



Fault Tolerance Adaptive Control Allocation for Unmanned Aerial Vehicles

M.T.E. (Mohab Tarek) Abdelbadie

BSc Report

Committee: Dr.ir. J.B.C. Engelen R. Hashem, MSc

August 2018

036RAM2018 Robotics and Mechatronics EE-Math-CS University of Twente P.O. Box 217 7500 AE Enschede The Netherlands

UNIVERSITY OF TWENTE.



BIOMEDICAL TECHNOLOGY AND TECHNICAL MEDICINE

Fault Tolerant Adaptive Control Allocation for Unmanned Aerial Vehicles

Mohab Tarek Elsayed Abdelbadie

August 29, 2018

Abstract

This thesis proposes an adaptive control allocation method to account for occurrence of failure in a fully actuated multi-copter system. The model used in this work is a hexa-rotor tilted in a certain configuration that allows it to be a fully actuated vehicle. Among many ways to account for air-crafts failure, adaptive theory is the one used in this work. The method used here has many advantages, it does not require exact knowledge of the fault, but it simply adds an adaptive term that estimates the effect of the failure in the control allocation part. Another advantage is that it only re-distributes the control effort among available actuators without the need to re-configure the whole controller. Furthermore, it can be applied on any baseline controller as long as it has a Lyapunov function whose derivative is negative definite. Backstepping control for position and attitude tracking was designed as a baseline controller. The approach was applied on it and it showed positive results. Simulations were carried out using 20-sim program to confirm the results given later.

<Mohab Abdelbadie> <The Netherlands, 29th of August 2018>

Acknowledgements

"My Lord, increase me in knowledge."

Quran surah Taha 114 (QS 20: 114)

First and foremost, I would like to thank God for his never-ending grace, mercy and provision during a five-month-journey that was a tough one and taught me a lot. This work is to finalize an intensive studying of engineering in general, control theory specifically. It is my first time to join the research world and I was so lucky to get the chance to work under the supervision of Eng. Ramy Rashad, he taught me a lot through our research period, he helped me know more about control theory and view the science behind it in a much better perspective. Special thanks to him and may we get the chance to work with each other again.

I would like to thank Dr. Johan and RAM group for giving me this great opportunity to be part of this team. Although we did not get the chance to work with each other directly, It was nice having those few discussions with him.

I owe this work to my parents and especially my father for his smart upbringing to us. I shall thank him for the rest of my life. My twin brother, thanks for the countless times you supported and encouraged me. I would like also to thank my friends who were close to me during this period (Khaled, Yaseen, and Mahmoud)

Sincere regards to my dutch athletics coach Jan Heusinkveld who was so supportive and helpful. I am so thankful for knowing him.

Contents

1	Intr	oduction	1	
	1.1	Motivation	1	
	1.2	Literature review and contribution	1	
	1.3	Problem Statement	2	
	1.4	Outline	2	
2	Bac	kground and System Modeling	4	
	2.1	Basic Information	4	
	2.2	Newton-Euler Formulation	5	
	2.3	Adaptive control theory	7	
3	Ada	ptive Control Allocation	9	
	3.1	Control Allocation	9	
	3.2	Adaptive Control Allocation	10	
4	Adaptive Back-stepping control			
	4.1	Motivation	15	
	4.2	Back-stepping Control design	15	
	4.3	Simulations for baseline controller	18	
	4.4	Adaptive Back-stepping control	21	
	4.5	Simulations and Results	23	
5	Con	clusion and Future work	25	
	5.1	Conclusion	25	
	5.2	Future work	25	
A	Appendix 1			
	A.1	First Attempt of adaptive control allocation	26	
	A.2	Adaptive back-stepping detailed derivation	26	
Bi	Bibliography			

List of Figures

1.1	Droneball cage for safety	1
2.1	Tilted Hexa-rotor vs Traditional configuration	4
2.2	Body fixed frame of tilted hexa-rotor	5
3.1	XY plane of the model showing direction of thrusts	10
3.2	Fault classification according to their modeling [1]	11
4.1	Altitude tracking results	19
4.2	x tracking results	19
4.3	y tracking results	19
4.4	Effect of gains' variation	20
4.5	Roll angle ϕ convergence to 0	20
4.6	Pitch angle θ convergence to 0	20
4.7	Yaw angle ψ convergence to 0	21
4.8	Effect of Euler Singularity on Attitude angles	21
4.9	Effect of failure on linear position	23
4.10	Effect of failure on linear velocity	23
4.11	Effect of failure on linear position after adaptation	24
4.12	Effect of failure on linear velocity after adaptation	24

1 Introduction

1.1 Motivation

Science and technology have led to the invention of what is called UAV (Unmanned Aerial Vehicle). In the nineteenth century, the first aircraft without a pilot aboard was used for warfighting. Since then, the use of unmanned air-crafts has been widely spread. Compared to manned crafts, UAVs were used for tasks where man can not be involved. First, they were used for military purposes and then they expanded to different aspects of life including agriculture, surveillance, inspection, aerial photography and many other scientific applications. Among many categories of UAVs, those with rotary wings take a heading position due to their privilege over fixed wings in some aspects such as VTOL (Vertical Take Off and Landing).

This thesis is concerned with a special class of UAVs with rotary wings.

Since aerial robots have a great contribution in many applications, their safety is a subject with crucial importance. One of safety aspects is hardware design, which simply means that the drone is designed to overcome any failure or at least, not to be completely damaged if it sud-denly drops. One way of achieving this is through designing cages for drones as shown in Figure (1.1). Another aspect on talking about UAVs' safety, is the matter of their fault tolerance, which simply shows how the flying drone will behave after the occurrence of failure in one of the parts of the drone. The failure can take place in many forms including partial damage of one of the propellers, decrease in the effectiveness of one of the motors or any other forms that will affect the behaviour of the drone while performing its task. The topic of fault tolerance is a wide one and contains many aspects such as fault detection, fault analysis and fault diagnosis. Some controllers focus on only fault detection and recognition, others focus on stabilizing the drone after the occurrence of failure without the need to identify the fault. Others can do both at the same time.



Figure 1.1: Droneball cage for safety

1.2 Literature review and contribution

1.2.1 Literature review

Flight control systems have gained much interest in the last decades due to their contribution in many applications[2]. The subject of fault tolerance control for air-crafts has been widely discussed. In this section literature review is given showing some of the past contributions of scientists in the field of fault tolerance control for multi-rotor systems.

One way of tolerating the fault was to construct a separate unit called Fault Detection and Isolation module(FDI). This unit was used to detect the fault and account for different failure scenarios. This method was widely used when failure scenarios are few and limited that one easily can switch between different cases[3]. However, if there exists some failure cases that can not be predicted, then using a separate FDI unit is not the best solution. Another disadvantage is when there are many failure scenarios, large computational load will be done online. An example of this method is constructing a box of different gains for the system and it switches between these gains according to the failure case.

Many control algorithms similar to the pre-mentioned one are gain scheduled Proportional-Integral-Derivative (PID) controllers. PID controllers with fixed gains fail to make the vehicle adapt after failure, so the idea was to make the gains time varying such that they are adjusted by adaptive algorithm to ensure system stability[4].

Presence of fault in a system indicate the presence of uncertainty because most faults are unknowns, that what led many researchers to view fault tolerance control problems as control problems with uncertainties and think of robust or adaptive control as a solution to this problem. Sliding mode control has shared a great contribution in this field [5]. Fault tolerance control was designed using sliding mode with online control allocation, without the need to reconfigure the whole controller in [6].

Since the great efforts of [7], adaptive control has proved to be an effective way to account for system uncertainties, actuator failures and external disturbances. It can be divided roughly into two main sets, the first one depends on a switching strategy through constructing an FDI module that has a bank of models and it chooses the equivalent one based on the failure case in adaptive way [8]. The other set depends on direct adaptive actuator failure compensation, through estimation of the failure in the system [9]. Recently, with advanced switching and selection functions, adaptive fault tolerance control can combine the two techniques together resulting in a better accounting for failure, the price paid is that the system is not able to detect the fault in the system, it only accounts for its effect.

