Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Modeling, Estimation and Attitude Control of an Octorotor Using PID and $\mathcal{L}_1$ Adaptive Control Techniques

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Kristoffer Bergman & Jonatan Ekström

LiTH-ISY-EX--14/4769--SE

Linköping 2014

Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Modeling, Estimation and Attitude Control of an Octorotor Using PID and $\mathcal{L}_1$ Adaptive Control Techniques

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Kristoffer Bergman & Jonatan Ekström

LiTH-ISY-EX--14/4769--SE

Handledare: **Isak Nielsen**
ISY, Linköpings universitet
Anders Pettersson
SAAB Dynamics AB

Examinator: **Daniel Axehill**
ISY, Linköpings universitet

Linköping, 11 juni 2014

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Datum
Date

2014-06-11

Språk
Language

□ Svenska/Swedish
☒ Engelska/English
□ ________________

Rapporttyp
Report category

□ Licentiatavhandling
☒ Examensarbete
□ C-uppsats
□ D-uppsats
□ Övrig rapport
□ ________________

URL för elektronisk version

http://www.ep.liu.se

ISBN

—

ISRN

LiTH-ISY-EX--14/4769--SE

Serietitel och serienummer
Title of series, numbering

ISSN

—

Titel
Title

Modellering, estimering och attitydreglering av en oktakopter med användning av PID- och $\mathcal{L}_1$-adaptiv teknik

Modeling, Estimation and Attitude Control of an Octorotor Using PID and $\mathcal{L}_1$ Adaptive Control Techniques

Författare
Author

Kristoffer Bergman & Jonatan Ekström

Sammanfattning
Abstract

A multirotor is a type of aerial vehicle that has attracted a lot of attention in recent years. Multirotors have found applications in a variety of different fields and they are used by scientists and researchers, commercial UAV companies and radio control enthusiasts alike. In this thesis a multirotor with eight rotors, also called an octorotor, is used.

A physical model of the octorotor has been developed using theory from rigid body mechanics and aerodynamics. The unknown parameters in this model have been found using several identification experiments. The model has been used for controller design and comparison in a simulation environment.

An attitude estimation algorithm has been designed and implemented on the target hardware. The algorithm is referred to as a nonlinear complementary filter and it uses a quaternion rotation representation and onboard measurements to compute an estimate of the current aircraft attitude.

Two different attitude controllers have been designed and evaluated. The first controller is based on PID techniques which are commonly used in multirotor flight stabilization systems. The second controller uses a novel control structure based on $\mathcal{L}_1$ adaptive control techniques. A baseline attitude PD controller is augmented with an $\mathcal{L}_1$ adaptive controller in the rate feedback loop. The two controller structures are compared using a simulation environment based on the developed model of the octorotor.

The results show that the proposed structure gives a notable performance increase with respect to robustness against modeling errors and input disturbance rejection compared to the PID controller. However, the $\mathcal{L}_1$ adaptive controller is more complex to implement and gives less noise attenuation. The PID controller has been implemented on the platform's hardware and initial flight tests have been performed with promising results.

Nyckelord
Keywords

Multirotor, Octorotor, VTOL, UAV, Modeling, Estimation, Attitude control, PID,
L1 adaptive control

Abstract

A multirotor is a type of aerial vehicle that has attracted a lot of attention in recent years. Multirotors have found applications in a variety of different fields and they are used by scientists and researchers, commercial UAV companies and radio control enthusiasts alike. In this thesis a multirotor with eight rotors, also called an octorotor, is used.

A physical model of the octorotor has been developed using theory from rigid body mechanics and aerodynamics. The unknown parameters in this model have been found using several identification experiments. The model has been used for controller design and comparison in a simulation environment.

An attitude estimation algorithm has been designed and implemented on the target hardware. The algorithm is referred to as a nonlinear complementary filter and it uses a quaternion rotation representation and onboard measurements to compute an estimate of the current aircraft attitude.

Two different attitude controllers have been designed and evaluated. The first controller is based on PID techniques which are commonly used in multirotor flight stabilization systems. The second controller uses a novel control structure based on $\mathcal{L}_1$ adaptive control techniques. A baseline attitude PD controller is augmented with an $\mathcal{L}_1$ adaptive controller in the rate feedback loop. The two controller structures are compared using a simulation environment based on the developed model of the octorotor.

The results show that the proposed structure gives a notable performance increase with respect to robustness against modeling errors and input disturbance rejection compared to the PID controller. However, the $\mathcal{L}_1$ adaptive controller is more complex to implement and gives less noise attenuation. The PID controller has been implemented on the platform's hardware and initial flight tests have been performed with promising results.

Sammanfattning

En multirotor är en typ av flygande farkost som har fått mycket uppmärksamhet under senare år. Multirotorer tillämpas inom flertalet områden och de används av bland annat forskare och vetenskapsmän, kommersiella UAV-företag samt hobbyentusiaster. I detta examensarbete används en multirotor med åtta rotorer, en så kallad oktakopter.

En fysikalisk modell av oktakoptern har tagits fram med hjälp av teori från stelkroppsmekanik och aerodynamik. De okända parametrarna i modellen har skattats med hjälp av ett flertal identifieringsexperiment. Modellen har använts för att designa och jämföra regulatorer i en simuleringsmiljö.

En algoritm för att skatta farkostens attityd har designats och implementerats på oktakopterns hårdvara. Algoritmen är en variant av ett olinjärt komplementärt filter och använder en kvaternionrepresentation av attitydvinklar och uppmätt sensordata för att skatta farkostens nuvarande attityd.

Två olika attitydregulatorer har designats och utvärderats. Den första regulatorn är baserad på PID-teknik vilket är vanligt för stabilisering av multirotorsystem. Den andra regulatorn använder en ny regulatorstruktur baserad på $\mathcal{L}_1$-adaptiv teknik. En grundregulator av PD-typ utökas med en $\mathcal{L}_1$-adaptiv regulator i vinkelhastighetsloopen. De två regulatorstrukturerna jämförs i en simuleringsmiljö baserad på den framtagna modellen av oktakoptern.

Resultaten visar att den föreslagna regulatorstrukturen ger en betydande prestandaökning gällande robusthet mot modellfel och undertryckning av ingångsstörningar jämfört med PID-regulatorn. Dock är $\mathcal{L}_1$-regulatorn mer komplex att implementera och den ger mindre brusundertryckning. PID-regulatorn har implementerats på plattformens hårdvara och inledande flygtester har genomförts med lovande resultat.

Acknowledgments

We would like to start by expressing our thanks to Torbjörn Crona at SAAB Dynamics AB for providing us with the opportunity to perform this master's thesis work. We would also like to thank our industrial supervisor Anders Pettersson as well as Stefan Thorstenson, Carl Nordheim and Magnus Johansson for their valuable input, guidance and support. Without their combined expertise we surely would not have got this far with our thesis. In addition, the people working at the mechanics and electronics workshop have helped us a lot with hardware related questions and test rig construction. For this we are very grateful.

Our examiner Daniel Axehill and academic supervisor Isak Nielsen have been very helpful to us during the course of the project. They have always been available for discussion regarding different control algorithms and solutions. A special thanks to Isak for his invaluable comments on our thesis report.

We also want to thank our fellow students at Linköping University who have helped make these last five years a really great experience. We have been very fortunate to be part of a tightly knit group of friends who have always supported each other through tough projects and examination periods.

Kristoffer would like to take the opportunity to show his gratitude to his father Lars, mother Maria and sister Emelie. You have been the single most important persons during my studies. I would never have made it without your support.

Finally, Jonatan would like to thank his family and fiancée Ida for their continuous support and for always believing in him. This marks the end of a seventeen year long academic career which I would never have accomplished without you.

Linköping, May 2014
Kristoffer Bergman & Jonatan Ekström

Contents

Notation

Acronyms

Acronym	Meaning
3D	Three-Dimensional
BL	Brushless
CPU	Central Processing Unit
DC	Direct Current
DFT	Discrete Fourier Transform
DoF	Degrees of Freedom
DTFT	Discrete Time Fourier Transform
EKF	Extended Kalman Filter
ESC	Electronic Speed Controller
FT	Fourier Transform
GPS	Global Positioning System
I^{2}C	Inter Integrated Circuit
JTAG	Joint Test Action Group
NED	North East Down
PCB	Printed Circuit Board
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation
RC	Radio Control
SISO	Single Input Single Output
TFD	Time-Frequency Description
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned Aerial Vehicle
UAVP-NG	Universal Aerial Video Platform-Next Generation
USB	Universal Serial Bus
VTOL	Vertical Takeoff and Landing

1 Introduction

This master thesis work is focused on modeling, state estimation and control of an unmanned multirotor aerial vehicle. A multirotor is a rotorcraft that uses more than two rotors as opposed to the standard helicopter configurations. Multirotors often use fixed pitch rotors and the vehicle is controlled by varying the rotation rate of the rotors according to some strategy. In this thesis a multirotor with eight rotors, also called an octorotor, is used. The thesis work is performed in cooperation with SAAB Dynamics AB in Linköping, Sweden.

1.1 Background

Multirotors have received a lot of interest in recent years and have become a popular choice for commercial, military and scientifical applications. Multirotors are capable of vertical takeoff and landing (VTOL) and they are highly maneuverable. This type of aircraft can be used for surveillance, transportation, mapping, photography, rescue operations and much more. Due to technological advances in batteries, sensors and electronics, high performance multirotors can now be manufactured at a relatively low cost [18]. This is one of the reasons for the increased popularity. Multirotors are also popular amongst radio control (RC) hobbyists and there exist several open source projects focused on development of multirotor autopilots [19]. A popular type of multirotor is the so called quadrotor which uses four rotors. A disadvantage with this structure is the lack of redundancy and low lift force compared to structures utilizing more rotors. By adding more rotors, it is possible to lift heavier payloads and also obtain a higher degree of redundancy.

In 2013, SAAB Dynamics AB decided to buy a multirotor for use as an experimental platform. It was deemed important that the aircraft should be able to lift relatively heavy loads. With this in mind an octorotor was bought as a ready to fly platform from a German company named HiSystems GmbH. Two students from Lund University wrote their master's thesis using this platform in 2013 (see [6]). The goal with the project was to develop and implement control algorithms for the aircraft. However, it was found that almost all of the source code for the flight and navigation control electronics was either closed or hard to modify. Hence, the students bought a new flight and navigation control card with open source code written in English. The new hardware was not integrated with the platform due to time constraints.

1.2 Purpose

The purpose of this thesis work can be divided into four different parts.

- The first part is to integrate the new flight and navigation card with the current platform so that it can be used for development and testing of new estimation and control algorithms.
- The second part is to develop a dynamic model of the aircraft which can be used for simulation and controller design. To get a complete model, several parameter identification experiments need to be carried out.
- The third part is to design and implement a state estimation algorithm that is needed by the closed-loop controllers designed for the system.
- The last part is to design and compare two different attitude controllers for the octorotor, a PID controller and a controller based on $\mathcal{L}_1$ adaptive control techniques. The controllers are compared using simulations based on the developed model with respect to reference tracking, input disturbance rejection and robustness against modeling errors. This is done to evaluate whether $\mathcal{L}_1$ adaptive control gives increased performance compared to the commonly used PID control architecture. $\mathcal{L}_1$ adaptive control is chosen since SAAB Dynamics AB is interested in how this controller structure performs when applied to different types of missiles and aircraft [25].

1.3 Related work

In the last decade there have been a lot of research done in the multirotor area. Two different modeling approaches are used, the first being traditional physical modeling using rigid body mechanics and aerodynamic relations to derive a dynamic model of the multirotor. This type of model is the most commonly used and it is thoroughly described in [21]. The second approach uses system identification techniques to obtain a model of the multirotor system based on collected data from identification experiments. This approach is described in [15].

Regarding attitude control algorithms, many different methods are described in the literature. PID controllers are very common, see e.g. [21], [19], [27] and [18]. Research results regarding $\mathcal{L}_1$ adaptive control applied to multirotors is much more sparse. However, in [22] variants of this controller architecture is used for attitude control of a multirotor.

1.4 Outline

The thesis is organized into seven chapters:

- Chapter 2 contains background information on a number of subjects that are covered in this thesis. Depending on the reader's background, some or all of the sections in Chapter 2 can be omitted.
- Chapter 3 presents the type of octorotor used in this thesis. The platform's hardware, motors and other equipment is described. In addition, some basic concepts in multirotor control are presented.
- Chapter 4 describes the physical model of the octorotor derived using rigid body mechanics and aerodynamics. This chapter also describes and presents the results from the different parameter identification experiments that were carried out on the platform.
- Chapter 5 describes how the sensors mounted on the octorotor can be used for attitude estimation together with the state estimation algorithm that is used in this thesis.
- Chapter 6 presents the results of the controller design performed in this thesis. The multirotor control architecture and the linearized model used in the controller design are described in this chapter. The design of the two different control structures are thoroughly detailed and the performance of the resulting control algorithms are compared using a simulation model based on the results from Chapter 4.
- Chapter 7 summarizes the main results of this thesis work and discusses what can be done in future projects.

2 Preliminaries

This chapter is intended to provide the necessary background information about different subjects covered in this thesis.

- Section 2.1-2.4 contains a description of the different coordinate systems that are used and how it is possible to map a coordinate representation in one coordinate system to the representation in another. Some important kinematic concepts for the octorotor are described and a detailed explanation of the rigid body motion equations that describe the motion of the octorotor in three-dimensional space is presented.
- Section 2.5 describes the mathematical framework used to analyze how the frequency content of a signal varies with time.
- Section 2.6, describes how a linear complementary filter is used to combine sensor data from different sensors into usable attitude estimates.
- Section 2.7 introduces the mathematical concept of quaternions which is used to make efficient implementations of attitude estimation algorithms.
- Section 2.8-2.9 presents some important concepts in linear control theory together with the theory behind the control algorithms that are used in the controllers designed in this thesis.

2.1 Coordinate systems

To describe the octorotor's movement in the three-dimensional airspace it is necessary to define suitable coordinate systems. In this section, two coordinate systems will be defined, the earth-fixed frame and the body-fixed frame.

The earth-fixed frame, in this thesis denoted with superscript E, has its origin fixed at a point on the earth's surface. In aviation, it is common to use the so called north, east, down coordinate system (NED) [29] and hence it will also be used in this thesis. To conform with this convention the x-axis of the earth-fixed frame points to the north, the y-axis points to the east and the z-axis points straight down. These axes will be denoted X^{E}, Y^{E} and Z^{E} respectively. The direction of the X^{E}- and Y^{E}-axes is tangential to the surface of the earth and hence the Z^{E}-axis points straight down towards the earth's center. See Figure 2.1 for an illustration.

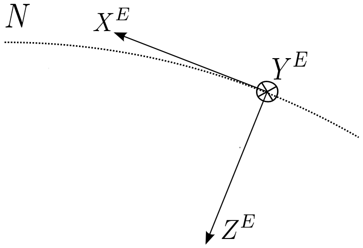


Figure 2.1: *Illustration of the earth-fixed coordinate system. The dotted line represents the surface of the earth.*

The earth-fixed frame is in this thesis assumed to be an inertial frame in which the absolute linear position of the octorotor is defined. This is an approximation since the earth constantly rotates around its own axis.

In addition to the earth-fixed frame a body-fixed coordinate system is defined. In this thesis the body-fixed coordinate system is denoted with superscript B and the origin of this coordinate system is fixed to the octorotor's center of gravity. The body-fixed coordinate system will move relative to the inertial frame when the octorotor moves through the air. The body-fixed frame is also chosen to conform to the north, east, down standard. Hence, the x-axis of the body-fixed coordinate system always points in the forward direction which is represented by the red colored arm in Figure 2.2. The y-axis points to the right and the z-axis points downwards from the octorotor's center of gravity. These axes will be denoted X^{B}, Y^{B} and Z^{B} respectively. Figure 2.2 provides an overview of the body-fixed coordinate system.

The body-fixed frame simplifies the formulation of the dynamic equations of motion for the platform. Using the body-fixed frame when deriving these equations is advantageous since (see [4]):

- the inertia matrix is time invariant
- the body symmetry can be used to simplify the equations
- measurements taken onboard are easily converted to the body-fixed frame
- control forces are given in the body-fixed frame.

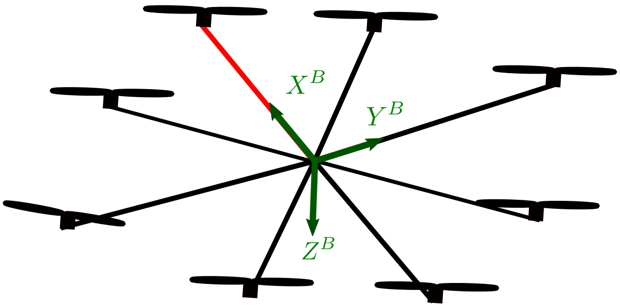


Figure 2.2: *Illustration of the body-fixed coordinate system. The* X*-axis points in the same direction as the red colored arm, which is seen as the front of the octorotor.*

2.2 Octorotor kinematics

This section describes some important kinematic concepts and definitions that are used when deriving the motion equations for the platform. To describe the location of the octorotor in space it is useful to define two concepts, the octorotor's linear and angular position. The linear position is defined as the position vector of the octorotor's center of gravity expressed with coordinates relative to the earth-fixed coordinate system. The linear position is defined by (2.1) and illustrated in Figure 2.3.

$$\mathbf{\Gamma}^{\mathrm{E}} \triangleq \begin{bmatrix} x^{\mathrm{E}} & y^{\mathrm{E}} & z^{\mathrm{E}} \end{bmatrix}^T \tag{2.1}$$

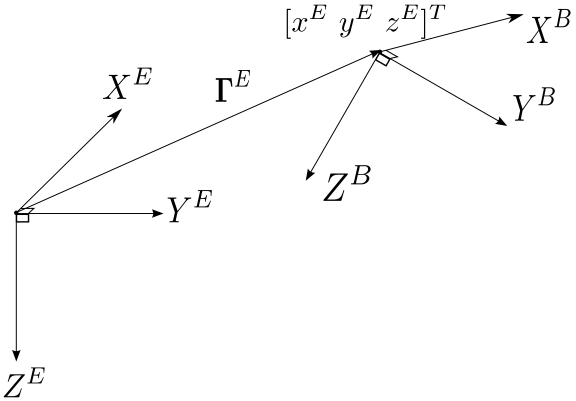


Figure 2.3: *The linear position* $\mathbf{\Gamma}^{\mathrm{E}}$ *of the octorotor is defined as the position of the octorotor's center of gravity (which coincides with the origin of the body-fixed frame) in the earth-fixed frame,* $[x^{\mathrm{E}}\ y^{\mathrm{E}}\ z^{\mathrm{E}}]^T$.

The angular position $\mathbf{\Theta}^{\mathrm{E}}$, also referred to as the attitude, is given by the orientation of the body-fixed frame with respect to the earth-fixed frame. The definition of the angular position is given by

$$\mathbf{\Theta}^{\mathrm{E}} \triangleq \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T . \tag{2.2}$$

The linear- and angular velocities of the platform with respect to the body-fixed frame, $\mathbf{v}^{\mathrm{B}}$ and ω^{B}, are defined as

$$\mathbf{v}^{\mathrm{B}} \triangleq \begin{bmatrix} u & v & w \end{bmatrix}^T , \tag{2.3}$$

$$\omega^{\mathrm{B}} \triangleq \begin{bmatrix} p & q & r \end{bmatrix}^T . \tag{2.4}$$

The three angles ϕ, θ and ψ are denoted roll, pitch and yaw, respectively and are referred to as Euler rotation angles [29]. These angles specify how a coordinate system which is initially aligned with the earth-fixed frame, needs to be rotated in order to be aligned with the body-fixed frame. This rotation is done in three separate steps. Let the original coordinate system be denoted Q, then let the resulting coordinate system after the first rotation be denoted Q′ and let the resulting system after the second rotation be denoted Q″. After the third rotation the rotated coordinate system will be aligned with the body-fixed frame, see Figure 2.4.

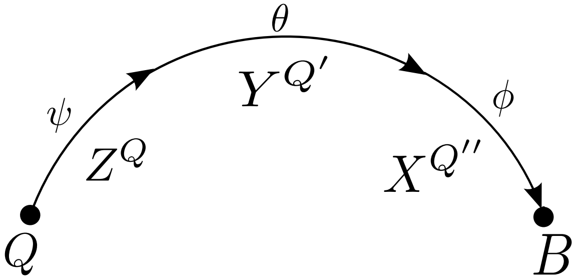


Figure 2.4: *Rotation steps in the definition of angular position. After the three rotations the* Q*-coordinate system is aligned with the* B*-coordinate system.*

In Figure 2.4, the angle ψ specifies rotation around the Z^{Q}-axis which is initially aligned with the Z^{E}-axis, the angle θ specifies subsequent rotation around the $Y^{\mathrm{Q}'}$-axis and the angle ϕ specifies the final rotation around the $X^{\mathrm{Q}''}$-axis. As mentioned previously, the body-fixed coordinate system moves and rotates with respect to the earth-fixed inertial system. How this is handled is described in the following section.

