Technische Universiteit
Eindhoven
University of Technology

Department of Mechanical Engineering
Den Dolech 2, 5612 AZ Eindhoven
P.O. Box 513, 5600 MB Eindhoven
The Netherlands

Author
Thomas Beumer (0766510)

External supervisor:
prof. L. Wang
Yuankang Zhu

Internal Supervisor:
prof. dr. H. Nijmeijer

Date
January 24, 2017

Fault Tolerant Control of a Quadrotor Unmanned Aerial Vehicles

Traineeship RMIT University
Department of Mechanical Engineering
Dynamics & Control
DC 2017.023

List of symbols

Notation

x	Vector
X	Matrix
x^T	Transpose of vector x
X^T	Transpose of matrix X
X^{-1}	Inverse of matrix X

Symbols

b	Thrust coefficient		
d	Distance	Meters	m
I_{xx}	Inertia on x-axis	Kilogram-square meters	Kgm^2
I_{yy}	Inertia on y-axis	Kilogram-square meters	Kgm^2
I_{zz}	Inertia on z-axis	Kilogram-square meters	Kgm^2
k	Drag coefficient		
p	Body frame angular velocity around the x-axis	Radians per second	rad/s
θ	Pitch angle	Radian	rad
q	Body frame angular velocity around the y-axis	Radians per second	rad/s
r	Body frame angular velocity around the z-axis	Radians per second	rad/s
T	Thrust	Newton	N
τ	Torque	Newton per meter	Nm
ϕ	Roll angle	Radian	rad
ψ	Yaw angle	Radian	rad
ω	Rotors angular velocity	Radians per second	rad/s

Technische Universiteit
Eindhoven
University of Technology

Table of contents

Title
Fault Tolerant Control of a Quadrotor
Unmanned Aerial Vehicles

Abstract

The interest in unmanned aerial vehicles of civilian and military applications has significantly increases the past couple of years. In particular quadrotor UAVs, that have the advantage of high manoeuvrability, a compact design, and relatively low costs, providing researchers a suitable research platform.
Safety, reliability and performance gain interest due to its importance and the challenging hardware limitations of a quadrotor UAV. This report investigates the control problem of a quadrotor UAV which is experiencing a partial or complete loss of effectiveness in one of its rotors. The used PID controllers are designed for fault free conditions using automatic tuning. The possibility of changing the original control strategy to a fault tolerant control strategy will be examined. This fault tolerant control strategy must be able to recover the quadrotor UAV to stable hover condition without changing the quadrotor dynamics. Further, the behaviour of the quadrotor in faulty conditions is also investigated for a model predictive control strategy. The performance of the quadrotor is evaluated through numerical simulations. It is shown that the complete loss of effectiveness comes with many difficulties that might be solvable when the control of one degree of freedom is sacrificed to maintain the control of the most important variables to land the quadrotor safely.

1 Introduction

Remote controlled and autonomous unmanned aerial vehicles (UAVs) are increasingly changing the world. UAVs do not require an onboard human pilot, which allows for a more compact, safer and cheaper design of the device. The tasks of full-size aerial vehicles, such as traffic and weather monitoring, inspection of dangerous or inaccessible areas, surveillances, as well as photo and video applications, can be executed by those smaller UAVs.
Commonly used UAVs are rotorcrafts which come in multiple configurations, including helicopters and multirotor UAVs. A quadrotor, of which an example is shown in 1.1, is a multirotor UAV equipped with four pitch fixed rotors producing a vertical thrust attached on a cross shaped frame. Basic motions are generated by variation of the four rotor speeds which makes them, compared to fixed wing UAVs, highly maneuverable and able to perform complex movements, making them useful for a wide range of applications. The ability to vertically take off and land, fly indoors and requires little human intervention, provide great advantages for military support in dangerous situations to create situation awareness in a responsible way. Another advantage is the relatively cheap equipment, making it an interesting platform for researchers to investigate new ideas in flight control theories, navigation, etc.

Figure 1.1: Example of a quadrotor UAV

Safety, reliability and performance are extremely important to operate the UAV properly and avoid accidents. One of the advantages of a quadrotor UAV is the mechanical simplicity. However, the quadrotor has six degrees of freedom (DOF) and only four actuators which makes it an under-actuated device with highly coupled states. When there is rotor damage, the absence of any kind of redundancy results directly in unbalanced torques, a loss of control, and shortly after, a crash. Damage can be caused by flying in extreme weather conditions or hitting objects causing the rotor to break. Partial or complete rotor failure, considered here as a partial or complete loss of effectiveness, cause undesired and unpredictable behaviour of the system when it is controlled as a fully operating system. Partial or complete rotor failure is often referred to as faulty conditions.

On the other hand, if it is possible to implement a fault-tolerant controller after fault detection, there might be a possibility to land the quadrotor safely and protect the electronic equipment. There exists a wide selection of literature proposing many control methods for quadcopter UAV control. Gain-Scheduled Proportional-Integral-Derivative (GS-PID) control [1] and sliding mode control [2] already showed the possibility of trajectory tracking under partial rotor failure without changing the dynamics of the quadrotor UAV. However, the literature on control strategies for complete rotor failure varified with experimental results is relatively low due to the commercial opportunities. Simulation based flight control of a quadrotor UAV subsequent to rotor failure has been addressed in [3] and [4], where a robust feedback linearization controller is developed to safely land the aerial vehicle after rotor failure. For this work, a PID controller was chosen to control the quadrotor UAV because it is often neglected as a fault tolerant controller. Furthermore, the hardware implementation of PID controllers do not pose difficulties. However, the behaviour of the quadrotor in faulty conditions is also investigated for a model predictive control strategy.

The purpose of this research report is to design a controller to stabilize the quadrotor using automatic tuning and propose a fault tolerant control strategy for the quadrotor that encounters partial or complete loss of effectiveness of one of its actuators. This fault tolerant controller must be able to recover the quadrotor UAV to stable hover condition without changing the quadrotors dynamics. This project initially aims to achieve this using a PID control approach using an automatic tuning technique. However, a model predictive control approach will also be examined.

First, the non-linear model of the quadrotor UAV is elaborated in chapter 2. The design of a PID controller for the quadrotor UAV without faulty conditions using an automatic tuning approach is discussed in Chapter 3. Partial loss and complete failure are investigated and a new control architecture is proposed in chapter 4. The results are discussed in chapter 5 and conclusions and recommendations are given at the end of the report.

2 Dynamic Model of the Quadrotor UAV

This chapter elaborates on the dynamic model of a quadrotor UAV considering the translation and rotation dynamics with their corresponding frames. Moreover, the chapter will investigate the actuator dynamics and physical constraints of the aerial vehicle.

2.1 Overview

The quadrotor UAV is employed with four dc-motors attached to a symmetric frame where the arms make an angle of 90°with each other. The dc-motors actuate each one rotor, resulting in a lifting force and the ability to move without adjusting the pitch on the rotor blades as they spin around. For conventional helicopters a tail rotor is needed to manoeuvre. However, the fact that only four actuators are used to control six degrees of freedom (DOF), makes it an underactuated system with strongly coupled states. The altitude of the quadrotor UAV is directly influenced by the four actuators. When the rotors provide sufficient thrust against the gravitational force, the quadrotor maintains its altitude. The rotations about the principal axes are produced by torques resulting from a variation in distribution of the aerodynamic force about the vehicle's center of mass. The attitude does not change when all rotors produce the same rotor speed. During the rest of this research the situation where the attitude and altitude of the vehicle are hold is referred to as 'hover condition'. The quadrotor accomplishes translations along the x- and y-axis by an interaction of the previous mentioned degrees of freedom.