1.2.2 Contribution

This thesis represents an adaptive way to account for failure without the need to have a FDI module or a switching technique to select the appropriate model. However it depends on adaptive parameter estimation after reformulating the problem of the control allocation with unknown control effectiveness. Furthermore, it works with any baseline controller as long as it has a Lyapunov function with negative definite derivative. One more thing is that fault tolerance is designed during allocating the virtual control input among the available actuators so the baseline controller is not reconfigured.

1.3 Problem Statement

The problem can be finally summarized in this section.

Given any baseline controller that provides the system with desired wrench, how can control allocation with adaptive theory play their role to account for failure in the system without the need to know this failure?

In other words, What adaptive modification should be added to any controller of a fully actuated vehicle such that the behaviour of the model is not affected after the occurrence of failure, without the need to reconfigure the controller or know any details about the failure?

1.4 Outline

For a better understanding of the content of this work, the thesis is organized as follows:

Chapter 2 shows the background information the reader needs to understand the rest of the thesis. Mathematical modeling of the tilted hexa-rotor is shown. Furthermore, The adaptive control theory behind the proposed approach is discussed.

Chapter 3 represents the core of this thesis. First it explains the problem of control allocation, then adaptive control laws behind the approach used are derived briefly.

In chapter 4, back-stepping control is designed from scratch using Newton-Euler formulation. Simulations of the baseline controller are presented. Adaptive back-stepping control is discussed briefly. Simulations of failure cases and how adaptive control allocation overcame these cases are shown.

Finally, chapter 5 provides summary of this thesis, future work and conclusion.

2 Background and System Modeling

This chapter presents the background knowledge the reader should have before going further in this thesis. First, the mathematical model is introduced, then adaptive control theory is discussed in a summarized way since its considered to be the core of this work. System modeling is done here using Newton-Euler formulation since it will be used later in chapter (4) for designing the back-stepping controller.

Basic information about the tilted hexa-rotor is first given in section 1, before getting deep into deriving equations of motion by Newton-Euler method in section 2. Section 3 is concerned with adaptive control theory and its relation with flight control systems.

2.1 Basic Information

The hexa-rotor system is a multicopter with six motors located at vertices of a planar hexagon, equidistant from the center of mass. Usually normal hexa-rotors have their thrusts in one direction parallel to each other, perpendicular to the plane of the drone, by this configuration the hexa-rotor is under-actuated system since it can allow forces and moments in four directions only (Three moments due to rotation about each axis and one force in one direction which is the thrust) while it has 6 Degrees of freedom . However, the model used in this work is a hexa-rotor having its motors tilted in a certain configuration allowing it to produce thrust in three directions, thus it becomes a fully actuated system. The tilting angle of each motor is 47 degrees, such that each two adjacent motors are tilted in opposite direction. The choice of this angle is based on optimization problem that is out of the scope of this thesis. For further details refer to [10],[11]. The following figure shows the difference between traditional hexa-rotor and the tilted one.



(a) Traditional hexa-rotor with unidirectional thrust



(b) Tilted hexa-rotor configuration

Figure 2.1: Tilted Hexa-rotor vs Traditional configuration

The tilted hexa-rotor has six motors, three rotating in a clockwise direction and the other three rotating in a counter-clockwise direction to balance the rotation of the drone about its vertical axis. They are arranged in a way such that each motor rotating in a clockwise direction is located between two motors rotating in a counter-clockwise direction.

By rotation of the propellers, a force and a moment are produced and both of them are proportional to the square of the spinning speed of the propeller. A force which is the thrust is produced normal to the plane of the rotating propeller. A drag moment is produced in a direction opposite to the directing of rotation of the propeller. For more information about the aerodynamics behind the thrust and drag resulting from rotation of propellers, the reader is kindly asked to refer to [12].

2.2 Newton-Euler Formulation

The hexa-rotor system is modeled as a rigid body in space affected by forces and torques in all directions i.e. have 6 degrees of freedom. The forces result from thrust of rotating propellers and gravitational force. Torques occur due to drag moments of propellers, moments due to thrust forces and gyroscopic effects. Mapping between forces and torques into speed of each propeller is done later in the control allocation part. For now, it is dealt with them as a virtual control input.

The model is presented using 12 states and 6 virtual control inputs. The states are linear positions, linear velocities, angular positions and angular velocities. The set of forces and moments affecting the body are the virtual control inputs. For simplicity, one can view the system as 4 states and 2 virtual inputs such that each state is a vector $\in \Re^3$ as discussed below.

Let Ψ_E : $\langle O_E, X_E, Y_E, Z_E \rangle$ denote the Inertial frame of reference while Ψ_B : $\langle O_b, X_b, Y_b, Z_b \rangle$ denotes the body fixed frame with its origin coincides with the center of mass of the drone and its axes are aligned on the body as shown in the figure.



Figure 2.2: Body fixed frame of tilted hexa-rotor

Before deriving equations of motion, there are some variables that need to be defined, some of them are considered to be system states and others are considered to be system inputs and outputs.

 $X = [x \ y \ z]^T$ holds the linear position of the body, it can be expressed in any frame by adding a sub-script of the initial of the frame, for example X_b stands for the position of the body expressed in body fixed frame.

 $\eta = [\phi \ \theta \ \psi]^T$ represents the attitude, i.e. angular position, of the body expressed in the Inertial Frame using Euler angles such that ϕ , θ and ψ are the roll, pitch and yaw angles respectively.

For the derivation of equations of motion, it is discussed in two parts, the first one for linear motion and the second one for rotational motion.

2.2.1 Linear Motion

Using Newton's 2nd law for translation motion:

$$m\dot{v}_E = F_E - \vec{g}_E,\tag{2.1}$$

where *m* is the mass of the body, v_E is the velocity of the body with respect to Ψ_E expressed in Ψ_E , F_E is the set of external forces affecting the body expressed in Ψ_E , such that $v_E and F_E \in \Re^3$ as mentioned above. \vec{g}_E is the gravity vector expressed in Ψ_E such that

$$\vec{g}_E := \begin{bmatrix} 0\\0\\g \end{bmatrix}, \tag{2.2}$$

where g is the gravitational constant.

Since thrust forces are always attached to the body, it is better to represent the forces expressed in Ψ_E as a function of the forces expressed in Ψ_B ,

$$F_E = R_b^E F_b, (2.3)$$

such that $R_b^E \in \Re^{3 \times 3}$ is the rotational matrix from Ψ_B to the Ψ_E and $F_b \in \Re^3$ is the set of forces affecting the body expressed in Ψ_B .

From equation(2.1) one can deduce that acceleration of the body expressed in Ψ_B without gravity compensation can be expressed as follows:

$$\ddot{X}_b = \frac{1}{m} F_b, \tag{2.4}$$

where $\ddot{X}_b \in \Re^3$ is the linear acceleration of the body expressed in Ψ_B .

Adding gravity effect and referring to equation(2.3), one can finally get equations of linear motion of the body expressed in Ψ_E .

$$\ddot{X}_E = \frac{1}{m} R_b^E F_b - \vec{g}_E.$$
(2.5)

One step is left, the linear position can be expressed as function of linear velocity as follows:

$$\dot{X}_E = \nu_E. \tag{2.6}$$

Attitude parameterization here is expressed using Euler angles with a sequence of Z-Y-X. Nevertheless, it is not the best way to do so and it has singularities leading to what is called Gimbal Lock. One way to avoid this problem is to use Unit Quaternions instead of Euler angles [13]. However, maneuvers that cause these singularities are not faced during simulations.

The well known Rotation matrix is defined as follows:

$$R_b^E(\eta) := \begin{pmatrix} c\theta c\psi & c\psi s\theta c\phi - c\phi s\psi & c\phi c\psi s\theta + s\phi s\psi \\ c\theta s\psi & c\phi c\psi + s\theta s\phi s\psi & c\phi s\theta c\psi - c\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix}$$
(2.7)

For the derivation of the Rotational matrix refer to [14].