2.3 Transformations between coordinate systems

The dynamic equations of motion for a rigid body are expressed with respect to an inertial frame [8] but the forces and torques acting on an octorotor are more easily expressed with respect to a body-fixed coordinate system. Due to this it is important to be able to convert vector coordinates between different frames. This can be done using rotation matrices.

2.3.1 Rotation matrix

A rotation matrix **R** is a matrix that has the following properties (see [10]):

- **R** is orthogonal, i.e. the relation $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$ holds.
- A sequence of rotations is computed as a product of rotation matrices, one matrix for each rotation.

The rotation matrix used for converting between a coordinate description in the body-fixed frame to a coordinate description in the earth-fixed frame is calculated using the current angular position $\mathbf{\Theta}^{\mathrm{E}}$. Each Euler angle corresponds to a fundamental rotation which has its own rotation matrix. The total rotation matrix is calculated as the product of these matrices.

In the following description it is assumed that an arbitrary vector **a** has the coordinates $[x\ y\ z]^T$ in the earth-fixed system (and therefore also in the Q-system), the coordinates $[x'\ y'\ z']^T$ in the Q′-system, the coordinates $[x''\ y''\ z'']^T$ in the Q″-system and the coordinates $[x^{\mathrm{B}}\ y^{\mathrm{B}}\ z^{\mathrm{B}}]^T$ in the resulting body-fixed coordinate system after the last rotation. The following rotation steps are made:

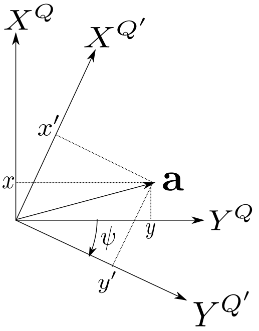


Figure 2.5: *Rotating the angle ψ around the Z^{Q}-axis to form the Q′-system*

1. When rotating the angle ψ around the Z^{Q}-axis according to Figure 2.5, the relationship between the vector **a**'s coordinates in the Q- and Q′-system is

$$\begin{aligned} x &= x'\cos\psi - y'\sin\psi, \\ y &= x'\sin\psi + y'\cos\psi, \\ z &= z'. \end{aligned} \tag{2.5}$$

This gives the following rotation matrix

$$\mathbf{R}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.6}$$

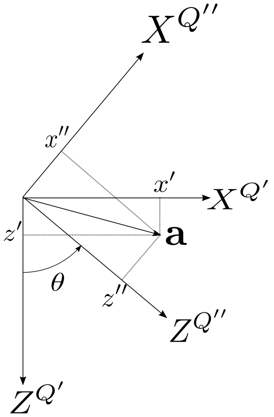


Figure 2.6: *Rotating the angle θ around the $Y^{\mathrm{Q}'}$-axis to form the* Q″*-system*

2. When rotating the angle θ around the $Y^{\mathrm{Q}'}$-axis as done in Figure 2.6 the second coordinate relationship becomes

$$\begin{aligned} x' &= x''\cos\theta + z''\sin\theta, \\ y' &= y'', \\ z' &= -x''\sin\theta + z''\cos\theta. \end{aligned} \tag{2.7}$$

The corresponding rotation matrix is

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{2.8}$$

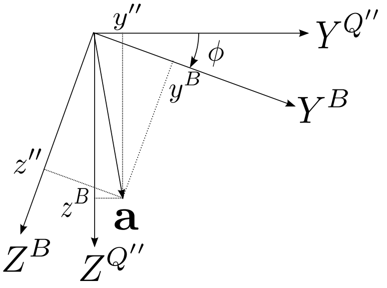


Figure 2.7: *Rotating the angle ϕ around the $X^{\mathrm{Q}''}$-axis to form the* B*-system*

3. The final rotation by the angle ϕ around the $X^{Q''}$-axis according to Figure 2.7 gives the following coordinate relationship

$$\begin{aligned} x'' &= x^{B}, \\ y'' &= y^{B}\cos\phi - z^{B}\sin\phi, \\ z'' &= y^{B}\sin\phi + z^{B}\cos\phi. \end{aligned} \tag{2.9}$$

The final rotation matrix hence is

$$\mathbf{R}_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}. \tag{2.10}$$

The total rotation matrix $\mathbf{R}_{B\to E}$ for the three sequential rotations can now be calculated as the product of the three matrices, $\mathbf{R}_{\psi}$, $\mathbf{R}_{\theta}$ and $\mathbf{R}_{\phi}$. Note that the matrix for each subsequent rotation is post-multiplied to the previous matrix, hence rotation order is important. The result is given by (2.11) and it is valid for the rotation convention used in this thesis. Other conventions give different rotation matrices

$$\mathbf{R}_{B\to E} \triangleq \mathbf{R}_{\psi}\mathbf{R}_{\theta}\mathbf{R}_{\phi} = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}. \tag{2.11}$$

Here s and c is a shorthand notation for sin and cos respectively.

The total rotation matrix is used when transforming the coordinate representation of a vector given in the body-fixed frame into the coordinate representation in the earth-fixed frame according to

$$\mathbf{a}^{E} = \mathbf{R}_{B\to E}\mathbf{a}^{B}. \tag{2.12}$$

In (2.12) it is assumed that the origin of the body-fixed and earth-fixed frame coincide. The conversion from an earth-fixed representation into a body-fixed representation is done in the same way by simply inverting the total rotation matrix. Since this matrix is orthogonal its inverse is given by the transpose

$$\mathbf{a}^{B} = (\mathbf{R}_{B\to E})^{-1}\,\mathbf{a}^{E} = (\mathbf{R}_{B\to E})^{T}\,\mathbf{a}^{E} = \mathbf{R}_{E\to B}\mathbf{a}^{E}, \tag{2.13}$$

where $\mathbf{R}_{E\to B}$ is defined as

$$\mathbf{R}_{E\to B} \triangleq \begin{bmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ -s\psi c\phi + c\psi s\theta s\phi & c\psi c\phi + s\psi s\theta s\phi & c\theta s\phi \\ s\psi s\phi + c\psi s\theta c\phi & -c\psi s\phi + s\psi s\theta c\phi & c\theta c\phi \end{bmatrix}. \tag{2.14}$$

2.3.2 Time derivatives

Since the orientation of the body-fixed system is not constant it is also necessary to derive an expression for the derivative of the Euler angles and the rotation matrix with respect to time.

Euler angles

The expression for the time-derivative of the Euler angles can be derived using the rotation matrices calculated in Section 2.3. The Q coordinate system is aligned with the earth-fixed system which does not rotate and hence its angular velocity vector is

$$\boldsymbol{\omega}^{\mathrm{Q}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.15}$$

The Q′-system rotates around its z-axis which is aligned with the z-axis of the Q-system. The angular velocity vector for the Q′-system is therefore

$$\boldsymbol{\omega}^{\mathrm{Q}'} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}. \tag{2.16}$$

For the Q″-system the angular velocity is composed of both (2.16) expressed in coordinates relative to the Q″-system and the pitch angular velocity. This can be expressed as

$$\boldsymbol{\omega}^{\mathrm{Q}''} = \mathbf{R}_\theta^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} = \begin{bmatrix} -\dot{\psi}\sin\theta \\ \dot{\theta} \\ \dot{\psi}\cos\theta \end{bmatrix}. \tag{2.17}$$

Finally, the angular velocity vector for the body-fixed coordinate system is composed of (2.17) expressed in coordinates relative to the B-system and the roll rotation velocity

$$\boldsymbol{\omega}^{\mathrm{B}} = \mathbf{R}_\phi^T \begin{bmatrix} -\dot{\psi}\sin\theta \\ \dot{\theta} \\ \dot{\psi}\cos\theta \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta \\ -\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta \end{bmatrix}. \tag{2.18}$$

Using the definition given in (2.4) the following relation must hold

$$\begin{aligned} p &= \dot{\phi} - \dot{\psi}\sin\theta, \\ q &= \dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta, \\ r &= -\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta. \end{aligned} \tag{2.19}$$

By using (2.19) it is possible to define the inverse transfer matrix, $\mathbf{T}_{\mathrm{E}\to\mathrm{B}}$, used to relate the time derivatives of the Euler angles to the octorotor's angular velocity. By inverting this matrix, as done in (2.21), it is possible to convert angular velocities measured in the body-fixed frame to the time derivatives of the corresponding Euler angles. This relationship is expressed in (2.22).

$$\mathbf{T}_{\mathrm{E}\to\mathrm{B}} \triangleq \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{2.20}$$

$$\mathbf{T}_{\mathrm{B}\to\mathrm{E}} \triangleq (\mathbf{T}_{\mathrm{E}\to\mathrm{B}})^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{2.21}$$

$$\dot{\mathbf{\Theta}}^{\mathrm{E}} = \mathbf{T}_{\mathrm{B}\to\mathrm{E}}\boldsymbol{\omega}^{\mathrm{B}} \tag{2.22}$$

Equation (2.21) and (2.22) show that the derivatives of the Euler angles are *not* equal to the angular velocities in the body-fixed frame.

In order to derive an expression for the time derivative of the rotation matrix the following important equation is needed (see [8])

$$\left(\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t}\right)_I = \left(\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t}\right)_{\mathrm{B}} + \boldsymbol{\omega} \times \mathbf{a}. \tag{2.23}$$

Equation (2.23) states that the derivative of an arbitrary vector $\mathbf{a}$ as seen by an observer fixed in an inertial system is equal to the derivative of the same vector seen by an observer fixed in the rotating system plus a correction term $\boldsymbol{\omega} \times \mathbf{a}$. This correction term depends on the rotational velocity vector $\boldsymbol{\omega}$ of the rotating coordinate system.

Rotation matrix

In this section an expression for the time derivative of the rotation matrix will be derived. Assume that a point in space has the position vector $\mathbf{r}$ and the velocity vector $\mathbf{v}$. Then $\mathbf{r}^{\mathrm{E}}$ is the position vector with its coordinates expressed with respect to the earth-fixed frame and $\mathbf{r}^{\mathrm{B}}$ is the same vector with its coordinates expressed with respect to the body-fixed frame. $\boldsymbol{\omega}^{\mathrm{B}}$ is the angular velocity vector of the body-fixed frame expressed relative to the body-fixed frame and $\mathbf{R}$ is the rotation matrix from the body frame to the earth frame as defined in (2.11). In addition, assume that $\left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_I^{\mathrm{E}}$ is the inertial time derivative of $\mathbf{r}$ with its coordinates given with respect to the earth-fixed frame and that $\left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{\mathrm{B}}^{\mathrm{B}}$ is the body-fixed time derivative of $\mathbf{r}$ with its coordinates given relative to the body-fixed frame. For notational purposes $\dot{\mathbf{x}} = \left(\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right)_I$ is now used to represent inertial time derivation. Then the following equations apply

$$\dot{\mathbf{r}}^{\mathrm{E}} = \mathbf{v}^{\mathrm{E}}, \tag{2.24}$$

$$\mathbf{r}^{\mathrm{E}} = \mathbf{R}\mathbf{r}^{\mathrm{B}}, \tag{2.25}$$

$$\mathbf{v}^{\mathrm{E}} = \mathbf{R}\mathbf{v}^{\mathrm{B}}. \tag{2.26}$$

Taking the derivative of (2.25) with respect to time gives

$$\dot{\mathbf{r}}^{\mathrm{E}} = \dot{\mathbf{R}}\mathbf{r}^{\mathrm{B}} + \mathbf{R}\left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{\mathrm{B}}^{\mathrm{B}}. \tag{2.27}$$

The velocity vector with respect to the body-fixed frame $\mathbf{v}^{\mathrm{B}}$ can be calculated by using (2.23)

$$\mathbf{v}^{\mathrm{B}} = \dot{\mathbf{r}}^{\mathrm{B}} = \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{dt}}\right)_{\mathrm{B}}^{\mathrm{B}} + \boldsymbol{\omega}^{\mathrm{B}} \times \mathbf{r}^{\mathrm{B}}. \tag{2.28}$$

By combining the equations above it is possible to derive the following equation as a relation between the inertial and body-fixed velocity

$$\dot{\mathbf{r}}^{\mathrm{E}} = \mathbf{v}^{\mathrm{E}} = \mathbf{R}\mathbf{v}^{\mathrm{B}} = \mathbf{R}\left(\left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{dt}}\right)_{\mathrm{B}}^{\mathrm{B}} + \boldsymbol{\omega}^{\mathrm{B}} \times \mathbf{r}^{\mathrm{B}}\right) = \dot{\mathbf{R}}\mathbf{r}^{\mathrm{B}} + \mathbf{R}\left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{dt}}\right)_{\mathrm{B}}^{\mathrm{B}}. \tag{2.29}$$

The last equality in (2.29) gives

$$\dot{\mathbf{R}}\mathbf{r}^{\mathrm{B}} = \mathbf{R}\left(\boldsymbol{\omega}^{\mathrm{B}} \times \mathbf{r}^{\mathrm{B}}\right). \tag{2.30}$$

It is now possible to define a skew-symmetric matrix $\boldsymbol{\omega}_{\times}^{\mathrm{B}}$ such that the relation

$$\boldsymbol{\omega}_{\times}^{\mathrm{B}}\mathbf{a} \triangleq \boldsymbol{\omega}^{\mathrm{B}} \times \mathbf{a} \tag{2.31}$$

holds for all vectors $\mathbf{a}$. This matrix is

$$\boldsymbol{\omega}_{\times}^{\mathrm{B}} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}. \tag{2.32}$$

The final expression for the time derivative of the rotation matrix is then given by

$$\dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}_{\times}^{\mathrm{B}}. \tag{2.33}$$

2.4 Newton-Euler equations of motion

The dynamic model of the octorotor's motion in space is based on rigid body mechanics. This section details how the dynamic equations of motion are derived for a general rigid body. To do this some important definitions are needed:

- The position vector of a rigid body's *center of mass* $\mathbf{r}_G$ is defined as

$$\mathbf{r}_G \triangleq \frac{1}{m}\int_B \mathbf{r}\,\rho\,\mathrm{dV}. \tag{2.34}$$

- The rigid body's *linear momentum* $\mathbf{p}$ is defined as

$$\mathbf{p} \triangleq \int_B \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{dt}}\right)_I \rho\,\mathrm{dV}. \tag{2.35}$$

- The rigid body's *angular momentum* with respect to the arbitrary point A, $\mathbf{h}_A$, is defined as

$$\mathbf{h}_A \triangleq \int_B \boldsymbol{\rho}_A \times \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{dt}}\right)_I \rho\, \mathrm{dV}. \tag{2.36}$$

Here, $\mathbf{r}$ is the position vector of a volume element dV, ρ is the density of the rigid body, m is the body's total mass and $\boldsymbol{\rho}_A$ is a vector from an arbitrary point A to the volume element.

The Newton-Euler equations of motion for a rigid body are (see [8])

$$\mathbf{F} = \left(\frac{\mathrm{d}\mathbf{p}}{\mathrm{dt}}\right)_I, \tag{2.37}$$

$$\boldsymbol{\tau}_A = \left(\frac{\mathrm{d}\mathbf{h}_A}{\mathrm{dt}}\right)_I. \tag{2.38}$$

Equation (2.37) states that the change in linear momentum with respect to an inertial frame is equal to the sum of all forces acting on the body. Similarly, (2.38) states that the change in angular momentum around an arbitrary fixed point A is equal to the total torque with respect to A. These equations can be rewritten using the fact that the mass of the rigid body does not change with time and by choosing the arbitrary point A to coincide with the body's center of gravity [8]. The simplified equations are

$$\mathbf{F} = m\left(\frac{\mathrm{d}\mathbf{v}_G}{\mathrm{dt}}\right)_I, \tag{2.39}$$

$$\boldsymbol{\tau}_G = \left(\frac{\mathrm{d}\mathbf{I}_G\boldsymbol{\omega}}{\mathrm{dt}}\right)_I. \tag{2.40}$$

Here, $\mathbf{v}_G$ is the velocity vector for the body's center of gravity, $\mathbf{I}_G$ is the body's inertia tensor with respect to the center of gravity and $\boldsymbol{\omega}$ is the angular velocity vector for the body.

The derivatives in (2.39) and (2.40) are computed and expressed in coordinates relative to the body-fixed frame using (2.23). This gives

$$\mathbf{F}^{\mathrm{B}} = m\left(\frac{\mathrm{d}\mathbf{v}_G}{\mathrm{dt}}\right)_I^{\mathrm{B}} = m\left(\left(\frac{\mathrm{d}\mathbf{v}_G}{\mathrm{dt}}\right)_{\mathrm{B}}^{\mathrm{B}} + \boldsymbol{\omega}^{\mathrm{B}} \times \mathbf{v}_{\mathrm{G}}^{\mathrm{B}}\right), \tag{2.41}$$

$$\boldsymbol{\tau}_{\mathrm{G}}^{\mathrm{B}} = \left(\frac{\mathrm{d}\mathbf{I}_G\boldsymbol{\omega}}{\mathrm{dt}}\right)_I^{\mathrm{B}} = \left(\frac{\mathrm{d}\mathbf{I}_G\boldsymbol{\omega}}{\mathrm{dt}}\right)_{\mathrm{B}}^{\mathrm{B}} + \boldsymbol{\omega}^{\mathrm{B}} \times \left(\mathbf{I}_{\mathrm{G}}^{\mathrm{B}}\boldsymbol{\omega}^{\mathrm{B}}\right) = \mathbf{I}_{\mathrm{G}}^{\mathrm{B}}\left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{dt}}\right)_{\mathrm{B}}^{\mathrm{B}} + \boldsymbol{\omega}^{\mathrm{B}} \times \left(\mathbf{I}_{\mathrm{G}}^{\mathrm{B}}\boldsymbol{\omega}^{\mathrm{B}}\right). \tag{2.42}$$

The last equality in (2.42) is due to the fact that the inertia tensor is constant when deriving relative to the body-fixed system [4].

2.5 Frequency description of signals

A signal can be described either in the time or frequency domain. The relationship between these different representations is given by the Fourier transform (FT) as

$$X(i\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}\mathrm{d}t, \tag{2.43}$$

where $x(t)$ is the time representation of the signal and $X(i\omega)$ is the frequency representation. Since computerized logging of signals is performed by sampling continuous-time signals, an expression corresponding to (2.43) for discrete-time signals is needed. This expression is given by the discrete-time Fourier transform (DTFT)

$$X_T(e^{i\omega T}) = T\sum_{k=-\infty}^{\infty} x[k]e^{-i\omega kT}, \tag{2.44}$$

where T is the sample rate and $x[k] = x(kT)$ is the sampled continuous-time signal. The transformation is periodic with period $2\pi/T$. The relation between the DTFT and the FT is given by Poisson's summation formula

$$X_T(e^{i\omega T}) = \sum_{r=-\infty}^{\infty} X(i(\omega + r\omega_s)) \quad \frac{-\omega_s}{2} \leq \omega \leq \frac{\omega_s}{2}, \tag{2.45}$$

where ω_s corresponds to the sampling frequency. If the sampled signal includes frequencies that are higher than half the sampling frequency, these will appear as wrong frequencies in the DTFT. This phenomenon is called folding or aliasing. To prevent this from happening, the sample frequency needs to be at least twice as high as the highest frequency content of the signal [11].

The expression for the DTFT can only be calculated for an infinitely long signal. Since only finite signals can be stored on a computer all the unknown samples in (2.44) are replaced by zeros, giving the formula

$$X_T^{(N)}(e^{i\omega T}) = T\sum_{k=0}^{N-1} x[k]e^{-i\omega kT}. \tag{2.46}$$

The truncation will lead to a phenomenon referred to as leakage, which means that the energy at one frequency is "leaking" to nearby frequencies. This gives worse frequency resolution. To minimize this effect, the truncated signal can be multiplied with a so called window function [11].

The truncated DTFT is a function of a continuous frequency variable ω. Computers work with sampled versions of continuous signals and therefore the last step in the frequency analysis is to sample the continuous frequency truncated DTFT.

This gives the discrete Fourier transform (DFT)

$$X[n] = \sum_{k=0}^{N-1} x[k]e^{-2\pi ikn/N}, \quad n = 0, 1, \ldots, N-1. \tag{2.47}$$

The frequency grid resolution $\omega_0 = \frac{2\pi}{NT}$ for this transform depends on how many samples and which sample frequency that are used. One way to increase the frequency grid resolution is to zero-pad the signal, i.e add a number of zeros to the end of the signal to get more samples [11].

A time-frequency description (TFD) shows how the frequency content in a signal varies over time. One way to create a TFD is to first divide the signal into segments of a specific length. Longer segments will give less frequency leakage in the TFD. To get a higher time resolution, the segments are allowed to overlap each other. Each segment is multiplied with a window function to mitigate leakage and zero-padded to increase the frequency grid resolution. Finally, the DFT is calculated for every segment, and all the DFT:s are stacked in a resulting vector. This method is called Batch-Wise DFT [11].

2.6 Linear complementary filter

A linear complementary filter is a type of filter that is used for attitude estimation in different multirotor projects [19]. The filter is based on the fact that several sensors with different frequency characteristics are available. The differing frequency characteristics can then be exploited by running each sensor output through a frequency selective filter and hence remove the unreliable frequency content for each sensor.