2.2 Coordinate system

In order to track the position and attitude of the quadrotor when in motion, two coordinate systems are required. The two frames are used to describe translation and rotation of the aerial vehicle in motion. Figure 2.1 gives a schematic overview of the quadrotor UAV and its coordinate systems where the body frame is denoted with the label b and is fixed in the centre of gravity of the quadrotor. The earth frame is taken as inertial frame and is attached to the earth. The quadrotor's attitude is defined by the Euler angles roll, pitch and yaw, which describe the rotation around the x,y and z-body axis respectively and is referred to as rotation vector $\zeta = [\phi, \theta, \psi]^T$, expressed in radians (rad). Furthermore, the angles for roll and pitch are bounded between $\pm\frac{\pi}{2}$ rad, to exclude it from making acrobatic manoeuvres. However, the heading of the vehicle can vary freely since the yaw angle ψ is unrestricted. The position vector $\xi = [x, y, z]^T$ is expressed in meters and relates the body fixed frame to the earth frame. The rotation matrix translates the angles in the body-fixed frame to the inertial frame and is given by R, defined as

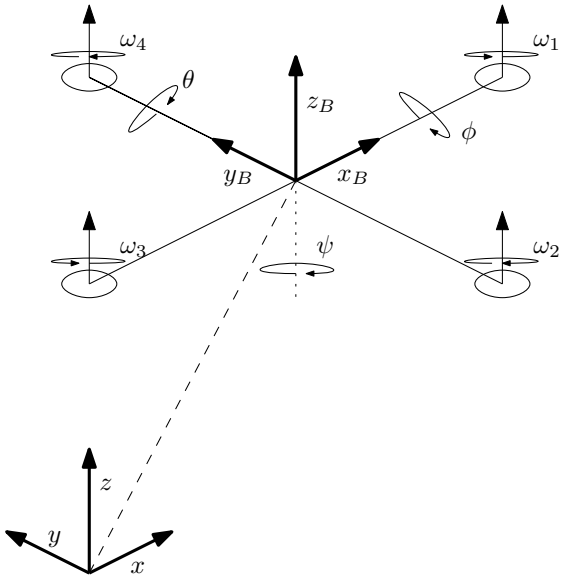

Figure 2.1: Quadrotor structure with inertial and body frame

$$R = \begin{bmatrix} C(\phi)C(\theta) & C(\phi)S(\theta)S(\phi) - S(\psi)C(\phi) & C(\psi)S(\theta)C(\phi) + S(\psi)S(\phi) \\ S(\psi)C(\theta) & S(\psi)S(\theta)S(\phi) + C(\psi)C(\phi) & S(\psi)S(\theta)S(\phi) - C(\psi)S(\phi) \\ -S(\theta) & C(\theta)S(\phi) & C(\theta)C(\phi) \end{bmatrix} \tag{2.1}$$

in which $S(i) = sin(i)$ and $C(i) = cos(i)$ with $i = \phi, \theta, \psi$ and where the inverse of R equals its transposed. This relation is presented to gain understanding of the quadrotor UAV but cannot be used in the rest of the research due to the absence of sensors needed to perform trajectory tracking. The relationships between the Euler angular velocities and the body frame angular velocities, $\omega = [p, q, r]^T$, are described in the following differential equation:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S(\phi)T(\theta) & C(\phi)T(\theta) \\ 0 & C(\phi) & -S(\phi) \\ 0 & S(\phi)/C(\theta) & C(\phi)/C(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.2}$$

Notice, when roll, pitch and yaw angles are small, $p \approx \dot{\phi}, q \approx \dot{\phi}, r \approx \dot{\phi}$. Due to small change of angles when in motion this angular boundary is commonly never reached. However, for a quadrotor that is subject to a fault, these conditions do not hold.

2.3 Actuator Dynamics

As stated at the beginning of this chapter, the quadrotor has four rotors, labelled in Figure 2.1 from 1 to 4. To achieve an accurate model, the forces and moments on the quadrotor needs to be taken into account. The total upward thrust equals the sum of the rotor angular speeds squared [5]

$$T = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{2.3}$$

where b is the thrust constant determined by air density ρ, the length of the blade l and the blade radius r. Pairwise differences in rotor angular speed ω_i with $i = 1, 2, 3, 4$, referring to the rotor number, results in the torques τ_x, τ_y and τ_z in the body frame and causes the aerial vehicle to rotate about the x,y or z-axis. The torques around the x-axis and y-axis are given by

$$\tau_x = db(\omega_4 - \omega_2) \tag{2.4}$$
$$\tau_y = db(\omega_3 - \omega_1) \tag{2.5}$$

where d is the distance from the motor to the mass center. The total torque around the z-axis, τ_z, is a result of the four rotors due to imbalance in clockwise and counter clockwise rotor speeds when the algebraic sum does not equal zero according to

$$\tau_z = k(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \tag{2.6}$$

The aerodynamic drag constant k is determined by the same factors as b. The moment of inertia of the quadrotor UAV determines the torque needed for a desired angular acceleration about the rotational axis and equals diagonal matrix I,

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \tag{2.7}$$

in which $I_{xx} = I_{yy}$ due symmetry of the quadrotor UAV. Euler's equation of motion

$$I \cdot \dot{\omega} + \omega \times (I \cdot \omega) = \tau \tag{2.8}$$

is used to define the body frame angular accelerations and gives

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})qr/I_{xx} \\ (I_{zz} - I_{xx})qr/I_{yy} \\ (I_{xx} - I_{yy})qr/I_{zz} \end{bmatrix} + \begin{bmatrix} 1/I_{xx} & 0 & 0 \\ 0 & 1/I_{yy} & 0 \\ 0 & 0 & 1/I_{zz} \end{bmatrix} \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{2.9}$$

with I_{xx}, I_{yy} and I_{zz} the moments of inertia for the three axes in x,y, and z directions. Note that the drag force is neglected in the computation of the moment based on the findings in [6] where it only causes a neglectable disturbance. Newtons's second law is used to describe the translational dynamics of the quadrotor in intertial frame

$$m\ddot{\xi} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix}. \tag{2.10}$$

An important assumption made in the dynamic derivation is the neglecting of aerodynamic changes during take-off and landing, the so called 'ground effects' [7]. For simplicity the quadrotor is already in hover condition at a set altitude when it encounters rotor failure. However, for future work where autonomous take off and landing is requested, these effects cannot be ignored.

2.4 Constraints

A simplified model will be used for control purposes. Nevertheless, the unmodelled dynamics should not be neglected. The blade gyroscopic effects which are generated by the difference in rotor speeds can be neglected for a normal functioning quadrotor, but might be of influence during failure. To take restriction regarding the dc-motors into account, an anti-windup mechanism is applied to the rotor speed of each motor

$$\omega_{min}^2 \leq \omega_i^2 \leq \omega_{max}^2 \text{ for } i = 1, 2, 3, 4. \tag{2.11}$$

Now the direction in which the rotor spins is fixed. Additionally, the rotors are rate limited which means the rotors need a certain time to reach the desired rotational speed defined as

$$-\Delta\omega_i^2 \leq \frac{d\omega_i^2(t)}{dt} \leq \Delta\omega_i^2 \tag{2.12}$$

where

$$\frac{d\omega_i^2(t)}{dt} \approx \frac{\omega_i^2(t_i) - \omega_i^2(t_{i-1})}{\Delta t}. \tag{2.13}$$

2.5 Conclusions

This chapter explains the basic concepts for the quadrotor in motion and the dynamics. Furthermore, coordinate systems are presented to describe the translation and rotation of a quadrotor UAV. The outputs of the control system are the three Euler angles, ϕ, θ, ψ, following the three reference signals ϕ^*, θ^*, ψ^*. The control variables are the three torques, τ_x, τ_y, τ_z. A complete representation of the system is given, where assumptions are made for simplicity, control purposes and feasibility. As a final note, position control is out of the scope of this research project because of the absence of sensors to track the translation of the quadrotor available for implementation.

3 Controller Design

The Quadcopter is essentially an under-actuated system with six degrees of freedom and only four actuators. Therefore, the control of all states simultaneously has been excluded. This section starts with a literature review to discuss the most proposed control techniques. Furthermore, the attitude control structure and an automatic tuning technique is discussed.

3.1 Literature review

Many control strategies have already been applied to control the attitude of the quadrotor UAV. PID control is a widely used control techniques and elaborated in [8] and [1] in combination with a gain scheduling control strategy. In [8] a fault tolerant control strategy is proposed using sliding mode control. This study shows promising results for a rotor losing half of its thrust needed to hover when encountering partial failure and is able to control the height. The quadrotor can land horizontally, sacrificing the control on yaw angle in faulty mode. The new control strategy does not take the faulty rotor into account to control roll and pitch angles, but is still capable of producing thrust to maintain a preferred height without changing the dynamics.

A feedback lineariztion approach to fault tolerance in quadrotor vehicles is elaborated in [9]. The attitude control problem is solved, taking the nonlinear dynamics of the vehicle into account. The quadrotor UAV under faulty conditions spins around the vertical axis with a constant rotational speed. The rotor couple which is not encountering failure is used to achieve a desired altitude by producing more thrust. Again the ability of controlling all torques and the altitude is lost. To maintain controllability of roll, pitch and altitude, the control of yaw is sacrificed. The quadrotor is able to stabilize after detecting the fault and can continue its flight with three rotors.