2.2.2 Rotational Motion

Newtons 2nd law for rotational motion is expressed as follows,

$$J_b \dot{\omega}_b = -\omega_b \times J_b \omega_b + M_b, \tag{2.8}$$

 $\omega_b = [\omega_{x_b} \ \omega_{y_b} \ \omega_{z_b}]^T$ is the set of angular velocities of the body with respect to Ψ_E expressed in Ψ_B . $M_b \in \Re^3$ is the set of external moments affecting the body expressed in Ψ_B . J_b is the diagonal inertia matrix $\in \Re^{3\times 3}$ such that

$$J_b = \begin{pmatrix} I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz} \end{pmatrix}.$$
 (2.9)

Unlike linear motion, transformation of angular velocities from Ψ_B to Ψ_E is expressed in the following equation,

$$\dot{\eta} = T(\eta)\omega_b,\tag{2.10}$$

where T is the transformation matrix,

$$T(\eta) := \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix}$$
(2.11)

Remark : It must be noted that there exist some moments affecting the body such as gyroscopic effects due to rotation of propellers, which were neglected due to their small contribution. For detailed modeling of the system, refer to [15].

Finally, equations (2.5, 2.6, 2.8, 2.10) represent the full equations of motion of the system, and it can be summarized as follows:

$$\begin{bmatrix} \dot{X} \\ m\ddot{X} \\ \dot{\eta} \\ J_b \dot{w} \end{bmatrix} = \begin{bmatrix} v \\ -m\vec{g}_E \\ T\omega \\ -\omega \times J_b \omega \end{bmatrix} + \begin{bmatrix} 0_3 & 0_3 \\ R_b^E & 0_3 \\ 0_3 & 0_3 \\ 0_3 & I_3 \end{bmatrix} * \begin{bmatrix} F_b \\ M_b \end{bmatrix}.$$
 (2.12)

It can be expressed as a non linear system in state space representation as follows

$$\dot{x} = f(x) + g(x)v,$$
 (2.13)

where $x \in \Re^{12}$ holds states of the system , $v \in \Re^6$ is the virtual control input which is the desired Wrench of the system \mathcal{W}_{des} .

2.3 Adaptive control theory

Adaptive control is a wide branch of control theory which mainly deals with systems having uncertainties, parameters varying with time or unknown disturbance[16]. The role of adaptive control comes from its name, it adapts the controller to these changes without the need to know these unknowns. A very simple example to illustrate this concept is to imagine having a point mass in space that wants to track a certain trajectory. The problem you are facing is that you do not know the mass of this body. However, you are not concerned with knowing its mass, your target is to minimize the tracking error whether you have knowledge of the body's mass or not. Of course, it would be a privilege to get the mass known, but its not the main concern. For this type of problems, adaptive control is considered to be one of few techniques to get it solved. Another example that shows advantages of adaptive control is to imagine an aircraft flying in air affected by wind, you have no knowledge of the wind's direction, its effect or its magnitude. If adaptive control did not intervene in the design of the aircraft controller, severe results will be shown when implementing this controller into a real aircraft system.

Unlike robust control which requires prior information about these unknowns such as their bounding, adaptive control requires no prior information. Nevertheless, adaptive control is used when these uncertainties are slowly varying with time, otherwise robust control must be used. In many cases, adaptive control alone is not enough to account for system unknowns and in these cases robust adaptive control must be used. One reason for this is something called parameter drift which simply indicates that the estimated parameter diverges until infinity without being bounded[17]. In this case, some robust adaptive techniques such as projection or leakage solve this problem. However, in the approach introduced in this thesis, there was no need to robust adaptive techniques and they are to be done in future work.

On talking about adaptive control, a very important term comes to mind which is parameter estimation. Parameter estimation which is a branch of system identification is considered to be the foundation of adaptive control. As the name states, adaptive control relies on estimating the unknown parameters and using these estimates to update the control law. There are many ways for parameter estimation such as Least square method, decent gradient and Lyapunov stability method. Least square method depends on solving an optimization problem to estimate the unknown parameter. Lyapunov stability method is the one used in this work and it simply derives the adaptive law to ensure stability of the system. So, while deriving the adaptive control law, it is already made sure that the system states will not diverge.

Adaptive controller can be divided into three main categories; direct, indirect and hybrid methods. Direct adaptive control is the method where the estimated parameters are used directly in the controller. In indirect methods, estimated parameters are used to calculate other required control parameters. Hybrid relies on both of parameter estimation and direct modification of the control law.

To sum up why adaptive control theory was the one used in this work, a multi-copter flying in air is subjected to many uncertainties, such as wind, unbounded disturbance, sensor noise or actuator failure. For this thesis, actuator failure is the reason for presence of uncertainties in the system. And its adaptive control role to account for this failure. It does not depend on direct estimation of this failure, but it estimates another parameter that will be used in the controller as discussed later. Parameter estimation is done through Lyapunov method to ensure convergence of system states and their stability.

3 Adaptive Control Allocation

3.1 Control Allocation

For every system, there must exist some inputs and outputs. For a drone flying in air, one can easily see that voltage supplying the motors is the input of the system. By varying the voltage, the speed of spinning will be varied resulting in a thrust that we can control to reach certain position and orientation in space, which are the system outputs. The relation between voltage to the motors and their rotational speed is out of the scope of this thesis [18].

Control allocation is the problem of distributing control effort among multiple actuators. In other words, transforming the set of forces and moments coming from the controller to the set of actuators available. In the context of this work, it can be viewed as mapping between the desired wrench affecting the body into the spinning speed of each motor. [19]

So, the actual input of the system is the rotational speed of the motors. In the earlier sections, it was dealt with the set of forces and moments resulting from rotation of motors as the virtual control input. Mapping between virtual control inputs and real inputs is shown in this section.

As was just mentioned earlier, a rotating propeller *i* produces thrust force λ_i and a drag moment τ_i . Since both thrust and drag moment are proportional to the square of the spinning speed, they can be written as follows,

$$\lambda_i = \mu \,\Omega_i^2,\tag{3.1}$$

$$\tau_i = \gamma \ \sigma_i \ \lambda_i, \tag{3.2}$$

where μ is a thrsut co-efficient, Ω_i is the rotational speed of *i* -th motor, γ is the drag-to-thrust ratio, σ is either 1 or -1 indicating the direction of the produced moment.

Remark 1:

It is obvious that each tilted propeller *i* produce thrust λ_i in 3D-space can be expressed in body fixed frame as $\lambda_i = [\lambda_{i_x} \ \lambda_{i_y} \ \lambda_{i_z}]^T$ and since tilting angle of propellers is 47 degrees for all motors, then one can deduce that the *z*-component will always be,

$$\lambda_{i_z} = \lambda_i \cos(47) \qquad \forall i. \tag{3.3}$$

x, *y*-components are derived using figure(3.1).

The following figure shows the X-Y plane of the body, where the vectors shown indicate the direction of the thrust of each propeller in the X-Y plane. Numbering of motors and their direction of rotation is also displayed. By using this figure and equation (3.1, 3.2), one can easily deduce how forces and moments are mapped into their thrusts.

The control allocation can be expressed in the following equation,

$$\mathcal{W}_b = \begin{pmatrix} F_b \\ M_b \end{pmatrix} = B \ \lambda, \tag{3.4}$$

where $B \in \Re^{6 \times 6}$ is the control allocation matrix.