Assume that two separate measurements of a signal x is available, each with additive noise in a certain frequency range

$$y_1 = x + \eta_1, \tag{2.48}$$

$$y_2 = x + \eta_2. \tag{2.49}$$

Here, η_1 represents high frequency and η_2 low frequency noise respectively. By designing a low-pass filter $G_{\text{LP}}(s)$ and a high-pass filter $G_{\text{HP}}(s)$ such that

$$G_{\text{LP}}(s) + G_{\text{HP}}(s) = 1, \ \forall s \tag{2.50}$$

holds, the filter estimate can be computed as

$$\hat{X}(s) = G_{\text{LP}}(s)Y_1(s) + G_{\text{HP}}(s)Y_2(s) = X(s) + G_{\text{LP}}(s)\eta_1(s) + G_{\text{HP}}(s)\eta_2(s). \tag{2.51}$$

The two filters should be designed such that the frequency content of the signal x is in the passband and that the two noises η_1 and η_2 are sufficiently attenuated [20].

For the octorotor, both the accelerometer and the gyroscope can provide attitude estimates but with different frequency characteristics. The accelerometer estimate requires low-pass filtering and the gyroscope estimate requires high-pass filtering in order to be useable. This will be explained in more detail in Section 5.1. Hence, a simple complementary filter can be used for attitude estimation by feeding the accelerometer measurement through a low-pass filter and the gyroscope measurement through a high-pass filter. The final attitude estimate is then taken as the sum of the two filtered signals. The filters' cutoff frequencies are tuned to get good performance. Figure 2.8 provides an illustration.

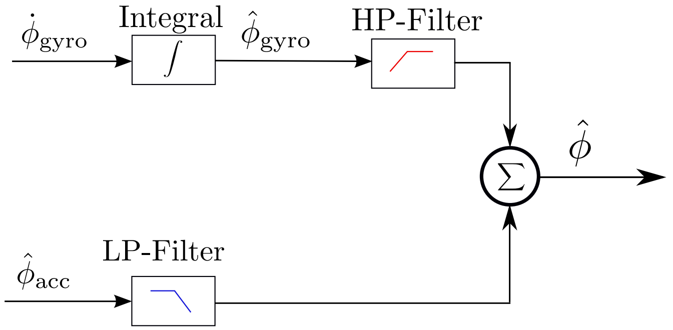

$\dot{\phi}_{\text{gyro}}$
Integral
$\int$
$\hat{\phi}_{\text{gyro}}$
HP-Filter
Σ
$\hat{\phi}$
$\hat{\phi}_{\text{acc}}$
LP-Filter

Figure 2.8: Schematic view of a complementary filter for attitude estimation. The variables $\hat{\phi}_{acc}$ and $\hat{\phi}_{gyro}$ corresponds to the accelerometer and gyroscope angle estimates. The output, $\hat{\phi}$, is the sum of the two separate estimates.

2.7 Quaternion rotation representation

When describing the attitude of an aircraft several different techniques can be used. The rotation matrix and Euler angle representation are described in Section 2.3. These representations suffer from different problems such as that they are incapable of describing certain orientations at the poles of the unit sphere. They are also inherently ambiguous since a rotation of α radians cannot be distinguished from a rotation of $\alpha \pm 2\pi$ radians. However, the Euler angles are intuitive to understand and visualize [10].

Rotation matrices are used in combination with Euler angles and they consist of nine elements which need to be updated in the state estimation algorithm. Using nine elements to represent rotation around three axes is not a minimal representation and hence it causes unnecessary computations. When updating a rotation matrix, (2.33) is used which is based on a matrix multiplication. This differential equation is actually a system of nine coupled differential equations that has to be solved. Another issue with the rotation matrix is that it has to be orthogonal in order to actually represent a proper rotation. The state estimation algorithms that are typically used does not automatically preserve this property and the matrix needs to be orthogonalized, which requires additional computational power [7].

To get around these problems, an alternative rotation representation is used that does not suffer from the same problems as the Euler angles and the rotation matrix. This representation is based on quaternions which are described in more detail in the remainder of this section.

The quaternion is a number system which can be seen as an extension of the complex number system with two more complex dimensions to form a hyper complex number of rank four. The quaternion can be represented in several ways, a natural way of doing this is to represent it similarly to a standard complex number in rectangular form or to use a vector representation

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}, \tag{2.52}$$

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T. \tag{2.53}$$

The first element of the quaternion, q_0 is referred to as the scalar part and the three complex elements q_1, q_2 and q_3 are referred to as the vector part. Here $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are imaginary basis vectors that follow the relation $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$ [7].

The quaternion has a conjugate expression similar to standard complex numbers. The conjugate of a quaternion is given by simply switching sign on all components of the vector part

$$\mathbf{q}^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^T. \tag{2.54}$$

Addition and subtraction of quaternions work in the same way as for standard complex numbers. Multiplication with a scalar works in the same way as for a four dimensional vector. Quaternion multiplication however is a complicated operation that is defined in (2.55) below and it is not commutative (as for matrix multiplication)

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} s_1 s_2 - \mathbf{v}_1^T \mathbf{v}_2 \\ s_1\mathbf{v}_2 + s_2\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2 \end{bmatrix}. \tag{2.55}$$

Here, s_1 and s_2 and $\mathbf{v}_1$ and $\mathbf{v}_2$ represent the scalar and vector part of quaternion $\mathbf{q}_1$ and $\mathbf{q}_2$ respectively [20].

In addition to the conjugate expression the quaternion also has a length norm like the standard complex numbers. The length of a quaternion $\mathbf{q}$ is defined as

$$||\mathbf{q}|| = \sqrt{\mathfrak{Re}(\mathbf{q} \otimes \mathbf{q}^*)} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \tag{2.56}$$

The inverse of a quaternion is given by (see [7])

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{||\mathbf{q}||^2}. \tag{2.57}$$

By using the length norm it is possible to define the concept of unit quaternions, i.e. quaternions with unit length, which can be used to represent rotations. Assume that $\mathbf{u}$ is a unit vector representing a rotation axis. Then the following holds

$$\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) & \sin\left(\frac{\alpha}{2}\right)\mathbf{u} \end{bmatrix}^T, \tag{2.58}$$

$$||\mathbf{q}|| = 1. \tag{2.59}$$

Equation (2.58) states that a unit quaternion $\mathbf{q}$ can be interpreted as a rotation of the angle α around an axis represented by $\mathbf{u}$. Hence, unit quaternions contain the same information as a set of Euler angles or a rotation matrix. A unit quaternion can describe how a frame A should be rotated in order to be aligned with another frame B. Two subsequent rotations are given by multiplying the quaternion for each rotation [29].

Since a quaternion contains information about the rotation of one frame with respect to another it can be used to transform the coordinate representation of a vector from one coordinate frame to the other, just like a rotation matrix. Assume that the earth-fixed coordinate system E has been rotated the angle α around an axis represented by $\mathbf{u}$ to form the body-fixed B-system. This makes it possible to compute a quaternion $\mathbf{q}_{\mathrm{B}\to\mathrm{E}}$ using (2.58) that can be used to transform a coordinate representation given in the body-fixed frame to a representation with respect to the earth-fixed frame. Similarly, the quaternion given by the conjugate of $\mathbf{q}_{\mathrm{B}\to\mathrm{E}}$ can be used to map coordinates in the earth-fixed frame to the body-fixed frame [29].

The coordinate transformation from the body-fixed frame to the earth-fixed frame of an arbitrary vector $\mathbf{a}$ is performed by first forming a so called *pure* quaternion from the body-fixed coordinate representation. The pure quaternion is a quaternion with a zero scalar value and a vector part equal to the coordinate representation of the vector

$$\mathbf{q}_{\mathbf{a}}^{\mathrm{B}} = \mathbf{p}\left(\mathbf{a}^{\mathrm{B}}\right) = \begin{bmatrix} 0 & a_x & a_y & a_z \end{bmatrix}^T . \tag{2.60}$$

Given the pure quaternion $\mathbf{q}_{\mathbf{a}}^{\mathrm{B}}$, the transformation between $\mathbf{a}$'s representation in the body-fixed frame to the earth-fixed frame is given by (see [29])

$$\mathbf{a}^{\mathrm{E}} = \mathbf{p}^{-1}\left(\mathbf{q}_{\mathbf{a}}^{\mathrm{E}}\right) = \mathbf{p}^{-1}\left(\mathbf{q}_{\mathrm{B}\to\mathrm{E}} \otimes \mathbf{q}_{\mathbf{a}}^{\mathrm{B}} \otimes \mathbf{q}_{\mathrm{B}\to\mathrm{E}}^{*}\right). \tag{2.61}$$

The rotation quaternion is multiplied with the pure quaternion representing the body-fixed coordinate representation and subsequently multiplied with the conjugate of the rotation quaternion. The result is a pure quaternion whose vector part is equal to the transformed vector's coordinates in the earth-fixed frame. In (2.61), $\mathbf{p}^{-1}$ represents the inverse operation of forming a pure quaternion from a three-dimensional vector.

It is also possible to calculate the rotation quaternion directly from the Euler angles instead of using (2.58) [29]. The relationship between the Euler angles and the rotation quaternion is given by

$$\mathbf{q}_{\mathrm{B}\to\mathrm{E}} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix}. \tag{2.62}$$

Since the rotation quaternion can be used for the same operation as a rotation matrix, it is possible to derive a relation between the rotation quaternion and the rotation matrix. By transforming the axis vector for each axis of the body-fixed frame using (2.61), removing the scalar part of the resulting quaternion and stacking the result horizontally in a matrix, the rotation matrix from the body-fixed frame to the earth-fixed frame is obtained [7]

$$\mathbf{R}_X(\mathbf{q}) = \mathbf{q} \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes \mathbf{q}^* = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1 q_2 + q_0 q_3) \\ 2(q_1 q_3 - q_0 q_2) \end{bmatrix}, \tag{2.63}$$

$$\mathbf{R}_Y(\mathbf{q}) = \mathbf{q} \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \otimes \mathbf{q}^* = \begin{bmatrix} 2(q_1 q_2 - q_0 q_3) \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_2 q_3 + q_0 q_1) \end{bmatrix}, \tag{2.64}$$

$$\mathbf{R}_Z(\mathbf{q}) = \mathbf{q} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes \mathbf{q}^* = \begin{bmatrix} 2(q_1 q_3 + q_0 q_2) \\ 2(q_2 q_3 - q_0 q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}, \tag{2.65}$$

$$\mathbf{R}_{\mathrm{B}\to\mathrm{E}} = \begin{bmatrix} \mathbf{R}_X(\mathbf{q}) & \mathbf{R}_Y(\mathbf{q}) & \mathbf{R}_Z(\mathbf{q}) \end{bmatrix}. \tag{2.66}$$

According to (2.66) the rotation matrix corresponding to the rotation quaternion is

$$\mathbf{R}_{\mathrm{B}\to\mathrm{E}}\left(\mathbf{q}_{\mathrm{B}\to\mathrm{E}}\right) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{2.67}$$

The expression for the time derivative of the quaternion is given by (2.68) and it is the quaternion equivalent of (2.22) and (2.33) [29]

$$\dot{\mathbf{q}}_{\mathrm{B}\to\mathrm{E}} = \frac{1}{2}\mathbf{q}_{\mathrm{B}\to\mathrm{E}} \otimes \mathbf{p}\left(\omega^{\mathrm{B}}\right). \tag{2.68}$$

2.8 Linear control theory

In this thesis, a linearized model of the octorotor is used when designing attitude control algorithms, see Section 6.2. This section provides an overview of some important concepts in linear control theory.

A general linear controller can be described by two transfer functions: $F_r(s)$ and $F_y(s)$ [28]. $F_r(s)$ can be viewed as a pre-filter for the reference signal $r(t)$ while $F_y(s)$ filters the measured output signal $y(t)$ to form a negative feedback compensation. The Laplace transform of the controller output $u(t)$ is

$$U(s) = F_r(s)R(s) - F_y(s)Y(s). \tag{2.69}$$

A general feedback loop is illustrated in Figure 2.9 where $G(s)$ represents the linear transfer function describing the controlled system's dynamics.

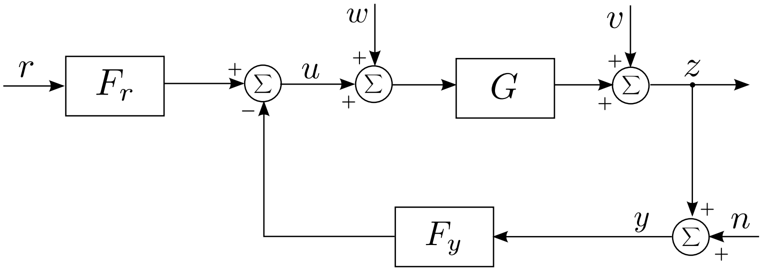


Figure 2.9: A block diagram describing the closed-loop system structure obtained when using a linear controller parameterized by the transfer functions F_r and F_y. The controlled system's dynamics is represented by the transfer function G.

- $r(t)$ is the reference signal
- $u(t)$ is the control signal
- $z(t)$ is the controlled variable
- $y(t)$ is the measurement signal
- $w(t)$ is an input disturbance signal
- $v(t)$ is an output disturbance signal
- $n(t)$ is measurement noise

The general block diagram in Figure 2.9 gives the following relation between the signals (see [28])

$$Z(s) = G_c(s)R(s) + S_u(s)W(s) + S(s)V(s) - T(s)N(s). \tag{2.70}$$

where the transfer functions are given by

$$G_c(s) = \frac{G(s)F_r(s)}{1 + G(s)F_y(s)} \tag{2.71}$$

$$S(s) = \frac{1}{1 + G(s)F_y(s)} \tag{2.72}$$

$$S_u(s) = \frac{G(s)}{1 + G(s)F_y(s)} = G(s)S(s) \tag{2.73}$$

$$T(s) = \frac{G(s)F_y(s)}{1 + G(s)F_y(s)} = 1 - S(s). \tag{2.74}$$

- $G_c(s)$ is the closed-loop transfer function that relates the reference input to the controlled variable.
- $S(s)$ is the sensitivity function that relates a disturbance on the output to the controlled variable. In addition, the sensitivity function describes how relative modeling errors will affect the controlled variable.

- $S_u(s)$ is the input sensitivity function that relates a disturbance on the input $W(s)$ to the controlled variable.
- $T(s)$ is the complementary sensitivity function that relates measurement noise to the controlled variable. The complementary sensitivity function is also related to the closed-loop system's sensitivity to modeling errors.

Ideally, $G_c(s) = 1$ and $S(s) = S_u(s) = T(s) = 0$ should hold. However, this is impossible since $S(s) + T(s) = 1$ and the size of the control signals are limited by the actuators used in the real system. In addition, several other fundamental limitations exist that restrict which performance is attainable by the controlled system given a certain controller structure. Hence, when performing controller design it is important to take these limitations into account and work towards attainable performance goals.

To attenuate low frequency disturbances it is often desirable to design the controller such that $S(s)$ is small for low frequencies. Conversely, $T(s)$ is designed such that it is small for higher frequencies. This is done to attenuate high frequency measurement noise. In addition, the model is often more uncertain for higher frequencies which makes it desirable to keep $T(s)$ small in this frequency region to improve robustness. These design choices makes it possible to keep within the boundaries set by the relation $S + T = 1$ and still obtain good performance [28].

A common choice is to set $F_r(s) = F_y(s)$ in the general control structure illustrated in Figure 2.9 [28]. By doing this, a degree of freedom is lost in the controller design and $T(s) = G_c(s)$. This means that the reference tracking is not decoupled from the system's robustness and disturbance rejection. On the other hand, choosing a controller that utilizes the more general structure makes it possible to separate the reference tracking design from the rest of the control design process.

2.9 Control algorithms

There are many different control algorithms that can be used when implementing and designing a closed-loop controller for a system. This section describes the theory behind the control algorithms used in the attitude controllers that are designed in this thesis.

2.9.1 The standard PID controller

Controllers that implement the PID structure are the most common type of controller used in the industry. Variants of the PID controller have been used since the 18$^{\text{th}}$ century [28]. In addition, PID controllers have been used in many different multirotor projects with good results [19].

The PID controller is based on the deviation between the desired and the measured output $e(t) = r(t) - y(t)$, according to

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\,\mathrm{d}\tau + K_D \frac{\mathrm{d}}{\mathrm{dt}} e(t). \tag{2.75}$$

Here, $u(t)$ is the control signal that is the output from the controller and provided as input to the controlled system. K_P, K_I and K_D are constant parameters that need to be tuned in order to get good closed-loop performance. By computing the Laplace transform of (2.75), the following transfer function is obtained

$$U(s) = \left(K_P + \frac{K_I}{s} + K_D s\right) E(s). \tag{2.76}$$

The first term of the control signal is proportional to the control error $e(t)$. By increasing the proportional gain, the system will respond faster but it will also become less robust and more oscillative. The second term is proportional to the integral of the control error. This means that the contribution from this term will continue to increase or decrease until the value of the control error is zero. This effect helps eliminate static errors in the output. However, adding integral action might reduce the stability of the system. The last term is proportional to the derivative of the control error and helps to decrease oscillations in the closed loop system [28].

Using the control law in (2.76) is equivalent to setting

$$F_r(s) = F_y(s) = F(s) = K_P + \frac{K_I}{s} + K_D s \tag{2.77}$$

in the general linear control structure illustrated in Figure 2.9. Hence, the standard PID structure does not allow the reference tracking to be tuned separately from the disturbance rejection.

2.9.2 Modified PID structure

A generalization of the standard PID structure in (2.75) is

$$u(t) = K_P(\alpha r(t) - y(t)) + K_I \int_0^t e(\tau)\,\mathrm{d}\tau + K_D \frac{\mathrm{d}(\beta r(t) - y(t))}{\mathrm{dt}}, \tag{2.78}$$

where α and β are design parameters with a value between zero and one [1]. By changing these parameters it is possible to vary the effect that the reference input has on the control signal. The modified PID controller given by (2.78) can be seen as a controller with two degrees of freedom because the reference and output signals are not treated in the same way. Hence, the controller can be described by two transfer functions, $F_r(s)$ and $F_y(s)$. The Laplace transform of the control signal is given by

$$U(s) = F_r(s)R(s) - F_y(s)Y(s) \tag{2.79}$$

and the transfer functions are

$$F_r(s) = \alpha K_P + \frac{K_I}{s} + \beta K_D s, \tag{2.80}$$

$$F_y(s) = K_P + \frac{K_I}{s} + K_D s. \tag{2.81}$$

Given these transfer functions, the closed-loop transfer function for the system becomes

$$G_c(s) = \frac{G(s)F_r(s)}{1 + G(s)F_y(s)} = \frac{(\beta K_D s^2 + \alpha K_P s + K_I)G(s)}{s + (K_D s^2 + K_P s + K_I)G(s)}. \tag{2.82}$$

From (2.82) it can be seen that the poles of the closed loop system (the roots of G_c's denominator polynomial) depend on how the parameters K_P, K_I, and K_D are selected, whereas they are unaffected by the values of α and β. The equation also shows that the PID controller introduces two additional zeros (the roots of G_c's numerator polynomial) to the closed loop system. These zeros will affect the reference tracking of the closed loop system differently depending on their position in the complex plane. By varying the values of α and β it is possible to move these zeros. Hence, the extra degree of freedom in the generalized PID controller can be used to affect the reference tracking of the output signal without modifying the closed-loop system's stability and robustness properties [1].

2.9.3 $\mathcal{L}_1$ adaptive control

Models used for controller design are often uncertain, due to e.g. unmodeled effects and uncertanties in the model parameters. Hence, the controlled system needs to be robust to these types of errors. Varying uncertainties make a system suitable for control using an $\mathcal{L}_1$ adaptive controller. A thorough description of $\mathcal{L}_1$ adaptive control theory is given in [17]. In this thesis an $\mathcal{L}_1$ adaptive controller of piecewise constant type is used. The same type of control architecture is also used in [22].

Problem formulation

Consider a system with the following dynamics

$$\dot{x}(t) = A_m x(t) + B_m u(t) + f(t, x(t)), \; x(0) = x_0, \tag{2.83}$$

$$y(t) = Cx(t), \tag{2.84}$$

where $x(t) \in \mathbb{R}^n$ is the system's state vector which is measured for feedback, $u(t) \in \mathbb{R}^m$ is the control signal, $y(t) \in \mathbb{R}^m$ is the controlled output, $A_m \in \mathbb{R}^{n\times n}$ is a matrix that specifies the desired closed-loop dynamics of the system, $B_m \in \mathbb{R}^{n\times m}$ is a known constant matrix with linearly independent columns, $C \in \mathbb{R}^{m\times n}$ is a known constant matrix of full rank and $f : \mathbb{R}\times\mathbb{R}^n \to \mathbb{R}^n$ is an unknown nonlinear function. In addition, (A_m, B_m) is controllable and (A_m, C) is observable.