In [10] a cascaded model predictive control (MPC) strategy is used to control the attitude and altitude of the quadrotor UAV under normal condition. This research will expand this model with the control strategy that will be proposed. This strategy has the ability to set constraints and predict the future path on roll and pitch angles which is a great advantage compared to feedback linearization and PID control. The most common literature investigating complete rotor failure is based on feedback linearization controllers such as in [11]. The FTC strategy is able to control the aerial vehicle with complete rotor failure and land it safely.

All control strategies use a double control loop architecture in which a fast inner loop controls the attitude angles and height. The slower outer loop modifies the desired attitude angles to set a trajectory.

This project aims to get the quadrotor UAV hovering, roll and pitch angle are zero degrees, under partial and complete loss of effectiveness of one rotor, without changing the dynamics. A PID control method is often neglected when it comes to a fault tolerant controller. Since PID excels in simplicity and is the most used control approach, it is worth looking into the possibilities. Furthermore, a strategy using model predictive control is implemented as well to investigate the proposed strategy and evaluate the behaviour of the quadrotor under faulty conditions. As mentioned before, trajectory tracking is not taken into account because of limitations regarding the quadrotor UAV that is available for experimentation. Therefore, position control will not be elaborated here.

3.2 Cascaded PID control strategy

Now, a PID controller design is proposed making use of an automatic tuning method to achieve appropriate controller values which is discussed in the next section. In [12] an automatic tuning method is proposed for UAVs with a cascaded control structure. The method is implemented on a fixed-wing UAV and will now be used to design PID controllers for the quadrotor UAV.

The mathematical model of quadrotor UAV considered here is an integrating system and is controlled by a cascade PI control structure. This structure can effectively deal with the two sets of nonlinear dynamic equations. For attitude control, the objective is to effectively control the three Euler angles ϕ, θ, ψ, following the three reference signals ϕ^*, θ^*, ψ^*, with ϕ, θ, ψ the outputs of the control system and the primary variable to achieve attitude control. The control signals are the three torques, τ_x, τ_y, τ_z. The bodyframe angular velocities p, q and r are the secondary variables because they are directly related to the manipulated variables τ_x, τ_y and τ_z. The quadrotor closed-loop control system block diagram in cascade control can be seen in Figure 3.1. The torques are manipulated by independently designed PI controllers, because the three axis are almost decoupled. The reference signal for this so called inner loop is generated by the slower outer loop controllers.

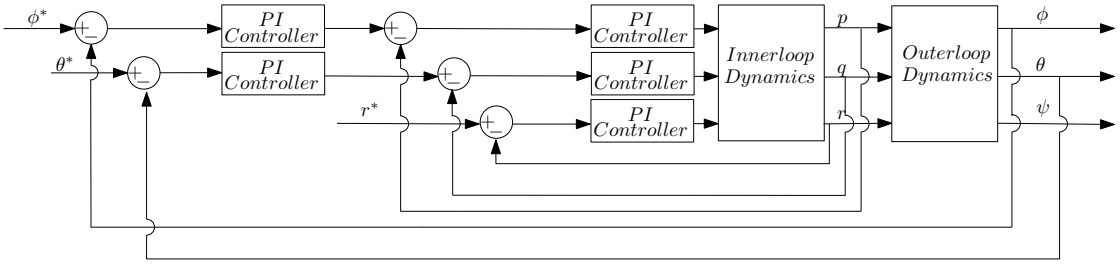

Figure 3.1: Quadrotor closed-loop control system block diagram in cascade control

The general PID control law for the inputs is stated as

$$u_i(t) = K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(\tau) d\tau + K_c \tau_D \frac{de(t)}{dt} \tag{3.1}$$

with τ_I the integral gain and τ_D the derivative gain. The error signal is represented by $e(t) = r(t) - y(t)$ with a set reference $r(t)$ and the output $y(t)$. However, in this case, only PI controllers

are used to simplify hardware implementation. The inner loop control system could be controlled using only proportional control. The use of PI controllers would eliminate the steady-state error, but slightly complicates hardware implementation. For a quadrotor, the manipulative inputs are the three torques τ_x, τ_y and τ_z, which are directly related to the rotors angular speed ω_i^2 following

$$\tau_x = db(\omega_4^2 - \omega_2^2) \tag{3.2}$$
$$\tau_y = db(\omega_3^2 - \omega_1^2) \tag{3.3}$$
$$\tau_z = k(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2). \tag{3.4}$$

Important to notice from Figure 3.1 is the absence of control on yaw angle. Instead, the yaw angular velocity is controlled, which can be accurately measured, which was not possible for the yaw angle due to the lack of equipment. For each individual case the control signals are calculated using

$$u_p(t) = K_c^p(p^* - p) + \frac{K_c^p}{\tau_I^p}\int_0^t (p^*(\tau) - p(\tau))d\tau \tag{3.5}$$
$$u_q(t) = K_c^q(q^* - q) + \frac{K_c^q}{\tau_I^q}\int_0^t (q^*(\tau) - q(\tau))d\tau \tag{3.6}$$
$$u_r(t) = K_c^r(r^* - r) + \frac{K_c^r}{\tau_I^r}\int_0^t (r^*(\tau) - r(\tau))d\tau. \tag{3.7}$$

The outer loop, controlling the roll and pitch angles, uses two PI controllers

$$u_\phi(t) = K_c^\phi(\phi^* - \phi) + \frac{K_c^\phi}{\tau_I^\phi}\int_0^t (\phi^*(\tau) - \phi(\tau))d\tau \tag{3.8}$$
$$u_\theta(t) = K_c^\theta(\theta^* - \theta) + \frac{K_c^\theta}{\tau_I^\theta}\int_0^t (\theta^*(\tau) - \theta(\tau))d\tau \tag{3.9}$$
$$\tag{3.10}$$

The next section discusses the use of an autotuner to find appropriate values for the control parameters.

3.3 Autotuner Design

Manually selection of control parameters of multiple PI(D) controllers is a time consuming and challenging activity. This section will propose a elaborate on an automatic tuning technique.

Automatic tuning is used to find the optimized parameters for the PI controllers. It can automatically find the mathematical model of the plant to be controlled [5]. A simple and easy to execute identification experiment is designed to ensure the collection of input and output data contains useful information for controller design. A closed-loop system is required to be stable for safety of equipment during the experiments. The relay feedback control diagram can be seen in Figure

3.2. As mentioned earlier, the system has a cascaded control structure. The PI controller for the inner loop is tuned first for stabilization of p, q and r, after which the outer loop can be found in a similar way. A proportional controller with known gain K_T is used to stabilize the integrating system. The relay feedback control experiment calculates the relay feedback error,

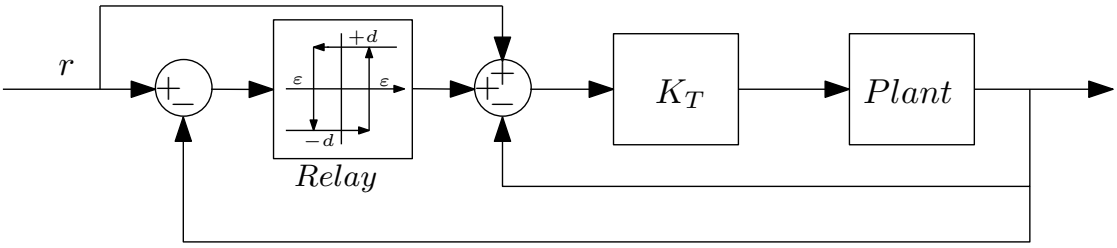

Figure 3.2: Relay feedback control for inner loop

$$e(t_k) = r(t_k) - \bar{y}(t_k), \tag{3.11}$$

and checks if

$$\text{if } |e(t_k)| \leq \varepsilon \text{ then } \bar{u}(t_k) = \bar{u}(t_{k-1}) \tag{3.12}$$
$$\text{if } |e(t_k)| > \varepsilon \text{ then } \bar{u}(t_k) = r(t_k) + a \times sign(e(t_k)) \tag{3.13}$$

in which the reference signal $r(t)$ represents a steady state operation of the plant [13]. ε is the hysteresis selected to avoid the possible random switches caused by the measurement noise and Ra is the amplitude of the relay. The actual output of the measurement is represented by $\bar{y}(t)$.

