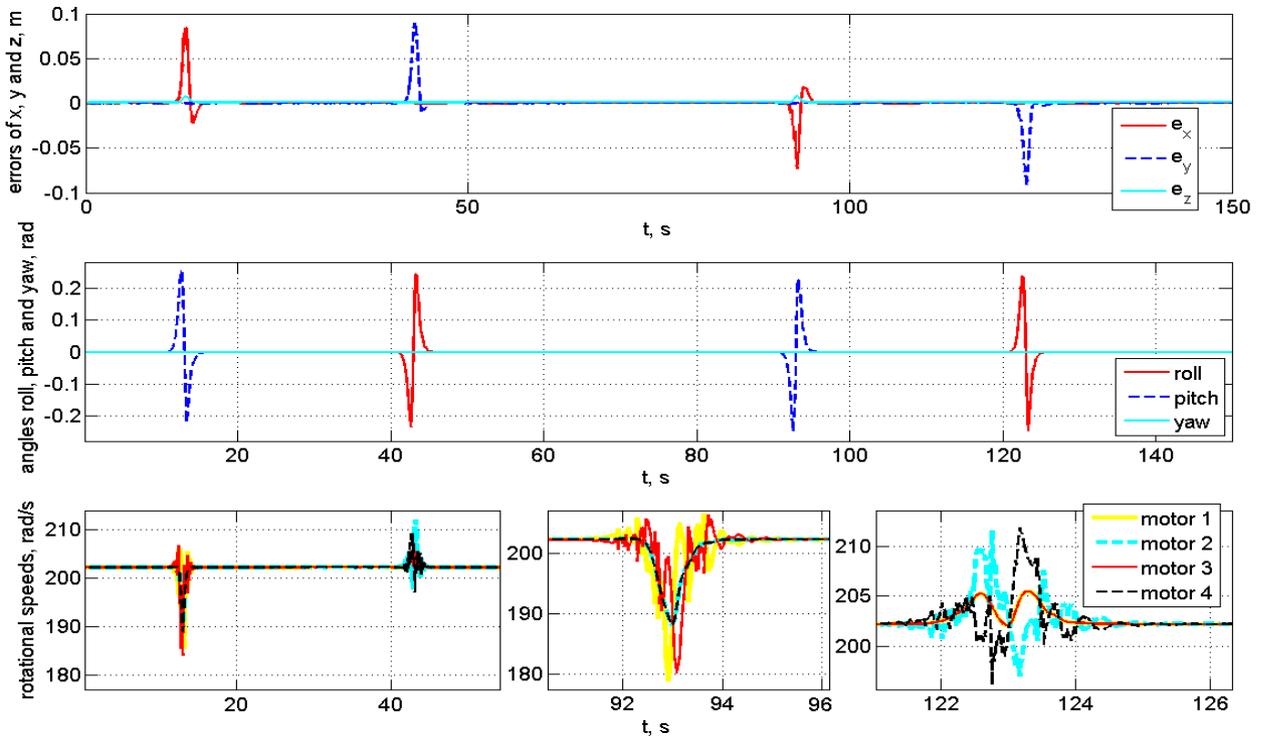


Figure 5.25: The backstepping control with sensor noises and the gains in Table 5.2.

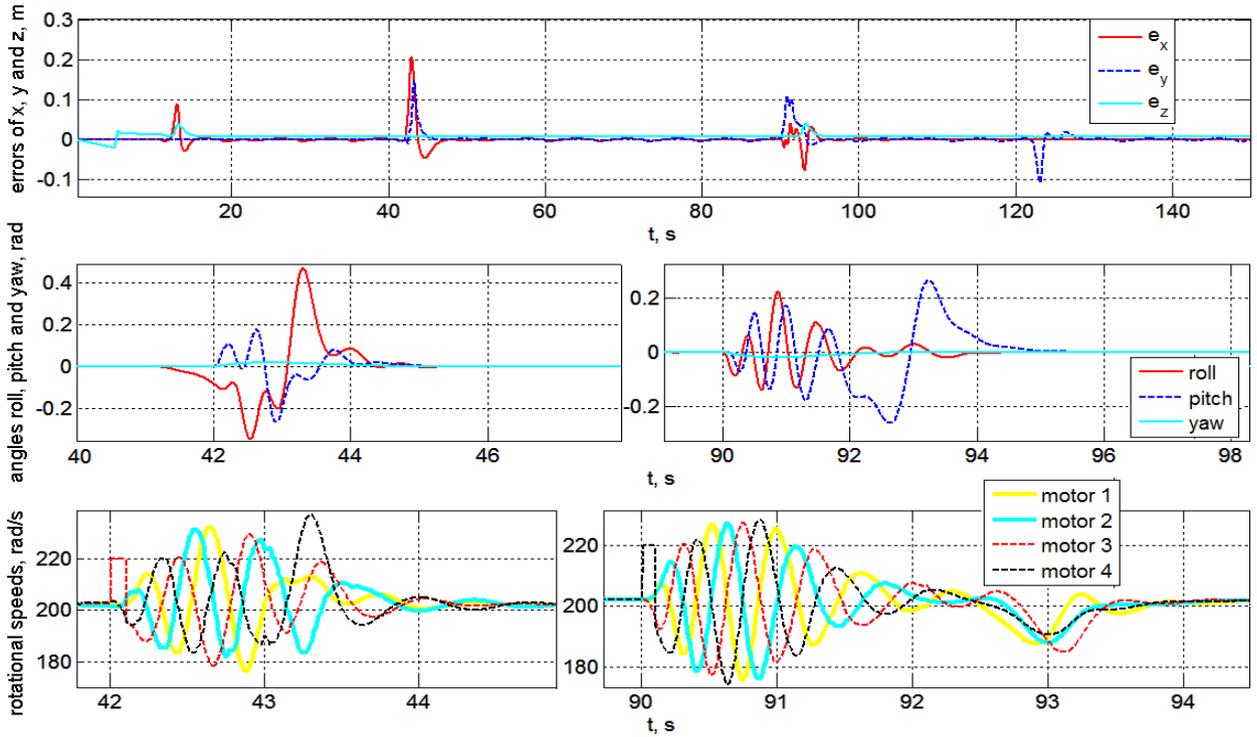


Figure 5.26: The backstepping control with actuator faults and the gains in Table 5.2.

faults are defined in Section 4.1. The control gains are in Table 5.10. The simulation results are in Figure 5.26.

Two actuator faults happened during the moving and hovering. In both cases, the backstepping control kept its stability. The actuator fault during the moving caused 0.2m maximum tracking error, comparing to 0.1m during hovering. The actuator fault caused the quadrotor tilt about 0.21rad during hovering using the backstepping control, while in the model free control the tilt angle is only 0.1rad. After the faults, the system found its stable position in about 2.5s. The responding time is bigger than the model free control which is about 0.5s. The actuation steps is 10020, while 9987 in the basic scenario.

### 5.2.3 Sliding Mode Control

In this subsection, we implement the sliding mode control method proposed by Bouabdallah [20]. The control laws are described in Section 4.3.3 in Chapter 4. The simulation time is 150s. The event triggering laws are defined at the beginning of this section.

### 1. Basic scenario

The sliding mode control is firstly implemented in the basic scenario without disturbance. The control gains are chosen as in Table 5.11. The simulation results in the basic scenario without disturbance are in Figure 5.27.

Table 5.11: The sliding mode control gains

$k_1$	1	$k_2$	5	$\alpha_1$	3	$k_3$	1	$k_4$	5	$\alpha_3$	3
$k_5$	1	$k_6$	2	$\alpha_5$	2	$k_7$	2	$k_8$	8	$\alpha_7$	4
$k_9$	0.5	$k_{10}$	2	$\alpha_9$	1.5	$k_{11}$	0.5	$k_{12}$	2	$\alpha_{11}$	1.5

In the basic scenario, the sliding mode control showed a good performance. The system stays stable during the task, and the maximum absolute tracking error is 0.1m, while 0.08m in the time triggered sliding mode control and 0.06m and 0.1m of the event triggered model free and backstepping controls in the basic scenario respectively. The actuation steps is 8493, while 9008 and 9987 in the event triggered model free and backstepping controls in the basic scenario.

### 2. Wind disturbance

In this subsection, the sliding mode control is simulated in the scenario with wind disturbance. The wind is depicted in Section 4.1. The control gains are always as in Table 5.11. The simulation results are in Figure 5.28.

During the wind disturbance, the control system has big tracking errors. The maximum absolute tracking error is 1.25m, which is 62.5% of the total length 2m, while 0.23m in the time triggered backstepping control. In the event triggered model free control, the maximum tracking error is only 0.06m. As the backstepping control, the sliding mode control is also vulnerable to the wind disturbance. After the wind disturbance, the system followed the reference trajectory, and kept the tracking error in a small limit as in the scenario without wind disturbance. However, the event triggered backstepping control should not be used in the applications with wind disturbance, because the tracking error is too high. The actuation steps is 12445, while 8493 in the basic scenario.

### 3. Parameter uncertainties

In the scenario with model parameter uncertainties which are defined in Section 4.1, the simulation results are in Figure 5.29. The gains are in Table 5.11.

The maximum absolute tracking error is 0.41m, while 0.11m in the time triggered scheme.

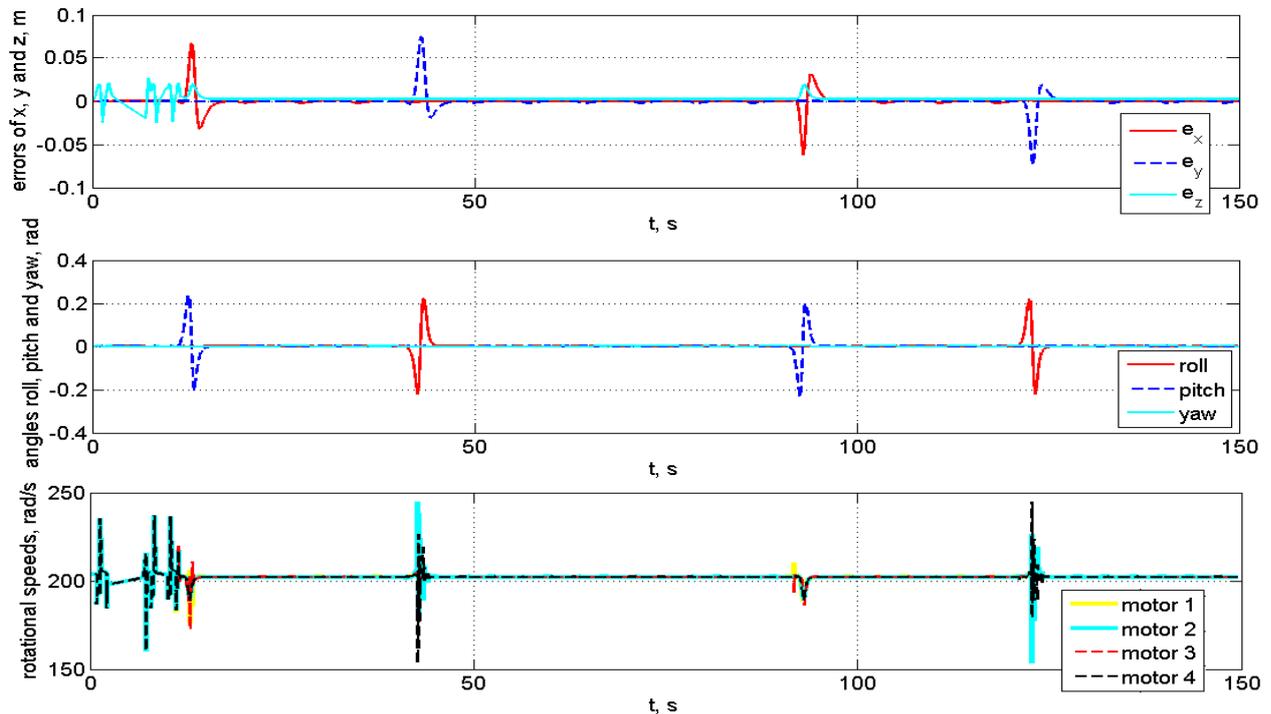


Figure 5.27: The sliding mode control in the basic scenario with the gains in Table 5.3.

All three methods show good performance in the condition with parameter uncertainties. The actuation steps is 14983, while 8493 in the basic scenario.

#### 4. Sensor noises

The sliding mode control is also simulated in the scenario with sensor noises. The sensor noises are defined in Section 4.1. The simulation results are in Figure 5.30. The selected gains are in Table 5.11.

In this scenario, the sensor noises caused many vibrations in the actuators, which are much more than the backstepping control and model free control. The sliding mode control is much more sensible to the sensor noises than the backstepping control. The reason is due to the switching feature of the sliding mode control which is less stable than the backstepping control. The actuation steps is 15000.

#### 5. Actuator faults

In the scenario with actuator faults which are defined in Section 4.1. The simulation results are in Figure 5.31. The selected gains are in Table 5.11.

The actuator faults did not cause bigger maximum tracking error during the flight. The

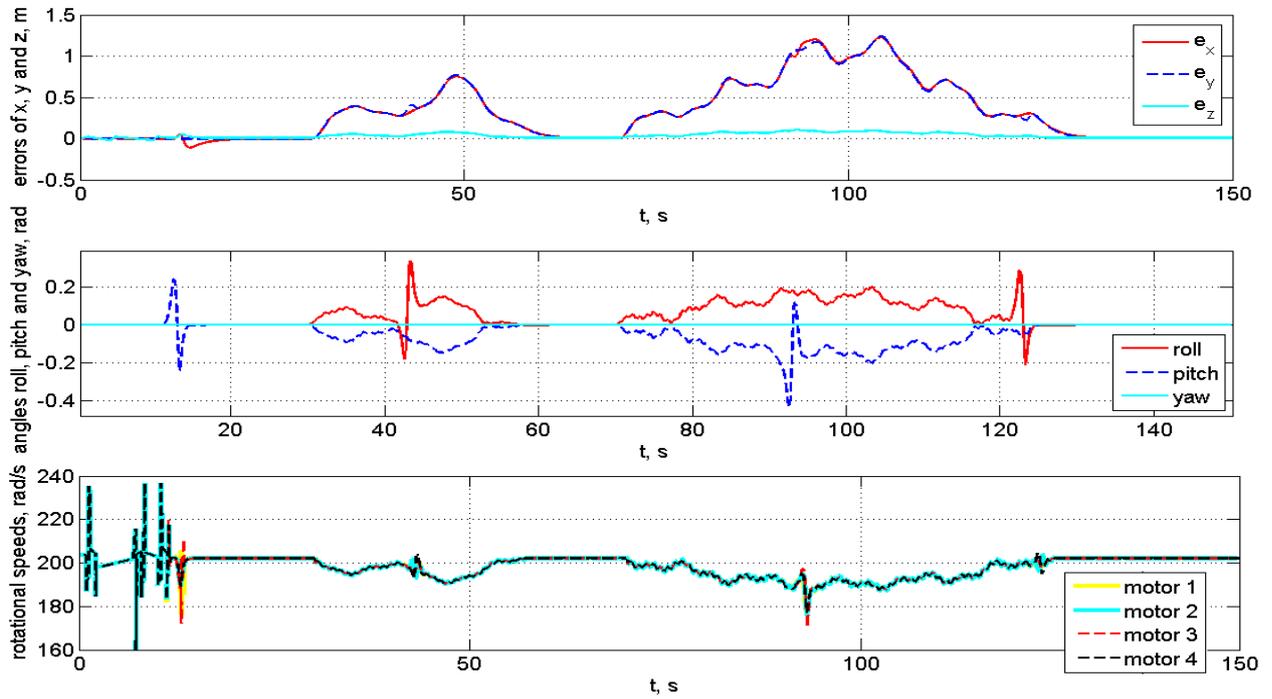


Figure 5.28: The sliding mode control with wind disturbance and the gains in Table 5.3.

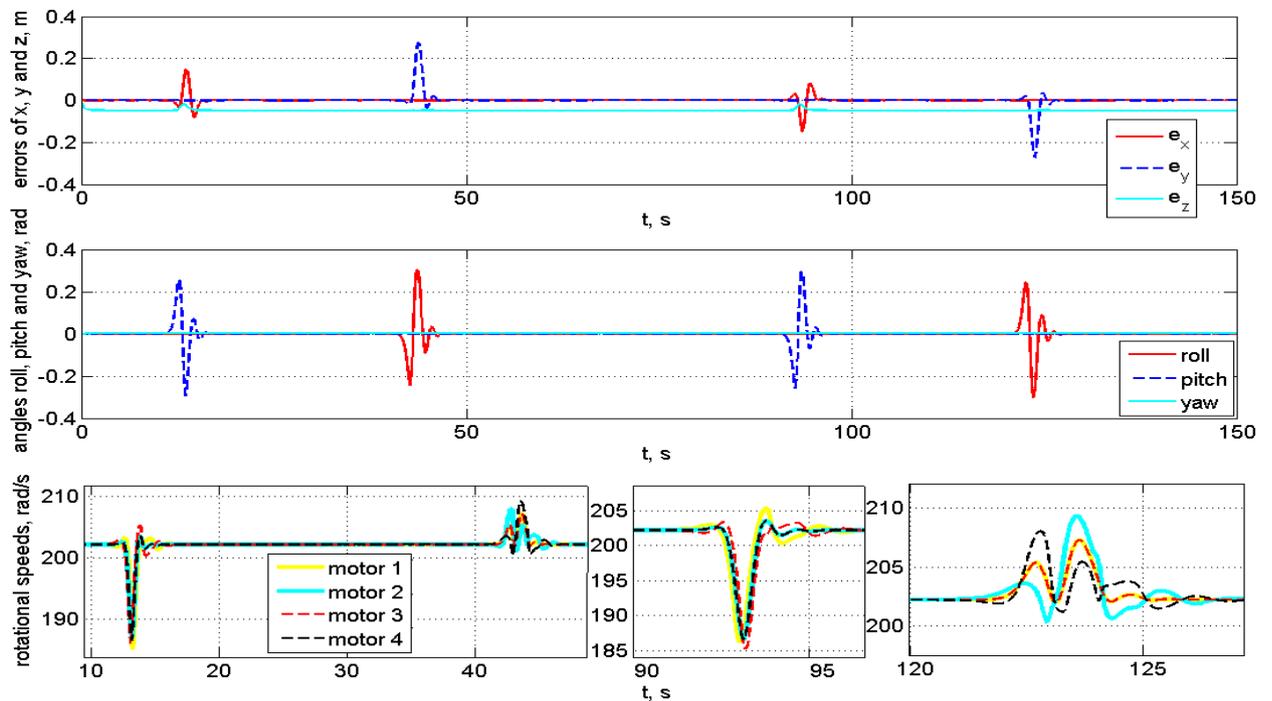


Figure 5.29: The sliding mode control with parameter uncertainties and the gains in Table 5.3.

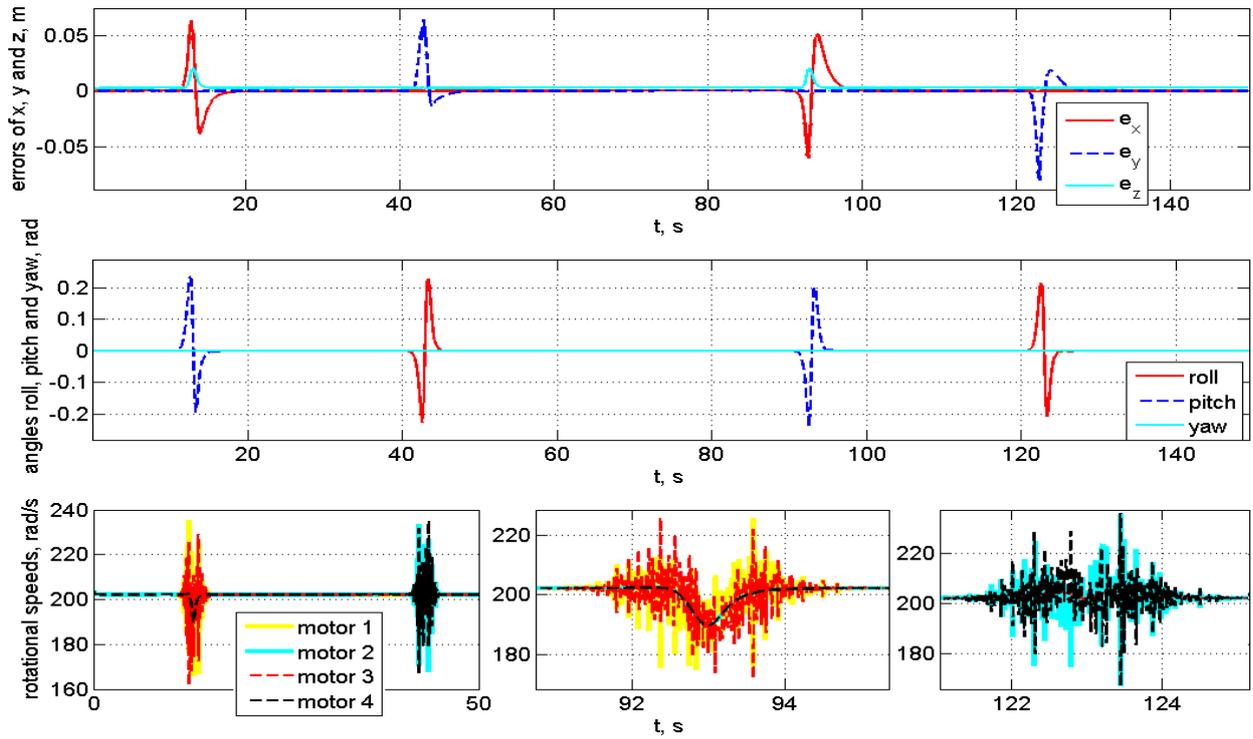


Figure 5.30: The sliding mode control with sensor noises and the gains in Table 5.3.

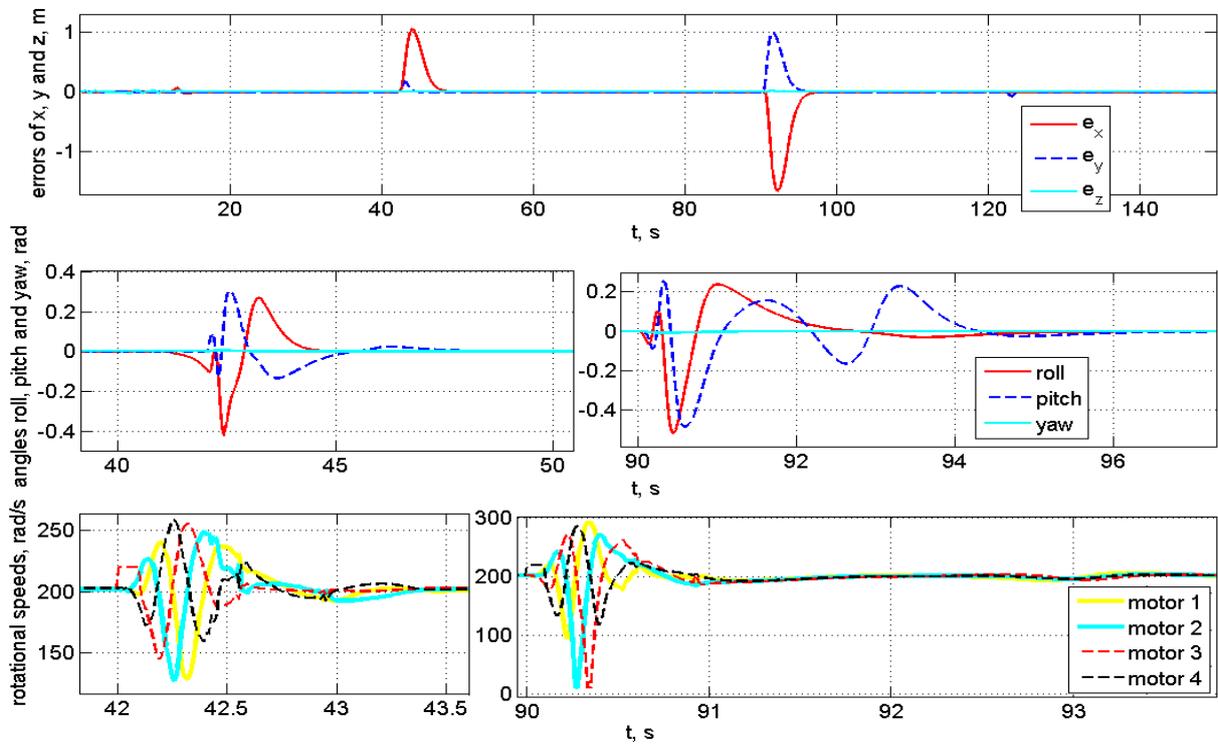


Figure 5.31: The sliding mode control with actuator faults and the gains in Table 5.3.

fault during the hovering caused 0.035m tracking error. During hovering, the actuator fault has been responded quickly, and in about 1s the system was back to normal stable state. However, the actuator fault during the moving caused a longer adjustment. While the responding time is 0.5s in the event triggered model free control and 1.5s in the event triggered backstepping control. The actuator fault caused the quadrotor tilt about 0.1rad during hovering, while 0.1rad in the event triggered model free control and 0.21rad in the event triggered backstepping control. The actuation steps is 8297, while 8493 in the basic scenario.