Using simple geometry of a regular hexagon and basics of mechanics one can easily represent thrusts shown in the above figure as x, y components only. By the same way, each thrust causes



Figure 3.1: XY plane of the model showing direction of thrusts

a rotating torque due to its arm. Writing each force and moment into three components $\{x, y, z\}$ leads to the following:

$$\begin{bmatrix} F_{b_x} \\ F_{b_y} \\ F_{b_z} \\ M_{b_x} \\ M_{b_y} \\ M_{b_z} \end{bmatrix} = \begin{bmatrix} -s & -s & \frac{s}{2} & \frac{s}{2} & \frac{s}{2} & \frac{s}{2} & \frac{s}{2} \\ 0 & 0 & \frac{-\sqrt{3s}}{2} & \frac{-\sqrt{3s}}{2} & \frac{\sqrt{3s}}{2} & \frac{\sqrt{3s}}{2} \\ c & c & c & c & c & c \\ -lc - \gamma s & lc + \gamma s & \frac{lc}{2} + \frac{\gamma s}{2} & \frac{-lc}{2} + \frac{-\gamma s}{2} & \frac{-lc}{2} + \frac{-\gamma s}{2} & \frac{lc}{2} + \frac{\gamma s}{2} \\ 0 & 0 & \frac{-\sqrt{3lc}}{2} + \frac{-\sqrt{3}\gamma s}{2} & \frac{\sqrt{3lc}}{2} + \frac{\sqrt{3}\gamma s}{2} & \frac{\sqrt{3lc}}{2} + \frac{\sqrt{3}\gamma s}{2} \\ -ls + \gamma c & +ls - \gamma c & -ls + \gamma c & ls - \gamma c & ls - \gamma c & -ls + \gamma c \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix}$$

$$(3.5)$$

Where *s*, *c* indicate $\sin 47^{\circ}$ and $\cos 47^{\circ}$ respectively. *l* is the length between any propeller and center of mass of the body.

To reach for the thrust of motors from the system wrench, one way to solve equation(3.4) is to get the pseudo-inverse of the control allocation matrix. However, in our case the system is a fully actuated one, which means that *B* always has full rank and non singular so it can be solved using inverse of a matrix directly [20].

$$\lambda = B^{-1} \mathcal{W}_b. \tag{3.6}$$

Spinning speed of each motor can be easily derived through,

$$\Omega_i = \sqrt{\frac{\lambda_i}{\mu}}.$$
(3.7)

Advantages of using control allocation can be easily observed here, by using any high-level control algorithm that provides the system with the desired wrench, one can get easily the real input to the motors.

3.2 Adaptive Control Allocation

As discussed earlier, control allocation is the mapping between virtual control inputs(forces and moments) into the real inputs of the system (thrust of each motor), expressed in equation (3.4) as

$$\mathcal{W}_b = \begin{pmatrix} F_b \\ M_b \end{pmatrix} = B \ \lambda. \tag{3.8}$$

Remark 1:

As presented in the previous chapter that the desired wrench coming from the controller is expressed in body fixed frame, so the sub-script is removed in the coming sections for simplicity.

Remark 2:

From the beginning of this section, due to the existence of failure in the system, the desired wrench that should affect the body does not equal to the actual one. A sub-script will be added such that W_{act} is the actual wrench, W_{des} is the desired one and W_{err} is the error between the desired and actual one,

$$\mathcal{W}_{err} = \mathcal{W}_{des} - \mathcal{W}_{act}.$$
(3.9)

Remark 3:



Figure 3.2: Fault classification according to their modeling [1]

Faults in any system can be classified according to their location into three categories; actuator faults, component fault or sensor fault. As mentioned earlier, this thesis is concerned with actuator faults. Furthermore, with respect to fault modeling, it can be classified as additive or multiplicative as shown in the above figure (3.2)[1].

Usually actuator faults are modeled using the multiplicative way, so one way of representing this fault is to multiply each thrust by a constant showing its degree of damage, such that

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix}_{eff} = \begin{bmatrix} \Lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Lambda_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Lambda_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Lambda_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Lambda_6 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix}_{calc}$$
(3.10)

where $0 \le \Lambda_i \le 1$, such that 0 represents complete damage of the rotor and 1 indicates that the rotor has no failure[21]. This extends equation (3.1) to be,

$$\mathcal{W}_{act} = \begin{pmatrix} F\\ M \end{pmatrix} = B \Lambda \lambda, \tag{3.11}$$

where $\Lambda \in \Re^{6 \times 6}$ is called the effectiveness matrix and it is considered to be unknown representing uncertainty in the system.

If the effectiveness matrix is known, then equation (3.11) could be easily solved using inverse of matrix ($B\Lambda$) as long as it has full rank. If not, which is the case in many over-actuated systems, then its solved as an optimization problem[20].

Many fault tolerance control allocation techniques depend on estimating of the effectiveness matrix, then solving an optimization problem. Estimation can be done through Kalmen filters or any other adaptive techniques[22].

Nevertheless, the method used here is also based on estimation techniques, but without the need to solve any optimization problems. Instead of using equation (3.6) to get the actual control input of the system, here comes the rule of adaptive estimation to solve this problem.

If there exists a term Θ such that $(B^{-1} + \Theta) = (B\Lambda)^{-1}$, then no need to have knowledge of the effectiveness matrix[23]. Lets call that

$$\lambda_o = B^{-1} \mathcal{W}_{des}, \tag{3.12}$$

the nominal control law i.e, the case where the effectiveness matrix is an identity matrix. In other words, the case where the system has no rotor failure. What if there exists some failure which is unknown in one or more of the rotors. The nominal control law must be augmented with adaptive part to account for this failure. In this case

$$\lambda = \lambda_o + \lambda_{ad}. \tag{3.13}$$

The adaptive augmentation is defined as follows,

$$\lambda_{ad} = \hat{\Theta} \, \mathcal{W}_{des},\tag{3.14}$$

where $\hat{\Theta} \in \Re^{6 \times 6}$ is an estimation of the ideal parameter Θ satisfying the condition

$$(B \Lambda) (B^{-1} + \Theta) = I_6, \tag{3.15}$$

where I_6 is a 6 × 6 identity matrix.

Finally the control law can be expressed as,

$$\lambda = (B^{-1} + \hat{\Theta}) \mathcal{W}_{des}.$$
(3.16)

If a very accurate estimate was obtained such that $\hat{\Theta} \longrightarrow \Theta$ then in this case,

$$\mathcal{W}_{act} = B \ \Lambda \ \lambda = (B \ \lambda) \ (B^{-1} + \Theta) \mathcal{W}_{des} = I_6 \ \mathcal{W}_{des} = \mathcal{W}_{des}$$
(3.17)

But unfortunately, there never exist the case of ideal estimation, there will always be an estimation error $\tilde{\Theta}$,

$$\tilde{\Theta} = \hat{\Theta} - \Theta, \tag{3.18}$$

which causes the wrench error and can be defined as follows,

$$\begin{split} \mathcal{W}_{err} &= \mathcal{W}_{des} - \mathcal{W}_{act}, \\ &= \mathcal{W}_{des} - (B \ \lambda) \ (B^{-1} + \hat{\Theta}) \mathcal{W}_{des}, \\ &= \mathcal{W}_{des} - (B \ \lambda) \ (B^{-1} + \tilde{\Theta} + \Theta) \mathcal{W}_{des}, \\ &= \mathcal{W}_{des} - \mathcal{W}_{des} - B \Lambda \tilde{\Theta} \mathcal{W}_{des}, \\ &= -B \Lambda \tilde{\Theta} \mathcal{W}_{des}. \end{split}$$
(3.19)

No failure case

Imagine having a baseline controller that provides the system (2.13)

$$\dot{x} = f(x) + g(x)\mathcal{W}_{des},\tag{3.20}$$

with the control law \mathcal{W}_{des} and this system has Lyapunov function and its derivative such that

$$\Psi_1(x) \le V_0(x, t) \le \Psi_2(x), \tag{3.21}$$

$$\dot{V}_0(x,t) = \frac{\partial V_0}{\partial t} + \left(\frac{\partial V_0}{\partial x}\right)^T (f(x) + g(x)\mathcal{W}_{des}) \le -\Psi_3(x).$$
(3.22)

Where $\Psi_1(x)$, $\Psi_2(x)$ and $\Psi_3(x)$ are continuous positive definite functions.