The data of the relay feedback control experiment is then used to identify the frequency response of the closed loop system. First the frequency response of the closed loop system, $T(e^{j\omega})$, is directly estimated using Fast Fourier Transform (FFT) after which the discrete-time open loop frequency response can be estimated by

$$G(e^{j\omega_d}) = \frac{1}{K_T} \frac{T(e^{j\omega_d})}{1 - T(e^{j\omega_d})}$$

with known K_T at frequency ω_d. The estimate of $T(e^{j\omega})$ uses Fast Fourier Transform. The FFT of the input signal $u(k)$, with $k = 1, 2, ..., L$, is

$$U(n) = \frac{1}{L} \sum_{k=1}^{L} u(k) e^{-j\frac{2\pi(k-1)(n-1)}{L}} \tag{3.14}$$

with data length L. The corresponding FFT of the output is

$$Y(n) = \frac{1}{L}\sum_{k=1}^{L} y(k)e^{-j\frac{2\pi(k-1)(n-1)}{L}} \tag{3.15}$$

where $n = 1, 2, ..., L$ and the discrete frequency $\omega_d = 0, ..., \frac{2\pi(L-1)}{L}$. The discrete-time frequency response $G(e^{j\omega_d})$ is a close approximation to its continuous-time frequency response under the assumption that the system operates in a fast sampling environment, where the equivalent continuous-time frequency is $\omega_1 = \frac{\omega_d}{\Delta_t}$ [13]. The continuous-time frequency response can be approximated as

$$G_p(j\omega_1) \approx G(e^{j\omega_d}). \tag{3.16}$$

For an integrating plus time delay system, a single frequency is sufficient to determine its gain K_p and time delay d. The approximate model of an integrating system is assumed to be of the form

$$G_p(s) = \frac{K_p e^{-ds}}{s} \tag{3.17}$$

K_p and d are found using the estimated $G_p(j\omega)$ as

$$K_p = \omega|G_p(j\omega)| \text{ , where } |e^{-jd\omega}| = 1 \tag{3.18}$$

$$d = -\frac{1}{\omega}tan^{-1}\frac{Im(jG_p(j\omega))}{Re(jG_p(j\omega))} \tag{3.19}$$

The controller parameter values are calculated by

$$\tau_{cl} = \beta d \tag{3.20}$$

$$K_c = \frac{\hat{K}_c}{dK_p} \tag{3.21}$$

$$\tau_I = d\hat{\tau}_I \tag{3.22}$$

$$\tau_D = d\hat{\tau}_D \tag{3.23}$$

for $\xi = 0.707$, with β a scaling factor for the desired closed-loop time constant and

$$\hat{K}_c = \frac{1}{0.328\beta^2 + 0.0786\beta + 0.6442} \tag{3.24}$$

$$\hat{\tau}_I = -3.7845\beta^2 + 10.2044\beta - 4.0298 \tag{3.25}$$

$$\hat{\tau}_D = \frac{1}{-1.9064\beta^2 + 6.1545\beta - 1.5875} \tag{3.26}$$

3.4 Control Validation

In this section the behaviour of the quadrotor UAV with the designed controller, is evaluated through numerical simulations under normal flight conditions. The quadrotor should follow a commanded trajectory. Since there is no position control to set a reference for the Euler angles, so a trajectory is set manually for both roll and pitch angle. As can be seen in Figure 3.3, the system can follow the set reference. A comparison between Figure 3.3 and 3.4 confirms that the so called inner loop, controlling p, q and r, is significantly faster than the outerloop, which primary objective is to achieve desired closed-loop performance. The control variables, τ_x, τ_y, τ_z and the total thrust as shown in Figure 3.5. The torques saturate at ± 0.1 Nm to provide a feasible solution to the rotors. Once the control signals are decided by the cascade feedback controllers, the rotor speed of each rotor will be uniquely determined and perform the desired control action. As can be seen in Figure 3.6, the rotor pairs 1-3 and 2-4 both equally contribute to the control action. All variables are controlled, except for the yaw angle due to the absence of suitable measurement equipment at the available quadrotor UAV. Instead, the yaw angular velocity is controlled, causing the quadrotor to stabilize at an angle.

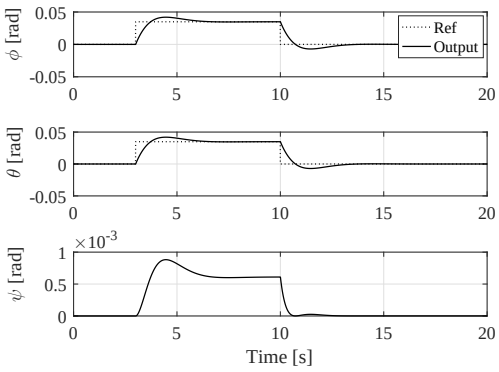

Figure 3.3: Euler angles of the quadrotor UAV following a commanded trajectory. The yaw angle cannot be controlled because there is no angle measurement.

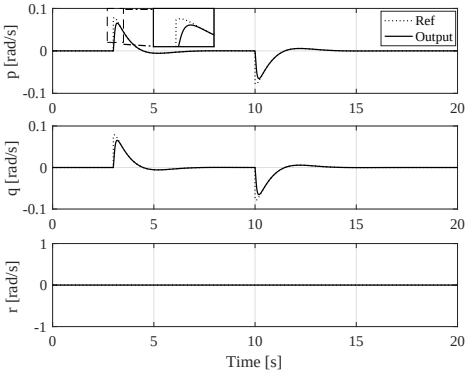

Figure 3.4: Angular velocity output following the reference set by the outerloop control.

This section proposed a cascaded control architecture, controlled by five PI controllers, designed by an automatic tuning technique. The behaviour of the quadrotor is evaluated for this controller design through numerical results and show sufficient performance under normal operating conditions. The next section investigates the behaviour of a quadrotor UAV encountering a loss of effectiveness in one of its rotors.

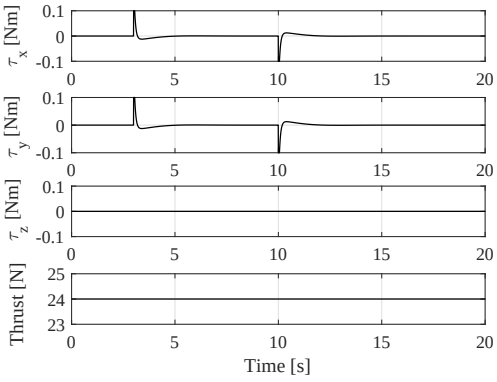

Figure 3.5: Control variables of the quadrotor system, directly related to p, q and r.

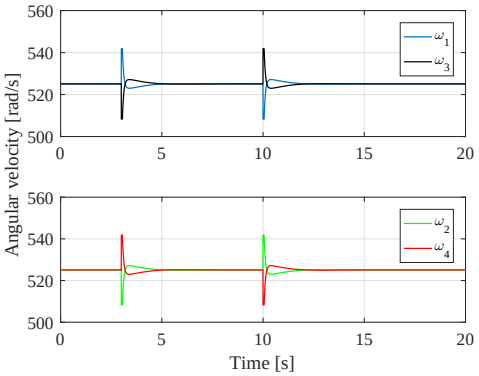

Figure 3.6: Angular velocity output following the reference set by the outerloop control.

4 Fault Tolerant Controller Design

This section first investigates the need for a fault tolerant controller by simulation of a rotor losing its effectiveness. Throughout this chapter it is assumed that the quadrotor UAV is in hovering at $t = 0$. In this situation, all rotors rotate at the same speed, resulting in both roll and pitch angles equal to zero and the gravitational force is counterbalanced.

4.1 Partial Loss of Effectiveness

A fault is considered here as a loss of effectiveness in one of the actuators, which generally means that the rotor speed differs from the rotor speed desired by the controller. Such fault injection represent motor failure, small damage to rotors caused by obstacles or extreme weather conditions. It is assumed here that the dynamics of the quadrotor do not change significantly as a result of the damage. A rotor encountering a loss of effectiveness is formulated as

$$\omega_i = \lambda \omega_{di} \text{ with } i = 1, 2, 3, 4 \text{ and } 0 \leq \lambda \leq 1 \tag{4.1}$$

where ω_i refers to the actual rotor speed and the i^{th} actuator. The desired rotor speed by the controller is referred to as ω_{di}. Recall the actuator dynamics from Chapter 2,

$$T = b(\omega_1 + \omega_2 + \omega_3 + \omega_4) \tag{4.2}$$
$$\tau_x = db(\omega_4 - \omega_2) \tag{4.3}$$
$$\tau_y = db(\omega_3 - \omega_1) \tag{4.4}$$
$$\tau_z = k(\omega_1 - \omega_2 + \omega_3 - \omega_4). \tag{4.5}$$

In hover, all torques equal zero and all rotors produce equal thrust. Because the actuators of the under-actuated system are highly coupled, the loss of effectiveness of one rotor, immediately affects the other actuators. To recover the quadrotor in hover, all rotors should rotate with the same speed. This results in a loss of total thrust, because the rotor speed is limited by the faulty rotor. This behaviour of the quadrotor UAV under faulty conditions is studied further through simulation results.