### 5.2.4 Discussion

As in the time triggered control scheme, in order to show the differences of these control methods explicitly, ten important criteria are listed as follows for the comparison:

- (1) The sum of the variances of the tracking errors in the x, y and z axis in basic scenario without disturbance:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (2) The sum of the variances of the tracking errors with the wind disturbance:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (3) The sum of the variances of the tracking errors with parameter uncertainties:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (4) The sum of the variances of the tracking errors with sensor noises:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (5) The sum of the variances of the tracking errors with actuator faults:  $\sum_{i=1}^3 \int_t e_i^2$ .
- (6) Adjusting time:  $t$ .
- (7) Maximum tilt angle:  $\phi$  or  $\theta$ .
- (8) The number of the actuation steps.
- (9) The computational complexity: the number of the addition and multiplication operations in the control law.
- (10) The energy consumption: the energy spent on motion control, the sum of the four motors' squared angular speeds:  $\sum_{i=1}^4 \int_t \omega_i^2$ .

In the event triggered scheme, the maximum absolute tracking error, the sum of the variances of the tracking errors, the energy consumption and the actuation steps of three methods in five scenarios are presented in Figure 5.32. From the figure, we can see the event triggered model free control has a better performance in general.

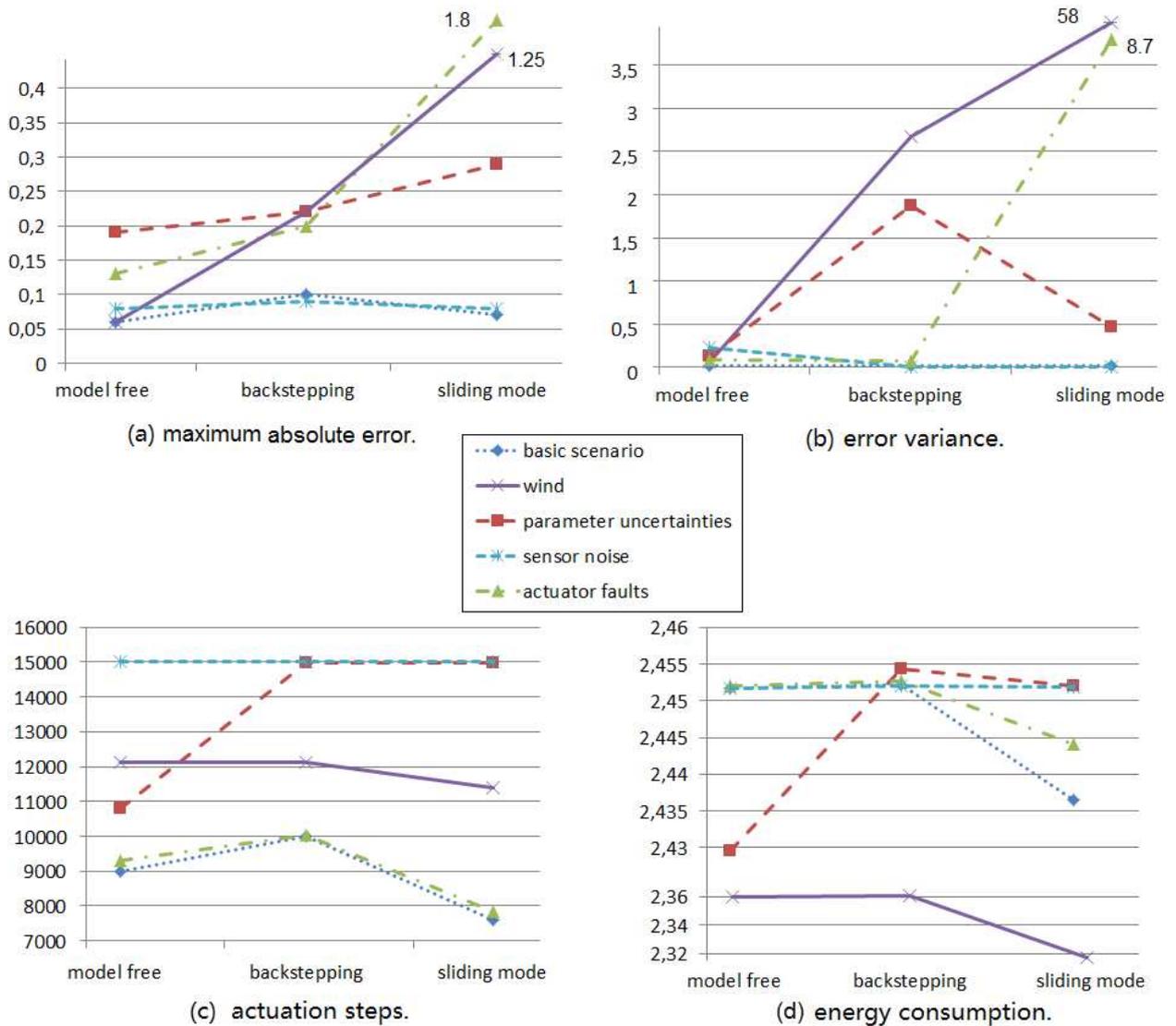


Figure 5.32: Event triggered three methods in all scenarios: (a) the maximum absolute tracking error. (b) the error variance. (c) the actuation steps. (d) the energy consumption  $\times 10^{-7}$ .

The comparison in the basic scenario is shown in Table 5.12. All the three methods are capable to keep their stability during the flight. There are not a big difference in the performance of three event triggered control methods. The event triggered backstepping control has the biggest actuation steps and maximum absolute error.

The comparison in the scenario with wind disturbance is shown in Table 5.13. The event triggered model free control can still well compensate the wind disturbance and has the smallest error variance, while the other two controls, especially sliding mode control, have big tracking errors. The maximum tracking error of the event triggered sliding mode

Table 5.12: The comparisons of the event triggered control methods in the basic scenario.  $n$  is  $n$ -digit number.

basic scenario	model free	backstepping	sliding mode
maximum absolute error	0.06	0.1	0.07
maximum tilt angle	0.21	0.21	0.21
error variance	0.02	0.02	0.0116
energy consumption	2.4517e+7	2.4522e+7	2.4365e+7
actuation steps	9008	9987	7603
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

Table 5.13: The comparisons of the event triggered control methods with wind disturbance.  $n$  is  $n$ -digit number.

wind	model free	backstepping	sliding mode
maximum absolute error	0.06	0.22	1.25
maximum tilt angle	0.4	0.4	0.405
error variance	0.0635	2.6679	58.5091
energy consumption	2.3598e+7	2.3604e+7	2.3559e+7
actuation steps	12110	12116	11400
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

control is 1.25m, which is 62.5% the desired path length 2m. The error variance of the event triggered sliding mode control is 58.591m, while 0.0635m and 2.6679m in other two methods respectively.

The comparison in the scenario with parameter uncertainties is shown in Table 5.14. The event triggered model free control has smallest tracking errors and actuation steps. The event triggered model free control saves the computational resources and at the same time has a better performance. The event triggered backstepping control has the biggest error variance in the system.

The comparison in the scenario with sensor noises is shown in Table 5.15. In this scenario, the control methods are triggered all the time, and the results are almost the same with the methods in the time triggered scheme. It shows the advantage of the event triggered scheme. If there are many disturbances, the control system will perform as in the time triggered scheme. If there are not many disturbances, the control system will select the control computation time according to the triggering laws.

The comparison in the scenario with actuator faults is shown in Table 5.16. Three event

Table 5.14: The comparisons of the event triggered control methods with parameter uncertainties.  $n$  is  $n$ -digit number.

parameter uncertainties	model free	backstepping	sliding mode
maximum absolute error	0.19	0.22	0.29
maximum tilt angle	0.34	0.49	0.3
error variance	0.1281	1.8675	0.4638
energy consumption	2.4295e+7	2.4543e+7	2.4521e+7
actuation steps	10803	14976	14983
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

Table 5.15: The comparisons of the event triggered control methods with sensor noises.  $n$  is  $n$ -digit number.

sensor noises	model free	backstepping	sliding mode
maximum absolute error	0.08	0.09	0.08
maximum tilt angle	0.22	0.23	0.21
error variance	0.2309	0.0029	0.0088
energy consumption	2.4517e+7	2.4520e+7	2.4518e+7
actuation steps	15000	15000	15000
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

Table 5.16: The comparisons of the event triggered control methods with actuator faults.  $n$  is  $n$ -digit number.

actuator faults	model free	backstepping	sliding mode
maximum error caused by fault	0.13	0.2	1.8
maximum tilt angle caused by fault	0.1	0.21	0.5
adjusting time after the fault	0.5	2.5	2.5
error variance	0.0918	0.0661	8.768
energy consumption	2.4520e+7	2.4527e+7	2.4441e+7
actuation steps	9298	10020	7857
computational complexity	$O(5n^2+3n)$	$O(6n^2+4n)$	$O(6n^2+4n)$

triggered control methods keep the stability after the actuator faults. The event triggered backstepping control and sliding mode control have oscillations in the acutator after the faults, and the system need about 2.5s to back to the stable state. The event triggered sliding mode control has the biggest maximum tracking error and error variance. The event triggered model free control has the smallest adjusting time and small tracking errors.

### 5.3 Discussion

In order to show the differences of these control methods explicitly, we choose six criteria for the comparison:

(1) The sum of the variances of tracking errors in x, y and z axis without wind disturbance:  

$$\sum_{i=1}^3 \int_t e_i^2.$$

(2) The sum of the variances of tracking errors with the wind disturbance:  $\sum_{i=1}^3 \int_t e_i^2.$

(3) The sum of the variances of tracking errors with rotation model error:  $\sum_{i=1}^3 \int_t e_i^2.$

(4) The number of the actuation steps in the basic scenario.

(5) The computational complexity: the number of the addition and multiplication operations in the control law.

(6) The energy consumption in the basic scenario: the energy spent on motion control, the sum of the four motors' squared angular speeds:  $\sum_{i=1}^4 \int_t \omega_i^2.$

The radar charts of the comparisons are shown in Figures 5.33 and 5.34.

The maximum absolute tracking errors of all the control methods in all the scenarios are presented in Table 5.17 and Figure 5.35. The time triggered model free has the smallest maximum absolute tracking error, and the event triggered model free has the smallest maximum absolute tracking error than other event triggered methods. The time and event triggered model free control have good performances especially in the scenario with wind disturbance. The tracking errors are much smaller than other methods. The wind disturbance has great influence on the time and event triggered backstepping and sliding mode control, while the time and event triggered model free control has almost the same performance as in the basic scenario. The sliding mode control does not match the event triggered scheme, and the tracking errors are much more bigger than other control methods.

The sum of the error variance of all the control methods in all the scenarios are presented in Table 5.18 and Figure 5.36. The time and event triggered model free control have the smallest error variance in general. The event triggered model free control has the error variance almost the same with the time triggered backstepping and sliding mode controls in five scenarios, or even better in the case with wind disturbance.

The actuation steps of all the control methods in all the scenarios are presented in Table 5.19 and Figure 5.37. The event triggered model free has reduced about 30% actuation steps in general while the tracking errors are not greatly increased. The actuation steps are much more reduced than other control methods in the scenario with parameter uncertainties, while

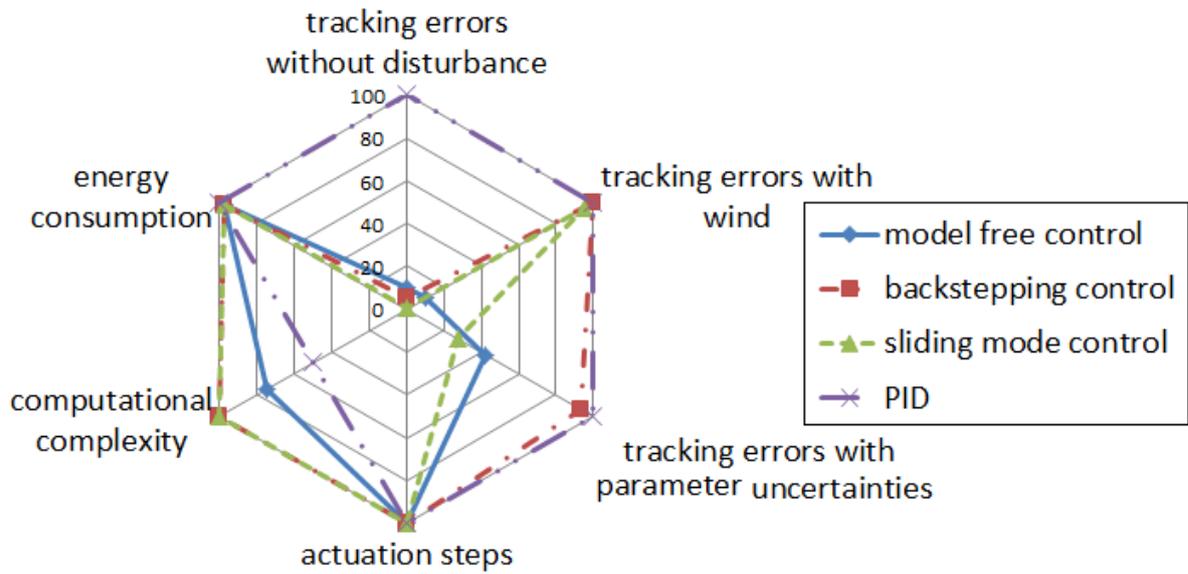


Figure 5.33: The comparison of the control methods in the time triggered scheme.

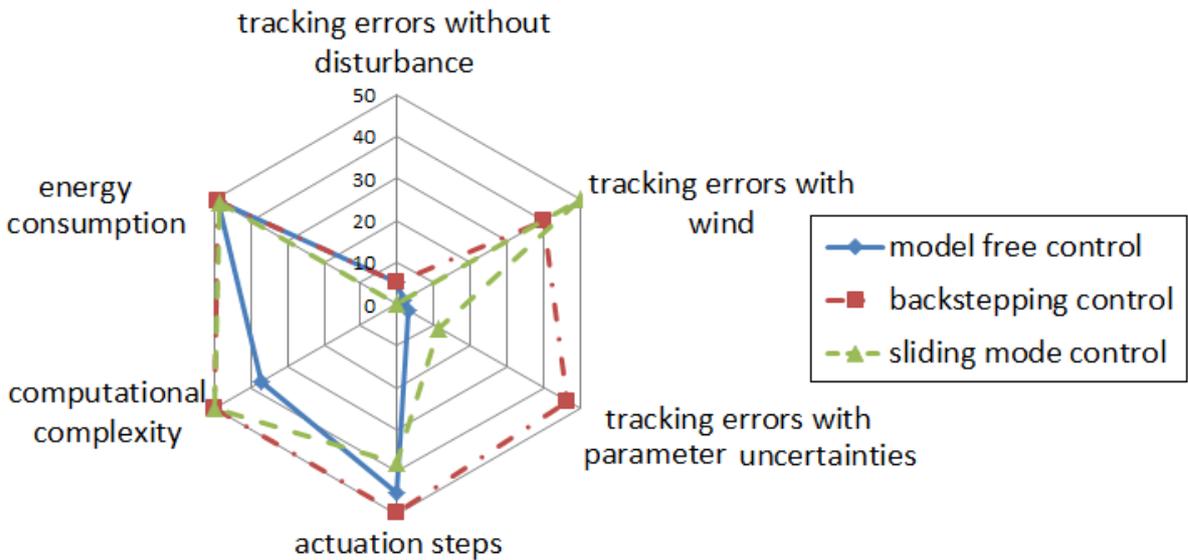


Figure 5.34: The comparison of the control methods in the event triggered scheme.

the error variance is almost the same with the time triggered model free control, backstepping control and sliding mode control.

The energy consumption of all the control methods in all the scenarios are presented in Table 5.20 and Figure 5.38. The energy consumptions are not changed too much in different methods. The event triggered sliding mode control has slightly less energy consumption, however the tracking errors are the biggest in the applications.

Table 5.17: The maximum absolute tracking errors of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control.

	time triggered			event triggered		
	MF	BS	SM	MF	BS	SM
basic scenario	0,042	0,09	0,07	0,06	0,1	0,07
wind	0,042	0,23	0,19	0,06	0,22	1,25
parameter uncertainties	0,075	0,11	0,055	0,19	0,22	0,29
sensor noise	0,08	0,09	0,08	0,08	0,09	0,08
actuator faults	0,04	0,06	0,21	0,13	0,2	1.8

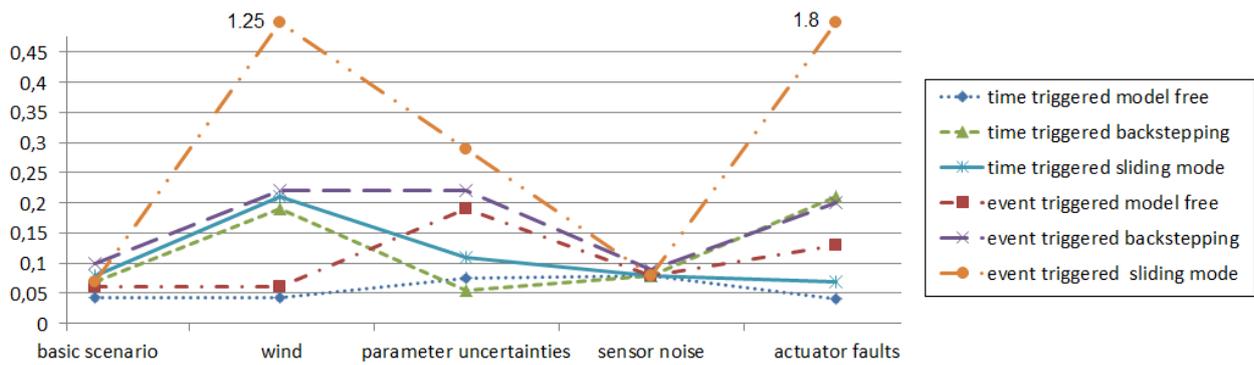


Figure 5.35: The maximum absolute tracking error of all the control methods in all scenarios.

Table 5.18: The sum of the error variances of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control.

	time triggered			event triggered		
	MF	BS	SM	MF	BS	SM
basic scenario	0,0049	0,0028	0,0003	0,2	0,2	0,0116
wind	0,0052	2,4723	0.8476	0,0635	2,6679	58.5091
parameter uncertainties	0,0212	0,0468	0,0138	0,1281	1,8675	0.4638
sensor noise	0,2309	0,0029	0,0088	0,2309	0,0029	0,0088
actuator faults	0,0327	0,0167	0,0585	0,0918	0,0661	8.768

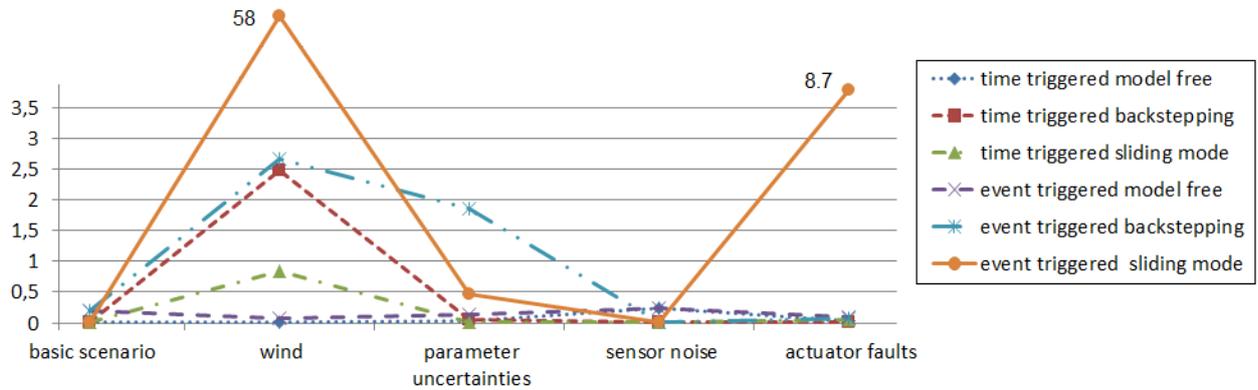


Figure 5.36: The sum of the error variances of all the control methods in all scenarios.

Table 5.19: The actuation steps of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control.

	time triggered			event triggered		
	MF	BS	SM	MF	BS	SM
basic scenario	15000			9008	9987	7603
wind	15000			12110	12116	11400
parameter uncertainties	15000			10803	14976	14983
sensor noise	15000			15000	15000	15000
actuator faults	15000			9298	10020	7857

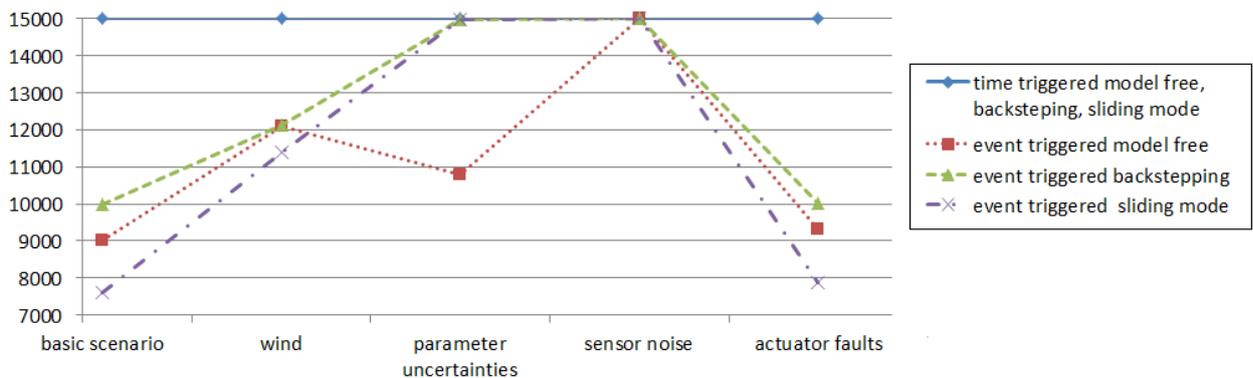


Figure 5.37: The actuation steps of all the control methods in all scenarios.

Table 5.20: The energy consumption of all the control methods in all scenarios. MF: model free control. BS: backstepping control. SM: sliding mode control.

$\times 10^7$	time triggered			event triggered		
	MF	BS	SM	MF	BS	SM
basic scenario	2,4518	2,4519	2,4518	2,4517	2,4522	2,4365
wind	2,3598	2,3600	2,3599	2,3598	2,3604	2,3559
parameter uncertainties	2,4523	2,4528	2,4517	2,4295	2,4543	2,4521
sensor noise	2,4517	2,4520	2,4518	2,4517	2,4520	2,4518
actuator faults	2,4518	2,4524	2,4522	2,4520	2,4527	2,4441

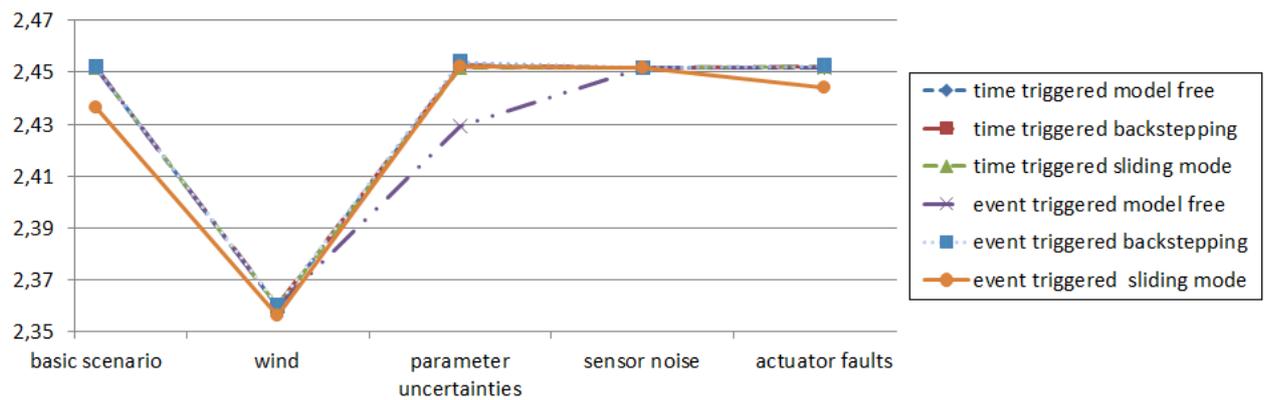


Figure 5.38: The energy consumption  $\times 10^{-7}$  of all the control methods in all scenarios.