Failure case

Due to the presence of failure in the system,

$$\begin{split} \dot{V}_{0} &= \frac{\partial V_{0}}{\partial t} + \left(\frac{\partial V_{0}}{\partial x}\right)^{T} (f(x) + g(x) \mathcal{W}_{act}), \\ &= \frac{\partial V_{0}}{\partial t} + \left(\frac{\partial V_{0}}{\partial x}\right)^{T} (f(x) + g(x) (\mathcal{W}_{des} - \mathcal{W}_{err})), \\ &\leq \Psi_{3}(x) - \left(\frac{\partial V_{0}}{\partial x}\right)^{T} (g(x) \mathcal{W}_{err}), \\ &\leq -\Psi_{3}(x) + \left(\frac{\partial V_{0}}{\partial x}\right)^{T} (g(x) B \Lambda \tilde{\Theta} \mathcal{W}_{des}). \end{split}$$
(3.23)

The main idea is to add a term in the Lyapunov function such that its derivative cancels the term $\left(\frac{\partial V_0}{\partial x}\right)^T (g(x)B\Lambda\tilde{\Theta}\mathcal{W}_{des})$ so \dot{V} is negative semi-definite function.

$$V = V_0 + V_{ad}, (3.24)$$

where V_{ad} carries information about parameter error $\tilde{\Theta}$,

$$V_{ad} = \frac{1}{2\delta} tr(\tilde{\Theta}\tilde{\Theta}^T \Lambda).$$
(3.25)

Remark : Since, the tilted hexa-rotor is a fully actuated vehicle and there is no redundancy in the system, it is impossible to tolerate a complete rotor failure. So, failure cases discussed only show rotor degradation $\Lambda_i > 0$, which imply that Λ is always positive definite matrix.

 δ is the adaptive gain $\in \Re > 0$,

 $tr(XY^T)$ is a way to present matrix multiplication in a scalar way in case X,Y are non-scalar square matrices, such that

 $tr(XY^T) = \sum_{i=1}^{n} x_i^T y_i$ where x_i^T is the *i* -th row of *X* Calculating the derivative leads to

$$\dot{V} = \dot{V}_0 + \dot{V}_{ad}, \leq -\Psi_3(x) + \left(\frac{\partial V_0}{\partial x}\right)^T (g(x) B \Lambda \tilde{\Theta} \mathcal{W}_{des}) + \frac{1}{\delta} tr(\tilde{\Theta} \dot{\tilde{\Theta}}^T \Lambda).$$
(3.26)

From equation (3.18) if the failure in the rotor is not time varying then $\dot{\Theta} = 0 \Longrightarrow \dot{\tilde{\Theta}} = \dot{\tilde{\Theta}}$

$$\dot{V} \leq -\Psi_3(x) + \left(\frac{\partial V_0}{\partial x}\right)^T g(x) B \Lambda \tilde{\Theta} \mathcal{W}_{des} + \frac{1}{\delta} tr(\tilde{\Theta} \dot{\tilde{\Theta}}^T \Lambda).$$
(3.27)

Using mathematical properties of tr() method,

$$\begin{split} & if \ A,B \ are \ column \ vectors \\ & then \ tr(A^TB) = A^TB, \\ & \dot{V} \leq -\Psi_3(x) + tr(\left(\frac{\partial V_0}{\partial x}\right)^Tg(x)B\Lambda\tilde{\Theta}\mathcal{W}_{des}) + \frac{1}{\delta}tr(\tilde{\Theta}\dot{\bar{\Theta}}^T\Lambda), \end{split}$$

tr(ABC) = tr(BCA) = tr(CAB)i.e. cyclic property of trace,

$$\leq -\Psi_{3}(x) + tr(\tilde{\Theta}\mathcal{W}_{des}\left(\frac{\partial V_{0}}{\partial x}\right)^{T}g(x)B\Lambda) + \frac{1}{\delta}tr(\tilde{\Theta}\dot{\Theta}^{T}\Lambda),$$

tr(AB) + tr(CD) = tr(AB + CD),

$$\leq -\Psi_{3}(x) + tr(\tilde{\Theta}\mathcal{W}_{des}\left(\frac{\partial V_{0}}{\partial x}\right)^{T}g(x)B\Lambda + \frac{1}{\delta}\tilde{\Theta}\dot{\Theta}^{T}\Lambda),$$

$$\leq -\Psi_{3}(x) + tr(\tilde{\Theta}(\mathcal{W}_{des}\left(\frac{\partial V_{0}}{\partial x}\right)^{T}g(x)B + \frac{\dot{\Theta}^{T}}{\delta})\Lambda).$$
(3.28)

Finally the adaptive law can be expressed as,

$$\dot{\hat{\Theta}}^{T} = -\delta \mathcal{W}_{des} \left(\frac{\partial V_{0}}{\partial x}\right)^{T} g(x) B, \qquad (3.29)$$

and the control allocation problem can be solved using equation(3.16).

4 Adaptive Back-stepping control

4.1 Motivation

The previous chapters presented the mathematical model of the tilted hexa-rotor using Newton-Euler formulation as a rigid body in space then a problem of control allocation was solved to allocate the control effort among the six actuators. In chapter three, the main approach was presented and it was said that for this approach to work, the baseline controller has to have a Lyapunov function whose derivative is negative definite. One thing to be mentioned, a trial was performed before on a baseline controller whose Lyapunov function derivative is negative semi-definite and it failed, refer to the appendix(A.1) for more information.

On thinking of a controller that has a lyapunov function whose derivative is negative definite, back-stepping seems to be a perfect choice since it is already derived through a series of Lyaounov functions. Back-stepping control is used here as the baseline controller for position and attitude tracking. Control input laws are derived in section 2 using the model obtained previously from Newton-Euler formulation. Results and simulations of tracking any arbitrary trajectory without failure is shown in section 3. The augmentation of the adaptive part is added in section 4. Section 5 shows final simulations after occurrence of failure and how it managed to overcome the two problems of the aforementioned controller.

4.2 Back-stepping Control design

Back-stepping is a recursive Lyapunov-based scheme proposed in the beginning of 1990s. The idea is to design a controller recursively by considering some of the state variables as virtual controls (stepping variables) and designing for them intermediate control laws[24].

As was mentioned before, the tilted hexa-rotor is a fully actuated vehicle so, position control can be achieved separately from attitude control. Control design is divided into two parts, one for tracking desired position and the other one for tracking desired attitude. Each one of them is done in two steps using one stepping variable.

Remark :

Variables in this section are expressed in Inertial frame unless mentioned otherwise. For example, *X* stands for position of body expressed in Inertial Frame.

Definitions :

 $X_d = [x_d \ y_d \ z_d]^T$ is the desired position expressed in Inertial Frame.

 $\eta_d = [\phi_d \ \theta_d \ \psi_d]^T$ is the desired attitude expressed in Inertial Frame.

Assumptions :

The desired trajectory { X_d , η_d } is twice differentiable and all its derivatives are bounded.

4.2.1 Position tracking

Let us first define position tracking error e_x , such that

$$e_x = X_d - X. \tag{4.1}$$

Starting with the first Lyapunov function,

$$V_1 = \frac{1}{2} e_x^T e_x > 0, (4.2)$$

whose derivative is

$$\dot{V}_1 = e_x^T(\dot{e}_x) = e_x^T(\dot{X}_d - \dot{X}) = e_x^T(\dot{X}_d - \nu).$$
(4.3)

Define the first and only back-stepping variable v such that $u_1 = v$. In other words, v is a virtual control input. *Let*

$$u_1 = \dot{X_d} + K_1 e_x, \tag{4.4}$$

such that $K_1 \in \Re^{3 \times 3}$, the term $K_1 e_x$ is used to stabilize e_x to make $\lim_{t \to \infty} e_x \to 0$.

$$\dot{V}_1 = -e_x^T K_1 e_x < 0. ag{4.5}$$

Now, define the error between stepping variable and state variable. $e_1 = u_1 - v$, if it was managed to stabilize this error, e_x will be stabilized.

$$e_1 = u_1 - v \Longrightarrow v = u_1 - e_1, \tag{4.6}$$

$$\dot{e_1} = \dot{u_1} - \dot{v} = \ddot{X_d} + K_1(\dot{e_x}) - \dot{v}.$$
(4.7)