In the first scenario, the quadrotor model is controlled by the PI controllers designed in chapter 3. After 2 seconds a loss of effectiveness fault in rotor 1 occurs, which can be seen in Figure 4.1. The Euler angles are stabilized, but the thrust decreases significantly which would cause a crash. This confirms the conclusions made from the analysis of the actuator dynamics. Under

faulty conditions one loses the ability to control all degrees of freedom. The roll and pitch angle of the vehicle are essential for a safe landing such as the control of total thrust. The yaw angle on the other hand has no contribution to a safe landing and therefore could be sacrificed to keep control of the other angles and thrust.

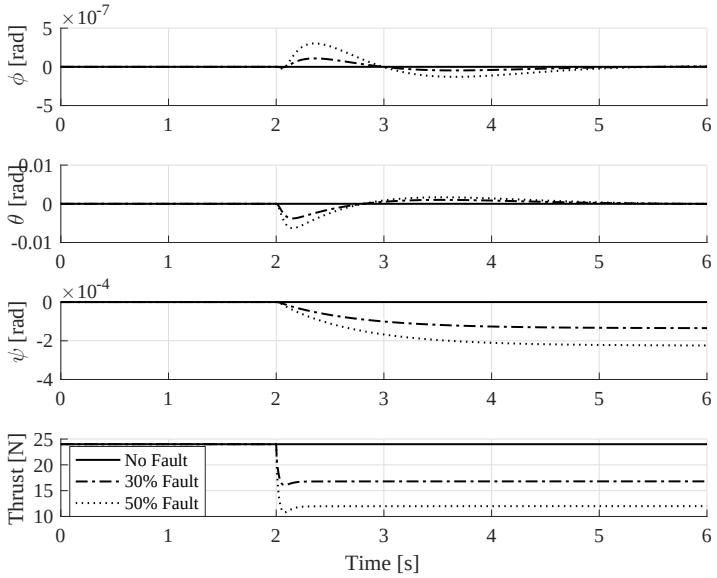

Figure 4.1: Response of the quadrotor UAV on a partial loss of effectiveness in rotor 1

A simulation under the same circumstances is performed but with implementation of a fault tolerant controller which compensates for loss of thrust and sacrifices the control of the yaw angle. The Euler angles and the produced thrust of the case with a 50% loss of effectiveness in rotor 1 is shown in Figure 4.2. As mentioned before fault detection is not the main focus of this research but is assumed to give a certain time delay to switch to the fault tolerant controller. The yaw angle goes to infinity, but the quadrotor is able to stabilize its roll and pitch angle and recover the loss in thrust. A closer look to the actuator outputs in Figure 4.3 show the coupling between the opposing rotors. After fault detection, the faulty rotor couple converge to the same rotor speed and the healthy rotor couple starts to compensate for the lost thrust.

The approach to this partial failure of rotors of the quadrotor UAV is adopted from the research done in [14], [2] and [15]. Those research paper can recover some of the control performances when a fault occurs. However, complete rotor failure is not taken into account. Complete failure of an actuator was discussed in [9] in which the same strategy for fault tolerant control is used as in [2]. The strategy proposed is again based on sacrificing the control on yaw angle which causes a horizontal rotation. Another interesting strategy is presented in [3] which claims to achieve stability and control of a quadrocopter despite the loss of one, two or three propeller, based on a defined fixed body axis with free rotation of the vehicle around this axis. The position of the vehicle is controlled by tilting of this axis combined with a variable thrust. The mentioned studies for complete loss of one or multiple actuators are based on a feedback linearization approach. The next section investigates the possibility of fault tolerant PID control designed using autotuning for complete loss of one actuator, adopting the strategy where the yaw angle

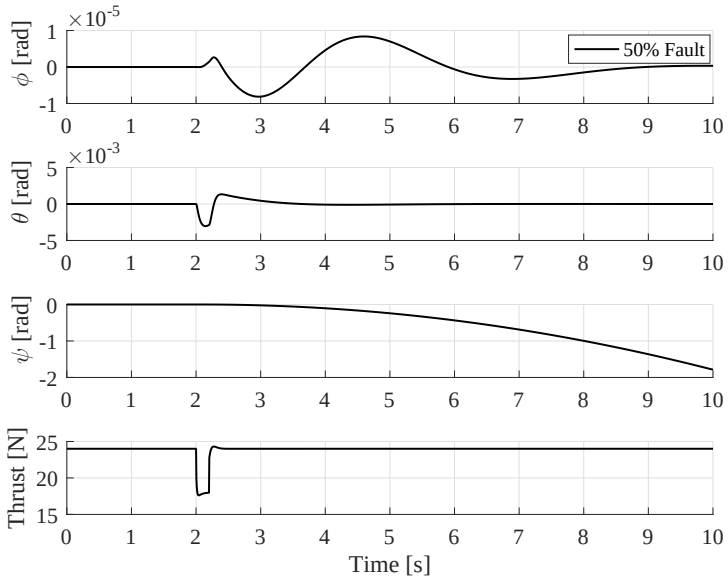

Figure 4.2: Response of the quadrotor UAV, with a fault tolerant controller, on a partial loss of effectiveness in rotor 1.

is sacrificed. This method would only change the control strategy and does not change the dynamics.

4.2 Complete loss of effectiveness

In this situation the quadrotor losses the complete effectiveness of one rotor, rotor 1. This would directly affects τ_y causing a negative value for the torque around the y-axis of the bodyframe since rotor 3 cannot react infinitely fast on the change in rotor speed of rotor 1. In the case of a partial fault, the faulty rotor was still able to produce some thrust to achieve zero torque and therefore the quadrotor could achieve hover condition. In case of complete rotor failure this is not possible and complicates the control of the Euler angles. It was already shown that the quadrotor starts to rotation around the z-axis of the body frame. According to equation 2.9 the quadrotor starts to rotate infinitely fast. However, this equation does not take the drag force of the aerial vehicle into account which is related to shape, velocity and material properties of the aerial vehicle according to [16]. This aerodynamic drag can be considered as a resistance torque depending on the rotational speed. Equation 2.9 can now be extended with this term, resulting in

$$\ddot{\psi} = -\frac{\gamma r}{I_{zz}} + pq\frac{I_{xx} - I_{yy}}{I_{zz}} + \frac{\tau_z}{I_{zz}}. \tag{4.6}$$

Now it can be seen that the angular acceleration converges to zero and the quadrotor will rotate with constant speed. Theoretically, if the rotation is constant it might be possible to use this rotation to stabilize the angle again and achieve hover. This is an unconventional way of flying but can be used to land it safely and protect the gear. A first simulation is done to see how

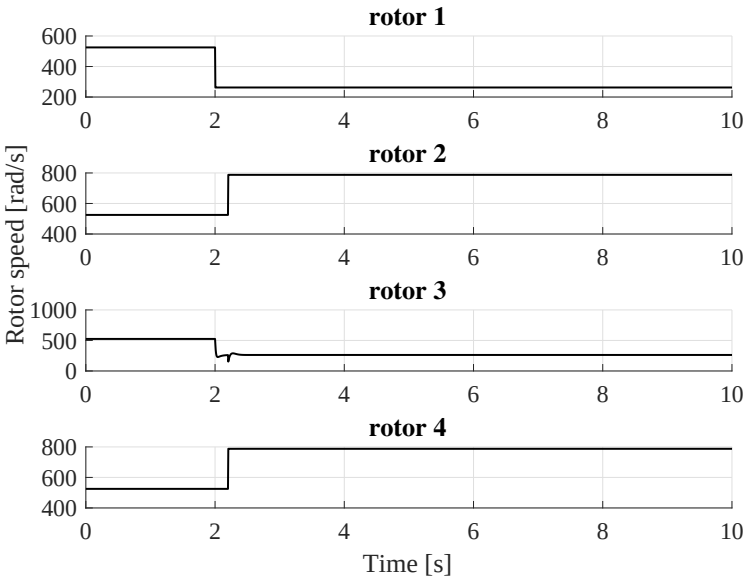

Figure 4.3: Rotor speed of the actuators, ω_1, ω_2, ω_3 and ω_4

the quadrotor behaves and how its actuated. To prevent the quadrotor from stabilizing up side down, the rotor speed of rotor 3 is limited. The constant rotor speed could not be measured, due to the absence of quadrotor and experimental setup, and is set to 3 rad/s based on [16] and is visualized in Figure 4.6. Figure 4.4 shows the rotor speed of the propellers when rotor 1 completely fails. Notice the small adjustments rotor 3 tries to make to stabilize the pitch angle. Unfortunately, the roll and pitch angle, shown in Figure 4.5, cannot be stabilized as was the first goals of this project. It is not possible to use the rotation to perfectly stabilize the quadrotor which makes the proposed control strategy not suitable for this case. Another remark must be made on the automatic tuning method to find parameters for the PI controller. The automatic tuner needs to find stabilizing controllers to work properly but could not deal with the sacrifice of a control input. As a result, handtuning is used to find suitable control parameters. Because a PI controller is a very simple controller, it might be possible to use the proposed control strategy with a model predictive control strategy which allows the use of constraints and tries to predict the optimal path.