# Chapter 6

## Conclusions

### 6.1 Conclusion

In the last decades, quadrotors become popular in scientific researches. Due to its simplicity in design and maneuver, quadrotors have many applications. However, there are many challenges in the control of quadrotors. This dissertation is therefore dedicated to these main challenges.

At the beginning, two inspiring examples are made to show the limits of a traditional PID control and the difference of the performance in ideal scenarios and realistic scenarios. Then, the challenges in quadrotor researches are thoroughly studied. Three main problems are listed. Firstly, a dynamic model which is simple for computation and close to quadrotor dynamics is needed. Secondly, a control method which demands small computational resources and at the same time is rather stable for disturbances and model uncertainties is required. At last, the results in realistic scenarios are needed to show the advantages and disadvantages of the control methods. The comparisons of several methods in the same scenarios are also demanded to verify which method is better in certain conditions. Therefore, these challenges are studied in this dissertation.

The existing quadrotors are intensively studied and presented. The configurations of quadrotors have limitations: Lithium Polymer batteries have limited capacities; embedded systems have limited computational resources, etc. Then, the applications of quadrotors are thoroughly studied, and necessary qualities of a control method are discussed. The existing quadrotor control methods are presented according to their research groups. Several control methods have been proposed, however we can not tell which one is better in which condition. The comparisons of control methods are rarely given in the literature. These studies of the

literature support the chosen main challenges.

In the modeling, the kinematic relations and dynamic models are proposed. The world frame, body frame and the Euler angles rotational sequence are selected based on the most generally used references. The relations of the positions, velocities and accelerations are clearly presented. Then, a dynamic model using the Newton Euler formalism is proposed. The aerodynamic forces and moments based on the Fay's work are studied and presented. The modeling is based on the Bouabdallah's modeling. The Bouabdallah's model is a rather realistic model in the literature. However, there are some errors in his modeling. For example, the hub force is calculated using the linear velocities along the  $x_w$  and  $y_w$  in the world frame. These are not exact. The hub force should be calculated using the linear velocities along the  $x_b$  and  $y_b$  in the body frame, as aerodynamic forces are caused by the air flow speed related to a quadrotor which changes when the quadrotor rotates. Moreover, some directions of the forces and moment are not correct, such as the direction of the rolling moment, etc. The frames defined in Bouabdallah's thesis are not the same as in his program, which is a little bit confusing. Therefore in this thesis, the chosen world and body frames, the relationship of linear and angular velocities, the aerodynamic forces and moments are clearly presented. The errors in Bouabdallah's thesis are corrected. Some unnecessary aerodynamic forces and moments are neglected.

Then, the dynamics of a quadrotor is analyzed. The dynamics has never been studied intensively in the literature. Firstly based on the normal form theory, the model of a quadrotor is simplified to a simplest form named the normal form which exhibits all possible properties of the original system. In the original model, the control inputs are coupled with the states, which makes the analysis of the system difficult. The relations between the system states can not be clearly seen. In the normal form, the system states are no longer coupled with the system inputs, and we can see the pitch angle  $\theta$  has more influence on the  $x$  axis and the roll angle  $\phi$  has more influence on the  $y$  axis. Then, the bifurcations of its normal form are studied. The system without input has twelve zero eigenvalues, and this means the stability of the system depends on the inputs. Given some general state feedbacks with some parameters, the changes of the eigenvalues which are caused by the changes of the parameters are proposed. Using the center manifold theory, the system can be further simplified at its bifurcation point. The twelve dimensional system is then simplified into a two dimensional system which exhibits all possible properties of the original system.

In Chapter 4, without losing generality, an aerial photography scenario in an open space is selected. Based on the necessary qualities in applications, five realistic scenarios are

presented: an ideal case, the scenario with wind disturbance, with parameter uncertainties, with sensor noises and with actuator faults. These realistic cases can show the stability and sensibility of a control method in difference aspects, and a control method can have varying performance in each aspect. These results make good references for choosing a control method for certain application. In order to further save the computational resources, the event triggered scheme is presented and used on all the control methods.

Then, a newly proposed method the model free control is presented. It is a simple but efficient technique for nonlinear, unknown or partially known dynamics. Instead of computing the dynamics of a system, the dynamics is evaluated as a variable  $F$  in real time by the system inputs and outputs. Therefore, the model free control can compensate well the modeling errors and real-time disturbances. The algorithm of this control method is rather simple as it does not fully compute the dynamic model. At the same time, the stability of this method is high as it compensates well the disturbances. In order to show its advantages and disadvantages, several other control methods are also proposed, such as a backstepping control and a sliding mode control. The time and event triggered schemes are implemented in all the control methods.

In Chapter 5, the simulation results of the time and event triggered all control methods in five scenarios are given. In order to close to the realistic cases, the control gains of each control method are the same in five different scenarios. Usually control gains are chosen beforehand and will not be changed during the applications when disturbances occur. The results are analyzed and compared thoroughly in different aspects. Ten important criteria discussed in Chapter 1 are listed, such as the maximum absolute tracking error, the error variance, the actuation steps, the energy consumption, etc. In each scenario, the values of these criteria are calculated and presented in tables and figures. The advantages and disadvantages of each control method can be seen clearly. These results make a good reference for choosing an appropriate control method for an application.

## 6.2 Future Work

Some of the challenges are discussed in this dissertation. However, there are still some work to finish in the future.

### (1) A simulation workbench.

Based on the programs of Bouabdallah, some simulation programs are written in this

thesis. It can be a good simulation workbench for future quadrotor studies. However in the modeling, many aerodynamic effects are not fully presented or neglected, such as the ground effect. The aerodynamic effects are definitely needed more intensive studies. Getting detailed aerodynamic mathematic expressions may be not useful for control system design, as the embedded system has limit computational resources. However, it is very useful to build a comprehensive simulation workbench. A standard simulation workbench and scenarios are useful for the comparison of control methods proposed by different authors.

**(2) Further research on the stability of the model free control.**

The model free control is proven practically stable in this thesis. To further prove the stability of the method, more information about a specific system is needed. As the part  $F$  is updated at each sampling time, the choice of the sampling period is important for the stability. In this dissertation, a sampling period of 10ms is selected based on the quadrotor dynamics and real control device. To select a sampling period for a general system, more researches are needed on the stability of the model free control.

**(3) Implementation on a real quadrotor.**

Many realistic cases are proposed in this thesis, and many control methods are simulated in MATLAB. From the simulation results, the advantages and disadvantages of each control method can be seen clearly. However due to the limit of the condition, these control methods are not implemented on a real quadrotor. Therefore, one of the future work is the implementation of the control methods on a real quadrotor.

# Appendix A

## MAPLE Code

```

> #####
> ##### normal form #####
> #####
> ## taylor expression
> restart:
> deg:=7: ## degree = deg-1
> sin(x):=series(sin(x),x,deg):
> cos(x):=series(cos(x),x,deg):
> ## substitute in the original system. eq 3 in the paper MNTS.
> arrayx:=seq(x||k=x[k],k=1..12):
> x1dot:=expand(subs(arrayx,x2)):
> x2dot:=mtaylor(subs(arrayx,expand(-(u1+g)*subs(x=x9,sin(x)))),[seq(x[k],k=1..12),u1],deg):
> x3dot:=expand(subs(arrayx,x4)):
> x4dot:=mtaylor(subs(arrayx,expand((u1+g)*subs(x=x9,cos(x))*subs(x=x7,sin(x)))),[seq(x[k],
> k=1..12),u1],deg):
> x5dot:=expand(subs(arrayx,x6)):
> x6dot:=mtaylor(subs(arrayx,expand((u1+g)*subs(x=x9,cos(x))*subs(x=x7,cos(x)))-g),[seq(x[k],
> k=1..12),u1],deg):
> x7dot:=expand(subs(arrayx,x8)):
> x8dot:=expand(subs(arrayx,u2)):
> x9dot:=expand(subs(arrayx,x10)):
> x10dot:=expand(subs(arrayx,u3)):
> x11dot:=expand(subs(arrayx,x12)):
> x12dot:=expand(subs(arrayx,u4)):

> ## change the equilibrium to the origin.
> x[1]:=y[1]; x[2]:=y[2]; x[3]:=y[3]; x[4]:=y[4]; x[5]:=y[5];
> x[6]:=y[6]; x[11]:=y[11]; x[12]:=y[12];
> x[7]:=y[7]/g; x[8]:=y[8]/g; x[9]:=y[9]/g; x[10]:=y[10]/g;
> x7dot:=x7dot*g;x8dot:=x8dot*g;x9dot:=x9dot*g;x10dot:=x10dot*g;
> u1:=v[1]; u2:=v[2]/g; u3:=v[3]/g; u4:=v[4];
> ## substitute into the system. eq 4 in the paper MNTS.
> dy:=Matrix(12,1):
> l:=dot:
> for i from 1 to 12 do
> dy[i,1]:=mtaylor(x||i||1,[seq(y[k],k=1..12),v[1],v[2],v[3],v[4]],deg):
> end:
> evalm(dy);

```

## Appendix A. MAPLE Code

---

```

> ## change * to . for matrix.
> product_to_Matrix_product:=proc(invh)
> local t,k,p,part;
> part:=0:
> for t from 1 to nops(invh) do
> if op(invh)[t]=1/iden then
> part:=part+1/iden:
> else
> p:=1;
> for k from 1 to nops(op(invh)[t]) do
> p:=p.op(op(invh)[t])[k];
> end do;
> part:=part+p;
> end if;
> end do;
> part
> end proc:

> ## expression of  $(I + dQ2 + \dots)^{-1}$ . eq 6 in the paper MNTS.
> with(LinearAlgebra):
> iden:=evalm(IdentityMatrix(12)):
> invhsymbol:=mtaylor(1/(iden+dQ2+dQ3+dQ4+dQ5+dQ6), [dQ2, dQ3, dQ4, dQ5, dQ6], deg):
> invh:=product_to_Matrix_product(invhsymbol):
> dQ3:=evalm(Matrix(12,12)):
> dQ4:=evalm(Matrix(12,12)):
> dQ5:=evalm(Matrix(12,12)):
> dQ6:=evalm(Matrix(12,12)):dQ7:=evalm(Matrix(12,12)):

> ##### 2th degree #####

> ## y dot by states z.
> hz1:=seq(y[i]=z[i], i=1..12):
> dybyz1:=evalm(subs(hz1, dy));
> ## calculate  $(I + dQ2 + \dots)^{-1}$ .
> dQ2:=evalm(Matrix(12, (i, j) -> a[i, j])):
> invh1:=evalm(invh):
> ## eq 6 in the paper MNTS.
> dz1:=evalm(invh1&*dybyz1):
> ## find the terms  $f(z, a)*u1$ .  $u1$  has degree 1. therefore, the 1st degree terms in  $f(z, a)$  should be 0 to
ensure the 2nd degree system has no coupled terms. the 1st degree in  $f(z, a)$  are either  $z$  or  $a$ .
> for m from 1 to 12 do
> eq[m]:=
> mtaylor(mtaylor(coeff(dz1[m, 1], u1), [seq(seq(a[i, j], i=1..12), j=1..12)], 1), [seq(z[i], i=1..12)], 2)
> -mtaylor(mtaylor(coeff(dz1[m, 1], u1), [seq(seq(a[i, j], i=1..12), j=1..12)], 1), [seq(z[i], i=1..12)], 1)
> +mtaylor(mtaylor(coeff(dz1[m, 1], u1), [seq(z[i], i=1..12)], 1), [seq(seq(a[i, j], i=1..12), j=1..12)], 2)
> -mtaylor(mtaylor(coeff(dz1[m, 1], u1), [seq(z[i], i=1..12)], 1), [seq(seq(a[i, j], i=1..12), j=1..12)], 1)=0;
> end;

0 = 0

> Ideal:=[seq(eq[i], i=1..12)]:
> S1:=solve(Ideal, select(has, indets(Ideal), a)):

> ##### 3rd degree #####

> #evalm(dy):
> Q2:=[seq(0, i=1..12)]:

```

```

> ## calculate Q2 from dQ2 (S1)
> with(linalg):
> for k from 1 to nops(S1) do
> if lhs(S1[k])<> rhs(S1[k]) then
> Q2[op(lhs(S1[k]))[1]]:=rhs(S1[k])*z[op(lhs(S1[k]))[2]]; end if;
> end do;
> Q2:
> dQ2:=jacobian(Q2,[z[1],z[2],z[3],z[4],z[5],z[6],z[7],z[8],z[9],z[10],z[11],z[12]]):

> ## y dot by states z.
> #hz2:=seq(y[i]=z[i]+Q2[i],i=1..12):
> #dybyz2:=evalm(subs(hz2,dy));
> ## calculate (I + dQ2 +..)^(-1).
> dQ3:=evalm(Matrix(12,(i,j)->a[i,j])):
> #dQ4:=evalm(Matrix(12,12)):dQ5:=evalm(Matrix(12,12)):
> invh2:=evalm(invh):

> ## eq 6 in the paper MNTS.
> dz2:=(evalm(invh2*&dybyz2)):

> ## find the terms f(z,a)*u1. u1 has degree 1. therefore, the 2nd degree terms in f(z,a) should be 0 to
ensure the 3rd degree system has no coupled terms. the 2nd degree in f(z,a) are either g(z)
> (quadratique terms of z) or a.
> for m from 1 to 12 do
> eq[m]:=mtaylor(mtaylor(coeff(dz2[m,1],u1),[seq(seq(a[i,j],i=1..12),j=1..12)],1),[seq(z[i],i=1..
> 12)],3)-mtaylor(mtaylor(coeff(dz2[m,1],u1),[seq(seq(a[i,j],i=1..12),j=1..12)],1),[seq(z[i],i=1
> ..12)],2)+mtaylor(mtaylor(coeff(dz2[m,1],u1),[seq(z[i],i=1..12)],1),[seq(seq(a[i,j],i=1..12),j
> =1..12)],2)-mtaylor(mtaylor(coeff(dz2[m,1],u1),[seq(z[i],i=1..12)],1),[seq(seq(a[i,j],i=1..12),
> j=1..12)],1)=0;
> end do;
> Ideal:=[seq(eq[i],i=1..12)]:
> S2:=solve(Ideal,select(has,indets(Ideal),a));

```



# Appendix B

## Papers

The papers published during three years are:

Journal paper:

- J. Wang, M. S. Geamanu, A. Cela, H. Mounier, S. I. Niculescu. *Event driven model free control and comparison with standard approaches in motion control*. IEEE transactions on Control Systems Technology. (1st submission 07/03/2013, 2nd submission 19/09/2013).

Conference papers:

- J. Wang, H. Mounier, A. Cela, S. I. Niculescu. *Event driven intelligent PID controllers with applications to motion control*. In Proceedings of the 18th IFAC World Congress, Milan, Italy, 2011.
- J. Wang, I. Boussaada, A. Cela, H. Mounier, S. I. Niculescu. *Analysis and control of quadrotor via a Normal Form approach*. In Proc. of 20th International Symposium on Mathematical Theory of Networks and Systems, Melbourne, Australia, 2012.
- J. Wang, M. S. Geamanu, A. Cela, H. Mounier, S. I. Niculescu. *Event driven model free control of quadrotor*. IEEE Conference on Control Applications (CCA), Hyderabad, India, 2013.

# Event driven intelligent PID controllers with applications to motion control

Jing Wang<sup>\*,\*\*\*</sup> Hugues Mounier<sup>\*</sup> Arben Cela<sup>\*\*</sup>  
Silviu-Iulian Niculescu<sup>\*</sup>

<sup>\*</sup> *Laboratoire des Signaux et Systèmes (L2S), CNRS, Supélec,  
Université Paris Sud 11, Supélec, 3, rue Joliot Curie, 91192 Gif sur  
Yvette Cedex (e-mails: jing.wang@lss.supelec.fr,*

*hugues.mounier@lss.supelec.fr, silviu.niculescu@lss.supelec.fr)*

<sup>\*\*</sup> *UPE, ESIEE Paris, Embedded System Department, 2 Bd Blaise  
Pascal, 93162 Noisy Le Grand Cedex (e-mail: celaa@esiee.fr)*

<sup>\*\*\*</sup> *Institut Polytechnique des Sciences avancées, 7-9 rue Maurice  
Grandcoing 94200 Ivry-sur-Seine (e-mail: wang@ipsa.fr)*

---

**Abstract:** A novel type of reduced complexity controller is proposed. It is the combination of model free control and event triggered control. The robustness of model free control, especially for badly known dynamics, is added to the event based scheme. The performances of the proposed method are illustrated in two motion controls, vehicular longitudinal control and quadrotor control. Comparisons with existing control schemes are also proposed.

Keywords: Intelligent PID, event driven control, model free control, reduced complexity controllers

---

## 1. INTRODUCTION

The trend to complex embedded control systems brings out a lot of new challenges. On one hand, the embedded character demands reduced complexity controllers. On the other hand, the complexity of the controlled systems enforces robustness of the proposed control schemes. Many constraints have to be taken into account, especially in distributed systems (see Murray et al. [2003]). Low computational cost control schemes which are able to deal with nonlinear systems with robustness are needed.

Model free control has been proven to be a simple but very efficient nonlinear feedback technique for the unknown or partially known dynamics (see Fliess et al. [2009], Choi et al. [2009]). We shall here use so-called *intelligent* PID (or *i*-PID). While retaining the PID reduced computational cost, it is able to cope with general types of nonlinearities. A precise relationship between *i*-PID and PIDs is given in d'Andréa-Novel et al. [2010]. It particularly emphasizes the ease of tuning of *i*-PID gains and gives a clearcut explanation of the performance of usual PIDs.

Contrarily to the time triggered control scheme which the control signals are sent to the actuator board every fixed sampling time, in the event based scheme, the control signals are sent only upon the triggering of an event (see Årzén [1999]). A typical event is that the tracking error goes beyond a specified limit. This type of scheme allows to go beyond the traditional Shannon sampling limit while still achieving asymptotic stability. We here propose an event based scheme for intelligent PID. The two techniques quoted above enable the efficiency and reduced complexity of the controller.

In the first section, the general setting of model free control and intelligent PID (*i*-PID) controllers are recalled. Then, event driven *i*-PID controllers are introduced. The simulations on the simplified models of longitudinal dynamics of a car and aerodynamics of quadrotor are then given.

## 2. MODEL FREE CONTROL

### 2.1 General setting

Model free control is a quite recent and very efficient technique for unknown and partially known systems (see Fliess et al. [2009]). The input-output behavior of the system is approximatively governed within its operating range by a partially known or totally unknown finite-dimensional ordinary linear or non-linear differential equation. For the sake of simplicity, the input and output are assumed to be mono-variable. The system is described implicitly as

$$E(y, \dot{y}, \dots, y^{(a)}, u, \dot{u}, \dots, u^{(b)}) = 0 \quad (1)$$

where  $E : \mathbf{R}^{a+1} \times \mathbf{R}^{b+1} \rightarrow \mathbf{R}$  is a sufficient smooth function of its arguments. Assume that for integer  $\nu$ ,  $0 < \nu \leq a$ ,  $\partial E / \partial y^{(\nu)} \neq 0$ . The implicit function theorem (see Krantz et al. [2002]) allows to express  $y^{(\nu)}$  locally

$$y^{(\nu)} = \mathfrak{E}(t, y, \dot{y}, \dots, y^{(\nu-1)}, y^{(\nu+1)}, \dots, y^{(a)}, u, \dot{u}, \dots, u^{(\kappa)})$$

with the function  $\mathfrak{E} : \mathbf{R} \times \mathbf{R}^{\nu} \times \mathbf{R}^{\kappa+1} \rightarrow \mathbf{R}$ .

Replace (1) by the following phenomenological model which is only valid in a very short time interval.

$$y^{(\nu)} = F + \alpha u \quad (2)$$

where

- $\alpha \in \mathbb{R}$  is a *non-physical* constant parameter, which is chosen by the engineer in such a way that  $F$  and  $\alpha u$  are of the same magnitude.

- The derivation order  $\nu$  is also an engineer's choice.
- $F$  is determined thanks to the knowledge of  $u$ ,  $\alpha$ , and of the estimate of  $y^{(\nu)}$ .

An estimate of  $F$  is obtained as follows:

$$\hat{F} = \hat{y}^{(\nu)} - \alpha \tilde{u} \quad (3)$$

where  $\hat{y}^{(\nu)}$  is an estimate of the  $\nu^{\text{th}}$  derivative of the measure  $y$  which is assumed available, and  $\tilde{u}$  is an approximate value of  $u$ , in order to avoid algebraic loops in the controllers. Among the existing possibilities,  $\tilde{u}$  can be chosen as a past value of the control variable  $u$ . The resulting controller is then

$$u = \frac{1}{\alpha} \left( y_r^{(\nu)} - \hat{F} + \Lambda(e^{(-\xi, \zeta)}) \right)$$

where

- $y_r$  is a reference trajectory which is selected as in flatness-based control (see Fliess et al. [1995]).
- $e = y_r - y$  is the tracking error.
- $e^{(-\xi, \zeta)} = (\int^\xi e, \int^{\xi-1} e, \dots, e, \dot{e}, e^{(\zeta)})$ ,  $\xi, \zeta \in [0, \nu]$ ,  $\int^k$  is the  $k$  iterated integral, and  $\Lambda$  is an appropriate function  $\mathbf{R}^{\xi+\zeta+1} \rightarrow \mathbf{R}$  such that the closed loop error dynamics

$$e^{(\nu)} = \Lambda(e^{(-\xi, \zeta)})$$

is asymptotically stable.

*Remarks 2.1.* a) The derivation order  $\nu$  is not necessarily equal to the derivation order  $a$  of  $y$  in Equation (1).

- The derivation order  $\nu$ , is often taken equal to 1 or 2, yielding so called intelligent PIDs or *i*-PID (see next subsection).
- A system may be partially unknown. It is straightforward to adapt the previous method.
- The estimate in (3) can be obtained for example through a simple first order filtering as

$$\mathcal{L}(\hat{y}) = \frac{s}{1 + T_f s} \mathcal{L}(y)$$

typically,  $1/T_f$  ranges from 8 to 20, and  $\mathcal{L}$  denotes the transformation to the operational domain.

It can also be given by efficient algebraic techniques (see Mboup et al. [2009]) yielding for example the following estimate for the first derivative

$$\hat{y} = \frac{-3!}{T^3} \int_0^T (T - 2\tau)y(\tau)d\tau$$

with  $T$  an integration window size which order of magnitude is 20 times the sampling time in a time triggered setting.

## 2.2 Intelligent PIDs

The desired behavior is obtained by implementing, for instance  $\nu = 2$ , the *intelligent PID controller (i-PID)* is

$$u = -\frac{\hat{F}}{\alpha} + \frac{\ddot{y}_r}{\alpha} + K_P e + K_I \int e + K_D \dot{e} \quad (4)$$

where  $K_P$ ,  $K_I$ ,  $K_D$  are the usual tuning gains.