The augmented Lyapunov function will be

$$V_2 = V_1 + \frac{m}{2} e_1^T e_1 > 0, (4.8)$$

$$\dot{V}_{2} = \dot{V}_{1} + me_{1}^{T} \dot{e}_{1}, = -e_{x}^{T} K_{1} e_{x} + e_{x}^{T} e_{1} + e_{1}^{T} (m \ddot{X}_{d} + m K_{1} \dot{e}_{x} - m \dot{\nu}), = -e_{x}^{T} K_{1} e_{x} + e_{1}^{T} (m \ddot{X}_{d} + m K_{1} \dot{e}_{x} - m \dot{\nu} + e_{x}),$$

$$(4.9)$$

from equation (2.5),

$$= -e_x^T K_1 e_x + e_1^T (m \ddot{X}_d + m K_1 \dot{e}_x - R_b^E F_{des} + m \vec{g}_E + e_x).$$

After reaching the real control input, F_b is chosen to stabilize the error e_1 , i.e. $\lim_{t\to\infty} e_1 \to 0$,

$$F_{des} = (R_b^E)^T (m\ddot{X}_d + mK_1\dot{e}_x + m\vec{g}_E + e_x + K_2e_1),$$

= $R_E^b (m\ddot{X}_d + mK_1\dot{e}_x + m\vec{g}_E + e_x + K_2e_1),$ (4.10)

where $K_2 \in \Re^{3 \times 3}$ is a gain matrix used to stabilize the error e_1 resulting in,

$$\dot{V}_2 = -e_x^T K_1 e_x - e_1^T K_2 e_1 < 0.$$
(4.11)

4.2.2 Attitude tracking

Going through same steps as previous section,

$$e_{\eta} = \eta_d - \eta. \tag{4.12}$$

Choosing Lyapunov function,

$$V_3 = \frac{1}{2} e_{\eta}^T e_{\eta} > 0, \tag{4.13}$$

whose derivative is,

$$\dot{V}_3 = e_{\eta}^T (\dot{e}_{\eta}) = e_{\eta}^T (\dot{\eta}_d - \dot{\eta}) = e_{\eta}^T (\dot{\eta}_d - T\omega_b),$$
(4.14)

where T is the transformation matrix defined in equation (2.11). let ω_b be a virtual control such that $\omega_b = u_2$,

$$u_2 = T^{-1}(\dot{\eta_d} + K_3 e_\eta), \tag{4.15}$$

leading to

$$\dot{V}_3 = -e_\eta^T K_3 e_\eta < 0. ag{4.16}$$

Introducing the error between stepping variable and state variable e_2

$$e_2 = u_2 - \omega_b \Longrightarrow \omega_b = u_2 - e_2, \tag{4.17}$$

$$\dot{e_2} = \dot{u_2} - \dot{\omega_b}.$$
 (4.18)

 \dot{u}_2 is derived below but used later as \dot{u}_2 for simplicity,

$$\dot{u}_2 = \frac{d}{dt} (T^{-1}(\dot{\eta_d} + K_3 e_\eta)),$$

= $-T^{-2}(\dot{\eta_d} + K_3 e_\eta) + T^{-1}(\ddot{\eta_d} + K_3 \dot{e_\eta}).$ (4.19)

The augmented Lyapunov function is,

$$V_4 = V_3 + \frac{1}{2}e_2^T J e_2 > 0, (4.20)$$

where J is the inertia matrix defined before in chapter 2.

$$\begin{split} \dot{V}_{4} &= \dot{V}_{3} + e_{2}^{T} J \dot{e}_{2}, \\ &= -e_{\eta}^{T} K_{3} e_{\eta} + e_{\eta}^{T} T e_{2} + e_{2}^{T} (J \dot{u}_{2} - J \dot{\omega}_{b}), \\ from \ equation \ (2.8), \\ &= -e_{\eta}^{T} K_{3} e_{\eta} + e_{\eta}^{T} T e_{2} + e_{2}^{T} (J \dot{u}_{2} + \omega_{b} \times J_{b} \omega_{b} - M_{des}), \\ &= -e_{\eta}^{T} K_{3} e_{\eta} + e_{2}^{T} (J \dot{u}_{2} + \omega_{b} \times J_{b} \omega_{b} - M_{des} + T e_{\eta}). \end{split}$$
(4.21)

Choosing,

$$M_{des} = J\dot{u}_2 + \omega_b \times J_b \omega_b + Te_\eta + K_4 e_2, \qquad (4.22)$$

leads to,

$$\dot{V}_4 = -e_\eta^T K_3 e_\eta - e_2^T K_4 e_2 < 0, \tag{4.23}$$

where $K_3, K_4 \in \Re^{3 \times 3}$ are gain matrices.

Stability analysis

Using Lyapounov theorem for stability [25], since

- V(x) is positive definite, (4.8, 4.20),
- $\dot{V}(x)$ is negative definite (4.11, 4.23),
- V(x) is radially unbounded, it can be clearly seen that $\lim_{|x|\to\infty} V(x) \to \infty$,

therefore, Global Asymptotic stability of e_x , e_1 , e_η , $e_2 = 0$ is guaranteed.

Before ending this section let us define the whole Lyapunov function of the system V_5 such that

$$V_5 = V_2 + V_4, \tag{4.24}$$

$$\dot{V}_5 = \dot{V}_2 + \dot{V}_4 = -e_x^T K_1 e_x - e_1^T K_2 e_1 - e_\eta^T K_3 e_\eta - e_2^T K_4 e_2.$$
(4.25)

In a summarized way there exists a Lyapunov function of the system $V_5(x, t)$, such that

$$0 \le V_5(x, t) \le \Psi_1(x), \tag{4.26}$$

$$\dot{V}_5(x,t) \le -\Psi_2(x),$$
 (4.27)

where $\Psi_1(x)$, $\Psi_2(x)$ are continuous positive definite functions.

The upper bounding $\Psi_1(x)$ of equation(4.26) can be proved as follows:

Since it was proved that global asymptotic stability of $\{e_x, e_1, e_\eta, e_2\} = 0$ is guaranteed. That means that they will never diverge, they will always have an upper bound. And since Lyapunov function is the summation of these terms, therefore it must also have an upper bound.

4.3 Simulations for baseline controller

As was mentioned in the above section, tracking a desired trajectory should be guaranteed with this controller, simulations will confirm these results in this section

4.3.1 Position tracking results

Simulations were carried out on both constant desired trajectory and time varying trajectories. In real life, constant desired trajectory means that the drone will just go from a place to another and the aim to hover at the desired location, while time-varying trajectories can be viewed as you are trying to reach a target which is continuously changing and moving.







Figure 4.2: *x* tracking results





Figures (4.1, 4.2, 4.3) show how tracking any trajectory was achieved for every linear motion; altitude and both longitudinal directions. Part (b) of each figure shows the case when the trajectory is time varying while part (a) shows constant trajectory tracking. Zero initial conditions were set to all the variables.

It is it to be noted that by decreasing the values of the gains (K_1, K_2) , the settling time of the output response decreases and vise verse. And this makes sense, the larger the values of the gains, the faster the convergence to the desired state. However, there were not any time specifications that should be met, the goal was to achieve tracking error = 0, and this was achieved. In addition to that for setting (K_1, K_2) to very low values (for example Identity matrix) an overshoot can be observed. The following Figure(4.4) shows the result after varying the gains. One can easily observe that for large values of gains, the settling time is so small (nearly less than 0.5 sec) and there is no overshoot. On the other hand, both overshoot and undershoot are observed for very low gain values and the settling time is around 7 seconds which is way slower than the previous case.



Figure 4.4: Effect of gains' variation

4.3.2 Attitude tracking results

Before showing the results, one thing has to be mentioned; in the previous section, two types of trajectories were shown and this was for a purpose. In real life, one task for a drone would be doing a circle maneuver in air and mathematically that implies that both x, y are sine or cosine waves shifted by constants, that's why the case of time varying trajectories was shown. While for the case of orientation, the most common case is that the drone stabilizes itself around a certain angle, no one would ever like to see his drone pitching up and down continuously. So attitude stabilization around zero angle is the case shown here with non-zero initial conditions.