The system given by (2.83) and (2.84) can be rewritten as

$$\dot{x}(t) = A_m x(t) + B_m (u(t) + f_1(t, x(t))) + B_{um} f_2(t, x(t)), \; x(0) = x_0, \tag{2.85}$$

$$y(t) = Cx(t), \tag{2.86}$$

where $B_{um} \in \mathbb{R}^{n\times(n-m)}$ is given by the null space of B_m^T with $\text{rank}([B_m\ B_{um}]) = n$ and $f_1 : \mathbb{R}\times\mathbb{R}^n \to \mathbb{R}^m$, $f_2 : \mathbb{R}\times\mathbb{R}^n \to \mathbb{R}^{n-m}$ are nonlinear functions that follow the relation

$$\begin{bmatrix} f_1(t, x(t)) \\ f_2(t, x(t)) \end{bmatrix} = B^{-1} f(t, x(t)), \tag{2.87}$$

$$B \triangleq \begin{bmatrix} B_m & B_{um} \end{bmatrix}. \tag{2.88}$$

The matrix B_m is called the matched input matrix and it is given by the system input model. This matrix relates the control signal and the matched uncertanties $f_1(\cdot)$ to the output. The matrix B_{um} is orthogonal to the direction of the matched input and relates the unmatched uncertainties $f_2(\cdot)$ to the output. The purpose of the $\mathcal{L}_1$ adaptive controller is to estimate and compensate for the uncertainties given by $f_1(\cdot)$ and $f_2(\cdot)$ and to make the output $y(t)$ track the output response of a system with the closed-loop dynamics given by A_m [17]. The $\mathcal{L}_1$ controller designed in this thesis (see Chapter 6) has only one state and hence no unmatched compensation.

Piecewise constant $\mathcal{L}_1$ adaptive controller

The main elements of a piecewise constant $\mathcal{L}_1$ adaptive controller are the following:

State predictor

The controller uses a state predictor according to

$$\dot{\hat{x}}(t) = A_m\hat{x}(t) + B_m(u(t) + \hat{\sigma}_1(t)) + B_{um}\hat{\sigma}_2(t), \tag{2.89}$$

where $\hat{\sigma}_1(t) \in \mathbb{R}^m$ and $\hat{\sigma}_2(t) \in \mathbb{R}^{n-m}$ are the estimates of the matched and unmatched uncertanties.

Adaptation law

The uncertanties are updated using a piecewise constant adaptation law described by

$$\begin{bmatrix} \hat{\sigma}_1(t) \\ \hat{\sigma}_2(t) \end{bmatrix} = \begin{bmatrix} \hat{\sigma}_1(kT_s) \\ \hat{\sigma}_2(kT_s) \end{bmatrix},\ t \in [kT_s,\ (k+1)T_s), \tag{2.90}$$

$$\begin{bmatrix} \hat{\sigma}_1(kT_s) \\ \hat{\sigma}_2(kT_s) \end{bmatrix} = -\begin{bmatrix} I_m & 0 \\ 0 & I_{(n-m)} \end{bmatrix} B^{-1}\Phi^{-1}e^{A_mT_s}\left(\hat{x}(kT_s) - x(kT_s)\right), \tag{2.91}$$

$$\Phi \triangleq A_m^{-1}\left(e^{A_mT_s} - I_m\right). \tag{2.92}$$

T_s is the sample time of the controller which ideally should be as low as possible.

Control law

The control signal from the $\mathcal{L}_1$ adaptive controller is formed in the frequency domain as

$$U(s) = C(s)\left(K_gR(s) - \hat{\sigma}_1(s) - H_m^{-1}(s)H_{um}(s)\hat{\sigma}_2(s)\right), \tag{2.93}$$

$$H_m(s) \triangleq C(sI_m - A_m)^{-1}B_m, \tag{2.94}$$

$$H_{um} \triangleq C(sI_m - A_m)^{-1} B_{um}, \tag{2.95}$$

where $C(s)$ is a low-pass filter and K_g is a constant gain matrix [17].

The low-pass filter is realized as

$$C(s) = (I + KD(s))^{-1} KD(s) \tag{2.96}$$

and its bandwidth is set by varying $KD(s)$. A simple choice is to set K as a diagonal matrix and $D(s)$ as an integrator. This gives a first order low-pass filter $C(s)$. The low-pass filter is used to restrict the $\mathcal{L}_1$ adaptive controller to only try to compensate for errors within the frequency range given by the low-pass filter's passband. In theory, the filter's cutoff frequency is limited by the highest frequency for which the model is accurate. In practice, it is a design parameter in the overall controller design.

3 Platform

This chapter provides an overview of the octorotor platform, its associated electronics and other equipment that is used in this thesis.

- Section 3.1 gives a general description of the platform and its mechanical structure.
- Section 3.2 desribes the octorotor's main electronics board that is used to control the aircraft during flight.
- Section 3.3 specifies what type of motor that is used on the platform to drive the rotors together with a description of the motor control electronics that are used to send commands to the DC motors.
- Section 3.4 describes the radio controller that is used by the pilot to control the aircraft.
- Section 3.5 describes the characteristics of the sensors mounted on the platform in more detail.
- Section 3.6 describes some basic concepts related to how the octorotor is controlled by varying the rotation rates of the rotors.

3.1 System overview

The platform that is used in this thesis is a Mikrokopter Okto-XL, manufactured by a German multirotor company called HiSystems GmbH [13]. The platform was bought by SAAB Dynamics AB in 2013 as a ready to fly product equipped with flight control electronics, navigation electronics, brushless motors, motor control electronics and more.

The source code for the software accompanying the flight and navigation electronics is either closed or hard to modify. Hence it was decided that a new flight and navigation card with software that could be more easily modified should be purchased and integrated onto the platform. This card was also bought by SAAB Dynamics AB and was available at the start of the thesis work. This card is described in Section 3.2.

Figure 3.1 shows a picture of the octorotor platform. The central part of the octorotor contains all the flight avionics and electronical control circuits as well as the battery pack. This part contributes the most to the total weight of the octorotor apart from the motors. To the center structure there are eight metal arms attatched, which are arranged in a symmetric pattern around the center structure with a 45 degree angle between each other. They are slim and light but still very rigid.

The rotors and motors are attached to the edge of each arm with the necessary cabling running inside the hollow structure of the arms. The platform uses fixed pitch motors which means that the octorotor cannot be controlled by tilting the rotor axis in different directions, instead the octorotor is controlled by varying the rotation speed of the rotors according to some strategy. For this particular octorotor each rotor axis is tilted three degrees in the plane that is perpendicular to the motor arm. The direction of tilt alternates between each rotor.

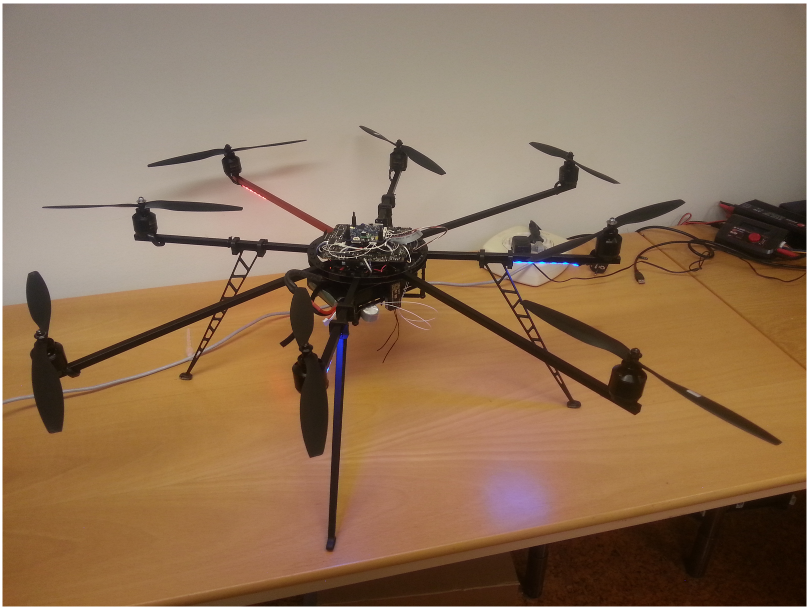


Figure 3.1: *An overview of the octorotor platform. The front of the octorotor is distinguished by the red arm. The motors are mounted at the end of each arm. All avionics are mounted on the central plate.*

The rotors are numbered from one to eight and each odd-numbered rotor is attached to one of the longer arms. The even-numbered rotors are attached to the shorter arms. Rotor number one is attached to an arm that is colored red to distinguish it as the front of the octorotor. To prevent the octorotor from rotating in the horizontal plane, the odd-numbered rotors spin clockwise and the even-numbered spin anti-clockwise. This is an important feature which removes the need of a tail rotor like the one used on a helicopter.

The octorotor can be illustrated as a symmetric cross structure with eight arms. At the end of each arm there is a rotor attached that can provide thrust and hence affect the platform's movement in 3D-space. Due to the rigidness of the structure it is also possible to model the aircraft using rigid body mechanics. During flight the octorotor has six degrees of freedom (6DoF) given by its position and orientation in 3D-space. It is not possible to directly control the octorotor's translational velocity in the horizontal plane. Instead, this has to be controlled through the dynamics of the system [21]. In this aspect, the octorotor is an underactuated system.

Basically, a multirotor's motion is controlled by varying the rotation rate of the rotors and hence the thrust generated by each rotor. This is however a very unintuitive way for a human operator to control the aircraft. Hence, the octorotor is instead controlled using four different control commands: throttle, roll, pitch and yaw. These four commands are related to the four basic movements that the multirotor can perform with respect to a body-fixed coordinate system, vertical movement and rotation around the three principal axes of the aircraft. In this aspect, the octorotor with its eight rotors is an overactuated system. By manipulating the four commands it is possible to make the multirotor attain a desired altitude and attitude. This is described in more detail in Section 3.6.

3.2 Flight and navigation control electronics

The new flight and navigation board mounted on the octorotor is a UAVP-NG Hardware 0.24-mini, constructed by a community-driven open source project [24]. It consists of one PCB with the following electronics:

- 32-bit LPC2148 ARM7 CPU 60 MHz
- three-dimensional gyroscope (MPU6000)
- two three-dimensional accelerometers (MPU6000 + LIS3LV02DQ)
- barometer (MS5611B)
- Venus GPS.

In addition, the electronics board contains outputs for up to 16 motors and five servos. The motors are controlled via the I^2C protocol by sending motor commands to the separate motor control electronics, see Section 3.3. Two onboard UART:s can be used for serial communication. One of them is used for ground station telemetry with a Bluetooth transciever. The other serial port is used by the GPS-unit. The communication with the remote control is handled by a PPM port. There is also a micro-USB connection which can be used to connect external devices, such as a PC, to the board via USB-cable [24]. A schematic overview of the electronics board with its components and connections, can be seen in Figure 3.2.

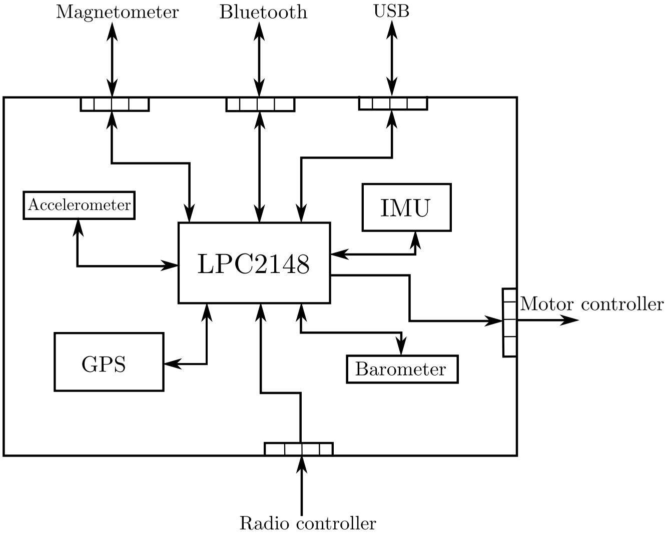


Figure 3.2: *Overview of the system hardware and the internal- and external flow of data between different components. The LPC2148 is the hub of all communication.*

The electronics board also ships with three separate circuit boards: a board with an attached Atmel ATmega644P CPU that can only be used for camera control, a JTAG-adapter and a board with an attached digital compass (HMC5883L).

3.3 Motors and motor controller

The motors that are used on the octocopter are of the type MK3638 and are manufactured specifically for the multirotors sold by HiSystems GmbH [14]. The motors are brushless DC motors which require special control electronics. Brushless motors are used in favor of brush-type motors because they have superior efficiency, longer life, smoother torque delivery and can operate at high speeds. A brushless DC motor has its wire coils in the core of the motor and the permanent magnet rotor attached to the outer housing of the motor. To be able to use these type of motors, the DC from the batteries must be converted to a triphasic alternating current. Also, the power output must be controlled in order to control the rotation rate of the motors as desired, which is done by the motor control electronics [2].

The Brushless-Control (BL-Ctrl) version 2.0 board is used to control the brushless DC motors. These electronics are designed and sold by HiSystems GmbH and they are designed for controlling the MK3638 brushless motor. The control board software is closed source. The board receives commands, which will be referred to as ESC commands in this thesis, via I^2C from the flight control board and translates the command to a desired rotation rate of the rotor. The ESC commands can be given in the interval $[0\ 2^{16}-1]$ but the command that is sent to the motor controller only consists of eight bits, hence the actual resolution is lower. The BL-Control board contains a closed-loop controller which makes the motors attain the desired rotation rate both fast and smoothly [12]. How the dynamic and static behaviour of the motors and the control board are modeled are described in more detail in Section 4.3 and 4.4.

3.4 Radio controller

To be able to send reference signals to the platform during flight, a Graupner MC-32 HoTT radio controller operating at 2.4 GHz is used. It contains a lot of features, e.g. telemetry analysis and transmitter display in real time, a warning function that notifies if connection with the platform is lost and it has 16 channels that can be used to send different information to the flight and navigation card [9]. Three of these channels are used to transmit reference signals to the closed-loop controllers. A fourth channel is used to control the throttle. This is illustrated in Figure 3.3.

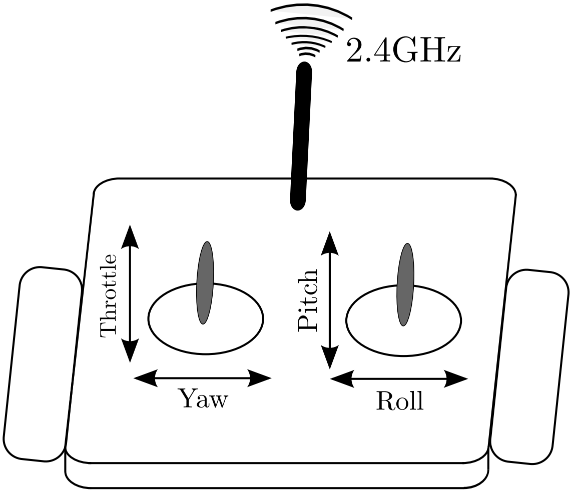


Figure 3.3: *Radio Controller with two joysticks. The left joystick is used to control the throttle by moving it vertically and the yaw rate by moving it horizontally. The right joystick is used to control the pitch demand by moving it vertically and the roll demand by moving it horizontally.*

3.5 Sensor characteristics

There are a variety of sensors attached to the octorotor's main electronics board and these sensors deliver measurements of different physical properties. This section gives a description of how some of these sensors work and what they measure.

3.5.1 Accelerometer

An accelerometer can be described as a device that measures forces acting on a small proof mass constrained to move together with it. When the accelerometer is subjected to an acceleration the proof mass will resist the movement due to its own inertia. This will exert forces that can be measured and related to the acceleration via the relation $F = ma$. In free fall, the accelerometer is not subjected to any external forces and will therefore report *zero* acceleration, even though the accelerometer is accelerating with $g \approx 9.82$ $[\mathrm{m/s^2}]$ towards the center of the earth. The accelerometer can therefore be seen as a device that measures accelerations in the body-fixed frame relative to free fall [29].

When the accelerometer is stationary, the downward force on the proof mass resulting from the gravitational acceleration is counteracted by the upward normal force that keeps the accelerometer in its place. Due to this, an accelerometer at rest on the surface of the earth will measure the *upward* acceleration of g $[\mathrm{m/s^2}]$. This is illustrated in Figure 3.4.

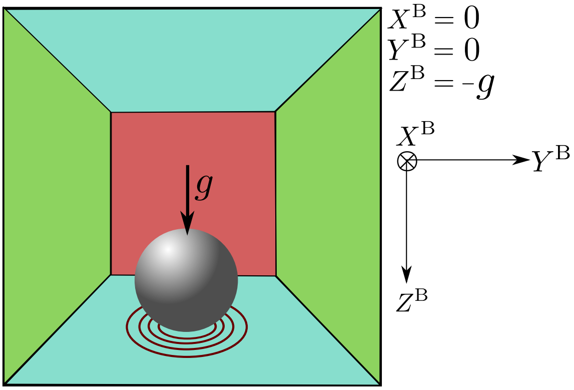


Figure 3.4: *Schematic illustration of an accelerometer at rest on the earth's surface. The proof mass inside is displaced in the positive Z^B-direction. The accelerometer measures an acceleration in the negative Z^B-direction.*

By combining three different accelerometers and mounting them orthogonal to each other it is possible to measure 3D-acceleration relative to a body-fixed system. The measurement equation that describes what an accelerometer mounted in a rigid body's center of mass measures is

$$\mathbf{a}^{\mathrm{B}}_{\mathrm{acc}} = \mathbf{a}^{\mathrm{B}}_{\mathrm{ext}} - \mathbf{R}_{\mathrm{E}\to\mathrm{B}}\mathbf{a}^{\mathrm{E}}_{g} + \mathbf{b}^{\mathrm{B}}_{\mathrm{acc}} + \boldsymbol{\eta}^{\mathrm{B}}_{\mathrm{acc}}. \tag{3.1}$$

Here, $\mathbf{a}^{\mathrm{B}}_{\mathrm{ext}}$ is the external accelerations affecting the accelerometer expressed in the body-fixed coordinate system, $\mathbf{a}^{\mathrm{E}}_{g}$ is the gravitational acceleration, $\mathbf{b}^{\mathrm{B}}_{\mathrm{acc}}$ is a bias term and $\boldsymbol{\eta}^{\mathrm{B}}_{\mathrm{acc}}$ is measurement noise [21].

3.5.2 Gyroscope

A gyroscope is a measurement device that can be used to measure either the angle turned by an object, a so called displacement gyroscope, or the angular rate of turn, a so called rate gyroscope [29]. The type of gyroscope that is mounted on the octorotor is a rate gyroscope and hence it measures angular velocities expressed in the body-fixed frame. The measurement equation for the gyroscope is similar to (3.1) and it is given by

$$\boldsymbol{\omega}^{\mathrm{B}}_{\mathrm{gyro}} = \boldsymbol{\omega}^{\mathrm{B}} + \mathbf{b}^{\mathrm{B}}_{\mathrm{gyro}} + \boldsymbol{\eta}^{\mathrm{B}}_{\mathrm{gyro}}. \tag{3.2}$$

Here, $\mathbf{b}^{\mathrm{B}}_{\mathrm{gyro}}$ is a slowly varying bias term and $\boldsymbol{\eta}^{\mathrm{B}}_{\mathrm{gyro}}$ is additive measurement noise [21]. The gyroscope provides accurate angular rate measurements and the signal to noise ratio is higher compared to the accelerometer readings. However, the slowly varying bias term causes problems when the gyroscope measurements are integrated to get angle estimates.

3.5.3 Magnetometer

A magnetometer, or digital compass, is a device capable of measuring the earth's magnetic field. The measurement equation for the magnetometer is given by

$$\mathbf{m}^{\mathrm{B}}_{\mathrm{mag}} = \mathbf{R}_{\mathrm{E}\to\mathrm{B}}\mathbf{m}^{\mathrm{E}} + \boldsymbol{\xi}^{\mathrm{B}} + \boldsymbol{\eta}^{\mathrm{B}}_{\mathrm{mag}}. \tag{3.3}$$

Here, $\mathbf{m}^{\mathrm{E}}$ is the earth's magnetic field expressed relative to the earth-fixed coordinate system, $\boldsymbol{\xi}^{\mathrm{B}}$ are local magnetic disturbances expressed in the body-fixed frame and $\boldsymbol{\eta}^{\mathrm{B}}_{\mathrm{mag}}$ is measurement noise [21]. The measurement noise is low for typical magnetometers but the local disturbances can be significant and sometimes render the measurements unuseable. The disturbances are caused by nearby magnetic fields induced by flowing electrical currents in the avionics and motor power wires. Nearby metallic objects can also cause significant disturbances in the local magnetic field. Hence, it is very important that the magnetometer is placed as far away from the motor power wires as possible [21].

3.6 Basic octorotor control concepts

A multirotor is controlled by varying the rotation rates of the motors. The change in rotation rates will exert forces and torques on the platform which will affect the linear and angular position of the multirotor. This section describes how the different control commands: throttle, roll, pitch and yaw rate can be realized.

3.6.1 Throttle

When the octorotor is in stationary horizontal flight in an undisturbed airspace no net forces and torques act on the octorotor and therefore it remains in its position in space. In order to achieve this the thrust force that is generated by the rotors have to counteract the downward acting force that is exerted by the gravitational acceleration. In addition, all rotors must provide a certain amount of thrust so that no net torque is generated with respect to one of the aircraft's principal axes.