Let us consider the following special cases:

- If  $\nu = 2$ , we may also employ an *intelligent PD controller (i-PD)*

$$u = -\frac{\hat{F}}{\alpha} + \frac{\ddot{y}_r}{\alpha} + K_P e + K_D \dot{e} \quad (5)$$

- If  $\nu = 1$ , we restrict ourselves to an *intelligent PI controller (i-PI)*

$$u = -\frac{\hat{F}}{\alpha} + \frac{\dot{y}_r}{\alpha} + K_P e + K_I \int e \quad (6)$$

or even to an *intelligent P controller (i-P)*

$$u = -\frac{\hat{F}}{\alpha} + \frac{\dot{y}_r}{\alpha} + K_P e \quad (7)$$

*Remarks 2.2.* a) If  $\nu = 2$  (resp. 1), plugging Equations (4) or (5) (resp. (6) or (7)) in Equation (2) yields the control of a pure double (resp. simple) integrator. This is why tuning the gains of our intelligent controllers is quite straightforward.

- It should be emphasized, if  $\nu = 2$  (resp. 1), that Equation (5) (resp. (7)) is mathematically sufficient for ensuring stability around the reference trajectory. The integral term  $K_I \int e$  in Equation (4) (resp. (6)) is however adding some well known robustness properties.

## 3. EVENT DRIVEN MODEL FREE CONTROL

The basic Årzén's event based controller consists of two parts: a time triggered event detector  $\tau_{ed}$  and an event triggered PID controller  $\tau_{ec}$ . See Årzén [1999]. The latter computes the control signal to be delivered to the actuator board. The former  $\tau_{ed}$  runs at a fixed sampling period  $h_{ed}$ , and upon fulfillment of a certain event triggering law  $L_{et}$ , sends events to  $\tau_{ec}$ . Upon reception of the event,  $\tau_{ec}$  computes the control signal and sends it to the actuator board.

Examples of event triggering laws  $L_{et}$  are:

- Error threshold law:

$$|e(t_k)| > e_{lim} \quad (8)$$

where  $e = y_r - y$  is the tracking error,  $t_k$  is the current discrete sensing time by  $\tau_{ed}$ , and  $e_{lim}$  is a fixed limit.

- Error difference threshold

$$|e(t_k) - e(t_{k-1})| > e_{lim} \quad (9)$$

- ISS based law:

$$e(t_k) = \sigma \frac{a}{b} |y(t_k)| \quad (10)$$

assuming the system can be rendered ISS (Input to State Stable) through static feedback (see Sontag [2007]).  $\sigma$  is chosen less than one to ensure an associated Lyapounov function decrease.  $a$  and  $b$  are chosen according to the Lipschitz constants of the  $\mathcal{K}_\infty$  (consisting of all functions  $\gamma: \mathbf{R}^+ \rightarrow \mathbf{R}^+$  which are continuous, strictly increasing, satisfying  $\gamma(0) = 0$  and  $\lim_{\xi \rightarrow \infty} \gamma(\xi) = \infty$ ). See, e.g., Sontag [2007]).

The present control goal is path tracking. We shall use geometric information on the reference trajectory  $y_r$ . Namely, we shall take the following event triggering scheme:

$$|e(t_k) - e(t_{k-1})| > e_{lim} \quad \wedge \quad t_k - t_{k-1} > \frac{\max(\sigma(\dot{y}_r)) \cdot h_M}{\sigma(\dot{y}_r(t_k))} \quad (11)$$

where  $\sigma$  is a saturation function, and  $h_M$  is the maximum sampling time ensuring stability in a time triggered scheme. We have chosen the following smooth saturating function

$$\begin{aligned}
\sigma(\xi) &= \frac{H-l}{2(\xi_H - \xi_l)} (\phi(\xi) + \psi(\xi)) + \frac{\xi_H + \xi_l}{2} \\
\phi(\xi) &= \frac{1}{\zeta} \ln(\cosh(\zeta(\xi - \xi_l))) \\
\psi(\xi) &= \frac{-1}{\zeta} \ln(\cosh(-\zeta(\xi - \xi_H)))
\end{aligned} \tag{12}$$

with  $l$  and  $H$  the low and high saturated values,  $\xi_l$  and  $\xi_H$  the beginning and ending abscissa of the linear part, and  $\zeta$  is a stiffness value. The  $\ln(\cosh(\xi))$  functions enable to have a linear part (when  $\xi \ll 0$ ,  $\cosh(\xi) \approx \exp(-\xi)/2$ , and  $\ln(\cosh(\xi)) \approx -\xi/2$ ; when  $\xi \gg 0$ ,  $\cosh(\xi) \approx \exp(\xi)/2$ , and  $\ln(\cosh(\xi)) \approx \xi/2$ ) with smooth transitions between the constant and linear parts.

#### 4. APPLICATION TO VEHICLE LONGITUDINAL CONTROL

##### 4.1 Model

We shall take a simplified model of longitudinal car dynamics as example. See Kiencke et al. [2005]. No attempt will be made to take longitudinal slip into account. Thus, the motor torque is supposed to be directly transmitted to the longitudinal dynamics.

The simplified model is as the following:

$$M\dot{V}_x = \frac{C}{r} - C_{ae}(V_x + V_w)|V_x + V_w| - Mg \sin(\theta) - MgC_{rr}\text{sign}(V_x) \cos(\theta) \tag{13}$$

where  $M$  the vehicle's mass,  $V_x$  the vehicle's longitudinal speed.  $C$  the traction torque which is taken as control input.  $r$  the wheel's mean radius.  $C_{ae}$  the aerodynamics coefficient.  $V_w$  the wind speed disturbance.  $g$  the gravity constant.  $\theta$  the road slope.  $C_{rr}$  the Rolling resistance coefficient.

The chosen values for the parameters are:  $M = 1200\text{kg}$ ,  $V_x = 0$  to  $36\text{m/s}$ ,  $r = 0.025\text{m}$ ,  $C_{ae} = 0.015\text{Ns}^2/\text{m}^2$ ,  $V_w = 0$  to  $14\text{m/s}$ ,  $\theta = 0$  to  $0.52\text{rad}$ ,  $C_{rr} = 0.15$ . In the second member of equation (13): The first term is the traction force. The second term is the aerodynamics force. The third term is the slope effect force, and the fourth term is the rolling resistance force.

##### 4.2 Model free setting

The model given in (13) can be expressed as

$$\dot{V}_x = F + \frac{1}{Mr} C \tag{14}$$

with

$$F = \frac{1}{M}(-C_{ae}(V_x + V_w)|V_x + V_w| - Mg \sin(\theta) - MgC_{rr}\text{sign}(V_x) \cos(\theta)) \tag{15}$$

which is of the form (2) with  $\alpha = 1/Mr$ . Thus, we have

$$\begin{aligned}
C &= Mr \left( \dot{V}_{xr} - \hat{F} - k_p e - k_i \int_0^t e(\tau) d\tau \right) \\
\hat{F} &= \hat{V}_x - \frac{1}{Mr} \tilde{C} \\
e &= V_x - V_{xr}
\end{aligned} \tag{16}$$

with  $V_{xr}$  the reference speed,  $\hat{V}_x$  an estimate of the derivative of  $V_x$ , and  $\tilde{C}$  a past value of  $C$  (an approximation of  $C$ ).

For instance, we can take the above form in discrete time

$$\begin{aligned}
C(t_k) &= C(t_{k-1}) + Mr(\hat{e}(t_k) + k_p e(t_k) + k_i I(t_k)) \\
\hat{e}(t_k) &= \dot{V}_{xr}(t_k) - \hat{V}_x(t_k) \\
e(t_k) &= V_{xr}(t_k) - V_x(t_k) \\
I(t_k) &= I(t_{k-1}) + h e(t_k) \\
\hat{V}_x(t_k) &= \frac{T_f}{T_f + h} \hat{V}_x(t_{k-1}) + \frac{1}{T_f + h} (V_x(t_k) - V_x(t_{k-1})) \\
h &= t_k - t_{k-1}
\end{aligned} \tag{17}$$

For comparison, a usual PID takes the following form

$$\begin{aligned}
C(t_k) &= K_p e(t_k) + K_i I(t_k) \\
e(t_k) &= V_{xr}(t_k) - V_x(t_k) \\
I(t_k) &= I(t_{k-1}) + h e(t_k) \\
h &= t_k - t_{k-1}
\end{aligned} \tag{18}$$

##### 4.3 Simulations: continuous ideal flatness based control

Supposing we have the full knowledge of the dynamics, the ideal flatness based control is of the form:

$$\begin{aligned}
C &= Mr \left( V_{xr} - F - k_p e - k_i \int_0^t e(\tau) d\tau \right) \\
F &= -C_{ae}(V_x + V_w)|V_x + V_w| - Mg \sin(\theta) - MgC_{rr}\text{sign}(V_x) \cos(\theta) \\
e &= V_x - V_{xr}
\end{aligned} \tag{19}$$

The error in the case of flatness based control is depicted in figure 1.

##### 4.4 Simulations: time triggered PI control and i-PID control

We first compare the cases of a time triggered PID and a time triggered *i*-PID.

Consider a fixed sampling time of  $h = 10\text{ms}$  (knowing that  $h = 35\text{ms}$  is the limit of stability). This yields 1976 actuation steps. We take a PI controller with gains  $k_p = 17000$  and  $k_i = 100$ . The reference trajectory and the tracking error are depicted in figure 1.

##### 4.5 Simulations: event triggered PI control and i-PID control

We now consider the event triggered controls. The event triggering scheme for PI control is the classical error difference of equation (9). The limit  $e_{lim}$  in (9) is taken as

$$e_{lim} = \frac{\max(y_r) - \min(y_r)}{200}$$

It yields 291 actuation steps and the tracking error is given in figure 1. The *i*-PI controller is with gains  $K_p = 60$  and  $K_i = 6$ .

##### 4.6 Discussion

Note that the maximum absolute tracking error is  $6.4 \cdot 10^{-2}$  m/s in the PI case, and  $3.2 \cdot 10^{-3}$  m/s in the *i*-PI case which

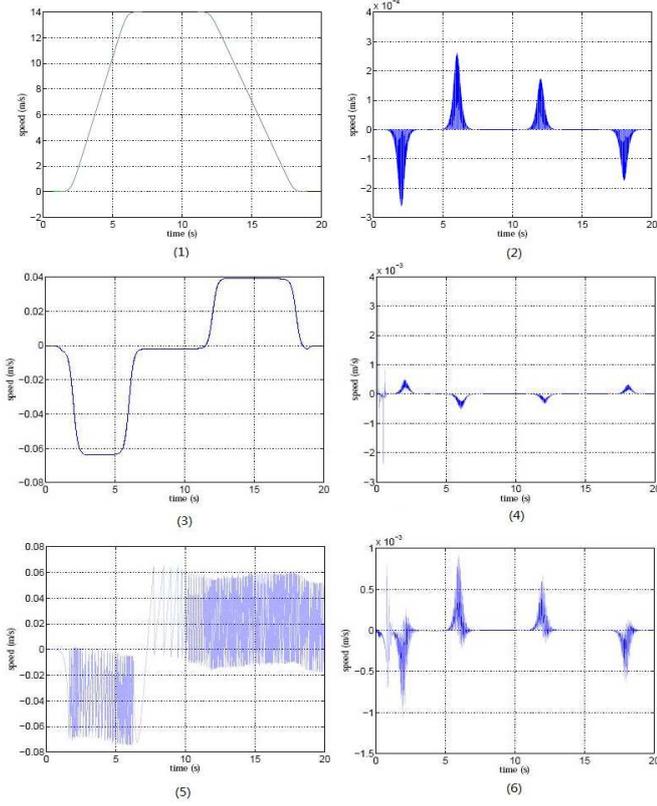


Fig. 1. (1) Reference trajectory. (2) Tracking errors of ideal flatness based control. (3) Tracking errors of time triggered PI control. (4) Tracking errors of time triggered  $i$ -PID control. (5) Tracking errors of event triggered PI control. (6) Tracking errors of event triggered  $i$ -PID control.

is 20 times less (2000%) than in the PI case. If we exclude the first second, the maximum absolute tracking error in the  $i$ -PI case is of  $5.2 \cdot 10^{-4}$ , which is 123 times less than in the PI case.

Consider now an  $i$ -PI control. The event triggered scheme is the one given in equation (11), with  $l = 1$ ,  $L = 20$ ,  $\xi_l = -2$ ,  $\xi_H = 2$ , and  $\zeta = 6$ . The corresponding tracking error is given in figure 1, and was performed in 569 actuation steps. The gain in performance, when using an  $i$ -PI instead of a PI, is 68.18 and the loss in actuation steps is 1.95.

Model free control has better performance than PI control. Using the event triggered schemes,  $i$ -PID can further reduce the number of actuation loops, which is very useful for real time control systems.

## 5. APPLICATION TO QUADROTOR CONTROL

### 5.1 Model

The chosen model of quadrotor is depicted in equations (20). See Bouabdallah [2007]. The rotation angles  $\phi$ ,  $\theta$  and  $\psi$  are along the world axis  $x$ ,  $y$  and  $z$  respectively, namely, roll, pitch and yaw.  $\Omega_r$  ( $i = 1..4$ ) are the angular velocities of each rotor, which are the real inputs of the quadrotor. The forces  $T_i$ ,  $H_i$  ( $i = 1..4$ ) are the thrust and hub forces

of each motor. The moments  $R_i$ ,  $Q_i$  ( $i = 1..4$ ) are the drag and rolling moments of each rotor. The quantities  $\dot{\omega}_1 \dot{\omega}_2 (I_{ii} - I_{jj})$ ,  $J_r \dot{\omega} \Omega_r$  ( $\omega = \phi, \theta, \psi$ ;  $i = x, y, z$ ) are the body gyroscopic effects and propeller gyroscopic effects. The notations  $c$  and  $s$  represent  $\cos$  and  $\sin$  respectively. The values of all the parameters can be found in Bouabdallah [2007].

$$\begin{aligned}
 I_{xx} \ddot{\phi} &= \dot{\theta} \dot{\psi} (I_{yy} - I_{zz}) + J_r \dot{\theta} \Omega_r + l(-T_2 + T_4) - \\
 &\quad h \left( \sum_{i=1}^4 H_{yi} \right) + (-1)^{i+1} \sum_{i=1}^4 R_{mxi} \\
 I_{yy} \ddot{\theta} &= \dot{\phi} \dot{\psi} (I_{zz} - I_{xx}) - J_r \dot{\phi} \Omega_r + l(T_1 - T_3) - \\
 &\quad h \left( \sum_{i=1}^4 H_{xi} \right) + (-1)^{i+1} \sum_{i=1}^4 R_{myi} \\
 I_{zz} \ddot{\psi} &= \dot{\theta} \dot{\phi} (I_{xx} - I_{yy}) + (-1)^i \sum_{i=1}^4 Q_i + \\
 &\quad l(H_{x2} - H_{x4}) + l(-H_{y1} + H_{y3}) \\
 m \ddot{z} &= -mg + (c\theta c\phi) \sum_{i=1}^4 T_i \\
 m \ddot{x} &= (s\psi s\phi + c\psi s\theta c\phi) \sum_{i=1}^4 T_i - \sum_{i=1}^4 H_{xi} - \frac{1}{2} C_x A_c \rho \dot{x} |\dot{x}| \\
 m \ddot{y} &= (-c\psi s\phi + s\psi s\theta c\phi) \sum_{i=1}^4 T_i - \sum_{i=1}^4 H_{yi} - \frac{1}{2} C_y A_c \rho \dot{y} |\dot{y}|
 \end{aligned} \tag{20}$$

The most important forces and moments are the thrust  $T$  and the rolling moments  $Q$ . Therefore, we can take

$$\begin{aligned}
 u_1 &= \sum_{i=1}^4 T_i & u_2 &= l(-T_2 + T_4) \\
 u_3 &= l(T_1 - T_3) & u_4 &= (-1)^i \sum_{i=1}^4 Q_i
 \end{aligned}$$

as control inputs to compute the needed torques for each rotor, and then use them to control the altitude  $z$ , position  $x, y$  and direction  $\psi$ .

### 5.2 Altitude $z$ control

The equation given in (20) related to  $z$  can be expressed as

$$m \ddot{z} = (c\theta c\phi) u_1 + F_z \tag{21}$$

where  $F_z$  can be considered as disturbances (e.g. the wind) or some parts of dynamics neglected in (20). In discrete time, the unknown part  $F_z$  can be expressed as following. The estimate of  $\ddot{z}(k)$  is denoted as  $\hat{\ddot{z}}(k)$ .

$$\hat{F}_z = m \hat{\ddot{z}}(k) - (c\theta c\phi) u_1(t_{k-1}) \tag{22}$$

Therefore, the chosen control input is

$$u_1(t_k) = u_1(t_{k-1}) + \frac{m}{c\theta c\phi} (\hat{e}_{2d}^z(t_k) + k_1^z e_d^z(t_k) + k_2^z e^z(t_k)) \tag{23}$$

with

$$\begin{aligned}\hat{e}_{2d}^z(t_k) &= \ddot{z}_r(t_k) - \hat{\ddot{z}}(t_k), \quad e_d^z(t_k) = \dot{z}_r(t_k) - \dot{z}(t_k) \\ e^z(t_k) &= z_r(t_k) - z(t_k) \\ \hat{\ddot{z}}(t_k) &= \frac{T_f}{T_f + h} \hat{\ddot{z}}(t_{k-1}) + \frac{1}{T_f + h} (z(t_k) - z(t_{k-1}))\end{aligned}$$

$\ddot{z}_r, \dot{z}_r, z_r$  are the reference acceleration, velocity and position of  $z$ . The variable sampling step is  $h = t_k - t_{k-1}$ .

### 5.3 Position $x, y$ control

We want to use  $u_2$  and  $u_3$  to control directly the position  $x, y$ . Therefore, we need to differentiate twice the equations related to  $x$  and  $y$  in (20) in order to appear the control inputs  $u_2$  and  $u_3$ . Since the equations in  $x$  and  $y$  are coupled, we get

$$\begin{aligned}x^{(4)} &= \frac{u_1}{mI_{xx}}(s\psi c\phi - c\psi s\theta s\phi)u_2 + \frac{u_1}{mI_{yy}}(c\psi c\theta c\phi)u_3 + F_x \\ y^{(4)} &= -\frac{u_1}{mI_{xx}}(c\psi c\phi + s\psi s\theta s\phi)u_2 + \frac{u_1}{mI_{yy}}(s\psi c\theta c\phi)u_3 + F_y\end{aligned}\quad (24)$$

where  $F_x, F_y$  are considered as the badly known parts. For simplicity, we define  $A = \frac{u_1}{mI_{xx}}(s\psi c\phi - c\psi s\theta s\phi)$ ,  $B = \frac{u_1}{mI_{yy}}(c\psi c\theta c\phi)$ ,  $C = -\frac{u_1}{mI_{xx}}(c\psi c\phi + s\psi s\theta s\phi)$  and  $D = \frac{u_1}{mI_{yy}}(s\psi c\theta c\phi)$ . Using the model free control scheme as before, we get

$$\begin{pmatrix} u_2(t_k) \\ u_3(t_k) \end{pmatrix} = \begin{pmatrix} u_2(t_{k-1}) \\ u_3(t_{k-1}) \end{pmatrix} + \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} \begin{pmatrix} \hat{e}_{4d}^x + \sum_{i=0}^3 k_i^x e_{id}^x \\ \hat{e}_{4d}^y + \sum_{i=0}^3 k_i^y e_{id}^y \end{pmatrix}\quad (25)$$

where  $\hat{e}_{4d}^x, \hat{e}_{4d}^y$  are the errors between the references  $x_r^{(4)}, y_r^{(4)}$  and the estimates of  $x^{(4)}, y^{(4)}$ .

### 5.4 Yaw control

For yaw control, we consider the equation of  $\psi$  as

$$I_{zz}\ddot{\psi} = u_4 + F_\psi\quad (26)$$

Then the control feedback is

$$u_4(t_k) = u_4(t_{k-1}) + I_{zz}(\hat{e}_{2d}^\psi(t_k) + k_1^\psi e_d^\psi(t_k) + k_2^\psi e^\psi(t_k))\quad (27)$$

where  $\hat{e}_{2d}^\psi$  is the error between the reference  $\ddot{\psi}_r$  and the estimate of  $\ddot{\psi}$ .

### 5.5 Simulation: time triggered control

The task is to follow a rounded square path with length of 2m while hovering at the altitude of 10m, which is given in (28). The desired length is  $h_d$ , and  $T_f$  is the time needed to reach the desired length. Here we choose  $h_d$  equals 2m, and  $T_f$  equals 6s. The reference trajectory is in figure 2.

In the time triggered  $i$ -PID control, the sampling time is 10ms, and it yields 2785 actuation steps. The results are given in figure 3. The red lines are the desired trajectories. The maximum errors in  $x$  and  $y$  are both less than 0.05m, that is, less than 2, 5% of the desired length.

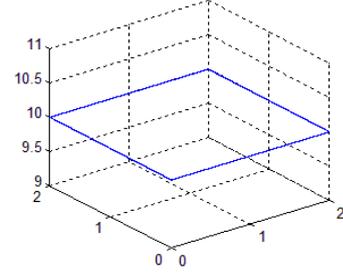


Fig. 2. Reference trajectory for the quadrotor.

$$\sigma(t) = \begin{cases} 0 & 0 \leq t \leq t_1, t_4 < t \leq 30 \\ h_d \frac{t^5}{t^5 + (T_f - t)^5} & t_1 < t \leq t_2 \\ 2 & t_2 < t \leq t_3 \\ h_d - h_d \frac{t^5}{t^5 + (T_f - t)^5} & t_3 < t \leq t_4 \end{cases}$$

$$\begin{aligned}h_d &= 2, T_f = 6 \\ x &= \sigma(t) \quad \text{with } t_1 = 3, t_2 = 9, t_3 = 15, t_4 = 21. \\ y &= \sigma(t) \quad \text{with } t_1 = 9, t_2 = 15, t_3 = 21, t_4 = 27. \\ z &= 10\end{aligned}\quad (28)$$

### 5.6 Simulation: event triggered control

In the event triggered  $i$ -PID control, the event triggering law is the absolute error limit. We set the error limit of  $z$  to be 0.1m. The error limit of yaw angle is 0.1rad. For  $x$  and  $y$ , we take the error limits both as 0.1m. The event triggered  $i$ -PID control yields 2389 actuation steps. The results are given in figure 4.

### 5.7 Discussion

The system mentioned in (20) is not complete. The aerodynamics of the system is complicated, and many more forces and moments will affect the system. Therefore, a control scheme which can adapt to the changes of the system is needed. The time triggered model free control performed nicely. It controls the system without the need of computing all the forces and moments in the system. In event triggered  $i$ -PID control, we set the the error limit to be 5% of the reference. It has 396 steps less comparing to the time triggered model free scheme while still achieving stability.

## 6. CONCLUSION

Event triggered model free controllers which yields strong robustness while needing few computing resources is proposed in this paper. It is very efficient to control the non-linear multi-input-output system which traditional PID is not able to. The  $i$ -PID control is also efficient to solve the partially known systems. From the simulation of a quadrotor model, we see that the  $i$ -PID control scheme avoids the heavy computations of the control laws, forces, moments and 4th derivatives of the variables. Moreover, the event triggered scheme enables to eliminate the small vibrations in the system while diminishing the number of actuation steps.

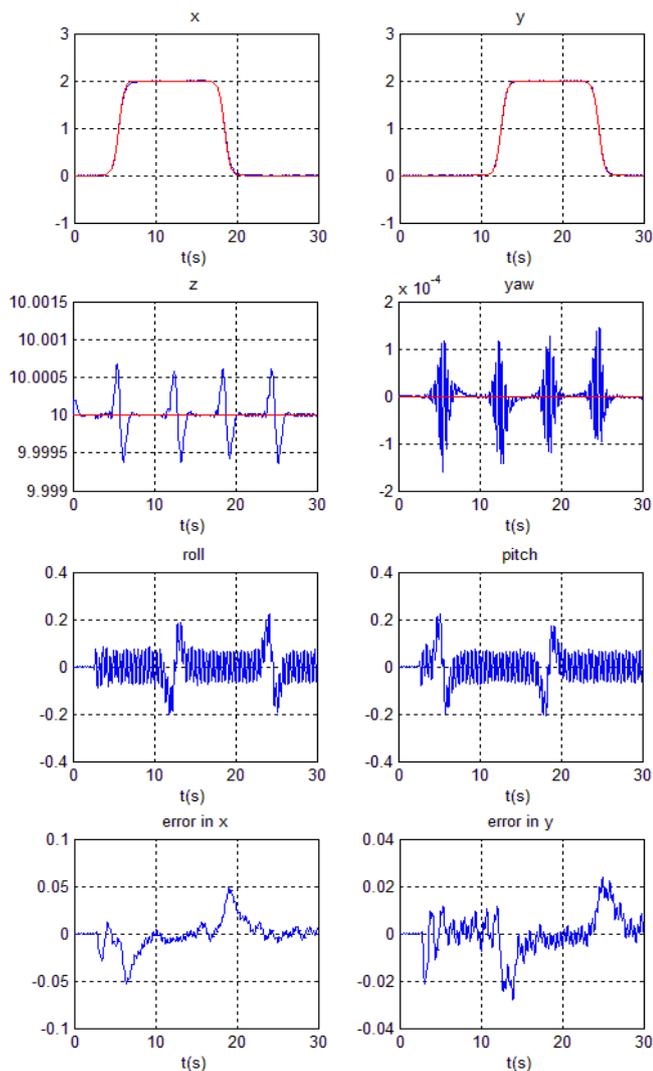


Fig. 3. Time triggered *i*-PID control of quadrotor

#### ACKNOWLEDGEMENTS

Jing Wang is financially supported by IPSA (Institut Polytechnique des Sciences Avancées).