Figure 4.5: Roll angle ϕ convergence to 0



Figure 4.6: Pitch angle θ convergence to 0



Figure 4.7: Yaw angle ψ convergence to 0

Remark:

As said previously in chapter(2), attitude parameterization is done in this thesis using Euler angles and this for sure has crucial drawbacks. The main one is having a singularity at $\theta = \pm \frac{\pi}{2}$, this results in not having a single solution of Euler angles or what is known by Gimbal Lock, and this can be observed here as in figure(4.8) by setting $\theta > \frac{\pi}{2}$, so its sure that it will pass by the singularity on its way to convergence.



Figure 4.8: Effect of Euler Singularity on Attitude angles

4.4 Adaptive Back-stepping control

In this section, adaptive control allocation is designed to account for actuator failure without modifying the baseline controller. Its crucial for the reader to read the section(3.2) before going further as it contains all the necessary information for the method applied here.

For the case of failure, the actual wrench is no longer the desired one due to the presence of error in the system.

The final Lyapunov function following same steps as previous chapter,

$$V = V_5 + \frac{1}{2\delta} tr(\tilde{\Theta}\tilde{\Theta}^T \Lambda) > 0.$$
(4.28)

As equation(3.26) states,

$$\dot{V} = -e_x^T K_1 e_x - e_1^T K_2 e_1 - e_\eta^T K_3 e_\eta - e_2^T K_4 e_2 + [e_1^T R_b^E \quad e_2^T] \quad \mathcal{W}_{err} + \frac{1}{\delta} tr(\tilde{\Theta}\tilde{\Theta}^T \Lambda).$$
(4.29)

Going through same steps for derivation of adaptive law, it is chosen as

$$\hat{\Theta}^T = \delta \quad \mathcal{W}_{des} \quad [e_1^T R_b^E \quad e_2^T] B, \tag{4.30}$$

resulting in

$$\dot{Y} = -e_x^T K_1 e_x - e_1^T K_2 e_1 - e_\eta^T K_3 e_\eta - e_2^T K_4 e_2 \le -\Psi_2(x).$$
(4.31)

And the control allocation problem is solved using equation(3.16) from previous chapter. *For the detailed derivation of equation(4.29, 4.30), refer to the Appendix(A.2).*

Stability Analysis

Since the Lyapunov function derivative is no longer negative definite, the traditional Lyapunov method for stability can not help here and its time to use Barbalt's Lemma.

Before using Lyabunov Like lemma "Barbalat's Lemma for Stability Analysis", Lets sum up everything here ,

$$V_{=}\frac{1}{2}e_{x}^{T}e_{x} + \frac{m}{2}e_{1}^{T}e_{1} + \frac{1}{2}e_{\eta}^{T}e_{\eta} + \frac{1}{2}e_{2}^{T}Je_{2} + \frac{1}{2\delta}tr(\tilde{\Theta}\tilde{\Theta}^{T}\Lambda) > 0, \qquad (4.32)$$

is a positive definite function (lower bounded),

$$\dot{V} = -e_x^T K_1 e_x - e_1^T K_2 e_1 - e_\eta^T K_3 e_\eta - e_2^T K_4 e_2 \le -\Psi_2(x), \tag{4.33}$$

is a negative semi definite function where $\Psi_2(x)$ is a continuous positive definite function,

• \ddot{V} is bounded, and this can be proven as follows,

$$\ddot{V} = 2 * (-e_x^T K_1 \dot{e}_x - e_1^T K_2 \dot{e}_1 - e_\eta^T K_3 \dot{e}_\eta - e_2^T K_4 \dot{e}_2), \qquad (4.34)$$

To prove the bounding of \ddot{V} , it has to show that all four terms on the right hand side are bounded, and each term is an error multiplied by its derivative. So, both the error and its derivative have to be bounded.

Through Lyapunov stability analysis, it can shown that each error is only stable, which imply that each error is bounded (e_x , e_1 , e_η , e_2).

Bounding of an error does not imply the bounding of its derivative, the error derivative has to be proved from system dynamics. Let us take e_x and e_1 as an example and it goes for the rest of the four errors.

$$\dot{e}_x = \dot{X}_d - \dot{X} = \dot{X}_d - \nu. \tag{4.35}$$

 \dot{X}_d is bounded from the assumption in the beginning of this chapter, v is already a system-state which was proven that it will never diverge, which means that it is bounded. This implies the bounding of \dot{e}_x .

$$\begin{split} \dot{e}_{1} &= \dot{u}_{1} - \dot{v} = \ddot{X}_{d} + K_{1}(\dot{e}_{x}) - \dot{v}, \\ &= \ddot{X}_{d} + K_{1}(\dot{e}_{x}) - \frac{1}{m} (R_{b}^{E} F_{des} - m \vec{g}_{E}), \\ &= \ddot{X}_{d} + K_{1}(\dot{e}_{x}) - \frac{1}{m} (R_{b}^{E} (R_{E}^{b} (m \ddot{X}_{d} + m K_{1} \dot{e}_{x} + m \vec{g}_{E} + e_{x} + K_{2} e_{1})) - m \vec{g}_{E}), \\ &= \frac{-1}{m} (e_{x} + K_{2} e_{1}), \end{split}$$
(4.36)

and e_x , e_1 are known to be bounded so \dot{e}_1 is bounded.

Following the same procedure for the other two errors, \ddot{V} is bounded.

By Barbalt's lemma, this leads to $\dot{V} \rightarrow 0$ as $t \rightarrow \infty$ which indicates the convergence of all errors in failure case, and the graphs in the next section confirm this result.

4.5 Simulations and Results

As mentioned earlier, the tilted hexa-rotor is a fully actuated system with no redundant actuators so there is no way for complete rotor failure. Situations of rotors degradation show that with adaptive fault tolerance, the system can stand partial failure of all rotors at the same time.

The trajectory used for testing failure is a circle maneuver at constant altitude, degradation of all rotors occur after 12 seconds as shown in the next figures(4.9, 4.10). It is clear from the graphs of figure(4.9) that failure affects all axes of linear motion, however the altitude is the one affected the most. Graphs of figure(4.10) show that even linear velocity of the system was tracking the desired velocity (which is \dot{X}_d) before occurrence of failure, after 12 seconds, an error appeared and it converges soon.



Figure 4.9: Effect of failure on linear position



Figure 4.10: Effect of failure on linear velocity

The following figures show the effect of adaptive control allocation after occurrence of failure. Figure(4.11) shows how the adaptive modification changes the whole behaviour of the system after failure, it nearly looks like the case with no failure. That is considered to be a great result of the approach proposed in this work.

It is to be noted that the effect of failure on the angular positions and velocities was not clear that's why simulations are shown for linear motion only.



Figure 4.11: Effect of failure on linear position after adaptation



Figure 4.12: Effect of failure on linear velocity after adaptation

5 Conclusion and Future work

5.1 Conclusion

In this thesis, an adaptive control allocation approach was applied to a position tracking problem of a class of fully actuated UAVs. The model used in this work was a hexa-rotor tilted in a certain configuration allowing it to be a fully actuated body. Due to the fact that there is no redundancy in the system, failure was viewed as a degradation of rotors without complete failure. The approach can be implemented on any baseline controller as long as it has a Lyapunov function with negative definite derivative. Adaptive control allocation was applied once on a baseline controller with a Lyapunov function whose derivative is negative semi-definite and it showed some drawbacks as mentioned in the Appendix(A.1). The method was then applied on a back-stepping controller since it was sure that it satisfies the above-mentioned condition. The approach used in this work has several advantages, first, it does not require any prior knowledge about the fault in the system. Furthermore, the baseline controller is not changed, only the virtual control input which is the wrench affecting the body is redistributed among available actuators without the need to reconfigure the whole controller. Results were confirmed using 20-sim as a simulation program. Simulations show that the system can stand partial failure of all rotors at the same time and this is considered to be an astonishing result.