The throttle command is realized by increasing or decreasing the rotation rate of all the platform's rotors. Since the amount of thrust that each rotor generates is related to the rotation rate this will generate a vertical force in the octorotor's body-fixed frame. If the octorotor is flying horizontally, i.e. the vertical direction of the octorotor's body-fixed frame is aligned with the vertical direction of the inertial frame, the platform will move vertically up or down. If this is not the case, the increased or decreased thrust provided by the given throttle command will generate both vertical and horizontal accelerations in the earth-fixed frame. Figure 3.5 shows how the thrust command works.

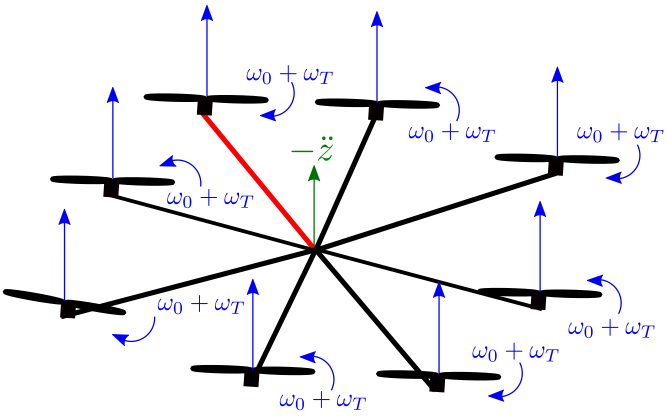


Figure 3.5: *Illustration of the thrust command*

3.6.2 Roll

The roll command is used to control the rotation around the x-axis of the octorotor. In order to realize a positive roll command a torque with respect to the x-axis has to be generated. This is done by increasing the rotation speed of rotor number six, seven and eight, and simultaneously decreasing the rotation speed of rotor number two, three and four. These rotors are attached at a non-zero perpendicular distance from the x-axis of the body-fixed frame and hence a torque will be generated. The torque will induce an angular acceleration around the x-axis and the platform will roll. The change in rotation speeds is supposed to be executed in such way that the total vertical thrust is kept constant, see Figure 3.6.

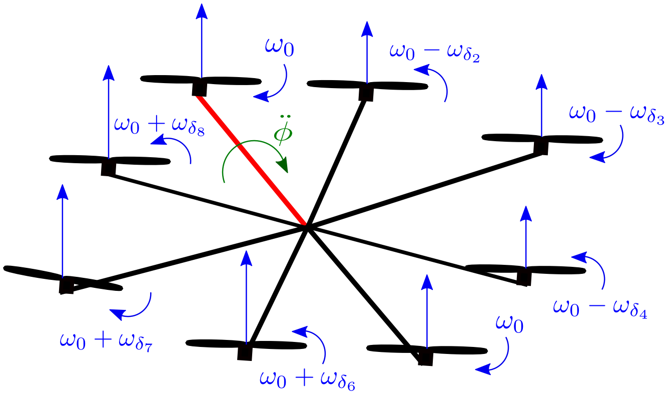

ω_0
$\omega_0 - \omega_{\delta_2}$
$\omega_0 - \omega_{\delta_3}$
$\omega_0 + \omega_{\delta_8}$
$\ddot{\phi}$
$\omega_0 - \omega_{\delta_4}$
ω_0
$\omega_0 + \omega_{\delta_7}$
$\omega_0 + \omega_{\delta_6}$

Figure 3.6: *Illustration of the roll command*

3.6.3 Pitch

The pitch command is given by the operator in order to control the octorotor's rotation around the y-axis. A positive pitch command is realized in a similar way to the roll command. The rotation speed, and hence the thrust, provided by rotor number four, five and six is decreased and the rotation speed of rotor number one, two and eight is increased. Since these rotors are attached at a non-zero perpendicular distance from the aircraft's y-axis, the change in thrust will exert a torque which in turn will create an angular acceleration around the same axis. As for the roll command, the rotation speed changes are performed in such a way that the total vertical thrust stays the same. Figure 3.7 illustrates the procedure.

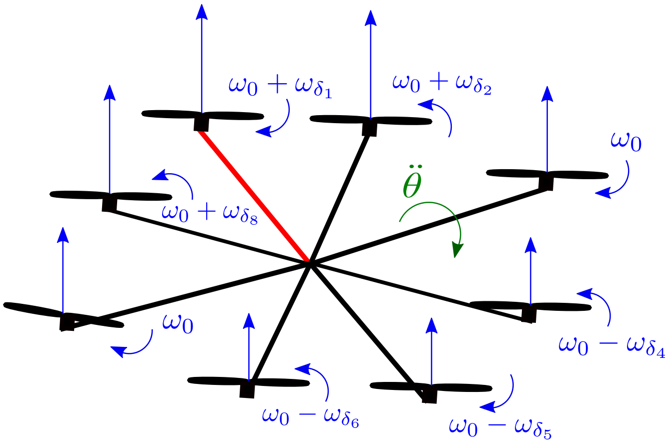

$\omega_0 + \omega_{\delta_1}$
$\omega_0 + \omega_{\delta_2}$
ω_0
$\ddot{\theta}$
$\omega_0 + \omega_{\delta_8}$
ω_0
$\omega_0 - \omega_{\delta_4}$
$\omega_0 - \omega_{\delta_6}$
$\omega_0 - \omega_{\delta_5}$

Figure 3.7: *Illustration of the pitch command*

3.6.4 Yaw rate

The aircraft's rotation around the z-axis of the body-fixed frame is controlled via the yaw rate command. This command is a bit different from the pitch and roll commands since it is not directly related to the upward thrust that is generated by each rotor. When the rotors turn through the air they generate a torque with respect to the rotor axis. The amount of torque that is generated is related to the rotation speed of the rotor. A higher rotation speed gives a higher torque. This torque leads to an angular acceleration around the octorotor's z-axis which makes the aircraft turn. Since the odd-numbered rotors rotate clockwise and the even-numbered rotors rotate anti-clockwise, the torque that is generated is balanced in stationary flight and the aircraft does not turn. By increasing (or decreasing) the rotation speed of the clockwise rotating rotors and simultaneously decreasing (or increasing) the rotation speed of the anti-clockwise rotating rotors, the aircraft's rotation around the z-axis can be controlled. As in the above cases, the total vertical thrust is kept constant so that the octorotor's altitude does not change. See Figure 3.8 for an illustration.

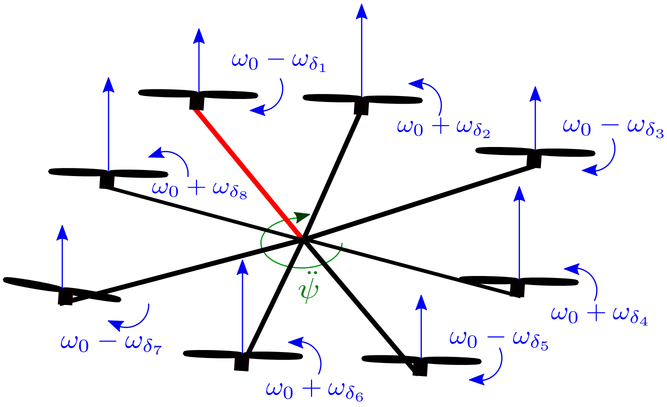

$\omega_0 - \omega_{\delta_1}$
$\omega_0 + \omega_{\delta_2}$
$\omega_0 - \omega_{\delta_3}$
$\omega_0 + \omega_{\delta_8}$
$\ddot{\psi}$
$\omega_0 + \omega_{\delta_4}$
$\omega_0 - \omega_{\delta_5}$
$\omega_0 - \omega_{\delta_7}$
$\omega_0 + \omega_{\delta_6}$

Figure 3.8: Illustration of the yaw rate command

4

Modeling and parameter identification

This chapter contains a derivation of the octorotor model that is used for simulation and controller design in Matlab/Simulink. By using the model equations it is possible to simulate and predict the octorotor's motion based on the current rotor speeds and state of the octorotor. This model depends on a variety of different model parameters. In order to get a complete model, these parameters need to be identified. How this was done is also described in this chapter.

- Section 4.1 provides an overview of the forces and torques that act on the octorotor platform. An understanding of these forces is needed to get a complete model of the octorotor.
- Section 4.2 summarizes the results and presents the complete dynamic and kinematic model for the octorotor airframe.
- Section 4.3 describes how the modeling of the motor dynamics was carried out and provides a model describing the motor dynamics.
- Section 4.4 contains a description of how the dynamic properties of the motors were determined.
- Section 4.5 details how the moment of inertia was estimated by weighing and measuring the different components of the octorotor.
- Section 4.6 provides a description of how the thrust constant for the propellers were estimated from static thrust tests using a test rig.
- Section 4.7 describes how the propellers' torque constant was estimated in a similar way to the thrust constant using a test rig.

4.1 Forces and torques acting on the octorotor

The Newton-Euler equations derived in Section 2.4 show how the octorotor's linear and angular position are affected when the octorotor is subjected to external forces and torques. These forces and torques can be grouped together into different categories:

- thrust and drag
- the force of gravity
- gyroscopic torque
- aerodynamic effects.

4.1.1 Thrust and drag

The primary forces and torques that affect the octorotor in flight are provided by the generated thrust and drag by the rotors. When a propeller rotates through the air it generates an upwards thrust aligned with the rotor axis. In addition, the movement of the propeller also generates a torque with respect to the rotor axis. Both the generated thrust and torque are related to the rotation rate of the rotor and are described in [21] as

$$\mathbf{F}_i^{\mathrm{B}} = F_i \mathbf{e}_i^{\mathrm{B}} = c_T \omega_i^2 \mathbf{e}_i^{\mathrm{B}}, \quad i = 1, 2, \ldots, 8, \tag{4.1}$$

$$\boldsymbol{\tau}_{Q_i}^{\mathrm{B}} = \tau_{Q_i} \mathbf{e}_i^{\mathrm{B}} = (-1)^{i+1} c_Q \omega_i^2 \mathbf{e}_i^{\mathrm{B}}, \quad i = 1, 2, \ldots, 8. \tag{4.2}$$

Here, ω_i is the rotation rate of rotor i and $\mathbf{e}_i^{\mathrm{B}}$ is the normalized rotor axis vector for rotor i expressed in the body-fixed frame. These equations are approximations that are valid for a hovering rotor in free air. The equations state that both the generated force and torque by a spinning rotor in free air are proportional to the square of the rotation rate of the rotor. The proportionality constants, c_T and c_Q, are referred to as the *thrust constant* and *torque constant* in this thesis. These constants depend on the rotor geometry and the density of air [21]. The values of these constants were obtained experimentally and how this was done is explained in Section 4.6 and 4.7.

The thrust forces that act on the octorotor will also generate a torque with respect to the octorotor's center of gravity, since the forces are applied at a distance from this point. This torque is given by the cross product of the displacement vector, $\mathbf{r}_i^{\mathrm{B}}$, and the force vector, $\mathbf{F}_i^{\mathrm{B}}$, i.e.

$$\boldsymbol{\tau}_{F_i}^{\mathrm{B}} = \mathbf{r}_i^{\mathrm{B}} \times \mathbf{F}_i^{\mathrm{B}}, \quad i = 1, 2, \ldots, 8. \tag{4.3}$$

The total torque that is generated due to the rotation of the motors' propellers is therefore a sum of $\boldsymbol{\tau}_{Q_i}^{\mathrm{B}}$ and $\boldsymbol{\tau}_{F_i}^{\mathrm{B}}$ given by (4.2) and (4.3).

The motors are tilted three degrees in the plane perpendicular to the arms on which the motors are mounted. Hence, the expressions for the rotor axis vectors $\mathbf{e}_i^{\mathrm{B}}$ become more complicated. If the motors would not have been tilted these vectors would only have a component in the Z^{B}-axis. The expressions for the rotor axis vectors are

$$
\begin{aligned}
\mathbf{e}_1^{\mathrm{B}} &= \begin{bmatrix} 0 & -\sin\alpha & -\cos\alpha \end{bmatrix}^T, \\
\mathbf{e}_2^{\mathrm{B}} &= \begin{bmatrix} -\frac{1}{\sqrt{2}}\sin\alpha & \frac{1}{\sqrt{2}}\sin\alpha & -\cos\alpha \end{bmatrix}^T, \\
\mathbf{e}_3^{\mathrm{B}} &= \begin{bmatrix} \sin\alpha & 0 & -\cos\alpha \end{bmatrix}^T, \\
\mathbf{e}_4^{\mathrm{B}} &= \begin{bmatrix} -\frac{1}{\sqrt{2}}\sin\alpha & -\frac{1}{\sqrt{2}}\sin\alpha & -\cos\alpha \end{bmatrix}^T, \\
\mathbf{e}_5^{\mathrm{B}} &= \begin{bmatrix} 0 & \sin\alpha & -\cos\alpha \end{bmatrix}^T, \\
\mathbf{e}_6^{\mathrm{B}} &= \begin{bmatrix} \frac{1}{\sqrt{2}}\sin\alpha & -\frac{1}{\sqrt{2}}\sin\alpha & -\cos\alpha \end{bmatrix}^T, \\
\mathbf{e}_7^{\mathrm{B}} &= \begin{bmatrix} -\sin\alpha & 0 & -\cos\alpha \end{bmatrix}^T, \\
\mathbf{e}_8^{\mathrm{B}} &= \begin{bmatrix} \frac{1}{\sqrt{2}}\sin\alpha & \frac{1}{\sqrt{2}}\sin\alpha & -\cos\alpha \end{bmatrix}^T.
\end{aligned}
$$

Here, α is the rotor tilt angle and the factor $1/\sqrt{2}$ is due to the fact that the motor mounting arms are spread symmetrically with an angle of $\pi/4$ between each arm and that $\sin\pi/4 = \cos\pi/4 = 1/\sqrt{2}$.

The displacement vectors $\mathbf{r}_i^{\mathrm{B}}$, which represent the distance from the origin in the body-fixed coordinate system to the motors, are calculated by choosing the center of the octorotor as a reference point and expressing a vector from this reference point to the motor in body coordinates. A displacement vector from this reference point to the octorotor's center of gravity (calculated in Section 4.5) is calculated and subtracted. The result is a displacement vector from the octorotor's center of gravity to the motor attachment point

$$
\begin{aligned}
\mathbf{r}_1^{\mathrm{B}} &= \begin{bmatrix} d_o & 0 & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}, \\
\mathbf{r}_2^{\mathrm{B}} &= \begin{bmatrix} \frac{1}{\sqrt{2}}d_e & \frac{1}{\sqrt{2}}d_e & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}, \\
\mathbf{r}_3^{\mathrm{B}} &= \begin{bmatrix} 0 & d_o & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}, \\
\mathbf{r}_4^{\mathrm{B}} &= \begin{bmatrix} -\frac{1}{\sqrt{2}}d_e & \frac{1}{\sqrt{2}}d_e & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}, \\
\mathbf{r}_5^{\mathrm{B}} &= \begin{bmatrix} -d_o & 0 & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}, \\
\mathbf{r}_6^{\mathrm{B}} &= \begin{bmatrix} -\frac{1}{\sqrt{2}}d_e & -\frac{1}{\sqrt{2}}d_e & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}, \\
\mathbf{r}_7^{\mathrm{B}} &= \begin{bmatrix} 0 & -d_o & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}, \\
\mathbf{r}_8^{\mathrm{B}} &= \begin{bmatrix} \frac{1}{\sqrt{2}}d_e & -\frac{1}{\sqrt{2}}d_e & 0 \end{bmatrix}^T - \mathbf{r}_G^{\mathrm{B}}.
\end{aligned}
$$

Here, d_o is the length of the odd-numbered rotor arms and d_e is the length of the even-numbered rotor arms.

Given these vectors, it is now possible to relate the exerted thrust and drag forces and torques on the platform to the motors' rotation speeds. By using the definitions

$$\mathbf{a}_i^{\mathrm{B}} \triangleq c_T \mathbf{e}_i^{\mathrm{B}}, \quad i = 1, 2, \ldots, 8, \tag{4.4}$$

$$\mathbf{b}_i^{\mathrm{B}} \triangleq \mathbf{r}_i^{\mathrm{B}} \times c_T \mathbf{e}_i^{\mathrm{B}} + (-1)^{i+1} c_Q \mathbf{e}_i^{\mathrm{B}}, \quad i = 1, 2, \ldots, 8, \tag{4.5}$$

$$\mathbf{\Lambda}^{\mathrm{B}} \triangleq \begin{bmatrix} \mathbf{a}_1^{\mathrm{B}} & \mathbf{a}_2^{\mathrm{B}} & \mathbf{a}_3^{\mathrm{B}} & \mathbf{a}_4^{\mathrm{B}} & \mathbf{a}_5^{\mathrm{B}} & \mathbf{a}_6^{\mathrm{B}} & \mathbf{a}_7^{\mathrm{B}} & \mathbf{a}_8^{\mathrm{B}} \\ \mathbf{b}_1^{\mathrm{B}} & \mathbf{b}_2^{\mathrm{B}} & \mathbf{b}_3^{\mathrm{B}} & \mathbf{b}_4^{\mathrm{B}} & \mathbf{b}_5^{\mathrm{B}} & \mathbf{b}_6^{\mathrm{B}} & \mathbf{b}_7^{\mathrm{B}} & \mathbf{b}_8^{\mathrm{B}} \end{bmatrix}, \tag{4.6}$$

$$\omega_{\mathrm{rot}}^2 \triangleq \begin{bmatrix} \omega_1^2 & \omega_2^2 & \omega_3^2 & \omega_4^2 & \omega_5^2 & \omega_6^2 & \omega_7^2 & \omega_8^2 \end{bmatrix}^T, \tag{4.7}$$

the relation can be written in matrix form as

$$\begin{bmatrix} \mathbf{F}_r^{\mathrm{B}} \\ \boldsymbol{\tau}_r^{\mathrm{B}} \end{bmatrix} = \mathbf{\Lambda}^{\mathrm{B}} \, \omega_{\mathrm{rot}}^2. \tag{4.8}$$

The rotor tilt in combination with the fact that not all arms are of equal length give raise to weak forces in the X^{B}- and Y^{B}-directions from the rotors' thrust. Hence, the constant matrix $\mathbf{\Lambda}^{\mathrm{B}}$ is a 6×8 matrix that describes how the rotation of the rotors are related to the applied forces and torques on the octorotor. If a platform without rotor tilt is used, $\mathbf{\Lambda}^{\mathrm{B}}$ simplifies to a 4×8 matrix.

4.1.2 The force of gravity

The gravitational acceleration and the resulting downward force on the octorotor is always aligned with the Z^{E}-axis of the earth-fixed frame. In addition, its application point is the octorotor's center of gravity (which coincides with the center of mass in a uniform gravitational field [8]) and hence the gravitational pull will not exert any torque with respect to this point. The gravitational force expressed in coordinates relative to the earth-fixed frame is given by

$$\mathbf{F}_g^{\mathrm{E}} = m \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T. \tag{4.9}$$

Here, g is the gravitational acceleration constant. By using the inverse (transpose since it is orthogonal) of the rotation matrix defined in (2.11) the gravitational force vector can be expressed in coordinates relative to the body-fixed coordinate system instead

$$\mathbf{F}_g^{\mathrm{B}} = m \mathbf{R}_{\mathrm{E} \to \mathrm{B}} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}. \tag{4.10}$$

4.1.3 Gyroscopic torque

Since the motors are constantly rotating during flight torques will be exerted on the octorotor when it changes its angular position according to Euler's law in (2.38). To derive an expression for this torque, it is assumed that a coordinate system R which is aligned with the rotor axis has its origin fixed inside the center of gravity of one of the octorotor's engines. Since this system is fixed on the octorotor it will rotate the same way as the body-fixed coordinate system and hence have the same angular velocity vector, $\boldsymbol{\omega}$. In addition, since the coordinate system R is fixed in the motor the inertia tensor for the engine with respect to the motors's center of gravity is constant. Using (2.40) together with (2.23) the torque around the motors's center of gravity can be expressed as

$$\boldsymbol{\tau}_{\text{gyr}_i}^{\text{R}} = \left(\frac{\mathrm{d}\mathbf{I}_{\text{rot}}\boldsymbol{\omega}_{\text{rot}_i}}{\mathrm{dt}}\right)_I^{\text{R}} = \left(\frac{\mathrm{d}\mathbf{I}_{\text{rot}}\boldsymbol{\omega}_{\text{rot}_i}}{\mathrm{dt}}\right)_{\text{R}}^{\text{R}} + \boldsymbol{\omega}^{\text{R}} \times \mathbf{I}_{\text{rot}}^{\text{R}}\boldsymbol{\omega}_{\text{rot}_i}^{\text{R}}, \quad i = 1, 2, \ldots, 8. \tag{4.11}$$

The motors and rotors are symmetric with respect to the rotor axis and the engines only rotate around the Z^{R}-axis. Thus, the inertia matrix is diagonal and the rotation velocity vector $\boldsymbol{\omega}_{\text{rot}_i}^{\text{R}}$ only has a component in the Z^{R}-axis.