#### REFERENCES

- K.E. Årzén. A simple event-based PID controller. *Proc. of 14th IFAC World Congress*, Beijing, PR China, 1999.
- B. d'Andréa-Novel, M. Fliess, C. Join, H. Mounier, B. Steux. A mathematical explanation via "intelligent" PID controllers of the strange ubiquity of PIDs. *Proc. of 18th Mediterranean Conference on Control and Automation*, Marrakech, Morocco, 2010.
- S. Bouabdallah. Design and Control of Quadrotors with Application to Autonomous Flying. *PhD thesis*, Ecole Polytechnique Federale De Lausanne, 2007.
- S. Choi, B. d'Andréa-Novel, M. Fliess, H. Mounier. Model-free control of automotive engine and brake for stop-and-go scenario. *Proc. of 10th Europ. Control Conf. (ECC'09)*, Budapest.
- M. Fliess, J. Levine, P. Martin and P. Rouchon. Flatness and defect of nonlinear systems : introductory theory

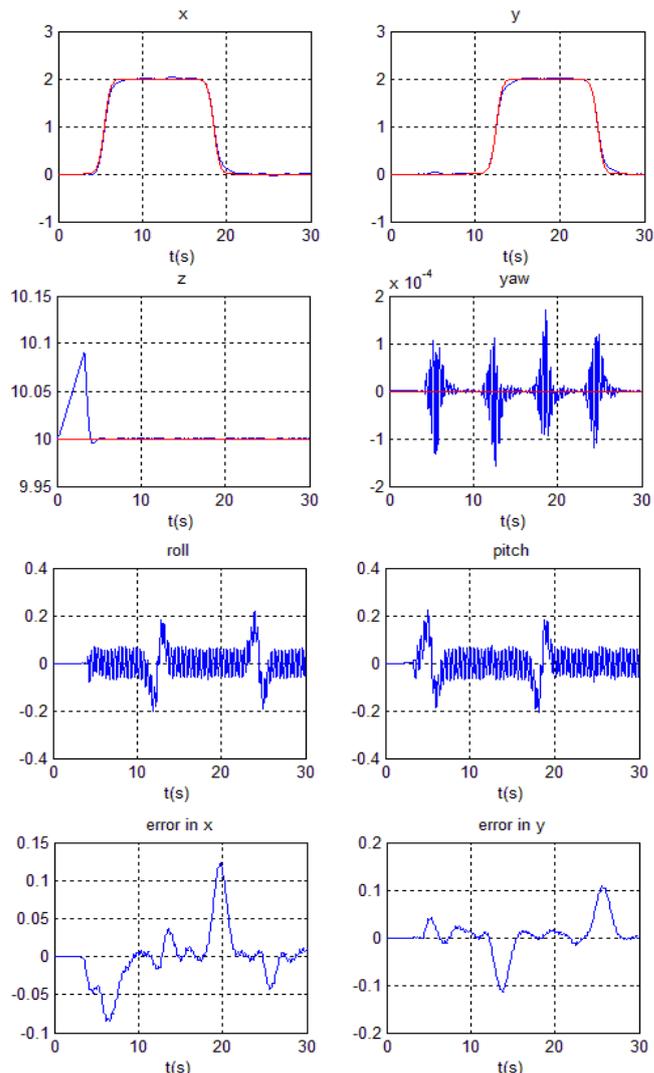


Fig. 4. Event triggered *i*-PID control of quadrotor

and examples. *Internat. J. Control*, Vol. 61, pp. 1327–1361, 1995.

- M. Fliess and C. Join. Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control?. *Proc. of 15th IFAC Symp. System Identif.*, Saint-Malo, France, 2009.
- U. Kiencke, L. Nielsen. *Automotive Control Systems: For Engine, Driveline, And Vehicle*. Springer, Berlin, 2005.
- S.G. Krantz, H.R. Parks. *The Implicit Function Theorem: History, Theory, and Applications*. Birkhser, Boston, United States, 2002.
- R.M. Murray, K.J. Astrom, S.P. Boyd, R.W. Brockett, G. Stein. Future Directions in Control in an Information-Rich World. *IEEE Contr. Syst. Mag.*, vol. 23, pp. 20–33, 2003.
- M. Mboup, C. Join, and M. Fliess. Numerical differentiation with annihilators in noisy environment. *Numer. Algor.*, vol. 50, pp. 439–467, 2009.
- R. Rajamani. *Vehicle Dynamics and Control*. Springer, New York, 2006.
- E.D. Sontag. Input to state stability: Basic concepts and results. *Nonlinear and Optimal Control Theory*, pp. 163–220, Springer-Verlag, Berlin, 2007.

# ANALYSIS AND CONTROL OF QUADROTOR VIA A NORMAL FORM APPROACH

JING WANG\*<sup>†</sup> ISLAM BOUSSAADA\*<sup>†</sup> ARBEN CELA<sup>‡</sup> HUGUES MOUNIER\* AND SILVIU-IULIAN NICULESCU\*

**Abstract.** This paper focuses on the analysis and control of some mathematical models representing the dynamics of a quadrotor. By using a normal form approach, the highly coupled parts in the quadrotor system are eliminated, while all possible properties of the original system are not changed. The bifurcations of the system are then analyzed. A two dimensional system is deduced at the origin which can determine the stability and possible local bifurcations of the system. Based on the normal form and indirect method of Lyapunov, we propose a state feedback control method with computational simplicity as well as practical implementation facility. Comparing to a standard PID control, the proposed method has faster response time and less tracking errors especially with wind disturbance.

**Key words.** Normal forms, Quadrotor control, Center manifold.

**AMS subject classifications.** 93C10, 93C35, 93D15

**1. Introduction.** The quadrotor (see in Figure 1.1) is a mini unmanned aerial vehicle (UVA) with four rotors, which has been widely studied in the last decades [1, 2, 3, 4]. It is a system with four inputs, six outputs and highly coupled states. Due to its simplicity both in mechanical structure and maneuver, it is widely used in surveillance, search and rescue, mobile sensor networks [1]. Many methods have been proposed for controlling quadrotors. For example, Bouabdallah et al.[2] have proposed a backstepping control used separately in two subsystems. Besnard et al.[3] have proposed a sliding mode control driven by a disturbance observer. Wang et al.[4] have presented an event driven model free control which can avoid heavy computation. However, to the best of the authors' knowledge, the bifurcation of the dynamical system have never been studied.

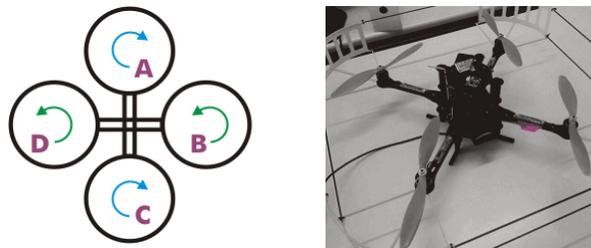


FIG. 1.1. *The quadrotor(right) from Ascending technologies(Available at ESIEE).*

The method of normal forms is an useful approach in studying the dynamical system properties [5]. Its purpose is employing successive coordinate transformations

---

\*Laboratoire des Signaux et Systèmes (L2S), CNRS, Supélec, Université Paris Sud 11, Supélec, 3 rue Joliot Curie, 91192 Gif Sur Yvette Cedex. (e-mails: {jing.wang, islam.boussaada, hugues.mounier, silviu.niculescu}@lss.supelec.fr).

<sup>†</sup>Institut Polytechnique des Sciences Avancées(IPSA), 7-9 rue Maurice Grandcoing 94200, Ivry sur Seine.

<sup>‡</sup>UPE, ESIEE Paris, Embedded system department, 2 Bd Blaise Pascal, 93162 Noisy Le Grand Cedex (e-mail: celaa@esiee.fr).

to construct the simplest form of the system. The normal form exhibits all possible properties of the original system. The normal forms of any degree with a single input were obtained by using change of coordinates and feedback [6]. For multi-input systems, the normal forms are deduced from the system with two inputs [7]. Based on the normal forms, the bifurcations and its control were studied by several authors [6, 8]. Center manifold is usually applied with the normal forms. It reduces the system to a center manifold associated with parts of the system with the eigenvalues with zero real parts at a bifurcation point [9].

To the best of our knowledge, the normal form and center manifold theories have never been used in the analysis and control of quadrotor. In this paper, the normal form of the quadrotor system is firstly calculated. By using such a methodology, the highly coupled parts in quadrotor system are eliminated. Under certain control laws, the normal form is reduced into a two dimensional system at the bifurcation point by using center manifold theory. Also, a simple control method based on the normal form using state feedback is proposed. The control laws are proposed to ensure the asymptotical stability of the system by moving all the eigenvalues of the system to the open left half plane. Comparing to a standard PID control, the proposed method has faster response time and less tracking errors especially when there is wind disturbance, as illustrated at the end of the paper. The interest of considering such control laws lies in the simplicity of the controller as well as in its practical implementation facility.

The paper is organized as follows: In Section 2, the model of quadrotor is given. In Section 3, the normal form of quadrotor is deduced. In Section 4, the bifurcation of the system under certain control laws is analyzed. In Section 5, simulations with and without wind disturbance using the proposed method and PID control are given.

**2. The quadrotor model.** The chosen model of quadrotor is depicted in equations (2.1). The rotation angles  $\phi$ ,  $\theta$  and  $\psi$  are along the world axis  $x$ ,  $y$  and  $z$  respectively, namely roll, pitch and yaw.  $w_i (i = 1..4)$  are the accelerations caused by four rotors, which are the inputs of the system. ( $g = 9.8\text{m/s}^2$  the gravity).

$$(2.1) \quad \begin{aligned} \ddot{x} &= -w_1 \sin\theta, & \ddot{y} &= w_1 \cos\theta \sin\phi, & \ddot{z} &= w_1 \cos\theta \cos\phi - g, \\ \ddot{\phi} &= w_2, & \ddot{\theta} &= w_3, & \ddot{\psi} &= w_4. \end{aligned}$$

We introduce the variables as  $x_1 = x$ ,  $x_2 = \dot{x}$ ,  $x_3 = y$ ,  $x_4 = \dot{y}$ ,  $x_5 = z$ ,  $x_6 = \dot{z}$ ,  $x_7 = \phi$ ,  $x_8 = \dot{\phi}$ ,  $x_9 = \theta$ ,  $x_{10} = \dot{\theta}$ ,  $x_{11} = \psi$ ,  $x_{12} = \dot{\psi}$ . Therefore, we can rewrite the system as:

$$(2.2) \quad \begin{aligned} \dot{x}_1 &= x_2, & \dot{x}_2 &= -w_1 \sin(x_9), & \dot{x}_3 &= x_4, & \dot{x}_4 &= w_1 \cos(x_9) \sin(x_7), \\ \dot{x}_5 &= x_6, & \dot{x}_6 &= w_1 \cos(x_9) \cos(x_7) - g, & \dot{x}_7 &= x_8, & \dot{x}_8 &= w_2, \\ \dot{x}_9 &= x_{10}, & \dot{x}_{10} &= w_3, & \dot{x}_{11} &= x_{12}, & \dot{x}_{12} &= w_4. \end{aligned}$$

**3. Normal form of the system.** It is easy to see that the equilibria of the system (2.2) are  $x_e = (c_1, 0, c_2, 0, c_3, 0, k\pi, 0, k\pi, 0, c_4, 0)$ ,  $w = (g, 0, 0, 0)$ , where  $k = 0, \pm 1, \pm 2, \dots$ ,  $c_i (i = 1..4) \in \mathbb{R}$  are constants and  $g$  is the gravity. Note in the real control system,  $\phi, \theta \in (-\pi/2, \pi/2)$  and  $\psi \in [0, \pi)$ . Therefore, without losing generality, only the equilibrium  $x_0 = (x, w) = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, g, 0, 0, 0)$  is considered. We move  $x_0$  to the origin by changing the coordinates of the inputs  $w_1 = u_1 + g, w_2 = u_2, w_3 = u_3, w_4 = u_4$ . Then, using the Taylor series of function  $\sin(x)$  and  $\cos(x)$  at  $x = 0$ . The system (2.2) can be written in polynomial form as follows. Here,  $O^5$  are the polynomials with 5th and higher degree:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -gx_9 - u_1x_9 + \frac{gx_9^3}{6} + \frac{u_1x_9^3}{6} + O^5,$$

$$\begin{aligned}\dot{x}_3 &= x_4, \quad \dot{x}_4 = gx_7 + u_1x_7 - \frac{gx_7^3}{6} - \frac{gx_9^2x_7}{2} - \frac{u_1x_7^3}{6} - \frac{u_1x_9^2x_7}{2} + O^5, \\ \dot{x}_5 &= x_6, \quad \dot{x}_6 = u_1 - \frac{gx_7^2}{2} - \frac{gx_9^2}{2} - \frac{u_1x_7^2}{2} - \frac{u_1x_9^2}{2} + \frac{gx_7^4}{24} + \frac{gx_9^2x_7^2}{4} + \frac{gx_9^4}{24} + O^5, \\ \dot{x}_7 &= x_8, \quad \dot{x}_8 = u_2, \quad \dot{x}_9 = x_{10}, \quad \dot{x}_{10} = u_3, \quad \dot{x}_{11} = x_{12}, \quad \dot{x}_{12} = u_4.\end{aligned}$$

Using the state and input transformation  $y_1 = x_1, y_2 = x_2, y_3 = x_3, y_4 = x_4, y_5 = x_5, y_6 = x_6, y_7 = gx_7, y_8 = gx_8, y_9 = gx_9, y_{10} = gx_{10}, y_{11} = x_{11}, y_{12} = x_{12}, v_1 = u_1, v_2 = gu_2, v_3 = gu_3, v_4 = u_4$ , we change the system (3.1) into Brunovsky form:

$$(3.1) \quad \begin{aligned}\dot{y}_1 &= y_2, \quad \dot{y}_2 = -y_9 - \frac{v_1y_9}{g} + \frac{y_9^3}{6g^2} + \frac{v_1y_9^3}{6g^3} + O^5, \\ \dot{y}_3 &= y_4, \quad \dot{y}_4 = y_7 + \frac{v_1y_7}{g} - \frac{y_7y_9^2}{2g^2} - \frac{y_7^3}{6g^2} - \frac{v_1y_7^3}{6g^3} - \frac{v_1y_7y_9^2}{2g^3} + O^5, \\ \dot{y}_5 &= y_6, \quad \dot{y}_6 = v_1 - \frac{y_7^2}{2g} - \frac{y_9^2}{2g} - \frac{v_1y_7^2}{2g^2} - \frac{v_1y_9^2}{2g^2} + \frac{y_7^4}{24g^3} + \frac{y_7^2y_9^2}{4g^3} + \frac{y_9^4}{24g^3} + O^5, \\ \dot{y}_7 &= y_8, \quad \dot{y}_8 = v_2, \quad \dot{y}_9 = y_{10}, \quad \dot{y}_{10} = v_3, \quad \dot{y}_{11} = y_{12}, \quad \dot{y}_{12} = v_4.\end{aligned}$$

The system (3.1) can be written as:

$$(3.2) \quad \dot{y} = f(y) + g(y)v = Ay + f^{(2)}(y) + f^{(3)}(y) + Bv + g^{(1)}(y)v + g^{(2)}(y)v + O^4$$

where  $A, B$  are the coefficients of the linear parts,  $f^{(2)}(y), g^{(1)}(y)v$  are the second degree homogeneous polynomials of the system,  $f^{(3)}(y), g^{(2)}(y)v$  are the third degree homogeneous polynomials.

We take a third-degree homogeneous transformation for example [10]:

$$(3.3) \quad y = z + \phi^{(2)}(z) + \phi^{(3)}(z)$$

which  $z$  are the new states of the system.  $\phi^{(2)}(z)$  is a second degree homogeneous polynomial and  $\phi^{(3)}(z)$  is a third degree homogeneous polynomial of the states  $z$ , whose coefficients will be defined later.

We get the derivative of equation (3.3). Therefore, the derivative of the new states  $z$  are:

$$(3.4) \quad \dot{z} = \left(I + \frac{d\phi^{(2)}}{dz} + \frac{d\phi^{(3)}}{dz}\right)^{-1}\dot{y}$$

where,

$$\left(I + \frac{d\phi^{(2)}}{dz} + \frac{d\phi^{(3)}}{dz}\right)^{-1} = I - \frac{d\phi^{(2)}}{dz} - \frac{d\phi^{(3)}}{dz} + \left(\frac{d\phi^{(2)}}{dz}\right)^2 + \left(\frac{d\phi^{(3)}}{dz}\right)^2 + 2\frac{d\phi^{(2)}}{dz}\frac{d\phi^{(3)}}{dz} \dots$$

In (3.2), we rewrite the  $f(y)$  and  $g(y)$  using the new states  $z$ .

$$f(y) = A(y) + f^{(2)}(y) + f^{(3)}(y) + O^4 = Az + A\phi^{(2)}(z) + f^{(2)}(z) + A\phi^{(3)}(z) + f^{(3)}(z) \dots$$

$$g(y) = B + g^{(1)}(y) + g^{(2)}(y) + O^3 = B + g^{(1)}(z) + g^{(1)}(\phi^{(2)}(z)) + g^{(2)}(z) \dots$$

Therefore, with the help of the equations (3.2), (3.4), by now we have the new system:

$$\begin{aligned}\dot{z} &= Az + Bv + A\phi^{(2)}(z) + f^{(2)}(z) + g^{(1)}(z)v - \frac{d\phi^{(2)}}{dz}Az - \frac{d\phi^{(2)}}{dz}Bv + A\phi^{(3)}(z) \\ &\quad + f^{(3)}(z) + g^{(2)}(z)v + g^{(1)}(\phi^{(2)}(z))v - \frac{d\phi^{(2)}}{dz}(A\phi^{(2)}(z) + f^{(2)}(z) + g^{(1)}(z)v) \\ &\quad - \frac{d\phi^{(3)}}{dz}(Az + Bv) + \left(\frac{d\phi^{(2)}}{dz}\right)^2(Az + Bv) + O^4\end{aligned}$$

For the simplicity of the system, the states  $z$  and the inputs  $v$  should be separated. In the third degree normal form, the polynomial  $g^{(1)}(z)v$ ,  $g^{(2)}(z)v$  should be canceled.

$$\begin{aligned} g^{(1)}(z) - \frac{d\phi^{(2)}}{dz}B &= 0 \\ g^{(2)}(z) + g^{(1)}(\phi^{(2)}(z)) - \frac{d\phi^{(2)}}{dz}g^{(1)}(z) - \frac{d\phi^{(3)}}{dz}B + \left(\frac{d\phi^{(2)}}{dz}\right)^2B &= 0 \end{aligned}$$

Therefore, the transformation in equation (3.3) should be:

$$\begin{aligned} \phi^{(2)}(z) &= (0, -\frac{z_6 z_9}{g}, 0, \frac{z_6 z_7}{g}, 0, 0, 0, 0, 0, 0, 0, 0), \\ \phi^{(3)}(z) &= (0, 0, 0, 0, 0, -\frac{z_6 z_7^2}{2g^2} - \frac{z_6 z_9^2}{2g^2}, 0, 0, 0, 0, 0, 0). \end{aligned}$$

Using the same method, we can calculate the normal form of any degree. A Maple package ‘QualitativeODE’ [11] has been made for calculating the normal form of quadrotor. Using this programme, we get the third degree normal form of the system (3.1) as:

$$(3.5) \quad \begin{aligned} \dot{z}_1 &= z_2 - \frac{z_6 z_9}{g}, & \dot{z}_2 &= -z_9 + \frac{z_6 z_{10}}{g} - \frac{z_9^3}{3g^2} - \frac{z_7^2 z_9}{2g^2} + O^4, \\ \dot{z}_3 &= z_4 + \frac{z_6 z_7}{g}, & \dot{z}_4 &= z_7 - \frac{z_6 z_8}{g} + \frac{z_7^3}{3g^2} + O^4, \\ \dot{z}_5 &= z_6 - \frac{z_6 z_7^2}{2g^2} - \frac{z_6 z_9^2}{2g^2}, & \dot{z}_6 &= v_1 - \frac{z_7^2}{2g} - \frac{z_9^2}{2g} + \frac{z_6 z_7 z_8}{g^2} + \frac{z_6 z_9 z_{10}}{g^2} + O^4, \\ \dot{z}_7 &= z_8, & \dot{z}_8 &= v_2, & \dot{z}_9 &= z_{10}, & \dot{z}_{10} &= v_3, & \dot{z}_{11} &= z_{12}, & \dot{z}_{12} &= v_4. \end{aligned}$$

#### 4. Bifurcation and simplification of the control system.

**4.1. Bifurcation of the roots.** It is easy to see that in the linear part of the equation (3.5),  $z_1$  is related only to  $z_2, z_9, z_{10}, v_3$ ;  $z_3$  is related to  $z_4, z_7, z_8, v_2$ ;  $z_5$  is related to  $z_6, v_1$ ;  $z_{11}$  is related to  $z_{12}, v_4$ . Therefore, the control laws can be defined as:

$$\begin{aligned} v_1 &= K_{11}z_5 + K_{12}z_6, & v_2 &= K_{21}z_3 + K_{22}z_4 + K_{23}z_7 + K_{24}z_8, \\ v_4 &= K_{41}z_{11} + K_{42}z_{12}, & v_3 &= K_{31}z_1 + K_{32}z_2 + K_{33}z_9 + K_{34}z_{10}. \end{aligned}$$

In this way, we can move the related eigenvalues in each group separately without changing the eigenvalues in other groups. Here, we define  $v_i (i = 1..4)$  as:

$$\begin{aligned} v_1 &= -256z_5 + K_{12}z_6, & v_2 &= -100z_3 - 308z_4 - 256z_7 - 32z_8, \\ v_4 &= -1024z_{11} + K_{42}z_{12}, & v_3 &= 100z_1 + 308z_2 - 256z_9 - 32z_{10}. \end{aligned}$$

The system has three equilibria  $P_1^e = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ ,  $P_2^e = (0, 0, 43.45, 0, -0.057, 0, -16.97, 0, 0, 0, 0, 0)$  and  $P_3^e = (0, 0, -43.45, 0, -0.057, 0, 16.97, 0, 0, 0, 0, 0)$ . However, only the origin  $P_1^e$  can be stable when  $K_{12}, K_{42}$  change.

At the equilibrium  $P_1^e$ , for simplicity  $K_{12} = K_{42}$ , when  $K_{12}$  changes, the real and imaginary parts of the eigenvalues are in Figure 4.1. When  $K_{12} < 0$ , the system has four eigenvalues with positive real parts, and the system becomes unstable. When  $K_{12} > 0$ , the system has all eigenvalues with negative real parts, and the system is asymptotically stable. When  $K_{12} = 0$ , the system has two pairs of pure imaginary eigenvalues  $\pm 16i$  and  $\pm 32i$ , and all other eigenvalues have negative real parts, which is a four dimensional center manifold. The stability cannot be determined by the linear part of the system. It depends on the nonlinearity of the system.