4.3 Conclusions

A fault tolerant controller can be designed for a quadrotor encountering a partial loss of effectiveness in one of its rotors. The FTC should compensate for the loss in thrust and needs to sacrifice the control on the yaw angle. Complete loss of effectiveness of one rotor complicates stabilization, where the actuators use the rotation around the z-axis of the body frame to stabilize roll and pitch angle. These findings emphasize the need of a fault detection system to detect the fault even before it can affect the quadrotor. The next chapter investigates the possibilities of a model predictive control strategy for complete loss of effectiveness.

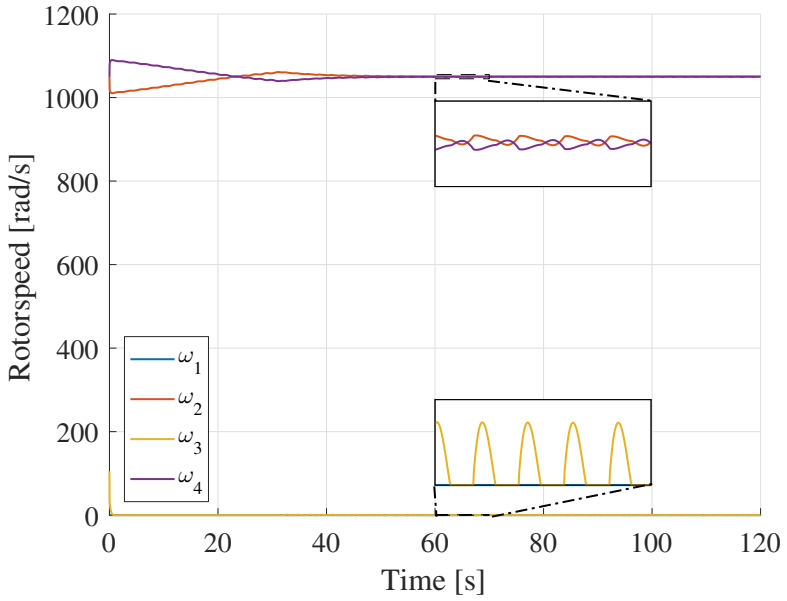

Figure 4.4: Rotorspeed of rotor 1,2,3 and 4. Rotor one is shut down from the beginning and therefore the rotorspeed is zero

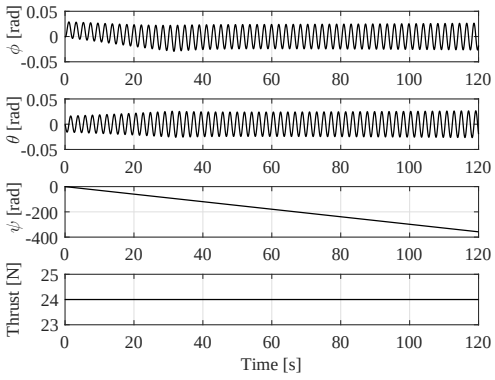

Figure 4.5: Euler angles of the quadrotor UAV rotating with constant speed where the system tries to stabilize roll and pitch angle to achieve hover condition.

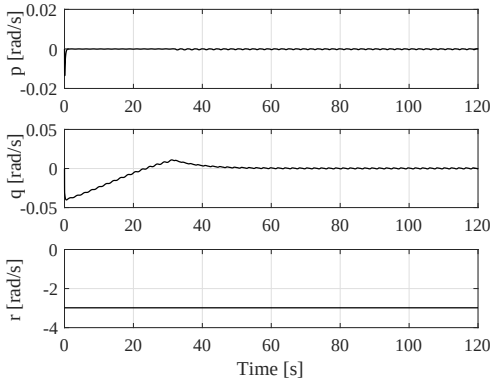

Figure 4.6: Angular velocity output, p, q and r, where r forced to an approximated value to simulated a the behaviour of the quadrotor.

5 Model Predictive Control Strategy

As concluded in the previous section, the proposed PID controller is not able to stabilize the quadrotor UAV. A discrete time model predictive control method (MPC) is now introduced, which has the advantage over PID that it is based on optimizing the future control trajectory and is capable of handeling constraints. The control strategy is based on the cascaded MPC controller design of [10] using Laguerre networks to capture the future control trajectory, which decreases the online computation demand. The control strategy proposed earlier, where the control on yaw is sacrificed to keep control on roll and pitch angle, is again applied. First, an introduction to the control strategy using MPC is given. The general idea is to introduce a discrete orthonormal basis function to generalize the design procedure and reformulate the predictive control problem.

The dynamics of the quadrotor are controlled by position- and angle control with cascaded structure. Since a linear controller is used to control this non-linear system, the controller will be most effective near the operating conditions. The state-space model for the angle control loop is linearised around the operating conditions $\phi \approx 0$, $\theta \approx 0$ and $\psi \approx 0$ and is given by

$$\dot{x}_a = A_a x_a + B_a u_a \tag{5.1}$$
$$y_a = C_a x_a \tag{5.2}$$

where the state, output and input vectors are

$$x_a = \begin{bmatrix} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \tag{5.3}$$
$$y_a = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \tag{5.4}$$
$$u_a = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \tag{5.5}$$

and the system matrices are given by

$$A_a = \begin{bmatrix} 0_{3\times3} & I_3 \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \qquad B_a = \begin{bmatrix} 0_{3\times3} \\ I \end{bmatrix}, \qquad C_a = \begin{bmatrix} I_3 & 0_{3\times3} \end{bmatrix} \tag{5.6}$$

with $I = diag(I_{xx}, I_{yy}, I_{zz})$, the inertia matrix. Notice, the state-space model for the angular control loop is here in continuous time and needs to be discretized to deal with the model predictive controller. The discrete time model of the form

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \tag{5.7}$$
$$y(k+1) = C_m x_m(k) \tag{5.8}$$

describes the plant to be controlled. The difference of respectively the state and control variable is defined by

$$\Delta x_m(k+1) = x_m(k+1) - x_m(k) \tag{5.9}$$
$$\Delta u(k) = u(k) - u(k-1). \tag{5.10}$$

The state space model can now be written as

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & 0_{m\times 1} \\ C_m A_m & I_{q\times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k). \tag{5.11}$$

For simplicity the SISO case is examined first, which will be later extended for MIMO cases. The control signal is modelled with a set of discrete Laguerre functions to approximate the sequence $\Delta u(k_i), \Delta u(k_i), \ldots, \Delta u(k_i + N_c - 1)$, with N_c is the control horizon and k_i is a certain time. The Laguerre functions are a set of discrete orthonormal basis functions of which the z-transformations can be written as

$$\Gamma_1(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}} \tag{5.12}$$
$$\Gamma_2(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}} \frac{z^{-1}-a}{1-az^{-1}} \tag{5.13}$$
$$\vdots \tag{5.14}$$
$$\Gamma_N(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}} (\frac{z^{-1}-a}{1-az^{-1}})^{N-1} \tag{5.15}$$

with a the pole of the discrete-time Laguerre network between 0 and 1 to guarantee stability [17]. However, instead of taking the z-transform of the Laguerre networks, the discrete time functions are found based on the state space realization of the networks. The set of discrete time Laguerre functions satisfies the following difference equation,

$$L(k+1) = A_l L(k), \tag{5.16}$$

where

$$L(k) = \begin{bmatrix} l_1(k) & l_2(k) & \ldots & l_N(k) \end{bmatrix}^T \tag{5.17}$$

and A_l is a $N \times N$ matrix and a function of parameter a. Here, $l_N(k)$ denotes the inverse z-transform of $\Gamma_N(z, a)$ etc. The initial condition is defined as

$$L(0)^T = \sqrt{1-a^2}\begin{bmatrix}1-a & a^2-a^3 & \dots & (-1)^{N-1} & a^{N-1}\end{bmatrix}. \tag{5.18}$$