The rotor-fixed coordinate system for each rotor is rotated with respect to the body-fixed coordinate system due to the three degree rotor axis tilt. It is hence necessary to be able to convert between coordinate representations in the two frames. This is done by using a rotation matrix. Each rotor has its own R-system R_i and therefore needs its own rotation matrix $\mathbf{R}_{\text{R}_i \to \text{B}}$. This rotation matrix is calculated using (2.11) and substituting correct values for ϕ, θ, and ψ. Equation (4.11) can hence be expressed as

$$\boldsymbol{\tau}_{\text{gyr}_i}^{\text{R}} = \left(\frac{\mathrm{d}\mathbf{I}_{\text{rot}}\boldsymbol{\omega}_{\text{rot}_i}}{\mathrm{dt}}\right)_{\text{R}}^{\text{R}} + \left(\mathbf{R}_{\text{B}\to\text{R}_i}\boldsymbol{\omega}^{\text{B}}\right) \times \mathbf{I}_{\text{rot}}^{\text{R}}\boldsymbol{\omega}_{\text{rot}_i}^{\text{R}}, \quad i = 1, 2, \ldots, 8. \tag{4.12}$$

The gyroscopic torque in (4.12) is expressed in coordinates relative to the rotor-fixed coordinate system. It is hence converted into body-fixed coordinates using $\mathbf{R}_{\text{R}_i \to \text{B}}$. The total gyroscopic torque is calculated as the sum of all the gyroscopic torques generated by each rotor's rotation

$$\boldsymbol{\tau}_{\text{gyr}}^{\text{B}} = \sum_{i=1}^{8} \boldsymbol{\tau}_{\text{gyr}_i}^{\text{B}} = \sum_{i=1}^{8} \mathbf{R}_{\text{R}_i \to \text{B}}\boldsymbol{\tau}_{\text{gyr}_i}^{\text{R}}. \tag{4.13}$$

4.1.4 Aerodynamic effects

In addition to all the above effects the octorotor is also affected by different aerodynamic effects as it moves through the airspace. These effects will also exert forces and torques on the octorotor and hence alter its movement. Some of these effects, referred to as blade flapping and induced drag, are explicitly modeled in [21]. However, the authors state that the main reason for including these effects in the model is to understand why certain state estimation algorithms provide good performance and that high performance control of multirotors have been demonstrated using models without these effects. The presented model is a simplification that depends on parameters related to the specific multirotor that is used. Most papers in the literature concerning multirotors use a physical model where these aerodynamic effects are not explicitly modeled [23]. Due to this, it was decided that these effects should not be explicitly modeled in this thesis and they are referred to as $\mathbf{F}^{\mathrm{B}}_{\mathrm{aero}}$ and $\boldsymbol{\tau}^{\mathrm{B}}_{\mathrm{aero}}$ respectively. The aerodynamic effects are instead viewed as external disturbances and noise.

4.2 Resulting model

By combining the results from Sections 2.1 through 2.4 and 4.1 it is now possible to write down the complete model in a matrix representation

$$\begin{bmatrix} m\dot{\mathbf{v}} \\ \mathbf{I}\dot{\omega} \end{bmatrix} = \begin{bmatrix} -\omega \times m\mathbf{v} \\ -\omega \times \mathbf{I}\omega \end{bmatrix} + \begin{bmatrix} \mathbf{F}_g \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau}_{\mathrm{gyr}} \end{bmatrix} + \boldsymbol{\Lambda}\omega^2_{\mathrm{rot}} + \begin{bmatrix} \mathbf{F}_{\mathrm{aero}} \\ \boldsymbol{\tau}_{\mathrm{aero}} \end{bmatrix}, \tag{4.14}$$

$$\begin{bmatrix} \dot{\boldsymbol{\Gamma}}^{\mathrm{E}} \\ \dot{\boldsymbol{\Theta}}^{\mathrm{E}} \\ \dot{\mathbf{R}}_{\mathrm{B}\to\mathrm{E}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{\mathrm{B}\to\mathrm{E}}\mathbf{v} \\ \mathbf{T}_{\mathrm{B}\to\mathrm{E}}\omega \\ \mathbf{R}_{\mathrm{B}\to\mathrm{E}}\omega_\times \end{bmatrix}. \tag{4.15}$$

In the above equations the index B have been dropped for notational brevity. It is assumed that all bold-faced letters without an index represent vectors or tensors with their coordinates expressed with respect to the body-fixed frame, or transformation matrices. A dot represents derivation with respect to time of the used *coordinate representation*. Equation (4.14) describes the octorotor's dynamics and (4.15) describes the kinematics. The linear and angular position of the octorotor is calculated by first solving the differential equations in (4.14) for the dynamics and then substituting the result into the kinematic equations in (4.15). Solving these equations yield the linear and angular position of the octorotor as a function of time.

4.3 Motor dynamics

The octocopter uses brushless DC-motors to drive its rotors, see Section 3.3. The motors are controlled via rotation rate commands sent over the $\mathrm{I}^2\mathrm{C}$-bus to the ESC:s. Each ESC has an internal closed-loop controller that makes the rotor attain and hold the commanded rotation rate [12]. The commanded rate is not reached immediately and hence a dynamic model describing the relation between the sent ESC command and the rotation rate is needed. It is sufficient to model the motor dynamics of a brushless DC-motor controlled by an ESC in closed-loop mode as a first order linear model with time delay [5]. Hence, the transfer function relating the given ESC command u_m and the obtained rotation rate ω_{rot} can be expressed as

$$\omega_{\mathrm{rot}}(s) = \frac{K_m}{1 + sT_m} e^{T_d s} \cdot U_m(s). \tag{4.16}$$

This linear model is parameterized by the static gain K_m, the time constant T_m and the time delay T_d.

The time constant in the linear model depends on the selected combination of

ESC, motor and propeller. In general, more powerful motors and smaller propellers give a lower time constant. The time delay in (4.16) is due to communication delays between the main CPU and the ESC:s. For ESC:s using I^{2}C communication, this delay is only a few milliseconds which is low compared to the time constant [5]. To simplify the model it was hence decided that the time delay should be neglected so that the motor dynamics could be modeled as a first order linear system with no time delay. The same model structure is also used in [16]. According to the previous discussion, the model for the motor dynamics used in this thesis is

$$\omega_{\text{rot}}(s) = \frac{K_m}{1 + sT_m} U_m(s). \tag{4.17}$$

4.4 Identification of motor model parameters

This section describes how the unknown parameters K_m and T_m in (4.17) was identified.

4.4.1 Static gain

In order to determine the static gain parameter in the linear model of the motor dynamics, it is sufficient to measure the obtained rotation speed in stationarity for some different values of the ESC commands. The motor control electronics does not provide any feedback information about the current rotation rate of the rotors. Hence, a digital tachometer was used to provide information about the rotation speed in stationarity. By attaching a reflective tape strip on the rotor, the rotation rate was measured by shining a laser beam towards the tape and counting the number of reflections per time unit. The tachometer can measure rotation rates in an interval of [2 99999] revolutions per minute [3]. Different ESC commands were sent to one of the motor controllers and the resulting rotation rate in stationarity was observed using the tachometer and converted to radians per second. The result can be found in Figure 4.1.

From Figure 4.1 it can be seen that the relationship between the angular velocity of the rotor and ESC command is approximately linear after about 200 rad/s. The behavior for lower rotation rates (the nonlinear part) is not of interest, since the rotor speeds will vary in an interval around the operating point given by the rotation speed in hover. An estimate of this value was calculated using the model described in Section 4.2. For this, the estimated values of the thrust and torque constants (see Section 4.6 and 4.7) were used. By setting the force generated from the motors equal to the negative gravitational force and the resulting torque to zero in (4.8) the rotation speed in hover is given by (4.18).

$$\omega_{\text{hover}}^2 = \boldsymbol{\Lambda}^{\dagger} \begin{bmatrix} \mathbf{F}_{\text{hover}} \\ \boldsymbol{\tau}_{\text{hover}} \end{bmatrix}. \tag{4.18}$$

Here $\boldsymbol{\Lambda}^{\dagger}$ represents the pseduo inverse of the matrix $\boldsymbol{\Lambda}$. By taking the square root of the result from (4.18), the theoretical rotation rates in hover were found

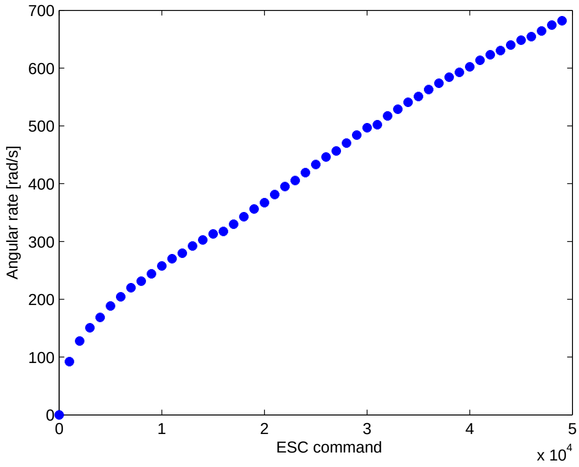


Figure 4.1: *Rotation rate of a rotor plotted against the corresponding ESC commands.*

to be close to 400 rad/s. In addition, results obtained from flight experiments suggest that this value is close to the actual hover rate. Using this information, a first order polynomial function was fitted to the data corresponding to the approximately linear interval including the operating point. This procedure was repeated for two different motors and the result of each experiment is shown by Figure 4.2 and 4.3 respectively.

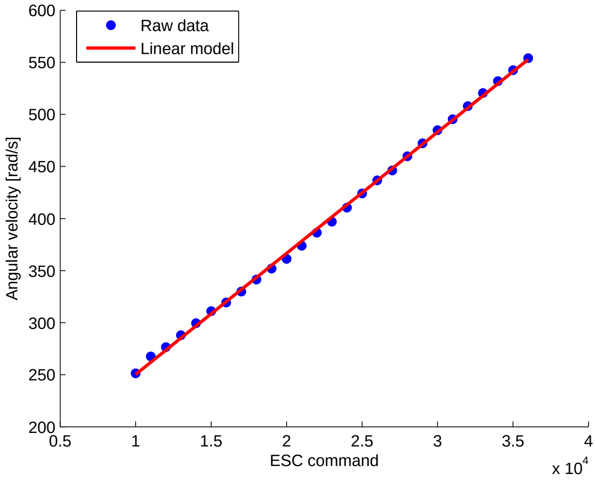


Figure 4.2: *The rotation rate of rotor 1 for different ESC commands (blue dots) and the corresponding linear model (red line).*

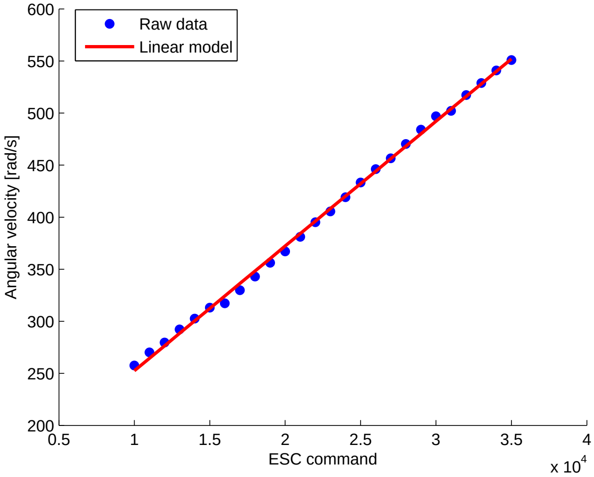


Figure 4.3: *The rotation rate of rotor 8 for different ESC commands (blue dots) and the corresponding linear model (red line).*

The equations for the linear models in Figure 4.2 and 4.3 is

$$\omega_{\text{rot}}^{1} = 0.012u_m + 134.0 \text{ [rad/s]}, \tag{4.19}$$

$$\omega_{\text{rot}}^{8} = 0.012u_m + 132.7 \text{ [rad/s]}. \tag{4.20}$$

Here, ω_{rot}^{i} is the angular velocity of rotor i and u_m is the ESC command sent to the motor controller. The slope for both linear models are equal, only the offset value changed slightly. One explanation could be that the battery voltage differed between the different experiments. Hence, the mean value of the offset was used in the final model. The relationship between given ESC command and the resulting rotation speed in stationarity is given by

$$\omega_{\text{rot}} = 0.012u_m + 133.4 \text{ [rad/s]}. \tag{4.21}$$

4.4.2 Time constant

The rotation rate of the rotors needed to be measured and logged to be able to identify the dynamic behavior of the motors. The digital tachometer described in Section 4.4.1 does not have a logging feature and hence it can only measure the static value of the rotation speed. It was hence necessary to develop another approach to make it possible to measure the dynamic behavior of the motors. Since the sound of the spinning rotors increases in pitch when the rotation speed is increased, it seemed reasonable that information about the motors' dynamic behavior could be found in a recording of the sound produced by the rotors. The sound from a spinning rotor was recorded using a microphone, and the data was then imported into Matlab for signal processing and data extraction. The goal was to obtain a description of how the rotation speed of the rotors change with time when a new rotation speed command is given. This can be done by analyzing the frequency content of the recorded signal using TFD-techniques described in Section 2.5.

The sound from the rotors were recorded using a microphone with a sample frequency of 192 kHz. The sample rate was chosen as high as possible to mitigate leakage and aliasing effects. The microphone signal was recorded using available Matlab functions. First, a recording of the rotors was done when the ESC command sent to the motors corresponded to the hovering rotation rate. The commanded rotation speed was kept constant throughout the recording. A DFT of the recorded signal was calculated and plotted to make sure that clear frequency peaks could be found in the signal. The resulting plot can be found in Figure 4.4.

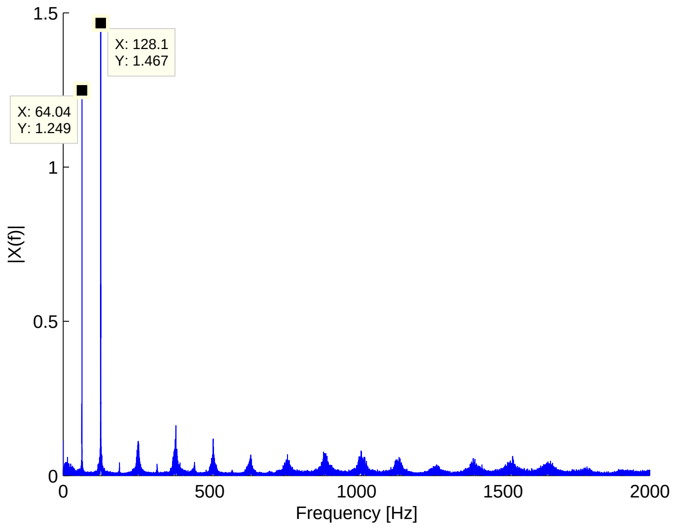


***Figure 4.4:** The absolute value of the DFT plotted against frequency for a measurement when the angular rate of the motor was kept constant.*

From Figure 4.4 it can be seen that the signal contains two clear frequency peaks, one at 64 Hz and one at 128 Hz, which corresponds to 402 rad/s and 804 rad/s respectively. The first peak comes from the rotation of the motor housing and corresponds to the true rotation rate. The second peak is due to the fact that the propellers have two blades and hence a sound corresponding to the double rotation rate is found in the signal. To obtain a time-frequency description of the signal, computations were performed according to the description in Section 2.5. A plot of the TFD can be found in Figure 4.5. Again, two clear frequencies can be found at 64 and 128 Hz during the whole measurement period.

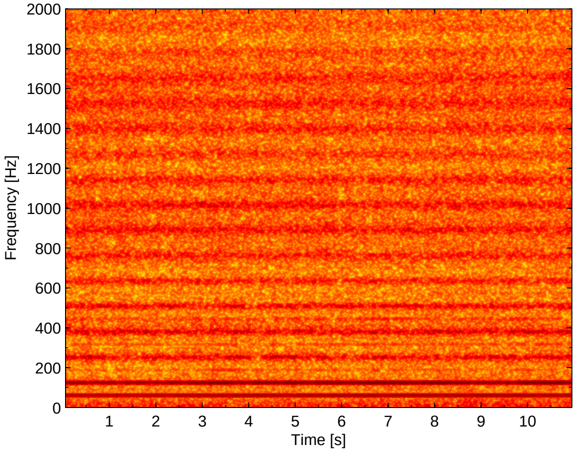


Figure 4.5: *The frequency content plotted against time of a recording with constant rotation rate of the rotor. A more reddish colour corresponds to a higher energy content for the particular frequency.*

To be able to calculate the time constant for the motors, a step response for the angular rates was recorded. The step command was sent to the motor controller after approximately six seconds. The resulting TFD can be found in Figure 4.6.

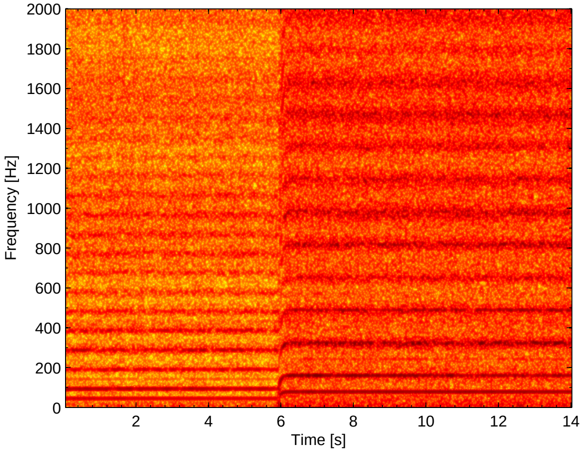


Figure 4.6: *The frequency content plotted against time of a recording with a positive step in desired rotation rate of the rotor at approximately 6 seconds. A more reddish colour corresponds to a higher energy content for the particular frequency.*

A clear change in frequency content can be seen after about six seconds. The exact time of the step application was not possible to identify. This is due to that the recording was performed using Matlab and the step command was sent over a Bluetooth connection to the platform using another software. This means that the time delay corresponding to the communication delay between the main CPU and the ESC:s cannot be identified from this experiment. However, this delay is low compared to the estimated time constant for the motor and is therefore neglected in the simplified model (4.17), see Section 4.3. To get a more detailed view of the step response, the result was zoomed in. The zoomed version of Figure 4.6 can be found in Figure 4.7.

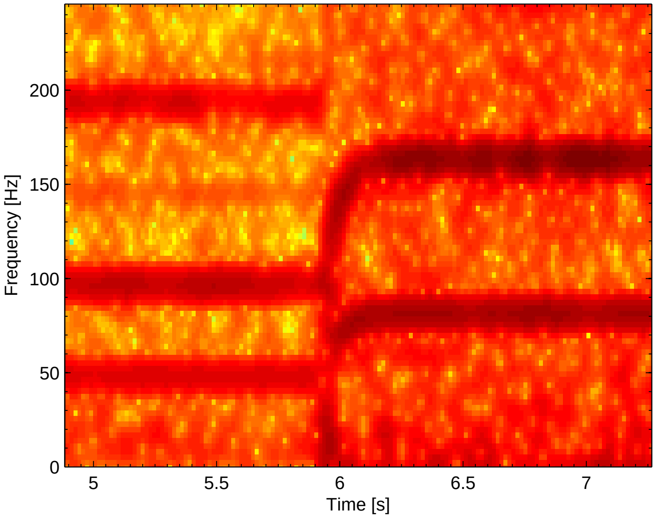


Figure 4.7: *The frequency content plotted against time of a recording with a positive step in desired rotation rate of the rotor. A more reddish colour corresponds to a higher energy content for the particular frequency.*

The step response in Figure 4.7 has the appearance of a typical first order step response. In order to find a matching time constant to the recorded step response, several first order step responses corresponding to systems with differing time constants, were simulated in Matlab and fitted manually to the recorded step response. The frequency corresponding to the sound of the rotor blades was used when fitting since it is more easily distinguished in the TFD:s. Since the rotors are attached to the motor housing it is equivalent to study the dynamic behavior of the rotor blades instead of the motor housing. The time constant that gave the best step response match was selected. Figure 4.8 shows the result for one recording.

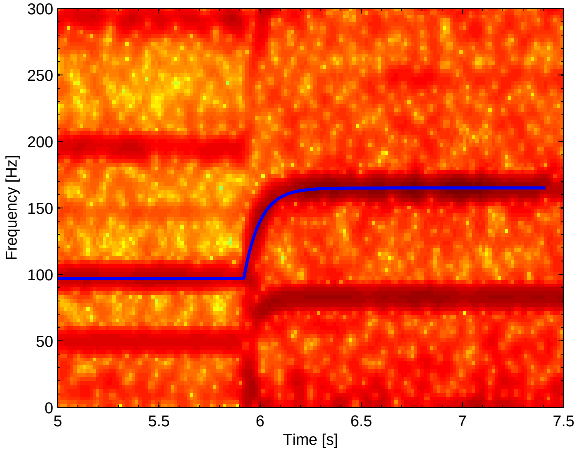


Figure 4.8: *The frequency content of a recorded step response plotted together with a simulated step response from a first order linear system with time constant 0.1 [s] (blue line). A more reddish colour corresponds to a higher energy content for the particular frequency.*

Next, a negative step response was recorded to find out if the motors behave in the same way for negative changes in the commanded rotor speed. The result from this measurement can be found in Figure 4.9.