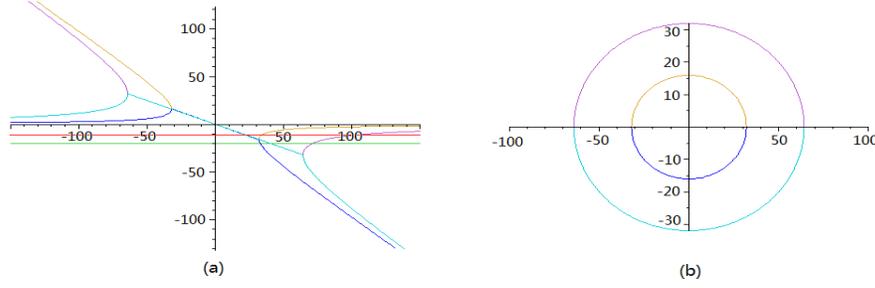


FIG. 4.1. The eigenvalues when  $K_{12}$  changes from -150 to 150: (a) the real parts. (b) the imaginary parts.

**4.2. Center manifold.** The aim of this part is to get the reduced system which can determine the stability and possible local bifurcations of the system at one bifurcation point [12]. A system can be written as:

$$\dot{x} = A(b)x + F(x), \quad x \in R^n$$

where  $b$  is a free parameter,  $b \in R$ .

At its origin  $x = [0, \dots, 0]$ ,  $J(b)$  is the Jordan form of the matrix  $A(b)$  and  $Q$  is a matrix which enables  $Q(b)J(b)Q^{-1}(b) = A(b)$ . Therefore, we have:

$$\dot{x} = Q(b)J(b)Q^{-1}(b)x + F(x) \Rightarrow Q^{-1}(b)\dot{x} = J(b)Q^{-1}(b)x + Q^{-1}(b)F(x)$$

we define  $y = Q^{-1}(b)x$ , then

$$(4.1) \quad \dot{y} = J(b)y + Q^{-1}(b)F(Q(b)y) = J(b)y + \tilde{F}(y)$$

we can separate the Jordan matrix  $J$  as matrices  $B$  and  $C$  whose eigenvalues have zero real parts and negative real parts respectively. Therefore, we can rewrite the system (4.1) at the origin with  $x = [0, \dots, 0]$ .

$$\dot{y}_0 = By_0 + f(y_0, y_-), \quad \dot{y}_- = Cy_0 + g(y_0, y_-).$$

Since the center manifold is tangent to  $E^c$  (the  $y_- = 0$  space), we define

$$(4.2) \quad y_- = h(y_0, b), \quad h(0, 0) = Dh(0, 0) = 0, \quad \dot{b} = 0.$$

We can calculate the function  $h(y_0, b)$  by using

$$\dot{y}_- = Dh(y_0, b)y_0 = Dh(y_0, b)[By_0 + f(y_0, h(y_0, b))] = Cy_0 + g(y_0, h(y_0, b))$$

Therefore, we can get the local evolution equations of  $y_0$  which can determine the stability of the original system.

In quadrotor center manifold analysis, the control laws are defined as:

$$\begin{aligned} v_1 &= -256z_5 - bz_6 - z_5^3, & v_2 &= -100z_3 - 308z_4 - 256z_7 - 32z_8, \\ v_4 &= -10z_{11} - 24z_{12}, & v_3 &= 100z_1 + 308z_2 - 256z_9 - 32z_{10}. \end{aligned}$$

The bifurcation of the system is like in previous subsection. When  $b < 0$ , the system has two eigenvalues with positive real parts. When  $b > 0$ , the system has all

eigenvalues with negative real parts. When  $b = 0$ , the system has two pure imaginary eigenvalues  $\pm 16i$ , and all other eigenvalues have negative real parts. The stability depends on the nonlinear parts of the system. We can use the center manifold theory to simplify the system, and further simplify the study of the bifurcation of the system. In this control system,  $y_0 = [y_1, y_2]^T = [z_5, z_6]^T$  and  $y_- = [y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}]^T = [z_1, z_2, z_3, z_4, z_7, z_8, z_9, z_{10}, z_{11}, z_{12}]^T$ .

We seek a quadratic center manifold ( $a$  are parameters to be defined later):

$$y_i = a_{i200}y_1^2 + a_{i020}y_2^2 + a_{i002}b^2 + a_{i110}y_1y_2 + a_{i101}y_1b + a_{i011}y_2b, \quad i = 3..12$$

Using the method mentioned before, we get  $h(y_0, b) = [-0.62b^2, -0.62b^2, -0.76b^2, -0.76b^2, 0, 0, 0, 0, -0.42b^2, -23.58b^2]$  in equation (4.2).

Therefore, the reduced system on the center manifold can be written:

$$(4.3) \quad \begin{aligned} \dot{y}_1 &= 16y_2 - 0.41b^4 - 0.011b^8 - (b + 0.057b^4)y_1 + 0.00024y_2^3 \\ \dot{y}_2 &= -16y_1 + 0.67b^4y_1 \end{aligned}$$

In the reduced system, when  $b$  is positive (negative), the origin is a stable (unstable) focus. When  $b = 0$ , the origin is a center. The phase portrait of equation (4.3) when  $b = -0.5$ ,  $b = 0$  and  $b = 0.5$  are depicted in Figure 4.2.

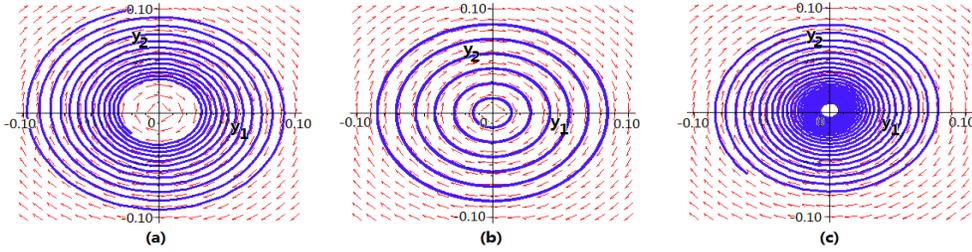


FIG. 4.2. The phase portrait of the reduced system: (a)  $b=-0.5$ . (b)  $b=0$ . (c)  $b=0.5$ .

**5. Quadrotor control.** Here we propose a control method based on the normal form and Lyapunov theory. In equation (3.5), the Jacobian matrix of the system can be easily found. If the system is time invariant, the indirect method of Lyapunov says that if the eigenvalues of Jacobian matrix of the system at the origin are in the open left half complex plane, then the origin is asymptotically stable. Therefore, we can define the state feedback as follows to move all the eigenvalues of the system to the open left half plane.  $x_r, y_r, z_r, \psi_r$  are the references.

$$\begin{aligned} v_1 &= -256(z_5 - z_r) - 32z_6, & v_2 &= -1700(z_3 - y_r) - 1000z_4 - 256z_7 - 32z_8 \\ v_4 &= -256(z_{11} - \psi_r) - 32z_{12}, & v_3 &= 1700(z_1 - x_r) + 1000z_2 - 256z_9 - 32z_{10} \end{aligned}$$

The simulation task is to let quadrotor follow a square path with the length of 2m while hovering at the altitude of 10m, which is given in Figure 5.1. The totally sample time is 20s. For comparison, the simulations using a standard PID control are also given.

**5.1. Simulation without wind disturbance.** The simulation results are given in Figure 5.2. The desired response time is 1s. We can see that the proposed method has better performance than a standard PID control.

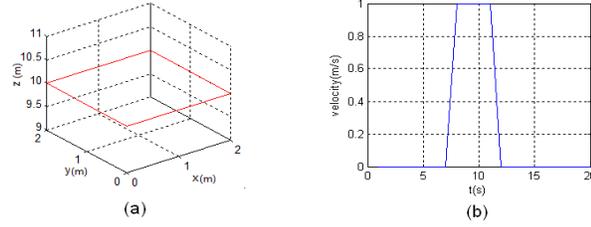


FIG. 5.1. (a) Reference trajectory for the quadrotor. (b) Wind disturbance.

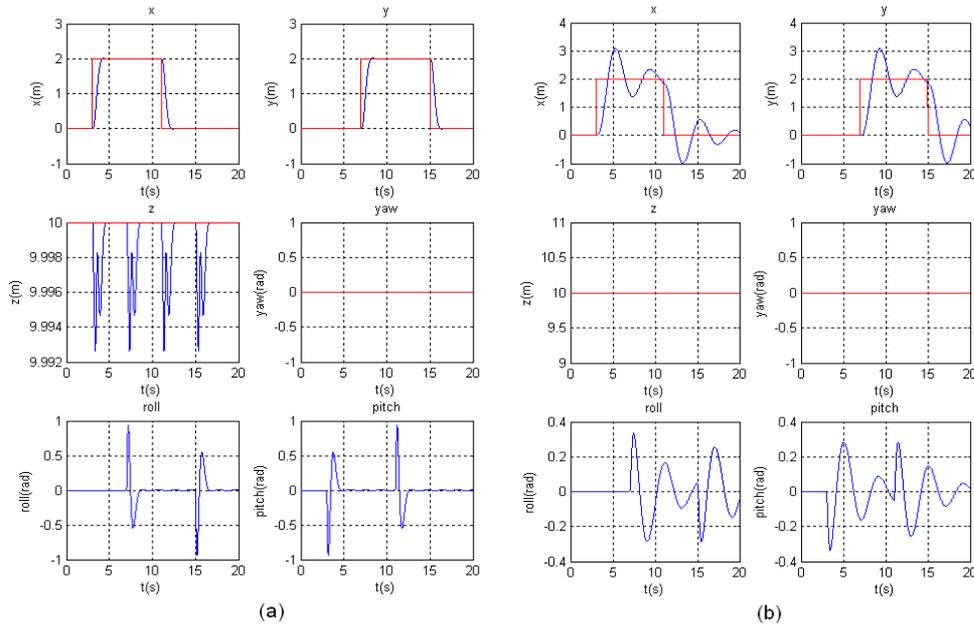


FIG. 5.2. The simulation without wind disturbance: (a) the proposed method. (b) a standard PID.

**5.2. Simulation with wind disturbance.** During the trajectory, there may have wind disturbance with velocity 1m/s as in Figure 5.1, which occurs in all x, y and z axis. The simulation results are given in Figure 5.3. The desired response time is 1s. The proposed method can keep the stability during the wind disturbance, and has better performance than a standard PID control.

**6. Conclusion.** In this paper, the normal form of quadrotor is deduced. A Maple package ‘QualitativeODE’ [11] has been written for calculating the normal form of any degree of the system. From equation (3.5), we can see that the highly coupled parts in quadrotor system are eliminated. This makes the analysis of the dynamical system easier. Under certain control laws, the system can be further deduced using center manifold theorem. A two dimensional system is deduced which can determine the stability and possible local bifurcations of the control system at the origin. Based on the normal form and indirect method of Lyapunov, we proposed a state feedback control method with computational simplicity as well as practical implementation facility. This method achieved good results. In the simulations, the system can remain stable with small tracking errors even if there is wind disturbance. Also, this

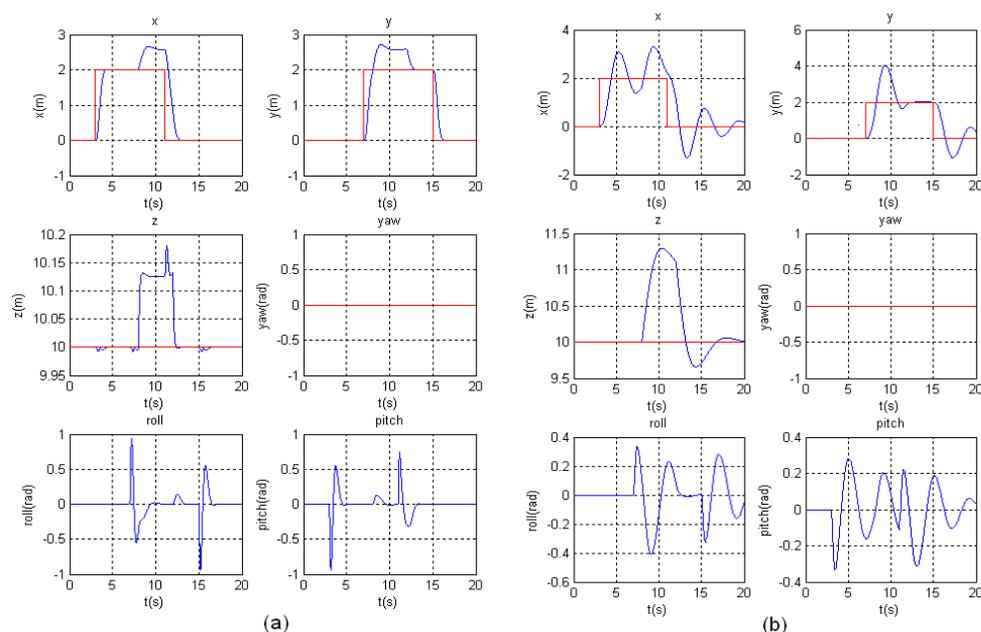


FIG. 5.3. The simulation with wind disturbance: (a) the proposed method. (b) a standard PID.

method has faster response time than a standard PID control.

#### REFERENCES

- [1] G. M. HOFFMANN, S. L. WASLANDER, C. J. TOMLIN, *Distributed Cooperative Search using Information-Theoretic Costs for Particle Filters with Quadrotor Applications*, Proceedings of the AIAA Guidance, Navigation, and Control Conference, Keystone, CO, Aug 2006.
- [2] S. BOUABDALLAH, R. STEGWART, *Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Apr 2005.
- [3] L. BESNARD, Y. SHTESSEL, B. LANDRUM, *Control of a quadrotor vehicle using sliding mode disturbance observer*, Proceedings of the American Control Conference, pp. 5230-5235, 2007.
- [4] J. WANG, H. MOUNIER, A. CELA, S.I. NICULESCU, *Event driven intelligent PID controllers with applications to motion control*, Proceedings of 18th IFAC World Congress, Milan, Aug 2011.
- [5] Y. KUZNETSOV, *Elements of applied bifurcation theory*, New York, Springer Verlag, 1998.
- [6] W. KANG, *Extended controller form and invariants of nonlinear control systems with a single input*, Journal of Mathematical Systems Estimation and Control, 6, 27-51, 1996.
- [7] I.A. TALL, W. RESPONDEK, *Normal forms for two-inputs nonlinear control systems*, Proceedings of the 41st IEEE Conference on Decision and Control, Dec 2002.
- [8] A.J. KRENER, W. KANG, *Control bifurcations*, IEEE Transactions on Automatic Control, pp. 1231-1246, Aug 2004.
- [9] J. CARR, *Applications of center manifold theory*, New York, Springer Verlag, 1981.
- [10] A.J. KRENER, S. KARAHAN, M. HUBBARD, R. FREZZA, *Higher order linear approximations to nonlinear control systems*, Proceedings of the 26th IEEE Conference on Decision and Control, 1987.
- [11] I. BOUSSAADA, *Analysis and control of quadrotor (French, in preparation, available upon request)*, Technical report IPSA, 2012.
- [12] J. GUCKENHEIMER, P. HOLMES, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, New York, Springer Verlag, 1983.

# Event driven model free control of quadrotor

Jing Wang, Marcel-Stefan Geamanu, Arben Cela, Hugues Mounier and Silviu-Iulian Niculescu

**Abstract**—In this paper we propose a new control approach, *event driven model free control*, which deals with the “trade-off” between computational cost and system performance. The model free control scheme demands low computational resources and has high robustness, which is especially suitable for systems with complex dynamics and/or affected by disturbances. Particularly for the embedded systems, the event driven model free control demands even fewer computational resources, since the actuation is allowed only when an event is triggered. The proposed method is implemented on a quadrotor model in different realistic scenarios with disturbances and uncertainties. Under the time and event triggered schemes, the model free control is compared with the backstepping and sliding mode controls in these scenarios.

## I. INTRODUCTION

Quadrotors are vertical take-off and landing (VTOL) aircrafts with four rotors, which have embedded microprocessors, micro-electro-mechanical (MEMS) sensors and Lithium Polymer (LiPo) batteries. Due to the simplicity in the design and maneuver, quadrotors have many applications, such as border patrol, surveillance, aerial photography, etc. The quadrotor system is nonlinear, which has twelve states highly coupled with the inputs (see eq. (14)). Its aerodynamics is complex and difficult to include all the parts in the modeling, which makes it a partially known system. In the applications, quadrotors are often affected by disturbances, such as wind, weather conditions, etc. Therefore, the quadrotor control systems bring out many challenges: control algorithm complexity reduction; energy consumption reduction; robustness to perturbations; fast response to environmental and system changes, etc.

Many control methods are proposed in literature: Castillo et al. have proposed a Lyapunov controller using a nested saturation algorithm [1]; S. Bouabdallah has implemented a backstepping control and a sliding mode control [2]; Mistler et al. have used a dynamic feedback control [3]; Mokhtari et al. have applied a mixed feedback linearization with linear  $\text{GH}_\infty$  controller. However, to the best of authors’ knowledge, these methods are tested in the ideal cases without

disturbances, and the comparison of the control methods have never been proposed on quadrotor in realistic scenarios.

The recently introduced model free control is proposed for the challenges in the control of quadrotor. A preliminary work can be found in [4]. It is a simple but efficient technique for the nonlinear, unknown or partially known dynamics [5]. While retaining the PID reduced computational cost, it is able to cope with general types of nonlinearities. The comparison between the model free control and traditional PID controllers is given in d’Andréa-Novel et al. [6]. Model free control has been implemented in some academic SISO systems [5], joint motion control in humanoid locomotion [7], non-minimum phase systems [8], etc.

In order to further save the computational resources and energy consumption, the event triggered scheme is proposed on the model free control. Contrarily to the time triggered control scheme, in the event based scheme the control signals are sent only upon the triggering of an event. The event driven control was firstly proposed by Ārzén [9]. The comparisons of the time driven and event driven control scheme for first order stochastic and nonlinear systems are proposed in [10] and [11] respectively.

The paper is organized as follows: In section II the event driven model free is presented; In section III the scenarios without and with wind disturbance are firstly presented. The model of quadrotor is then given. The model free control, backstepping control and sliding mode control laws are presented; In section IV the simulation results of the control laws in two scenarios are presented in both time and event driven schemes. The comparisons results in other realistic scenarios are also given.

## II. EVENT DRIVEN MODEL FREE CONTROL

A finite dimensional SISO system can be described implicitly as

$$E(y, \dot{y}, \dots, y^{(a)}, u, \dot{u}, \dots, u^{(b)}) = 0, \quad (1)$$

where  $E : \mathbf{R}^{a+1} \times \mathbf{R}^{b+1} \rightarrow \mathbf{R}$  is a sufficient smooth function of its arguments. Assume that for integer  $\nu$ ,  $0 < \nu \leq a$ ,  $\partial E / \partial y^{(\nu)} \neq 0$ . The implicit function theorem allows to express  $y^{(\nu)}$  locally

$$y^{(\nu)} = E(t, y, \dot{y}, \dots, y^{(\nu-1)}, y^{(\nu+1)}, \dots, y^{(a)}, u, \dot{u}, \dots, u^{(b)}), \quad (2)$$

with the function  $E : \mathbf{R} \times \mathbf{R}^a \times \mathbf{R}^{b+1} \rightarrow \mathbf{R}$ . No matter the system is linear or not, we can rewrite the system (1) as following phenomenological model which is only valid in a very short time interval:

$$y^{(\nu)} = F + \alpha u, \quad (3)$$

J. Wang, M.S. Geamanu, H. Mounier and S.I. Niculescu are with the Laboratory of Signals and Systems (L2S), CNRS, Supélec, Université Paris Sud 11, Supélec, 3, rue Joliot Curie, 91192 Gif sur Yvette Cedex (e-mails: {jing.wang, marcel.geamanu, hugues.mounier, silviu.niculescu}@lss.supelec.fr).

A. Cela is with Computer Science and Telecommunication Department, UPE, ESIEE Paris, 2 Bd Blaise Pascal, 93162 Noisy Le Grand Cedex (e-mail: celaa@esiee.fr).

J. Wang is also with Institut Polytechnique des Sciences avancées (IPSA), 7-9 rue Maurice Grandcoing 94200 Ivry-sur-Seine.

M.S. Geamanu is also with Institut Français du Pétrole et Énergies Nouvelles (IFPEN), 4 avenue du Bois Preau, 92500 Rueil-Malmaison.

where  $\alpha \in \mathbb{R}$  is a non-physical constant parameter, which is chosen by the engineer in such a way that  $F$  and  $\alpha u$  are of the same order of magnitude. The derivation order  $\nu$  is also an engineer's choice.

Here,  $F$  stands for the neglected parts of the system. It can be determined by the knowledge of  $u$ ,  $\alpha$  and  $y$ . An estimate of  $F$  is obtained as follows:

$$\hat{F} = \hat{y}^{(\nu)} - \alpha \tilde{u}, \quad (4)$$

where  $\hat{y}^{(\nu)}$  is an estimate of the  $\nu^{\text{th}}$  derivative of the measure  $y$  which is assumed available, and  $\tilde{u}$  is an approximate value of  $u$ . Among the existing possibilities,  $\tilde{u}$  can be chosen as a past value of the control variable  $u$ .

The resulting controller is then defined as:

$$u = \frac{1}{\alpha} \left( y_r^{(\nu)} - \hat{F} + \Lambda(\mathbf{e}) \right), \quad (5)$$

where  $y_r$  is a reference trajectory. The variable  $\mathbf{e} = y_r - y$  is the tracking error and  $\Lambda$  is an appropriate function such that the closed loop error dynamics  $e^{(\nu)} = \Lambda(\mathbf{e})$  is asymptotically stable.

From above, we can see that the derivation order  $\nu$  is not necessarily equal to the derivation order  $a$  of  $y$  in eq. (1). The derivation order  $\nu$  is often taken equal to 1 or 2.

The estimate of  $y^{(\nu)}$  in (4) can be obtained for example through a cascade of first order filter as:

$$\mathcal{L}(\hat{y}) = \frac{s}{1 + T_f s} \mathcal{L}(y). \quad (6)$$

Typically,  $1/T_f$  ranges from 8 to 20, and  $\mathcal{L}$  denotes the transformation to the operational domain.

Here, we choose the function  $\Lambda(\mathbf{e})$  as PID controller. The desired behavior is obtained by implementing the so-called intelligent PID controller (for instance  $\nu = 2$ ):

$$u = -\frac{\hat{F}}{\alpha} + \frac{\dot{y}_r}{\alpha} + K_P e + K_I \int e + K_D \dot{e}, \quad (7)$$

where  $K_P$ ,  $K_I$ ,  $K_D$  are the usual tuning gains.

The basic Ārzén's event based controller consists of two parts: a time triggered event detector  $\mathcal{C}_t$  and an event triggered PID controller  $\mathcal{C}_e$  [9]. The latter computes the control signal to be delivered to the actuators. The former  $\mathcal{C}_t$  runs at a fixed sampling period  $h_e$ , and upon fulfillment of a certain event triggering law  $L_e$ , sends events to  $\mathcal{C}_e$ . Upon reception of the event,  $\mathcal{C}_e$  computes the control signal and sends it to the actuators.

Usual event triggering laws  $L_e$  include:

(1) Error threshold law:

$$|e(t_k)| > e_{lim}, \quad (8)$$

where  $e = y_r - y$  is the tracking error,  $t_k$  is the current discrete sensing time by  $\mathcal{C}_e$ , and  $e_{lim}$  is a fixed limit.

(2) Error difference threshold:

$$|e(t_k) - e(t_{k-1})| > e_{lim}. \quad (9)$$

(3) ISS based law:

$$e(t_k) = \sigma \frac{a}{b} |y(t_k)|, \quad (10)$$

assuming the system can be rendered ISS (Input to State Stable) through static feedback [12]. Here,  $\sigma$  is chosen less than one to ensure an associated Lyapounov function decreases,  $a$  and  $b$  are chosen according to the Lipschitz constants of  $\mathcal{K}_\infty$  [12]).

In order to ensure the stability of the system, a maximum sampling period  $h_M$  is defined in [10]. The time interval between two events must then be smaller than  $h_M$ :

$$t_k - t_{k-1} < h_M. \quad (11)$$

Other conditions ensuring the stability of the system have also proposed, such as the forgetting factor used in event driven PID control [11].