5.2 Future work

It is well known from literature that uncertainties or noise in any system may lead to what is known as parameter drift. Which simply means that adaptation will not end, furthermore it will diverge resulting in an unstable system. So, the solution for this is to use robust adaptive techniques to ensure that the adaptive estimate will never diverge. From the robust modifications to be used is projection or leakage which simply bound the adaptive estimate and also allow adaptation only when needed.

One last thing is to try the back-stepping controller with adaptive control allocation in a real experiment. This requires to re-configure the flight controller, adjust the gains and trials will be conducted.

A Appendix 1

A.1 First Attempt of adaptive control allocation

The method was first tried on a controller with Lyapunov function whose derivative is negative semi-definite and it showed severe drawbacks. The baseline controller used was a spatial impedance controller. Stability of system states was proven with no information about convergence to the desired trajectory. It was also proven mathematically and during simulations that without gravity compensation, the actual wrench of the system tend to 0 during adaptation which simply indicates that the body will hover in its place without moving. One impractical solution to this problem is to make the failure varying with time, in this sense the actual wrench of the system will not tend to 0. The last drawback that appeared only during simulation that adaptation has to work only if there is failure, otherwise the drone will end up hovering in its place without moving even if there is no failure in the system. A solution to this problem maybe that feedback about failure time is given to the system so it can start adaptation after failure occurrence but this is not practical.

A.2 Adaptive back-stepping detailed derivation

keeping in mind that due to the presence of failure the virtual control input (\mathcal{W}) is the actual wrench and not the desired one. Which means that actual forces and moments are to be shown.

So jumping back to equation (4.9) and changing F_{des} to be F_{act} such that,

$$F_{act} = F_{des} - F_{err},\tag{A.1}$$

leads to

$$\dot{V}_2 = -e_x^T K_1 e_x + e_1^T (m \ddot{X}_d + m K_1 \dot{e}_x - R_b^E F_{des} + R_b^E F_{err} + m \vec{g}_E + e_x),$$
(A.2)

and using same control law as in (4.10)

$$F_{des} = R_E^b \ (m\ddot{X}_d + mK_1\dot{e_x} + m\vec{g_E} + e_x + K_2e_1), \tag{A.3}$$

leads to

$$\dot{V}_2 = -e_x^T K_1 e_x - e_1^T K_2 e_1 + e_1^T R_b^E F_{err},$$
(A.4)

Following the same steps for the moments, changing M_{des} to be M_{act} in equation (4.21) such that,

$$M_{act} = M_{des} - M_{err} \tag{A.5}$$

leads to

$$\dot{V}_4 = -e_{\eta}^T K_3 e_{\eta} + e_2^T (J \dot{u}_2 + \omega_b \times J_b \omega_b - M_{des} + M_{err} + T e_{\eta}),$$
(A.6)

and choosing same control law

$$M_{des} = J\dot{u}_2 + \omega_b \times J_b \omega_b + Te_\eta + K_4 e_2, \tag{A.7}$$

leads to

$$\dot{V}_4 = -e_\eta^T K_3 e_\eta - e_2^T K_4 e_2 + e_2^T M_{err}.$$
(A.8)

From equations (A.4 and A.8) it is clear that the final Lyapunov function derivative is

$$\dot{V}_5 = \dot{V}_2 + \dot{V}_4 = -e_x^T K_1 e_x - e_1^T K_2 e_1 - e_\eta^T K_3 e_\eta - e_2^T K_4 e_2 + e_1^T R_b^E F_{err} + e_2^T M_{err},$$
(A.9)

$$\begin{aligned} \dot{V}_5 &\leq -\Psi_2(x) + e_1^T R_b^E F_{err} + e_2^T M_{err}, \\ &\leq -\Psi_2(x) + [e_1^T R_b^E - e_2^T] \ \mathcal{W}_{err}. \end{aligned}$$
(A.10)

Recalling from equation (3.19) that $\mathcal{W}_{err}=-B\Lambda\hat{\Theta}\mathcal{W}_{des}$,

$$\dot{V}_5 \le -\Psi_2(x) - [e_1^T R_b^E \quad e_2^T] \quad B\Lambda \tilde{\Theta} \mathcal{W}_{des}.$$
(A.11)

Following same methedology as previous chapter, equation (A.11) is same as equation (3.23) and the term $\left(\frac{\partial V_0}{\partial x}\right)^T g(x)$ turns out to $-[e_1^T R_b^E e_2^T]$, so as equation(3.29) states, the final adaptive law is defined as:

$$\dot{\Theta}^T = \delta \quad \mathcal{W}_{des} \quad [e_1^T R_b^E \quad e_2^T] B.$$
(A.12)

Bibliography

- [1] Michel Verhaegen, Stoyan Kanev, Redouane Hallouzi, Colin Jones, Jan Maciejowski, and Hafid Smail. Fault tolerant flight control a survey. 2010.
- [2] Haider Al-Lami, Ameer Aslam, and Tanya Quigley. The evolution of flight control systems technology development, system architecture and operation. 2015.
- [3] Guillaume J.J. Ducard. Fault-tolerant Flight Control and Guidance Systems. 2009.
- [4] Abderrahmen Bouguerra, Djamel Saigaa, and Kamel Kara. Fault-tolerant lyapunov-gainscheduled pid control of a quadrotor uav. 2015.
- [5] Chee Pin Tan, Christopher Edwards, and Halim Alwi. *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. 2011.
- [6] Halim Alwi and C Edwards. Fault tolerant control using sliding modes with on-line control allocation. 2008.
- [7] Jack Goldberg, Ira Greenberg, and Thomas F. Lawrence. Adaptive fault tolerance. 1993.
- [8] J.D. Boskovic, J.A. Jackson, R.K. Mehra, and N.T. Nguyen. Multiple-model adaptive faulttolerant control of a planetary lander. 2009.
- [9] G. Tao, S.M. Joshi, and X.L. Ma. Adaptive state feedback control and tracking control of systems with actuator failures. 2001.
- [10] Ramy Rashad, Johan B.C. Engelen, and Stefano Stramigioli. Disturbance observer-based geometric control on se(3) of a fully-actuated hexarotor. 2018.
- [11] Johan Engelen Ramy Rashad, Petra Kuipers and Stefano Stramigioli. Design, modeling, and geometric control on se(3) of a fully-actuated hexarotor for aerial interaction. 2017.
- [12] The Federal Aviation Administration. *Helicopter Flying Handbook*, chapter 2. Federal Aviation Administration, 2012.
- [13] V.artale, Angela Ricciardello, and C. Milazzo. Mathematical modeling of hexacopter. 2013.
- [14] Thomas S. Alderete. Simulator aero model implementation. 1995.
- [15] Samir BOUABDALLAH. design and control of quadrotors with application to autonomous flying, 2007.
- [16] E. Lavretsky. Adaptive control: Introduction, overview, and applications. 2008.
- [17] ioannousun. Robust adaptive control. 2003.
- [18] DANIELLE COLLINS. The relationship between voltage and dc motor output speed, 2015.
- [19] Marc Bodson and Susan A. Frost. Load balancing in control allocation. 2011.
- [20] Tor A.Johansen and Thor I.Fossen. Control allocation a survey. 2013.
- [21] Ramy Rashad. Robust model reference adaptive control of a quadrotor unmanned aerial vehicle, 2015.
- [22] G.P. FalconÃň, J. Angelov, and F Holzapfel. Hexacopter outdoor flight test results using adaptive control allocation subject to an unknown complete loss of one propeller. 2016.
- [23] G.P. FalconÃň and F Holzapfel. Adaptive fault tolerant control allocation for a hexacopter system. 2016.
- [24] Carlos A. Arellano-Muro, Luis F. Luque-Vega, B. Castillo-Toledo, and Alexander G. Loukianov. Backstepping control with sliding mode estimation for a hexacopter. 2013.
- [25] Hassan Khalil. Non-Linear Systems. Prentice Hall, 2002.