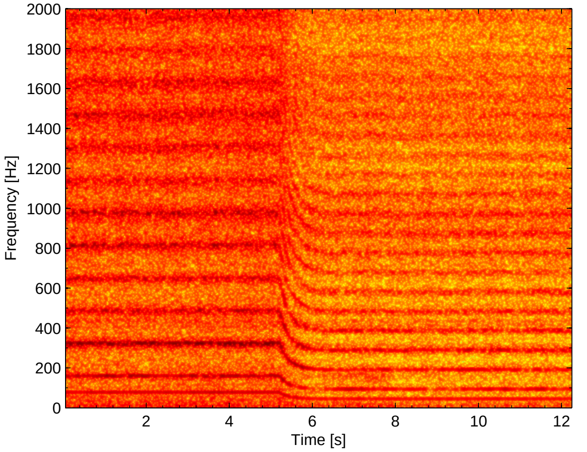


Figure 4.9: *The frequency content plotted against time of a recording with a negative step in desired rotation rate of the rotor. A more reddish colour corresponds to a higher energy content for the particular frequency.*

Here it can be seen that it takes longer time for the rotation rate to reach its final value compared to when the change in rotation rate was positive. The same results are obtained in [5] and are due to that the control algorithm used in the ESC:s does not actively apply a break to the rotors. Instead, the control algorithm simply waits for the rotor speeds to slow down. To find the time constant for the negative step response the same procedure used for the positive steps was repeated. The result can be found in Figure 4.10.

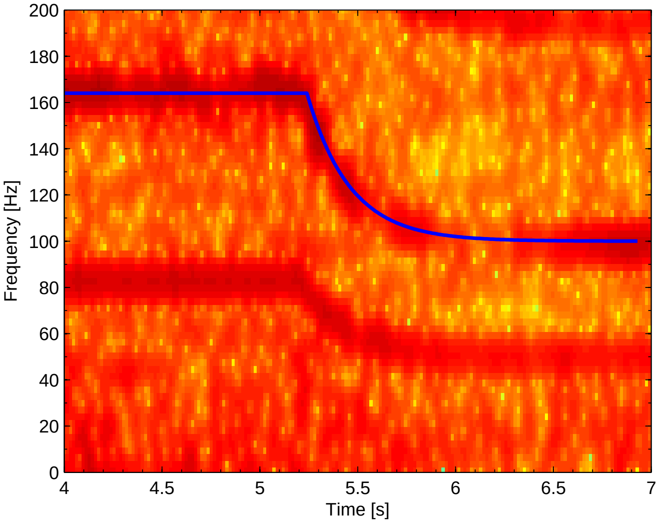


Figure 4.10: *The frequency content of a recorded step response plotted together with a simulated step response from a first order linear system with time constant 0.22 [s] (blue line). A more reddish colour corresponds to a higher energy content for the particular frequency.*

Additional positive and negative step responses were measured and the time constant for every step was found as previosly described. Table 4.1 contains the obtained results. From Table 4.1 it can be seen that the time constant for positive steps was the same independent of the step size. However, for negative steps the time constant increased when the size of the step increased. The step sizes in Table 4.1 are much larger than what the designed controllers in Section 6.3 usually use, most of the time small changes in rotation rate is commanded. Smaller step changes than the ones in Table 4.1 could not be used since the frequencies in the TFD were not distinguishable.

Table 4.1: *Calculated time constants in* (4.17) *for some different positive and negative step responses.*

Start Frequency [Hz]	Final Frequency [Hz]	$\Delta\omega$ [rad/s]	Time Constant [s]
55	73	113.1	0.10
48	80	201.1	0.10
32	96	405.3	0.097
65	60	-34.56	0.20
73	55	-113.1	0.21
82	49	-204.2	0.22
96	32	-405.3	0.3

The dynamics of the rotors were modeled as two different first order linear systems with different time constants but the same static gain. Which one of these two that is used depends on the sign of the derivative of the rotor angular velocities. If the rotor is increasing its rotation rate, a time constant of 0.10 seconds is used, T_{pos}. If the rotor is decreasing its rotation rate, a time constant of 0.20 is used, T_{neg}, corresponding to the smallest negative step size. Equation (4.17) can be used to express the differential equation that relates the ESC command to the output angular rate

$$\dot{\omega} = \frac{1}{T_m}(K_m u_m - \omega). \tag{4.22}$$

Equation (4.22) shows that it is the difference between $K_m u_m$ and ω that determines the sign of the derivative. This expression can therefore be used to select the correct value of the time constant, T_m. Figure 4.11 illustrates the model of the motor dynamics.

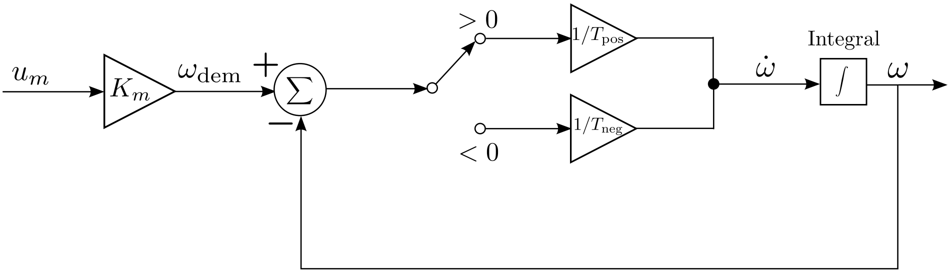


Figure 4.11: *Schematic illustration of how the motor dynamics was modeled, with two different time constants. Which one that is used depends on the sign of the derivative of the rotation rate.*

4.5 Estimating the moment of inertia matrix

To get an estimate of the moment of inertia matrix, both masses and spatial distributions of the components were measured. The masses were obtained by weighing the different components with a laboratory digital balance scale. The results can be found in Table 4.2.

Table 4.2: *Component masses and quantity*

Component	Quantity	Mass [kg/item]
Motor	8	0.110
Propeller	8	0.013
Battery holder	1	0.118
Battery	1	0.707
Avionics	1	0.053
Support	4	0.028
Long arm	4	0.070
Short arm	4	0.055
Central frame	1	0.216
Cover	1	0.037

The spatial distributions were obtained by measuring the components with a measuring tape. The results are listed in Table 4.3.

Table 4.3: *The spatial distribution of the components*

Component	Height [mm]	Width [mm]	Depth [mm]	Radius [mm]
Long arm	10	10	460	-
Short arm	10	10	360	-
Battery holder	106	145	54	-
Avionics	2	55	55	-
Support	10	10	300	-
Central frame	23	-	-	90

To simplify the calculations when estimating the moment of inertia matrix, the following simplifications were made:

- The eight motors together with their propellers were modeled as point masses.
- The central plate was seen as an infinitely thin circular plate with its radius according to Table 4.3.
- The octorotor's arms and supports were modeled as infinitely thin rods with depth according to Table 4.3.
- The battery and battery holder were modeled as a solid cuboid with spatial distribution according to Table 4.3.

- The avionics was considered as an infinitely thin quadratic plate with side-length according to Table 4.3.
- The cover was modeled as a spherical shell with its radius equivalent to the central plate.

All the components were then divided into a set of point masses, according to the specification above. The total mass of every component described in Table 4.2 was equally divided between its corresponding point masses, hence the sum of all point masses M is equal to the total mass of the octocopter

$$M \triangleq \sum_{i=1}^{n} m_i. \tag{4.23}$$

A plot of the octocopter modeled with point masses can be found in Figure 4.12.

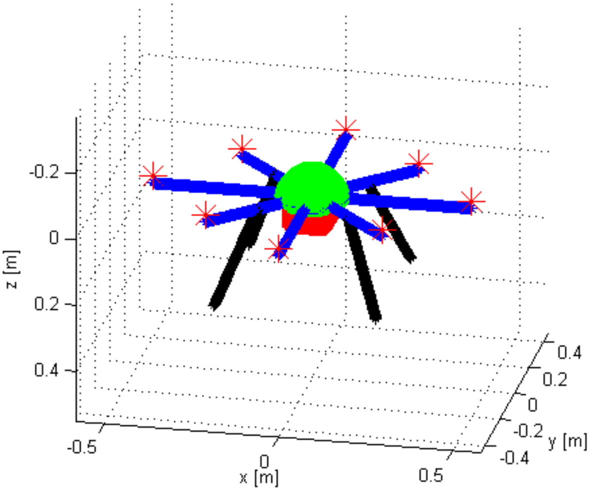


Figure 4.12: *The octorotor modeled as a set of point masses. The blue points represent the arms, the red points represent the battery and its holder, the green points represent the cover and the black points represent the supports. The red stars represent the motors.*

Since the center of mass for the octocopter was initially unknown, the center of the center plate was chosen as reference point R when measuring distances to the different components. To get an estimate of where the center of mass appears on the aircraft relative to the chosen reference point, the following formula was used

$$\mathbf{r}_G = \frac{1}{M} \sum_{i=1}^{n} m_i \mathbf{r}_i. \tag{4.24}$$

This equation is the point mass equivalent of (2.34). Here, $\mathbf{r}_i$ is the distance from R to the point mass i and n is the total amount of mass points.

The inertia matrix for a system of concentrated masses measured relative to a reference point R is given by

$$\mathbf{I}_{\mathrm{R}} = \sum_{i=1}^{n} m_i \left(r_i^2 \mathbf{I}_{3\times 3} - \mathbf{r}_i \mathbf{r}_i^T \right) = -\sum_{i=1}^{n} m_i \left((\mathbf{r}_i)_\times \right)^2 . \tag{4.25}$$

Here, $(\mathbf{r}_i)_\times$ denotes the skew-symmetric matrix, which means that $(\mathbf{r}_i)_\times \boldsymbol{\nu} = \mathbf{r}_i \times \boldsymbol{\nu}$ for the vector cross product and any vector $\boldsymbol{\nu} \in \mathbb{R}^3$ [8].

To obtain the inertia matrix relative to the center of mass, the parallel axis theorem was used

$$\mathbf{I}_G = \mathbf{I}_R - M \left(r_G^2 \mathbf{I}_{3\times 3} - \mathbf{r}_G \mathbf{r}_G^T \right) = \mathbf{I}_R + M((\mathbf{r}_G)_\times)^2. \tag{4.26}$$

Here, $\mathbf{r}_G$ and $\mathbf{I}_R$ are given by (4.24) and (4.25) respectively [8].

The number of mass points was increased until no difference in the four most significant figures of the resulting inertia matrix could be observed. The final estimate of the moment of inertia matrix is

$$\mathbf{I}_G = \begin{bmatrix} 0.1090 & 0 & 0 \\ 0 & 0.1083 & 0 \\ 0 & 0 & 0.2079 \end{bmatrix}. \tag{4.27}$$

As seen in (4.27), the resulting matrix is diagonal. This is because the octorotor's mass distribution is symmetric with respect to the body-fixed coordinate system.

4.6 Estimating the thrust constant

The thrust constant c_T defines the relation between the steady state thrust generated by the motors when hovering and the angular velocity of the rotors. To find this constant, one motor was soldered off from the motor control board and unmounted from its arm. Then it was soldered back on to be able to send control signals to it. The motor was then mounted on a rig consisting of four metal rods retained on a quadratic metallic plate. This was done to minimize the influence of airflow from the rotor on the measurement equipment and to make the test enviroment more similar to a hovering position. The metallic plate was in turn retained by a vise and then placed on a digital balance scale. The rig was sufficiently stiff to ensure that the motor kept its position when the rotation rate of the motor increased. Figure 4.13 shows a picture of the test rig.

When the rig was first tested, it was noticed that the airflow from the propeller affected the measured weight more than what was first expected. As a remedy, a cardboard box was placed between the motor and the digital balance scale to completely lead away the airflow. This eliminated the error caused by the airflow pushing down on the balance scale.

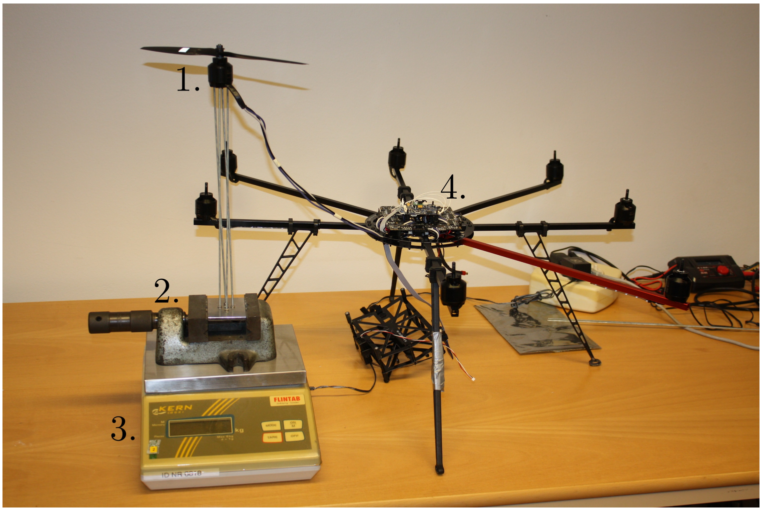


Figure 4.13: *The test rig used to estimate the thrust constant including a motor unmounted from the platform (1), a vise (2), a digital balance scale (3) and the platform (4).*

The measured weight of the total rig when the motor was switched off, m_{ref}, was used as a reference value. To calculate the steady state thrust, F_m, produced by the motor the following formula was used

$$F_m = (m_{\text{ref}} - m_{\text{meas}})g. \tag{4.28}$$

Here, m_{meas} is the measured weight for a specific angular velocity of the rotor and g is the gravitational constant. This will be equal to the resulting force since $(m_{\text{ref}} - m_{\text{meas}})$ will be the weight that the motor is lifting, and the motor's thrust was assumed to be pointing in the opposite direction of the gravitational force.

The angular velocity and corresponding measured weight was observed for some different ESC commands, using the tachometer described in Section 4.4.1. The measured thrust was calculated according to (4.28) for each measurement. Since it was known from Chapter 4 that the thrust is proportional to the square of the angular velocity, a first orded polynomial function with the constant term fixed to zero was fitted using the aquired data. This linear model together with raw data for a 12-inch plastic propeller can be found in Figure 4.14. The slope of the estimated linear model was used as the estimate of the thrust constant. The result is

$$c_T = 2.2 \cdot 10^{-5} \text{ [Ns}^2\text{]}. \tag{4.29}$$

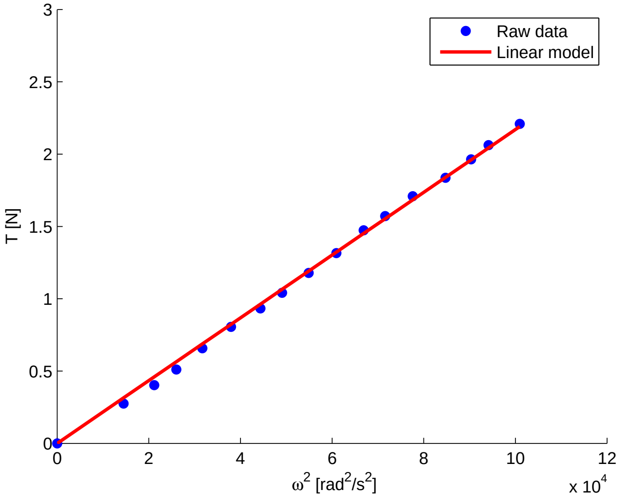


Figure 4.14: *Thrust plotted against the square of the angular velocity of the rotor (blue dots) together with a linear model (red line) when using a 12-inch plastic propeller and the rig from Figure 4.13.*

4.7 Estimating the torque constant

The torque constant c_Q relates the angular velocity of a rotor in hover to the torque produced by the rotor with respect to the motor axis. To find an estimate of this constant, the motor that was unmounted from the platform was used. This motor was attached to a rig consisting of four metallic rods retained on a metallic plate. At one end of the plate a ball bearing and a screw was attached. This enabled the metallic plate to move freely around the axis given by the attached screw. The screw was then fixed in a vise. As a first attempt the motor was placed horizontally which enabled the metallic plate to move freely in the vertical plane around the screw. The edge of the plate was put on the digital scale used in Section 4.5. The plate and the digital scale was placed in level so that the resulting force was directed straight down on the digital scale. A picture of the rig is shown in Figure 4.15.

When the rotor's angular velocity increases in this case, a resulting torque will try to rotate the plate. This will lead to an increased downward force on the digital scale and therefore also an increased mass reading. The initial mass, m_{ref}, obtained when the motor was turned off was used as reference value. The resulting force was calculated as

$$F_\tau = (m_{\text{meas}} - m_{\text{ref}})g. \tag{4.30}$$

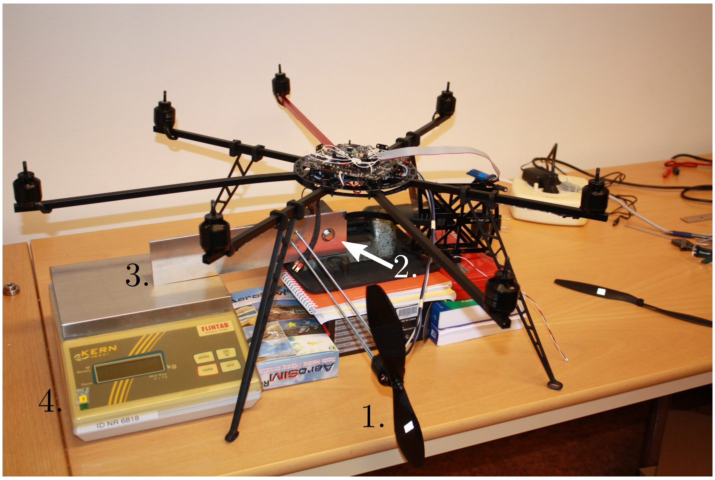


Figure 4.15: *The test rig used in the first attempt to estimate the torque constant. The motor is placed horizontally (1), attached to a metallic plate (3) which can rotate around a ball bearing (2). The metallic plate is placed on a digital scale (4).*

To calculate the resulting torque, the distance between the center of the screw and the edge of the plate, R_{lever}, was measured. Since the resulting force in this case is perpendicular to the lever, the resulting torque could be calculated as

$$\tau = R_{\text{lever}} F_{\tau}. \tag{4.31}$$

Some different angular velocities for the rotor was commanded and measured using the digital tachometer as in Section 4.4.1. The resulting torque was calculated according to (4.30) and (4.31). A data set using a plastic 12-inch propeller was collected and a first order polynomial function with the constant term fixed to zero was fitted to the measured data. The result from the experiment can be seen in Figure 4.16

The slope of this linear model, c_Q^v is the resulting estimate of the torque constant from this experiment

$$c_Q^v = 4.25 \cdot 10^{-7} \text{ [Nms}^2\text{]}. \tag{4.32}$$

When these measurements were done, it was noticed that the measured mass on the digital scale was sensitive to the position of the plate's edge. During the test, the placement changed a bit due to forces and air disturbances from the motor. This is a probable explanation for the relatively bad fit in Figure 4.16. Hence, the rig was modified such that the motor was instead placed vertically and the plate placed horizontally. Now, the plate could move freely in the horizontal plane instead. A picture of the rig can be seen in Figure 4.17.