### III. THE QUADROTOR MODEL: SCENARIO AND CONTROL

#### A. Scenario

The proposed scenario is a photo shooting at an outdoor garden. An autonomous quadrotor with limited energy is used. The quadrotor needs to follow a square path with length of 2m while hovering at the altitude of 10m, which is given in Figure 1. At each corner, the quadrotor will hover about 15s to take photos. The total simulation time is 150s. The reference trajectory is expressed as:

$$\sigma(t) = \begin{cases} 0 & 0 \leq t \leq t_1, \\ h_d \frac{(t-t_1)^5}{(t-t_1)^5 + (T_f - t + t_1)^5} & t_1 < t \leq t_2 \\ 2 & t_2 < t \leq t_3 \\ h_d - h_d \frac{(t-t_3)^5}{(t-t_3)^5 + (T_f - t + t_3)^5} & t_3 < t \leq t_4 \\ 0 & t_4 < t \leq 150s \end{cases}$$

$$h_d = 2\text{m}, \quad T_f = 6\text{s}.$$

$$x = \sigma(t), \quad \text{with } t_1 = 10\text{s}, t_2 = 16\text{s}, t_3 = 90\text{s}, t_4 = 96\text{s}.$$

$$y = \sigma(t), \quad \text{with } t_1 = 40\text{s}, t_2 = 46\text{s}, t_3 = 120\text{s}, t_4 = 126\text{s}.$$

$$z = 10\text{m}.$$

(12)

As the shooting takes place at an outdoor garden, there may

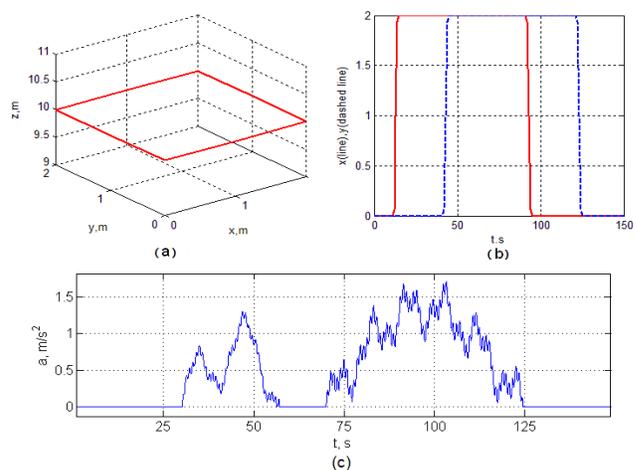


Fig. 1. (a) and (b) Reference trajectory for the quadrotor. (c) The wind disturbance w.r.t time.

be wind during the shooting. The wind is represented as the extra acceleration and affects all  $x$ ,  $y$  and  $z$  axis, which is depicted in Figure 1.

$$a(t) = \begin{cases} 0 & 0 \leq t \leq 30, \\ 0.8 \sin\left(\frac{\pi(t-30)}{31}\right) + 0.056 \sin\left(\frac{24\pi(t-30)}{11}\right) & 30 < t \leq 57, \\ +0.4 \sin\left(\frac{\pi(t-30)}{7}\right) + 0.08 \sin\left(\frac{\pi(t-30)}{2}\right) & 57 < t \leq 70, \\ 0 & 70 < t \leq 124, \\ 1.35 \sin\left(\frac{\pi(t-70)}{55}\right) + 0.105 \sin\left(\frac{24\pi(t-70)}{11}\right) & 124 \leq t \leq 150, \\ +0.15 \sin\left(\frac{\pi(t-70)}{2}\right) + 0.225 \sin\left(\frac{\pi(t-70)}{5}\right) & \\ 0 & \end{cases} \quad (13)$$

### B. Model

The model of the quadrotor is a six d.o.f. system with twelve states and four inputs which is depicted in eq. (14). The notations  $c$  and  $s$  represent  $\cos$  and  $\sin$  respectively. The rotation angles  $\phi$ ,  $\theta$  and  $\psi$  are along the  $x$ ,  $y$  and  $z$  axis respectively, namely roll, pitch and yaw. The parameters in the system can be found in the footnote.<sup>1</sup>

$$\begin{aligned} I_{xx}\ddot{\phi} &= \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(-T_2 + T_4) + (-1)^{i+1} \sum_{i=1}^4 R_{mxi}, \\ I_{yy}\ddot{\theta} &= \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + l(T_1 - T_3) + (-1)^{i+1} \sum_{i=1}^4 R_{myi}, \\ I_{zz}\ddot{\psi} &= \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + (-1)^i \sum_{i=1}^4 Q_i \\ m\ddot{z} &= s\theta \sum_{i=1}^4 H_{xi} - s\phi c\theta \sum_{i=1}^4 H_{yi} + c\phi c\theta \sum_{i=1}^4 T_i - mg + \rho g V_{vol}, \\ m\ddot{x} &= -c\theta c\psi \sum_{i=1}^4 H_{xi} - (s\phi s\theta c\psi - c\phi s\psi) \sum_{i=1}^4 H_{yi}, \\ m\ddot{y} &= -c\theta s\psi \sum_{i=1}^4 H_{xi} - (s\phi s\theta s\psi + c\phi c\psi) \sum_{i=1}^4 H_{yi}. \end{aligned} \quad (14)$$

Here,  $T_i$  and  $H_i$  are the thrusts and hub forces of each motor;  $Q_i$  and  $R_i$  are the drag and rolling moments;  $\omega_i$  are the rotational speeds of the four motors.

$$\begin{aligned} T_i &= C_T \rho A \omega_i^2 R_{rad}^2, & H_i &= C_H \rho A \omega_i^2 R_{rad}^2, \\ Q_i &= C_Q \rho A \omega_i^2 R_{rad}^3, & R_i &= C_R \rho A \omega_i^2 R_{rad}^3, \quad i = 1, \dots, 4. \end{aligned}$$

In eq. (14),  $\dot{\phi}\dot{\psi}(I_{zz} - I_{xx})$ ,  $\dot{\theta}\dot{\psi}(I_{yy} - I_{zz})$ ,  $\dot{\theta}\dot{\phi}(I_{xx} - I_{yy})$  are the body gyro effect moments;  $J_r\dot{\theta}\Omega_r$ ,  $J_r\dot{\phi}\Omega_r$  ( $\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$ ) are the propeller gyro effect moments;  $(-1)^{i+1} \sum_{i=1}^4 R_{mxi}$ ,  $(-1)^{i+1} \sum_{i=1}^4 R_{myi}$  are the rolling moments due to the sideward flight;  $(-1)^i \sum_{i=1}^4 Q_i$  is the unbalanced counter torque;  $\rho g V_{vol}$  is Archimedes' force;

<sup>1</sup> $I_{xx}, I_{yy} = 6.228 \times 10^{-3} \text{kg} \cdot \text{m}^2$ ,  $I_{zz} = 1.121 \times 10^{-2} \text{kg} \cdot \text{m}^2$ ,  $J_r = 6.01 \times 10^{-5} \text{kg} \cdot \text{m}^2$ ,  $l = 0.232 \text{m}$ ,  $h = 0.058 \text{m}$ ,  $m = 0.53 \text{kg}$ ,  $C_x, C_y, C_z = 1.32$ ,  $A_c = 0.005 \text{m}^2$ ,  $\rho = 1.293 \text{kg/m}^3$ ,  $b = 3.13 \times 10^{-5} \text{N} \cdot \text{s}^2$ ,  $d = 7.5 \times 10^{-7} \text{N} \cdot \text{s}^2$ ,  $A = 0.0707 \text{m}^2$ ,  $V_{vol} = 3.04 \times 10^{-4}$ .

The system is controlled by the rotational speeds of the four motors  $\omega_i$ . In the control system, we define

$$\begin{aligned} u_1 &= \sum_{i=1}^4 T_i, & u_2 &= l(T_2 - T_4), \\ u_3 &= l(-T_1 + T_3), & u_4 &= (-1)^{i+1} \sum_{i=1}^4 Q_i. \end{aligned} \quad (15)$$

Then, the rotational speeds  $\omega_i$  can be computed using  $u_i$  through:

$$\begin{aligned} u_1 &= b \sum_{i=1}^4 \omega_i^2, & u_2 &= bl(-\omega_2^2 + \omega_4^2), \\ u_3 &= bl(\omega_1^2 - \omega_3^2), & u_4 &= (-1)^{i+1} d \sum_{i=1}^4 \omega_i^2. \end{aligned} \quad (16)$$

Further details about the model can be found in [13].

### C. Model free control

Firstly, we control the altitude  $z$ . We rewrite the vertical dynamics in (14) as:

$$m\ddot{z} = (c\theta c\phi)u_1 + F_z. \quad (17)$$

where  $F_z$  includes the neglected vertical dynamics in eq. (14). In discrete time, the unknown part  $F_z$  can be expressed as following, where  $\hat{z}(k)$  is an estimate of  $z(k)$ :

$$\hat{F}_z = m\hat{z}(t_k) - (c\theta c\phi)u_1(t_{k-1}). \quad (18)$$

Thus, the chosen control law is:

$$\begin{aligned} u_1(t_k) &= u_1(t_{k-1}) + \frac{m}{c\theta c\phi} \left( \hat{e}_{2d}^z(t_k) + k_1^z e_d^z(t_k) + k_0^z e^z(t_k) \right), \\ \hat{e}_{2d}^z(t_k) &= \ddot{z}_r(t_k) - \hat{\ddot{z}}(t_k), & e_d^z(t_k) &= \dot{z}_r(t_k) - \hat{\dot{z}}(t_k), \\ e^z(t_k) &= z_r(t_k) - z(t_k), \\ \hat{\dot{z}}(t_k) &= \frac{T_f}{T_f + h} \hat{\dot{z}}(t_{k-1}) + \frac{1}{T_f + h} \left( \dot{z}(t_k) - \dot{z}(t_{k-1}) \right), \\ \hat{\ddot{z}}(t_k) &= \frac{T_f}{T_f + h} \hat{\ddot{z}}(t_{k-1}) + \frac{1}{T_f + h} \left( \ddot{z}(t_k) - \ddot{z}(t_{k-1}) \right), \end{aligned} \quad (19)$$

where  $\ddot{z}_r$ ,  $\dot{z}_r$ ,  $z_r$  are the reference acceleration, velocity and position of  $z$ . The variable  $h$  is the sampling period,  $h = t_k - t_{k-1}$ .

Then we control the position  $x$  and  $y$ . As the input  $u_1$  is already used in the control of the altitude  $z$ , we now use  $u_2$  and  $u_3$  to control the positions  $x$  and  $y$ . Therefore, we need to differentiate twice the equations related to  $x$  and  $y$  in eq. (14) in order to get the control inputs  $u_2$  and  $u_3$ . Then, we obtain:

$$\begin{aligned} x^{(4)} &= \frac{u_1}{mI_{xx}} (s\psi c\phi - c\psi s\theta s\phi)u_2 + \frac{u_1}{mI_{yy}} (c\psi c\theta c\phi)u_3 + F_x, \\ y^{(4)} &= -\frac{u_1}{mI_{xx}} (c\psi c\phi + s\psi s\theta s\phi)u_2 + \frac{u_1}{mI_{yy}} (s\psi c\theta c\phi)u_3 + F_y, \end{aligned} \quad (20)$$

where  $F_x, F_y$  are the remaining parts of the horizontal and lateral system. For further simplicity, here we define

$A = \frac{u_1}{mI_{xx}}(s\psi c\phi - c\psi s\theta s\phi)$ ,  $B = \frac{u_1}{mI_{yy}}(c\psi c\theta c\phi)$ ,  $C = -\frac{u_1}{mI_{xx}}(c\psi c\phi + s\psi s\theta s\phi)$  and  $D = \frac{u_1}{mI_{yy}}(s\psi c\theta c\phi)$ .

We implement the model free control scheme in a similar manner as previous:

$$\begin{pmatrix} u_2(t_k) \\ u_3(t_k) \end{pmatrix} = \begin{pmatrix} u_2(t_{k-1}) \\ u_3(t_{k-1}) \end{pmatrix} + \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} \begin{pmatrix} \hat{e}_{4d}^x + \sum k_i^x e_{id}^x \\ \hat{e}_{4d}^y + \sum k_i^y e_{id}^y \end{pmatrix}, \quad (21)$$

where  $\hat{e}_{4d}^x, \hat{e}_{4d}^y$  are the errors between the references  $x_r^{(4)}, y_r^{(4)}$  and the estimates of  $x^{(4)}, y^{(4)}$ ,  $i = 0, \dots, 3$ .

For the yaw control, we consider the equation of  $\psi$  as:

$$I_{zz}\ddot{\psi} = u_4 + F_\psi. \quad (22)$$

Then the needed control law is:

$$u_4(t_k) = u_4(t_{k-1}) + I_{zz} \left( \hat{e}_{2d}^\psi(t_k) + k_1^\psi e_d^\psi(t_k) + k_0^\psi e^\psi(t_k) \right), \quad (23)$$

where  $\hat{e}_{2d}^\psi$  is the error between the reference  $\ddot{\psi}_r$  and the estimate of  $\ddot{\psi}$ .

#### D. Backstepping control

For the purpose of comparison, a backstepping control proposed by S. Bouabdallah et al. [2] is also used on the quadrotor system. In order to simplify the control laws, some parts in the model in eq. (14) are neglected, such as the rolling moments and the hub forces. The system is written into the state space form using the state vector  $(x_1, \dots, x_{12})$  with  $x_1 = \phi, x_2 = \dot{\phi}, x_3 = \theta, x_4 = \dot{\theta}, x_5 = \psi, x_6 = \dot{\psi}, x_7 = z, x_8 = \dot{z}, x_9 = x, x_{10} = \dot{x}, x_{11} = y, x_{12} = \dot{y}$ .

The system is separated into the angular and position subsystems. The angular subsystem is thus firstly controlled, and then the position subsystem is controlled by using the angles from the angular subsystem.

In the angle subsystem, the control laws are defined as:

$$\begin{aligned} u_2 &= \frac{1}{b_1} \left( z_1 - a_1 x_4 x_6 - a_2 x_4 \omega - \alpha_1 (z_2 + \alpha_1 z_1) - \alpha_2 z_2 \right), \\ u_3 &= \frac{1}{b_2} \left( z_3 - a_3 x_2 x_6 - a_4 x_2 \omega - \alpha_3 (z_4 + \alpha_3 z_3) - \alpha_4 z_4 \right), \\ u_4 &= \frac{1}{b_3} \left( z_5 - a_5 x_2 x_4 - \alpha_5 (z_6 + \alpha_5 z_5) - \alpha_6 z_6 \right), \end{aligned} \quad (24)$$

with  $z_1 = x_{1d} - x_1, z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1, z_3 = x_{3d} - x_3, z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3, z_5 = x_{5d} - x_5, z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5$  and  $a_1 = (I_{yy} - I_{zz})/I_{xx}, a_2 = J_r/I_{xx}, a_3 = (I_{zz} - I_{xx})/I_{yy}, a_4 = J_r/I_{yy}, a_5 = (I_{xx} - I_{yy})/I_{zz}, b_1 = 1/I_{xx}, b_2 = 1/I_{yy}, b_3 = 1/I_{zz}$ . All the  $\alpha_i (i = 1, \dots, 12)$  are the control gains.

In the position subsystem, the control laws are defined as:

$$\begin{aligned} u_1 &= \frac{m}{cx_1 cx_3} \left( z_7 + g - \alpha_7 (z_8 + \alpha_7 z_7) - \alpha_8 z_8 \right), \\ u_x &= \frac{m}{u_1} \left( z_9 - \alpha_9 (z_{10} + \alpha_9 z_9) - \alpha_{10} z_{10} \right), \\ u_y &= \frac{m}{u_1} \left( z_{11} - \alpha_{11} (z_{12} + \alpha_{11} z_{11}) - \alpha_{12} z_{12} \right), \end{aligned} \quad (25)$$

with  $z_7 = x_{7d} - x_7, z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7, z_9 = x_{9d} - x_9, z_{10} = x_{10} - \dot{x}_{9d} - \alpha_9 z_9, z_{11} = x_{11d} - x_{11}, z_{12} = x_{12} - \dot{x}_{11d} - \alpha_{11} z_{11}$  and  $u_x = s\psi s\phi + c\psi s\theta c\phi, u_y = -c\psi s\phi + s\psi s\theta c\phi$ .

#### E. Sliding mode control

For comparison, a sliding mode control proposed by S. Bouabdallah et al. [2] is also proposed. The state variables are defined in the backstepping control. The sliding surfaces are chosen as:

$$\begin{aligned} S_\phi &= z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1, & z_1 &= x_{1d} - x_1, \\ S_\theta &= z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3, & z_3 &= x_{3d} - x_3, \\ S_\psi &= z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5, & z_5 &= x_{5d} - x_5, \\ S_x &= z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7, & z_7 &= x_{7d} - x_7, \\ S_y &= z_{10} = x_{10} - \dot{x}_{9d} - \alpha_9 z_9, & z_9 &= x_{9d} - x_9, \\ S_z &= z_{12} = x_{12} - \dot{x}_{11d} - \alpha_{11} z_{11}, & z_{11} &= x_{11d} - x_{11}, \end{aligned} \quad (26)$$

where  $\dot{x}_{id} (i = 1, 3, 4, 5, 7, 9, 11)$  are the references. The control laws are:

$$\begin{aligned} u_2 &= \frac{1}{b_1} \left( -k_1 \text{sign}(S_\phi) - k_2 S_\phi - a_1 x_4 x_6 - a_2 x_4 \omega + \ddot{\phi}_d - \alpha_1^2 z_1 \right), \\ u_3 &= \frac{1}{b_2} \left( -k_3 \text{sign}(S_\theta) - k_4 S_\theta - a_3 x_2 x_6 - a_4 x_2 \omega + \ddot{\theta}_d - \alpha_2^2 z_3 \right), \\ u_4 &= \frac{1}{b_3} \left( -k_5 \text{sign}(S_\psi) - k_6 S_\psi - a_5 x_2 x_4 + \ddot{\psi}_d - \alpha_3^2 z_5 \right), \\ u_1 &= \frac{m}{cx_1 cx_3} \left( -k_7 \text{sign}(S_z) - k_8 S_z + g + \ddot{z}_d - \alpha_7^2 z_7 \right), \\ u_x &= \frac{m}{u_1} \left( -k_9 \text{sign}(S_x) - k_{10} S_x + \ddot{x}_d - \alpha_9^2 z_9 \right), \\ u_y &= \frac{m}{u_1} \left( -k_{11} \text{sign}(S_y) - k_{12} S_y + \ddot{y}_d - \alpha_{11}^2 z_{11} \right). \end{aligned} \quad (27)$$

The definition of all the parameters can be found in the previous backstepping control section III-D.

## IV. COMPARISON OF THE CONTROL LAWS

### A. Basic scenario: time and event triggered schemes

In the time triggered scheme, the sampling period is 10ms, and it yields 15000 actuation steps. The simulation results of the three control methods are given in Figure 2. Three methods have followed the reference trajectory nicely. The maximum absolute tracking errors are 0.042m, 0.09m and 0.08m for the model free, backstepping and sliding mode control respectively, which are 2.1%, 4.5% and 4% of the desired length 2m.

In the event triggered scheme, the chosen event triggering law is the error difference threshold as in eq. (9). Here, we set the error difference threshold of  $z$  0.02m, yaw angle 0.1rad,  $x$  and  $y$  0.001m. Every 10ms, the control system verifies the tracking error differences, and decides the corresponding actions. The simulation results are shown in Figure 2. The event triggered model free control, backstepping control and sliding mode control yield 9008, 9987 and 8493 actuation steps, which are 60%, 66.6% and 56.6% of 15000 actuation steps in the time driven scheme. The maximum absolute tracking errors are 0.06m, 0.1m and 0.1m in the three control methods, which increased 0.43%, 0.11% and 0.25% with respect to their results in the time triggered scheme.

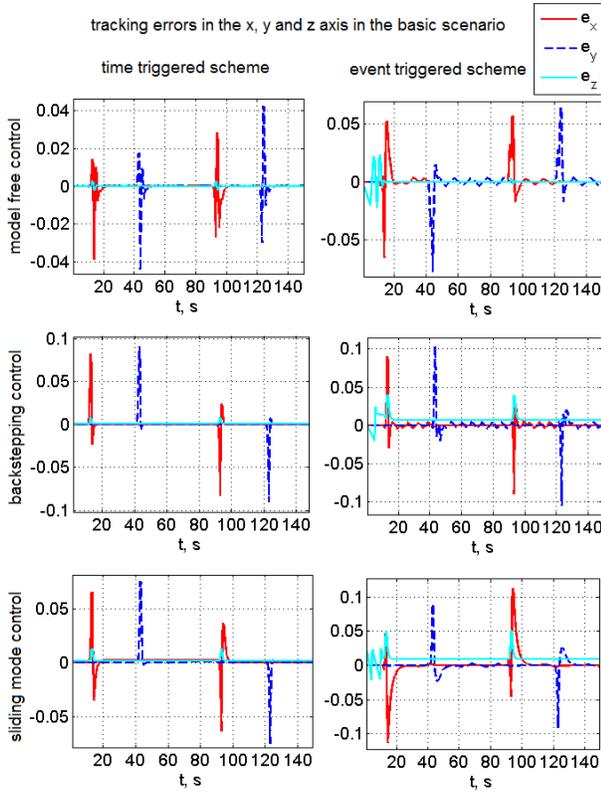


Fig. 2. The tracking errors along the x, y and z axis in the basic scenario. The first row: the model free control. The second row: the backstepping control. The third row: the sliding mode control. The first column: the time triggered scheme. The second column: the event triggered scheme.

### B. Scenario with wind disturbance: time and event triggered schemes

All the control methods are simulated in the scenario with wind disturbance. The simulation results are in Figure 3. The chosen event triggering law is the same as in the basic scenario.

In the time triggered scheme, the maximum absolute tracking errors are 0.042m, 0.23m and 0.21m in three control methods, which increased 0%, 155.6% and 162.5% with respect to the time triggered controls in the basic scenario. The wind does not have great influence in the model free control, however it has highly affected the backstepping and sliding mode controls.

In the event triggered scheme, the model free, backstepping and sliding mode controls yield 12110, 12116 and 12445 actuation steps, which are 80.7%, 80.8% and 83.0% of 15000 actuation steps. The maximum absolute tracking errors are 0.11m, 0.22m and 1.25m respectively in three control methods, which increased 83.3%, 120% and 1150% with respect to the event triggered controls in the basic scenario.

### C. Discussion

In order to get a comprehensive evaluation, all the methods are implemented in other realistic scenarios with parameter uncertainties, with sensor noises and with actuator faults.

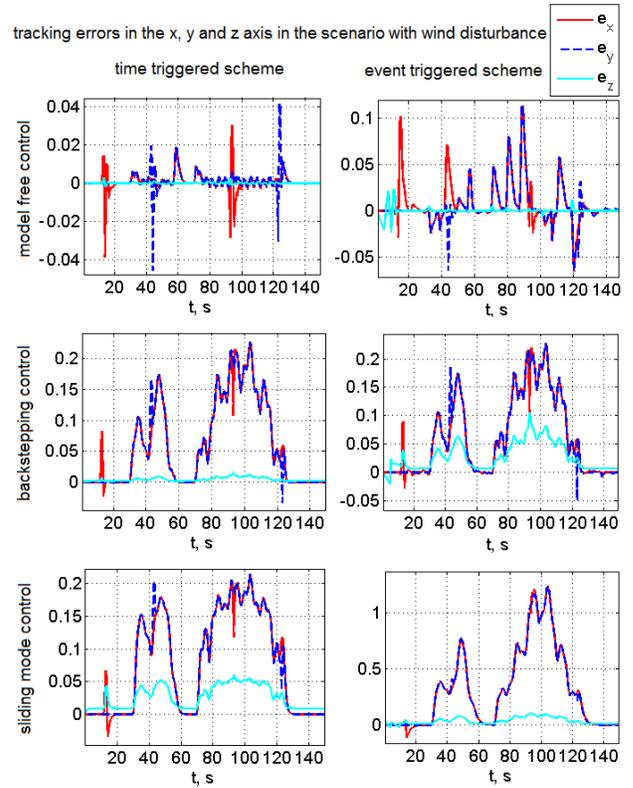


Fig. 3. The tracking errors along the x, y and z axis in the scenario with wind disturbance. The first row: the model free control. The second row: the backstepping control. The third row: the sliding mode control. The first column: the time triggered scheme. The second column: the event triggered scheme.

Further details can be found in [13]. The maximum absolute tracking errors, the sum of the error variances and the actuation steps of all the control methods in all scenarios can be found in Table I and Figure 4.