Depending on N, for example $N=3$, A_l becomes

$$A_l = \begin{bmatrix} a & 0 & 0 \\ 1-a^2 & a & 0 \\ -a(1-a^2) & 1-a^2 & a \end{bmatrix}, \quad L(0) = \sqrt{1-a^2}\begin{bmatrix}1\\-a\\a^2\end{bmatrix} \tag{5.19}$$

For a SISO system, Δu can now be described as

$$\Delta u(k_m+k) = L(k)^T\eta, \quad \eta = \begin{bmatrix}c_1 & c_2 & \dots & c_N\end{bmatrix}^T \tag{5.20}$$

if the future control signals are considered as the impulse response of the stable system, where $c_1, c_2, \dots, c_N$ are the coefficients to be determined from the system data. Using the Laguerre functions, the prediction of future state variable at sample m, becomes

$$x(k_i+m|k_i) = A^m x(k_i) + \sum_{i=0}^{m-1} A^{m-i-1}BL(i)^T\eta \tag{5.21}$$

where $\Delta u(k_i+i) = L(i)^T\eta$ and for the output of the plant

$$y(k_i+m|k_i) = CA^m x(k_i) + \sum_{i=0}^{m-1} CA^{m-i-1}BL(i)^T\eta. \tag{5.22}$$

Now the coefficient vector η has become an optimization variable instead of ΔU. Next, MIMO systems are considered where the future incremental control $\Delta u(k_m+k)$ becomes

$$\Delta u(k_m+k) = \begin{bmatrix} L_1(k)^T & 0 & \cdots & 0 \\ 0 & L_2(k)^T & \cdots & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_p(k)^T \end{bmatrix}. \tag{5.23}$$

with input matrix B as $B = \begin{bmatrix}B_1 & B_2 & \dots & B_p\end{bmatrix}$. Now, the coefficient vector η needs to be found to minimizes the cost function defined as

$$J = \eta^T\Omega\eta + 2\eta^T\Psi x(k_i) + \sum_{m=1}^{N_p} x(k_i)^T(A^T)^m Q A^m x(k_i), \tag{5.24}$$

where Ω and Ψ are defined as

$$\Phi(m)^T = \sum_{j=0}^{m-1} A^{m-j-1} \begin{bmatrix} B_1 L_1(j)^T & B_2 L_2(j)^T & \cdots & B_p L_p(j)^T \end{bmatrix} \tag{5.25}$$

$$\Omega = \sum_{m=1}^{N_p} \Phi(m) Q \Phi(m)^T + R_L \tag{5.26}$$

$$\Psi = \sum_{m=1}^{N_p} \Phi(m) Q A^m \tag{5.27}$$

$Q = C^T \times C$ is a weighting matrix for the state variables, R_L is a diagonal weighting matrix $(N \times N)$ for control variables η. The constraints are applied on the amplitude of roll and pitch. Minimum and maximum are set on the rotor speed based on the used motor. The minimized object function, a quadratic programming problem, is described by

$$J = \frac{1}{2} x^T E x + 2 x^T F \tag{5.28}$$

$$Mx \leq \gamma \tag{5.29}$$

with E, F, M and γ matrices and vectors in the QP problem. To solve the quadratic problem Hildreths QP procedure is used

$$\gamma_i^{m+1} = \mathsf{max}(0, \omega_i^{m+1}) \tag{5.30}$$

$$\omega_i^{m+1} = -\frac{1}{h_{ii}} \left[k_i + \textstyle\sum_{j=1}^{i-1} h_{ij} \lambda_j^{m+1} + \sum_{j=1}^{i-1} h_{ij} \lambda_j^m \right] \tag{5.31}$$

with h_{ii} as the ijth element of $H = ME^{-1}M^T$ and k_i the ith element of $K = \gamma + ME^{-1}F$ and λ the lagrange multiplier. The minimization over x, including the set constraints, is defined as

$$x = -E^{-1}(F + M^T \lambda^*) \tag{5.32}$$

where λ^* is the converged Lagrange multiplier after the iteration which is always greater or equal to zero.

5.1 Simulation Results

The simulation executed here is an attempt to fly with three rotors, controlling roll, pitch and thrust. In contrast with PID control, model predictive control should be able to find the optimal solution under constraints. Again, the first simulations showed a very strong reaction of the rotor laying on the opposite end of the rotor encountering failure. Therefore the maximum rotor speed of this rotor speed is limited. The rotor can only produce just enough lifting force to modulate the pitch angle to zero. The rotor speed of the four rotors is shown in Figure 5.1. It can be seen that rotor 3 is only used to stabilize the pitch angle. Similar to the PID controlled case, rotor 3 uses small rotor speed changes to stabilize the affected Euler angle. However, the change of rotor speed is achieved gradually and the Euler angles are no longer continuously oscillating around zero degree. Figure 5.2 shows the stable Euler angles roll and pitch, and the yaw angle which goes to infinity. The stabilization takes place within a second but a couple of remarks should be made. First, the yaw angle goes to infinity due to the aerodynamic drag of the vehicle, which is not taken into account in this simulation due to the use of a MATLAB SIMULINK block, describing the dynamics of the quadrotor UAV. Important to notice is the yaw angular velocity within the first second, where rotorspeed changes are made, shown in Figure 5.3. The rotational speed around the z- axis of the bodyframe already exceeds -25 rad/s in half a second because of this unmodelled aerodynamic drag. The input signals are stated in Figure 5.4 and show the τ_z stabilizing at a certain value, resulting in acceleration around the z-axis. However, the MPC method seems to find a solution, even with high angular velocity around the z-axis, something which was not possible using PID control.

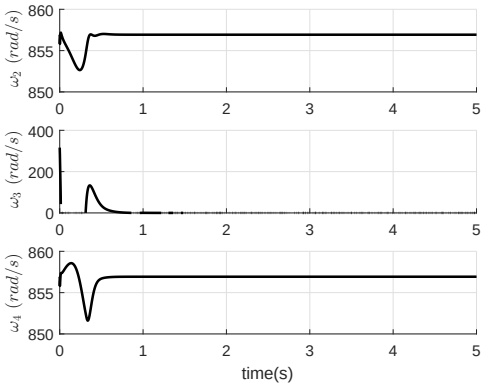

Figure 5.1: Response of rotor speed of rotor 2,3 and 4 when rotor 1 is not able to produce any thrust.

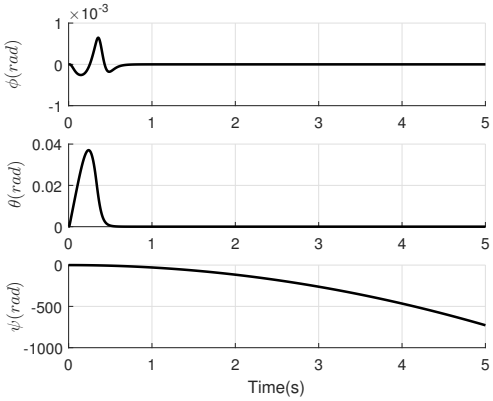

Figure 5.2: Euler angles of the quadrotor UAV when rotor 1 is shut down, causing a undesired angle on the pitch angle.

A model predictive control strategy is proposed, sacrificing the control on yaw. According to the simulations, the MPC method is able to get the quadrotor UAV in hover position. However, the angular velocities are very high and may cause problems when the controller is implemented on the hardware for experimental results. Further research is needed to validate these simulation results.

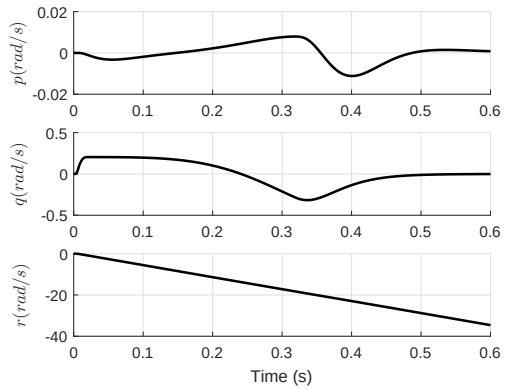

Figure 5.3: Angular velocities as a function of time for the first 0.6 seconds where stabilization actions are made.