In different realistic scenarios, the model free control has the smallest tracking errors than other control methods, and the event triggered model free control has the smallest errors than other event triggered methods. In the model free control, the part  $F$  is evaluated at each actuation step using the system measurement  $\hat{y}$  and the last time input  $\tilde{u}$ . The model free control avoids the time-consuming computation of the full control model and has an algorithm complexity  $O(5n^2+3n)$ , while the backstepping and sliding mode controls have to compute the full model and have an algorithm complexity  $O(6n^2+4n)$ . The disturbances and uncertainties are also considered into  $F$  in the real time control, and that is why the model free control compensate them well.

The event triggered scheme matches the model free control. While the maximum tracking errors are bounded in desirable limits, the actuation steps are reduced more than one third. The event triggered model free control works well in the realistic scenarios with disturbances. The sliding mode control does not match with the event triggered scheme. The tracking errors are too high to be accepted in the scenarios with disturbances.

TABLE I

THE MAXIMUM ABSOLUTE TRACKING ERRORS, THE SUM OF THE ERROR VARIANCES AND THE ACTUATION STEPS OF ALL THE CONTROL METHODS IN ALL SCENARIOS. MF: MODEL FREE CONTROL. BS: BACKSTEPPING CONTROL. SM: SLIDING MODE CONTROL.

max absolute errors	time triggered			event triggered		
	MF	BS	SM	MF	BS	SM
basic scenario	0,042	0,09	0,08	0,06	0,1	0,1
wind	0,042	0,23	0,21	0,11	0,22	1,25
parameter uncertainties	0,075	0,11	0,11	0,19	0,22	0,41
sensor noise	0,08	0,09	0,08	0,08	0,09	0,08
actuator faults	0,04	0,06	0,07	0,13	0,2	0,35
error variance						
basic scenario	0,0049	0,0028	0,0058	0,2	0,2	0,077
wind	0,0052	2,4723	3,5238	0,0635	2,6679	65,4849
parameter uncertainties	0,0212	0,0468	0,2805	0,1281	1,8675	3,5517
sensor noise	0,2309	0,0029	0,0088	0,2309	0,0029	0,0088
actuator faults	0,0327	0,0167	0,0188	0,0918	0,0661	0,554
actuation steps						
basic scenario	15000			9008	9987	8493
wind	15000			12110	12116	12445
parameter uncertainties	15000			10803	14976	14983
sensor noise	15000			15000	15000	15000
actuator faults	15000			9298	10020	8297

## V. CONCLUSION

In this paper, event driven model free controllers have been proposed and applied on quadrotor system in different realistic scenarios. The proposed method avoids the time-consuming computation of the full control model by evaluating the part  $F$  at each actuation step using the system measurement  $\hat{y}$  and the last time input  $\hat{u}$ . The part  $F$  can also take the disturbances and uncertainties into account in the real time control, therefore the model free control has a better performance in the realistic scenarios with disturbances.

Under the event driven scheme, the model free control has smaller tracking errors and lower actuation steps than backstepping and sliding mode control. In the realistic scenarios with disturbances and uncertainties, the event driven model free control has achieved almost the same results as in the scenarios without disturbance, which proves its robustness to perturbations. In the same scenarios, the backstepping and sliding mode controls have higher tracking errors.

The proposed method gives promising results in terms of control algorithm complexity reduction, computational resources reduction and robustness to perturbations, which is appropriate for implementation in embedded systems.

## REFERENCES

[1] P. Castillo, A. Dzul and R. Lozano. *Real-time stabilization and tracking of a four-rotor mini rotorcraft*. IEEE Transactions on Control Systems Technology, vol. 12, no 4, p. 510-516, 2004.

[2] S. Bouabdallah, *Design and control of quadrotors with application to autonomous flying*. PhD thesis, Ecole Polytechnique Federale De Lausanne, 2007.

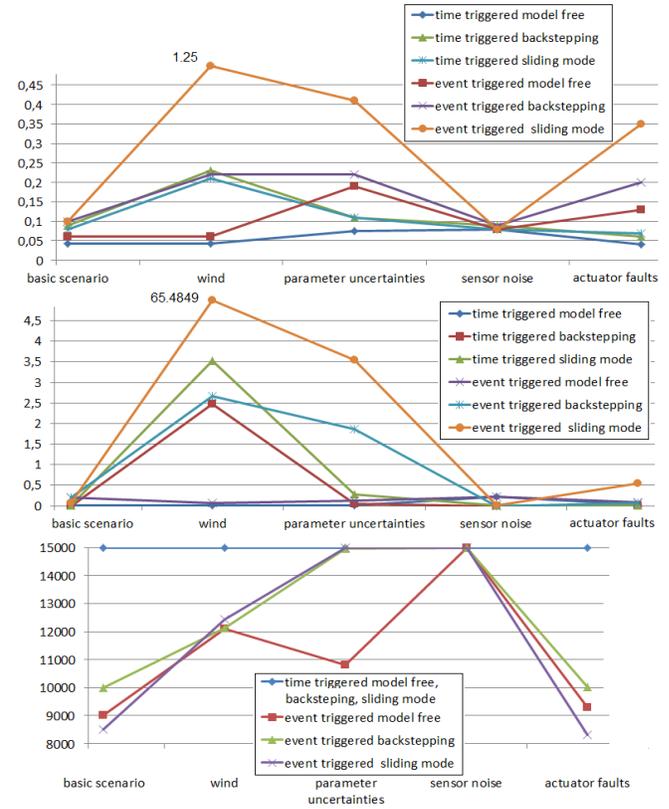


Fig. 4. The First figure: the maximum absolute tracking error. The second figure: the sum of the error variances  $\times 10^7$ . The third figure: the actuation steps.

[3] V. Mistler, A. Benallegue and N.K. M'Sirdi. *Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback*. 10th IEEE International Workshop on Robot and Human Interactive Communication, p. 586-593, 2001.

[4] J. Wang, H. Mounier, A.Cela and S.I. Niclescu, *Event driven intelligent PID controllers with applications to motion control*. Proc. of 18th IFAC World Congress, Milan, Italy, 2011.

[5] M. Fliess, C. Join and H. Sira-Ramirez, *Complex continuous nonlinear systems: their black box identification and their control*. Proc. of 14th IFAC Symp. System Identif. (SYSID-2006), Newcastle, Australia, 2006.

[6] B. d'Andréa-Novet, M. Fliess, C. Join, H. Mounier and B. Steux, *A mathematical explanation of "intelligent" PID controllers of the strange ubiquity of PIDs*. Proc. of 18th Mediterranean Conference on Control and Automation, Marrakech, Morocco, 2010.

[7] L. Villagra and C. Balaguer, *A model-free approach for accurate joint motion control in humanoid locomotion*. International Journal of Humanoid Robotics, Vol. 8, No. 1 (2011).

[8] S. Andary and A. Chemori, *A dual model-free control of non-minimum phase systems for generation of stable limit cycles*. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, December 12-15, 2011.

[9] K.E. Årzén, *A simple event-based PID controller*. Proc. of 14th IFAC World Congress, Beijing, PR China, 1999.

[10] K.E. Årzén and B. Bernhardsson, *Comparison of Riemann and Lebesgue sampling for first order stochastic systems*. Proc. of 41st IEEE Conference on Decision and Control, 2002.

[11] S. Durand and N. Marchand, *Further results on event-based PID controller*. Proc. of the European Control Conference, Hongrie, 2009.

[12] E.D. Sontag, *Input to state stability: basic concepts and results*. Nonlinear and Optimal Control Theory, pp. 163-220, Springer-Verlag, Berlin, 2007.

[13] J. Wang, *Analysis and control of quadrotor*. PhD thesis (in preparation), Laboratory of signals and systems (L2S), CNRS, Supélec, University of Paris 11, 2013.



# Bibliography

- [1] K. H. Ang, G. Chong, Y. Li, *PID Control System Analysis, Design, and Technology*. IEEE Trans. on control systems technology, Vol. 13, No.4, July 2005.
- [2] Ascending Technologies. [www.asctec.de](http://www.asctec.de).
- [3] AR. Drone. [www.ardrone2.parrot.com](http://www.ardrone2.parrot.com).
- [4] E. Altug, J. P. Ostrowski, R. Mahony. Control of a quadrotor helicopter using visual feedback. *IEEE International Conference on Robotics and Automation*, p. 72-77, 2002.
- [5] E. Altug, J. P. Ostrowski, C. J. Taylor. Control of a quadrotor helicopter using dual camera visual feedback. *The International Journal of Robotics Research*, 24(5), 329-341, 2005.
- [6] K. Alexis, G. Nikolakopoulos, A. Tzes. Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10), 1195-1207, 2011.
- [7] K.E. Årzén, *A simple event-based PID controller*. Proc. of 14th IFAC World Congress, Beijing, PR China, 1999.
- [8] K.J. Astrom and B. Bernhardsson, *Comparison of Riemann and Lebesgue sampling for first order stochastic systems*. Proc. of 41st IEEE Conference on Decision and Control, 2002.
- [9] A. Albert, *Comparison of event-triggered and time-triggered concepts with regard to distributed control systems*. Embedded World, 235-252, 2004.
- [10] B. d'Andréa-Novel, M. Fliess, C. Join, H. Mounier and B. Steux, *A mathematical explanation via "intelligent" PID controllers of the strange ubiquity of PIDs*. Proc. of 18th Mediterranean Conference on Control and Automation, Marrakech, Morocco, 2010.
- [11] S. Andary and A. Chemori, *A dual model-free control of non-minimum phase systems for generation of stable limit cycles*. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, December 12-15, 2011.

- [12] Arducopter. [code.google.com/p/arducopter](http://code.google.com/p/arducopter).
- [13] Aeroquad. [www.Aeroquad.com](http://www.Aeroquad.com).
- [14] Breguet-Richet Gyroplane No.1. [www.aviastar.org/breguet\\_gyro.php](http://www.aviastar.org/breguet_gyro.php).
- [15] P. J. Bristeau, F. Callou, D. Vissière, N. Petit. The Navigation and Control technology inside the AR. Drone micro UAV. *18th IFAC world congress*, p. 1477-1484, Milan, August 2011.
- [16] M. Bangura, R. Mahony. Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor. *Proceedings of Australasian Conference on Robotics and Automation*, 3-5 Dec 2012.
- [17] S. Bouabdallah, P. Murrieri, R. Siegwart. Design and control of an indoor micro quadrotor. *IEEE International Conference on Robotics and Automation*, (Vol. 5, pp. 4393-4398), 2004.
- [18] S. Bouabdallah, R. Siegwart. Full control of a quadrotor. *IEEE/RSJ international conference on Intelligent robots and systems*, (pp. 153-158), 2007.
- [19] S. Bouabdallah, M. Becker, R. Siegwart. Autonomous miniature flying robots: coming soon!-research, development, and results. *Robotics & Automation Magazine*, 88-98., 2007.
- [20] S. Bouabdallah. Design and control of quadrotors with application to autonomous flying. *PhD thesis*, Ecole Polytechnique Federale De Lausanne, 2007.
- [21] S. Bouabdallah, A. Noth, R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. *In Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2451-2456. IEEE, 2004.
- [22] S. Bouabdallah, R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. *In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2247-2252. IEEE, 2005.
- [23] A. Benallegue, A. Mokhtari, L. Fridman. Feedback linearization and high order sliding mode observer for a quadrotor UAV. *International Workshop on Variable Structure Systems*, (pp. 365-372), 2006.
- [24] A. Benallegue, A. Mokhtari, L. Fridman. High order sliding mode observer for a quadrotor UAV. *International Journal of Robust and Nonlinear Control*, 18(4-5), 427-440, 2008.
- [25] L. Besnard, Y.B. Shtessel, B. Landrum. Control of a quadrotor vehicle using sliding mode disturbance observer. *American Control Conference*, 2007.
- [26] L. Besnard, Y.B. Shtessel, B. Landrum. Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. *Journal of the Franklin Institute*, 349(2), 658-684, 2012.

- [27] A. R. Bramwell, D. Balmford, G. Done. Bramwell's helicopter dynamics. Butterworth-Heinemann, 2001.
- [28] S. Bertrand, N. Guenard, T. Hamel, H. Piet-Lahanier, L. Eck. A hierarchical controller for miniature VTOL UAVs: design and stability analysis using singular perturbation theory. *Control Engineering Practice*, 19(10), 1099-1108, 2011.
- [29] Convertawings. [www.aviastar.org/helicopters\\_eng/convertawings.php](http://www.aviastar.org/helicopters_eng/convertawings.php).
- [30] Curtiss-Wright VZ-7. [www.aviastar.org/helicopters\\_eng/curtiss\\_vz-7.php](http://www.aviastar.org/helicopters_eng/curtiss_vz-7.php).
- [31] Curtiss-Wright X-19. [www.aviastar.org/helicopters\\_eng/curtiss\\_x-19.php](http://www.aviastar.org/helicopters_eng/curtiss_x-19.php).
- [32] P. Castillo, A. Dzul, R. Lozano. Real-time stabilization and tracking of a four-rotor mini rotorcraft. *IEEE Transactions on Control Systems Technology*, 12(4), 510-516, 2004.
- [33] P. Castillo, R. Lozano, A. Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, 25(6), 45-55, 2005.
- [34] P. Castillo, P. Albertos, P. Garcia, R. Lozano. Simple real-time attitude stabilization of a quad-rotor aircraft with bounded signals. *45th IEEE Conference on Decision and Control*, p. 1533-1538., 2006.
- [35] P. Castillo, R. Lozano, A. Dzul. Modelling and control of mini-flying machines. *Springer*, 2005.
- [36] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, A. K. Cooke. A prototype of an autonomous controller for a quadrotor UAV. *European Control Conference*, (pp. 1-8), 2007.
- [37] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, A. K. Cooke. Direct method based control system for an autonomous quadrotor. *Journal of Intelligent & Robotic Systems*, 60(2), 285-316, 2010.
- [38] J. Carr. Applications of center manifold theory. New york, Springer Verlag, 1981.
- [39] Draganfly Innovations Inc. [www.draganfly.com](http://www.draganfly.com).
- [40] S. Durand and N. Marchand, *Further results on event-based PID controller*. Proc. of the European Control Conference, Hongrie, 2009.
- [41] *Discrete wind gust model block*. MATLAB Simulink.
- [42] G. Fay. Derivation of the aerodynamic forces for the mesicopter simulation. Stanford University, USA, 2001.
- [43] M. Fliess, C. Join and H. Sira-Ramirez Complex continuous nonlinear systems: their black box identification and their control. *Proc. of 14th IFAC Symp. System Identif. (SYSID-2006)*, Newcastle, Australia, 2006.

- [44] M. Fliess, C. Join. Intelligent PID controllers. *In Control and Automation, 2008 16th Mediterranean Conference on*, (pp. 326-331), 2008.
- [45] M. Fliess, C. Join. Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control?. *arXiv preprint arXiv:0904.0322.*, 2009.
- [46] M. Fliess, C. Join, S. Riachy. Revisiting some practical issues in the implementation of model-free control. *18th IFAC World Congress*, 2011.
- [47] M. Fliess, J. Levine, P. Martin and P. Rouchon. Flatness and defect of nonlinear systems : introductory theory and examples. *Internat. J. Control*, Vol. 61, pp. 1327–1361, 1995.
- [48] D. Gurdan, J. Stumpf, M. Achtelik, K. M. Doth, G. Hirzinger, D. Rus. Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz. *2007 IEEE International Conference on Robotics and Automation*, pp. 361-366. IEEE, 2007.
- [49] N. Guenard, T. Hamel, L. Eck. Control laws for the tele operation of an unmanned aerial vehicle known as an x4-flyer. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
- [50] N. Guenard, T. Hamel, R. Mahony. A practical visual servo control for an unmanned aerial vehicle. *Robotics, IEEE Transactions on*, 24(2), 331-340, 2008.
- [51] W. Gawronski, *Three models of wind-gust disturbances for the analysis of antenna pointing accuracy*. IPN Progress Report (2002): 42-149.
- [52] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, C. J. Tomlin. The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC). *Digital Avionics Systems Conference*, Vol. 2, p. 12., 2004.
- [53] M. H. Hoffmann, H. Huang, S. L. Waslander, C. J. Tomlin. Precision flight control for a multi-vehicle quadrotor helicopter testbed. *Control engineering practice*, 19(9), 1023-1036, 2011.
- [54] M. H. Hoffmann, H. Huang, S. L. Waslander, C. J. Tomlin. Quadrotor helicopter trajectory tracking control. *In AIAA Guidance, Navigation and Control Conference and Exhibit*, (pp. 1-14), 2008.
- [55] M. H. Hoffmann, H. Huang, S. L. Waslander, C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. *In Proc. of the AIAA Guidance, Navigation, and Control Conference*, pp. 1-20. 2007.
- [56] C. Join, G. Robert, M. Fliess. Model-free based water level control for hydroelectric power plants. *IFAC Conference on Control Methodologies and Technologies for Energy Efficiency*, 2010.
- [57] C. Join, F. Chaxel, M. Fliess. Intelligent" controllers on cheap and small programmable devices. *arXiv preprint arXiv:1307.4841*, 2013.

- [58] Y. Kuznetsov. Elements of applied bifurcation theory. New york, Springer Verlag, 1998.
- [59] A. J. Krener. Normal forms for linear and nonlinear systems. *Contemp. Math*, 68, 157-189, 1987.
- [60] A.J. Krener, S. Karahan, M. Hubbard, R. Frezza. Higher order linear approximations to nonlinear control systems. *Proceedings of the 26th IEEE Conference on Decision and Control*, 1987.
- [61] A.J. Krener, W. Kang. Control bifurcations. *IEEE Transactions on Automatic Control*, pp. 1231-1246, Aug 2004 .
- [62] W. Kang, A. J. Krener. Extended quadratic controller normal form and dynamic state feedback linearization of nonlinear systems. *SIAM Journal on Control and Optimization*, 30(6), 1319-1337, 1992.
- [63] W. Kang. Extended controller form and invariants of nonlinear control systems with a single input. *Journal of Mathematical Systems Estimation and control*, 27-51, 1996.
- [64] S.G. Krantz and H.R. Parks, *The Implicit function theorem: history, theory, and applications*. Birkhäuser, Boston, United States, 2002.
- [65] KKmulticopter. [www.KKmulticopter.com](http://www.KKmulticopter.com).
- [66] N. Michael, D. Mellinger, Q. Lindsey, V. Kumar. The grasp multiple micro-uav testbed. *IEEE Robotics & Automation Magazine*, vol. 17, no 3, p. 56-65, 2010.
- [67] D. Mellinger, N. Michael, V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, 31(5), 664-674, 2012.
- [68] D. Mellinger, Q. Lindsey, M. Shomin, V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 2668-2673), 2011.
- [69] V. Mistler, A. Benallegue, N. K. M'Sirdi. Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. *10th IEEE International Workshop on Robot and Human Interactive Communication*, p. 586-593, 2001.
- [70] A. Mokhtari, A. Benallegue. Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. *IEEE International Conference on Robotics and Automation*, (Vol. 3, pp. 2359-2366), 2004.
- [71] A. Mokhtari, A. Benallegue, B. Daachi. Robust feedback linearization and  $\text{GH}_\infty$  controller for a quadrotor unmanned aerial vehicle. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1198-1203, 2005.
- [72] A. Mokhtari, A. Benallegue, Y. Orlov. Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle. *International Journal of Robotics and Automation*, 21(1), 39-49, 2006.

- [73] L. Michel, C. Join, M. Fliess, P. Sicard, A. Cheriti. Model-free control of dc/dc converters. *IEEE 12th Workshop on Control and Modeling for Power Electronics (COMPEL)*, (pp. 1-8), 2010.
- [74] T. Madani, A. Benallegue. Control of a quadrotor mini-helicopter via full state backstepping technique. *45th IEEE Conference on Decision and Control*, 2006.
- [75] Mikrokoetter. [www.Mikrokoetter.de](http://www.Mikrokoetter.de).
- [76] Multiwii. [www.Multiwii.de](http://www.Multiwii.de).
- [77] D. Nesic, A.R. Teel, P.V. Kokotovic, *Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations*. *Systems and Control Letters*, 38(4-5), 1999.
- [78] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules, 2nd ed.* Imperial College Press, 2006.
- [79] Oehmichen No.2. [www.en.wikipedia.org/wiki/Etienne\\_Oehmichen](http://www.en.wikipedia.org/wiki/Etienne_Oehmichen).
- [80] Openpilot. [www.Openpilot.org](http://www.Openpilot.org).
- [81] Paparazzi. [paparazzi.enac.fr](http://paparazzi.enac.fr).
- [82] Pixhawk. [Pixhawk.ethz.ch](http://Pixhawk.ethz.ch).
- [83] P. Pounds. Design, construction and control of a large quadrotor micro air vehicle. *PhD thesis*, Australian National University, 2007.
- [84] P. Pounds, R. Mahony, P. Corke. Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7), 691-699, 2010.
- [85] P. Pounds, R. Mahony, P. Corke. Modelling and control of a quad-rotor robot. *In Proceedings Australasian Conference on Robotics and Automation 2006*, Australian Robotics and Automation Association Inc., 2006.
- [86] O. Purwin, R. D'Andrea. Performing aggressive maneuvers using iterative learning control. *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009.
- [87] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, J. K. Hedrick. An overview of emerging results in cooperative UAV control. *IEEE Conference on Decision and Control*, Vol. 1, pp. 602-607, Atlantis, 2004.
- [88] G. V. Raffo, M. G. Ortega, F. R. Rubio. An integral predictive nonlinear  $H_\infty$  control structure for a quadrotor helicopter. *Automatica*, 46(1), 29-39, 2010.
- [89] G. V. Raffo, M. G. Ortega, F. R. Rubio. Backstepping/nonlinear  $H_\infty$  control for path tracking of a quadrotor unmanned aerial vehicle. *American Control Conference*, (pp. 3356-3361), 2008.

- [90] S. Riachy, M. Fliess, C. Join. High-order sliding modes and intelligent PID controllers: First steps toward a practical comparison. *18th IFAC World Congress*, 2011.
- [91] E.D. Sontag, *Input to state stability: basic concepts and results*. Nonlinear and Optimal Control Theory, pp. 163-220, Springer-Verlag, Berlin, 2007.
- [92] A. Tayebi, S. McGilvray. Attitude stabilization of a four-rotor aerial robot. *43rd IEEE Conference on Decision and Control*, (Vol. 2, pp. 1216-1221), 2004.
- [93] A. Tayebi, S. McGilvray. Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3), 562-571, 2006.
- [94] A. Tayebi. Unit quaternion-based output feedback for the attitude tracking problem. *Automatic Control, IEEE Transactions on*, 53(6), 1516-1520, 2008.
- [95] I.A. Tall, W. Respondek. Normal forms for two-inputs nonlinear control systems. *Proceedings of the 41st IEEE Conference on Decision and Control*, Dec 2002.
- [96] L. Villagra, C. Balaguer, A model-free approach for accurate joint motion control in humanoid locomotion. *International Journal of Humanoid Robotics*, Vol. 8, No. 1, 2011.
- [97] L. Villagra, B. Vinagre, I. Tejado, Data-driven fractional PID control: application to DC motors in flexible joints. *Proceedings of the IFAC conference on advances in PID control PID*,(Vol. 12, pp. 28-30), 2012.
- [98] S. L. Waslander, G. M. Hoffmann, J. S. Jang, C. J. Tomlin. Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning. *IEEE/RSJ International Conference Intelligent Robots and Systems*, p. 3712-3717, 2005.
- [99] X. Wang and M.D. Lemmon, *Event-triggering in distributed networked control systems*. IEEE Trans. on Automatic Control, 56(3), 586-601, 2011.
- [100] J. Wang, H. Mounier, A. Cela, S.I. Niculescu, *Event driven intelligent PID controllers with applications to motion control*. Proc. of 18th IFAC World Congress, Milan, Italy, 2011.
- [101] J. Wang, A. Cela, H. Mounier, S.I. Niculescu, *Analysis and control of quadrotor via a Normal Form approach* In Proc. of 20th International Symposium on Mathematical Theory of Networks and Systems, 2012.
- [102] J. Wang, M.S. Geamanu, A. Cela, H. Mounier, S.I. Niculescu, *Event driven model free control of quadrotor* IEEE Conference on Control Applications (CCA), 2013.