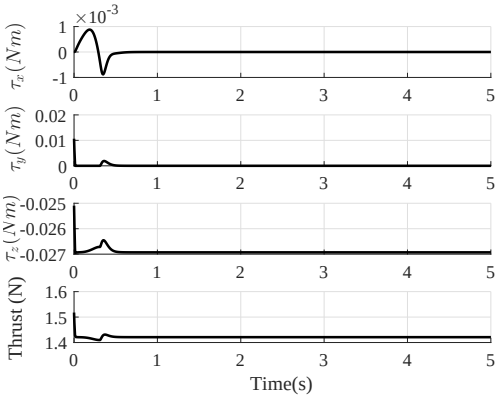

Figure 5.4: Input signals as a function of time, when the control on yaw angle is sacrificed and where τ_z becomes a constant value.

6 Conclusion

A control strategy has been presented applicable to partial and complete rotor failure of a quadrotor UAV. The fault tolerant controller should perform a safe landing of the quadrotor UAV, protecting its electronic equipment, without changing the dynamics. The performance of a fault tolerant controller using PID control and model predictive control are evaluated by simulations.

As a first step, a model of the quadrotor is presented with the corresponding coordinate systems, kinematics, actuator and quadrotor dynamics. PID autotuning control techniques are used to design controllers for the nonlinear model. Simulations are performed in MATLAB Simulink and show satisfactory performance of the designed controller when all rotors are fully operating.

Two types of fault are discussed, namely partial- and complete loss of effectiveness. The effect of each fault on the behaviour of the quadrotor is studied and a fault tolerant controller is proposed to recover the quadrotor from the loss of effectiveness. This fault tolerant controller takes the under-actuation of the system into account and needs to sacrifice one of its six degrees of freedom. Because the control of roll angle, pitch angle and thrust is essential for a safe landing an obvious choice is to sacrifice the control of the yaw angle which results in a rotation with constant speed around the z-axis of the body frame. This is a constant rotation because the aerodynamic drag of the aerial vehicle becomes a resistance torque limiting the angular acceleration. The loss of thrust is compensated by the rotor couple which is not encountering failure.

The original designed PID controller could stabilize the quadrotor for partial loss of effectiveness but to deal with complete failure, a specific controller should be designed. Unfortunately, automatic tuning could not deal with the loss of control on the yaw angle and therefore could not find suitable control parameters. The proposed autotuner could not be used during the parameter tuning procedure because the relay identification test is based on a proportional controller stabilizing the plant, which could not be found. Handtuned PID controllers resulted in a quadrotor producing sufficient thrust, but could not stabilize roll and pitch angle and therefore could not bring the quadrotor in stable hover condition using the constant rotational speed combined with the healthy rotor on opposite side of the failing rotor.

Another control technique, model predictive control, was applied because of its ability to handle constraints and predict the future path. These numerical simulations showed promising results, stabilizing roll and pitch angles at the desired angle while losing the control on yaw angle. However, the results need to be verified with experiments. From experience it is known that flying with two rotors brings challenges due to its sensitivity.

7 Recommendation

This work has made multiple assumptions to investigate the behaviour of a quadrotor. The dynamics of the quadrotor do no change due to rotor failure, in this case a broken propeller is assumed to have the same influence on the quadrotor as a stationary rotor. It is clear that a more accuracy is obtained by completely model the defect on a rotor. The dynamic model used for this research is not taking into account the failure of a rotor caused by breaking, but is assuming the rotor can no longer produce thrust, not taking the dynamics into account. Further, faults only occur in actuators, neglecting faults in sensors of the system which can also affect performance.

The simulations used in this research only investigate the *behaviour* of the quadrotor during failure. These simulations show promising results regarding stabilization of the quadrotor to hover condition, but should be verified with experimental results. Therefore, a specific quadrotor needs to be selected for experiments. In that case, it is possible to determine the maximum partial loss of effectiveness of a rotor. Physical limits and and bounds on the actuators are considered shortly but need to be more specific to make the simulation more realistic and closer to an actual quadrotor system. For example, the feasibility of the rotorspeed of the rotor on the opposite side of the faulty rotor. This rotor only produces very small peaks of rotor speed. This might not be possible to achieve for a quadrotor in an experimental setup but could work according to simulation results.

Eventually, the quadrotor UAV must be able to detect a fault itself and switch to another control strategy. However, the fault detection mechanism is not elaborated in this research report but is essential for achieving a safe landing. One possibility to detect if a quadrotor is encountering a faulty rotor is to look at the rotational speed around the z-axis of the quadrotor. Exceeding a certain angular speed could be caused by a failing rotor and therefore trigger a fail safe strategy. From the simulation it could be concluded that detection time is extremely important due to nonlinearity of the system. The highly couples actuators complicate the stabilization of the affected angle. Fast detection could solve this by gradually decreasing the rotor couple to zero degree and simplify the landing procedure because the hover condition can be achieved properly.

Another important part which should be edit is trajectory tracking. The quadrotor was not attached with sufficient sensors to do so. The path the quadrotor follows after failure is completely ignored and should be taken into account. However, if its possible to achieve stable hover, it should also be possible to follow a trajectory by adapting the unaffected rotor couple.

A last recommendation for future work is to concentrate on one particular failure. This work provides ideas on recovery of faults but does not contain a complete control strategy which can

be directly implemented into the quadrotor. The fault tolerant controller should be applicable for every partial fault to be effective.

Bibliography

[1] Youmin Zhang Alaeddin Milhim and Camille-Alain Rabbath. "gain scheduling based pid controller for fault tolerant control of quad-rotor uav". *AIAA Infotech@Aerospace 2010, Infotech@Aerospace Conferences*, 2010.

[2] Farid Sharifi, Mostafa Mirzaei, Brandon W Gordon, and Youmin Zhang. Fault tolerant control of a quadrotor uav using sliding mode control. In *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 239–244. IEEE, 2010.

[3] Mark W Mueller and Raffaello D'Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 45–52. IEEE, 2014.

[4] CeÌĄdric De Crousaz, Farbod Farshidian, Michael Neunert, and Jonas Buchli. Unified motion control for dynamic quadrotor maneuvers demonstrated on slung load and rotor failure tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2223–2229. IEEE, 2015.

[5] Liuping Wang and Xi Chen. Pid control of quadrotor. *School of Engineering, Royal Melbourne Institute of Technology University, Australia, Lecture*, 2016.

[6] Gabriel M Hoffmann, Haomiao Huang, Steven L Waslander, and Claire J Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, volume 2, page 4, 2007.

[7] Michael David Schmidt. Simulation and control of a quadrotor unmanned aerial vehicle. *University of Kentucky Master's Theses. Paper 93.*, 2011.

[8] Iman Sadeghzadeh, Ankit Mehta, Abbas Chamseddine, and Youmin Zhang. Active fault tolerant control of a quadrotor uav based on gainscheduled pid control. In *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pages 1–4. IEEE, 2012.

[9] Alessandro Freddi, Alexander Lanzon, and Sauro Longhi. A feedback linearization approach to fault tolerance in quadrotor vehicles. *IFAC Proceedings Volumes*, 44(1):5413–5418, 2011.

[10] Xi Chen and Liuping Wang. Cascaded model predictive control of a quadrotor uav. In *Control Conference (AUCC), 2013 3rd Australian*, pages 354–359. IEEE, 2013.

[11] Alexander Lanzon, Alessandro Freddi, and Sauro Longhi. Flight control of a quadrotor vehicle subsequent to a rotor failure. *Journal of Guidance, Control, and Dynamics*, 37(2):580–591, 2014.

[12] Pakorn Poksawat, Liuping Wang, and Abdulghani Mohamed. Automatic tuning of attitude control system for fixed-wing unmanned aerial vehicles. *IET Control Theory & Applications*, 2016.

[13] Liuping Wang and Xi Chen. Tuning rules and auto-tuners for pid controllers. *School of Engineering, Royal Melbourne Institute of Technology University, Australia, Lecture*, 2016.

[14] Hojjat A Izadi, Youmin Zhang, and Brandon W Gordon. Fault tolerant model predictive control of quad-rotor helicopters with actuator fault estimation. *IFAC Proceedings Volumes*, 44(1):6343–6348, 2011.

[15] Mina Ranjbaran and Khashayar Khorasani. Fault recovery of an under-actuated quadrotor aerial vehicle. In *49th IEEE Conference on Decision and Control (CDC)*, pages 4385–4392. IEEE, 2010.

[16] Raif C Gomes and George AP Th. Pid-based fail-safe strategy against the break of opposite motors in quadcopters. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 109–114. IEEE, 2015.

[17] Liuping Wang. *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, pages 85–108, 2009.