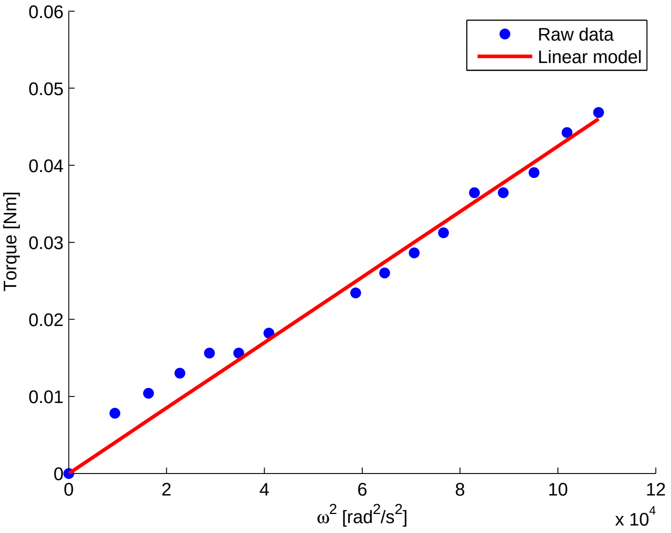


Figure 4.16: *Torque plotted against the square of the angular velocity of the rotor (blue dots) together with a linear model (red line) when using a 12-inch plastic propeller and the rig from Figure 4.15.*

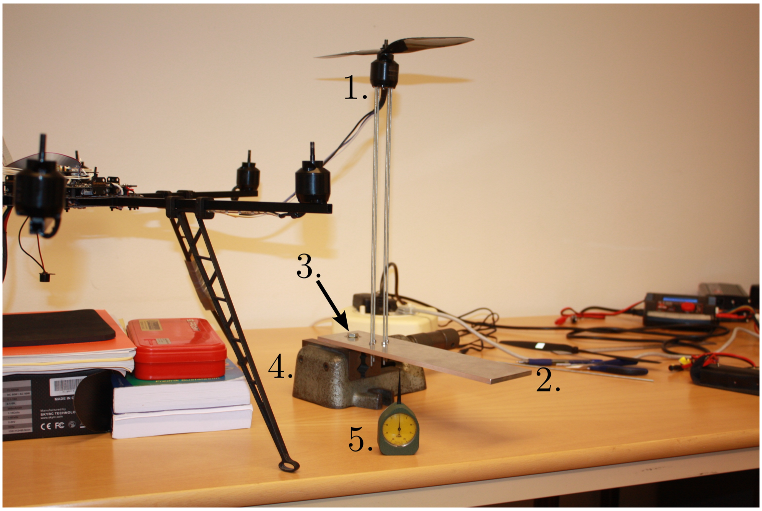


Figure 4.17: *A test rig for estimating the torque constant. The motor is placed vertically (1), attached to a metallic plate (2) that can move freely in the horizontal plane around a ball bearing (3). A screw was fitted to the ball bearing and fixed in a vise (4). An analog dynamometer was used to measure the force (5).*

For this experiment an analog dynamometer, which is seen in Figure 4.17, was used to measure the resulting perpendicular force. The dynamometer presents a value m_τ in grams between [0 0.02] kg, which can be converted to a force according to

$$F_\tau = m_\tau g. \tag{4.33}$$

Different rotor angular velocities were commanded and the resulting force was calculated using (4.33). The torque was calculated according to (4.31). A data set using the plastic 12-inch propeller was collected and a first order polynomial function with the constant term fixed to zero was fitted to the data. The result of the experiment can be found in Figure 4.18.

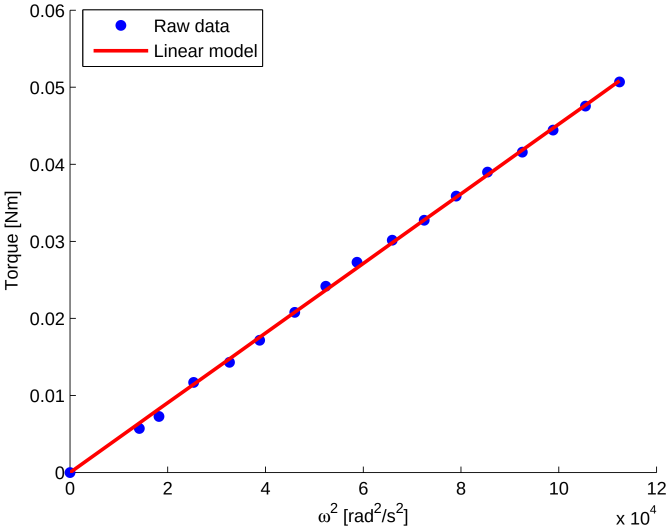


Figure 4.18: *Torque plotted against the square of the angular velocity of the rotor (blue dots) together with a linear model (red line) when using a 12-inch plastic propeller and the rig from Figure 4.17.*

The slope of this linear model, c_Q^h is the resulting estimate of the torque constant from this experiment

$$c_Q^h = 4.52 \cdot 10^{-7} \ [\mathrm{Nms}^2]. \tag{4.34}$$

In this experiment, the resulting fit was much better which indicates that the disturbances did not affect the result as much as compared to the first experiment. Hence, the estimate from the second experiment was selected as the torque constant for the 12-inch plastic propeller.

$$c_Q = 4.52 \cdot 10^{-7} \ [\mathrm{Nms}^2]. \tag{4.35}$$

5
State estimation

In order to control the attitude and position of the octorotor, these variables need to be estimated. This can be done using the available sensor measurements delivered by the platform's sensors and running a state estimation algorithm. The different sensors on the platform measure different physical properties and each sensor has its own special characteristics that must be taken into consideration. Section 5.1 provides a detailed explanation of the sensors on a multirotor that are typically used for attitude estimation and how this is done.

The process of combining sensor data from multiple sources in order to get a better information than what would be possible by only using the sensors separately is referred to as sensor fusion [10]. There are a lot of different algorithms that can be used to achieve this. Section 5.2 describe the filter algorithm that was implemented on the platform in detail. This section also contains plots illustrating the filter performance.

5.1 Attitude estimating sensors

For a multirotor, common sensors used for attitude estimation are the accelerometer, gyroscope and magnetometer [21]. The accelerometer measures the external accelerations of the platform as well as the gravitational acceleration according to (3.1). The gyroscope measures angular velocities in the body-fixed frame according to (3.2). The magnetometer measures the earth's magnetic field mapped to the body-fixed frame as explained by (3.3). All three sensors can be used to compute measurements of the pitch, roll and yaw angle using different techniques.

5.1.1 Accelerometer

According to (4.14) the octorotor is subjected to three forces: the force of gravity, thrust forces from the rotors and aerodynamic forces. Since the sum of all forces is equal to mass times acceleration according to Newton's second law, then the following relation must hold for the octorotor

$$m\mathbf{a}_{\text{ext}}^{\text{B}} = \mathbf{F}_g^{\text{B}} + \mathbf{F}_{\text{rot}}^{\text{B}} + \mathbf{F}_{\text{aero}}^{\text{B}}. \tag{5.1}$$

If this equation is combined with the measurement equation for the accelerometer (3.1) and neglecting noise and bias, the following result is obtained

$$\mathbf{a}_{\text{acc}}^{\text{B}} = \frac{1}{m}\left(\mathbf{F}_{\text{rot}}^{\text{B}} + \mathbf{F}_{\text{aero}}^{\text{B}}\right). \tag{5.2}$$

In a hovering state in an undisturbed airspace, the aerodynamic forces are zero and the thrust force from the rotors counteract the force of gravity. This means that the accelerometer gives a measure of the negative gravitational vector expressed in the body-fixed frame. If the platform is tilted in the air the thrust from the rotors will not only compensate for the gravitational pull but also generate a linear acceleration with components in the X^{B}- and Y^{B}-directions. This linear acceleration is also affected by aerodynamic forces referred to as rotor flapping and induced drag [23]. In stationarity, these effects will counteract the linear accelerations and hence the low frequency content of the accelerometer signal can be used to provide an estimate of the negative gravitational vector expressed in the body-fixed frame (see [21])

$$\mathbf{a}_{\text{acc}}^{\text{B}} \approx -\mathbf{a}_g^{\text{B}}. \tag{5.3}$$

Since it is known that the gravitational acceleration vector always has the coordinates $\begin{bmatrix}0 & 0 & g\end{bmatrix}^T$ in the earth-fixed frame, the roll and pitch angle can be estimated from the accelerometer measurements using basic trigonometry. This approach is used in various multirotor projects as well as in commercial attitude sensors [23], giving

$$\mathbf{a}_{\text{acc}}^{\text{B}} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \approx -\mathbf{R}_{\text{E}\to\text{B}}\mathbf{a}_g^{\text{E}} = \begin{bmatrix} g\sin\theta \\ -g\sin\phi\cos\theta \\ -g\cos\phi\cos\theta \end{bmatrix}. \tag{5.4}$$

According to (5.4) it is possible to estimate the roll angle and pitch angle by using arctangent and arcsine,

$$\hat{\phi}_{\text{acc}} = \arctan\frac{a_y}{a_z}, \tag{5.5}$$

$$\hat{\theta}_{\text{acc}} = \arcsin\frac{a_x}{g}. \tag{5.6}$$

The yaw angle, ψ, is however not possible to estimate using the accelerometer since it is not included in (5.4).

5.1.2 Gyroscope

The relationship between the angular velocities in the earth-fixed frame and the angular velocities in the body-fixed frame are given by (2.22). If rotation is only performed around one axis in the body-fixed frame at a time, i.e. all the other Euler angles are zero, then the rotation measured in the body-fixed frame is equal to the current Euler angle rate. Hence the gyroscope can be used as a device that measures the time derivative of the roll, pitch and yaw angle. In this case, it is possible to get an estimate of the attitude by integrating the gyro measurements

$$\hat{\phi}_{\text{gyro}} = \int \dot{\phi}_{\text{gyro}} \, \mathrm{dt}, \tag{5.7}$$

$$\hat{\theta}_{\text{gyro}} = \int \dot{\theta}_{\text{gyro}} \, \mathrm{dt}, \tag{5.8}$$

$$\hat{\psi}_{\text{gyro}} = \int \dot{\psi}_{\text{gyro}} \, \mathrm{dt}. \tag{5.9}$$

This method of estimating the attitude is problematic in a similar way to the accelerometer estimate. According to (3.2), the gyroscope is affected by a slowly varying bias term. This means that the gyroscope measurement will be non-zero even after the rotation has stopped. The bias is included in the integral which will drift away from the correct angle estimate. Hence, the gyroscope estimate is accurate in the short but not in the long term.

5.1.3 Magnetometer

The magnetometer is similar to the accelerometer in that it provides a vectorial measurement that can be compared to a known vector, in this case the earth's magnetic field. The magnetic field of the earth expressed in the earth-fixed frame, here denoted $\mathbf{m}^{\mathrm{E}}$, has no component in the east-west direction. Hence, the y component of $\mathbf{m}^{\mathrm{E}}$ is always zero. This can be used to provide an estimate of the local magnetic field before takeoff. If the magnetometer on the octorotor is placed horizontally and no magnetic disturbances are present, the relationship between the earth-frame representation of the local magnetic field and the magnetometer measurement is

$$\mathbf{m}^{\mathrm{E}} = \begin{bmatrix} m_x^{\mathrm{E}} \\ m_y^{\mathrm{E}} \\ m_z^{\mathrm{E}} \end{bmatrix} = \begin{bmatrix} \sqrt{m_x^2 + m_y^2} \\ 0 \\ m_z \end{bmatrix}. \tag{5.10}$$

A calibration of the magnetometer can be performed by collecting a number of samples when the octorotor is horizontal prior to takeoff. The local magnetic field vector can be estimated from these samples and then used for attitude estimation.

According to (3.3), in a disturbance free environment the magnetometer measures

$$\mathbf{m}_{\text{mag}}^{\mathrm{B}} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \mathbf{R}_{\mathrm{E}\to\mathrm{B}} \mathbf{m}^{\mathrm{E}} = \begin{bmatrix} m_x^{\mathrm{E}} c\theta c\psi - m_z^{\mathrm{E}} s\theta \\ m_x^{\mathrm{E}}(-c\phi s\psi + s\phi s\theta c\psi) + m_z^{\mathrm{E}} s\phi c\theta \\ m_x^{\mathrm{E}}(s\phi s\psi + c\phi s\theta c\psi) + m_z^{\mathrm{E}} c\phi c\theta \end{bmatrix}. \tag{5.11}$$

If the octorotor is placed exactly horizontal, i.e. if ϕ and θ are zero, then (5.11) simplifies to

$$\mathbf{m}^{\mathrm{B}}_{\mathrm{mag}} = \begin{bmatrix} m^{\mathrm{E}}_x \cos\psi \\ -m^{\mathrm{E}}_x \sin\psi \\ m^{\mathrm{E}}_z \end{bmatrix}, \tag{5.12}$$

and the magnetometer works just like a compass. An absolute yaw angle (heading) estimate is provided by computing

$$\hat{\psi}_{\mathrm{mag}} = -\arctan\frac{m_y}{m_x}. \tag{5.13}$$

5.2 Estimation algorithm

This section describes the type of attitude estimation algorithm that was implemented on the octorotor platform in more detail and also why this algorithm was chosen.

There are three types of attitude estimation algorithms that are used by larger scale open source multirotor projects. The algorithms are either implemented as EKF:s, linear or nonlinear complementary filters. The EKF is the most computationally intensive algorithm. The nonlinear complementary filter is less intensive but the theory behind the algorithm is relatively complex. The linear complementary filter is the simplest algorithm which is easy to implement and understand [19]. The linear complementary filter is described in Section 2.6.

It was decided that a nonlinear complementary filter algorithm should be designed and implemented on the platform. This type of filtering algorithm is thoroughly tested in [20] and proven to give good results. The same type of algorithm is also recommended in [21] where it is considered to be the preferred candidate for attitude estimation for multirotors compared to other filters, such as e.g. the EKF.

5.2.1 Nonlinear complementary filter

The nonlinear complementary filter works similarly to a linear complementary filter (see Section 2.6) in the sense that it combines gyroscope measurements and measurements of known vectors provided by other types of sensors. The gyro measurements are used to update the current rotation matrix according to (2.33) and the estimate is then corrected using an innovation term formed from data collected from the other sensors. In addition, the filter algorithm estimates the gyro bias online and compensates for it.

The complete filter algorithm is given by the following equations

$$\dot{\hat{\mathbf{R}}} = \hat{\mathbf{R}}\left(\left(\boldsymbol{\omega}_{\mathrm{gyr}} - \hat{\mathbf{b}}\right)_{\times} + k_P \boldsymbol{\alpha}_{\times}\right), \quad \hat{\mathbf{R}}(0) = \hat{\mathbf{R}}_0, \tag{5.14a}$$

$$\dot{\hat{\mathbf{b}}} = -k_I \boldsymbol{\alpha}, \tag{5.14b}$$

$$\boldsymbol{\alpha} = \sum_{i=1}^{n} k_i \mathbf{v}_i \times \hat{\mathbf{v}}_i, \quad k_i > 0, \tag{5.14c}$$

$$\hat{\mathbf{v}}_i = \hat{\mathbf{R}}^T \mathbf{v}_i^0. \tag{5.14d}$$

Here, a subscripted × represents the skew-symmetric matrix formed from a three-dimensional vector as in (2.32). $\hat{\mathbf{R}}$ is the estimated rotation matrix $\hat{\mathbf{b}}$ is the estimated gyro bias and $\boldsymbol{\alpha}$ is the innovation term. The innovation term is calculated by transforming known vectorial directions $\mathbf{v}_i^0$ into the body-fixed frame using the estimated rotation matrix and comparing the result to the vectorial measurements $\mathbf{v}_i$ according to (5.14c) and (5.14d). If more than one sensor is used to compute the innovation term the sensor measurements are weighted with the parameter k_i. This parameter represents the relative confidence in the measurements [20].

Since the absolute yaw angle ψ is not explicitly controlled in this project the implemented filter algorithm did not make use of the magnetometer. It is stated in [20] that the filter algorithm performs well in the case that only one vectorial direction is measured. Hence, n is set to one in (5.14c) and $\mathbf{v}_1^0 = -\mathbf{a}_g^{\mathrm{E}}$. However, if ψ is of interest, the magnetometer measurement can easily be added to (5.14c) which will improve the filter estimate of this angle.

The filter is tuned by varying the parameters k_P and k_I. A higher value of the gain k_P makes the filter estimate rely more on the accelerometer estimate of the current attitude. This corresponds to less low-pass filtering of the accelerometer measurements. Conversely, a lower value of k_P suggests more confidence in the integrated gyro measurements and hence less high-pass filtering of these values. The value of k_P should be selected such that the filter estimate tracks the high frequency part of the integrated gyro measurement and follows the low frequency part of the accelerometer estimate. This can be interpreted as tuning the cut-off frequency for a standard linear complementary filter even though there is no clear frequency interpretation due to the nonlinearities. The second filter gain, k_I, is related to how fast the gyro bias is tracked. It is usually tuned to an order of magnitude lower than k_P to track the slowly varying bias and decouple it from the attitude estimate [21].

A problem with the filter algorithm in (5.14) is that it is based on estimating rotation matrices. As discussed in Section 2.7, rotation matrices are non-minimal representations of the attitude and it is complicated to keep them orthogonal. Hence, the filter algorithm was implemented using a quaternion representation instead. The resulting filter algorithm corresponding to (5.14) is given by

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \mathbf{p}\left(\omega_{\text{gyr}} - \hat{\mathbf{b}} + k_P \boldsymbol{\alpha}\right), \quad \hat{\mathbf{q}}(0) = \hat{\mathbf{q}}_0, \tag{5.15a}$$

$$\dot{\hat{\mathbf{b}}} = -k_I \boldsymbol{\alpha}, \tag{5.15b}$$

$$\boldsymbol{\alpha} = -\mathrm{vex}\left(\sum_{i=0}^{n} \frac{k_i}{2}\left(\mathbf{v}_i\hat{\mathbf{v}}_i^T - \hat{\mathbf{v}}_i\mathbf{v}_i^T\right)\right), \quad k_i > 0, \tag{5.15c}$$

$$\hat{\mathbf{v}}_i = \hat{\mathbf{R}}^T(\hat{\mathbf{q}})\mathbf{v}_i^0. \tag{5.15d}$$

Here, vex is the inverse operation of the subscript × operator [20].

The filter algorithm that is described in (5.15) was implemented on the platform. In order to do this the differential equations describing the filter were discretized using a first order Euler approximation. Algorithm 1 describes the implemented filter.

Algorithm 1 The nonlinear complementary filter algorithm that is implemented on the platform.

```
$k \leftarrow 0$
$\hat{\mathbf{q}}_0 \leftarrow \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$
$\hat{\mathbf{b}}_0 \leftarrow \mathbf{0}_{3\times1}$
loop
    $\mathbf{v}_k \leftarrow \mathbf{a}^{\mathrm{B}}_{\mathrm{acc}}$
    $\hat{\mathbf{v}}_k \leftarrow \hat{\mathbf{R}}^T(\hat{\mathbf{q}}_k)(-\mathbf{a}^{\mathrm{E}}_g)$
    $\boldsymbol{\alpha}_k \leftarrow -\mathrm{vex}\left(\mathbf{v}_k\hat{\mathbf{v}}_k^T - \hat{\mathbf{v}}_k\mathbf{v}_k^T\right)$
    $\hat{\mathbf{q}}_{k+1} \leftarrow \hat{\mathbf{q}}_k + T_s \cdot \frac{1}{2}\hat{\mathbf{q}}_k \otimes \mathbf{p}\left(\omega_{\mathrm{gyr}} - \hat{\mathbf{b}}_k + k_P\boldsymbol{\alpha}_k\right)$
    $\hat{\mathbf{q}}_{k+1} \leftarrow \frac{1}{\|\hat{\mathbf{q}}_{k+1}\|} \cdot \hat{\mathbf{q}}_{k+1}$
    $\hat{\mathbf{b}}_{k+1} \leftarrow \hat{\mathbf{b}}_k - T_s \cdot k_I\boldsymbol{\alpha}_k$
    Compute attitude estimates from the estimated quaternion $\hat{\mathbf{q}}_k$
    $k \leftarrow k + 1$
end loop
```

In each iteration the innovation term is first computed according to (5.15c) and (5.15d). The estimated quaternion is used to express the gravitational acceleration vector in coordinates relative to the body fixed frame. When doing this, it is possible to utilize the fact that the earth-fixed frame representation only has a component in the Z^{E}-axis. The computation of $\hat{\mathbf{v}}_k$ is hence performed as

$$\hat{\mathbf{v}}_k = -g \cdot \hat{\mathbf{R}}^T(\hat{\mathbf{q}}_k)\begin{bmatrix}0\\0\\1\end{bmatrix} = -g \cdot \begin{bmatrix}2(q_1q_3 - q_0q_2)\\2(q_2q_3 + q_0q_1)\\q_0^2 - q_1^2 - q_2^2 + q_3^2\end{bmatrix}. \tag{5.16}$$

The result is multiplied with the most recent accelerometer measurement and the innovation term is formed. Next, the quaternion estimate is updated using the innovation term together with the latest gyro measurements. The quaternion estimate is subsequently normalized and the next bias estimate is formed using the innovation term.

The final step in the filtering process is to provide an estimate of the current roll and pitch angle, $\hat{\phi}$ and $\hat{\theta}$, since this is what the attitude control algorithm uses to perform its task. The method of determining these estimates based on the current quaternion estimate is given by comparing (2.67) with (2.11)

$$\tan\phi = \frac{2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}, \tag{5.17}$$

$$\sin\theta = 2(q_0q_2 - q_1q_3). \tag{5.18}$$

According to the above equations the roll and pitch estimates can be calculated by computing

$$\hat{\phi} = \arctan\left(\frac{2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right), \tag{5.19}$$

$$\hat{\theta} = \arcsin\left(2(q_0q_2 - q_1q_3)\right). \tag{5.20}$$

5.2.2 Tuning and results

The initial filter tuning was performed by collecting raw sensor data from the accelerometer and gyroscope mounted on the octorotor and then simulating the filter algorithm in Matlab with the logged data as input. The parameters that gave the best results in simulations were initially chosen as the default parameters on the platform. These parameters were then tuned even more based on received log data from the implemented filter on the platform.

In order to tune the filter, the filter angle estimate for each axis was compared with the raw angle estimates from the accelerometer and gyroscope. The raw accelerometer estimate was calculated as in (5.5) and (5.6). The raw gyroscope measurement for each axis was computed by subtracting the estimated bias from the angular rates measured by the gyroscope and integrating the result

$$\hat{\phi}_{\text{acc}} = \arctan\frac{a_y}{a_z}, \tag{5.21}$$

$$\hat{\theta}_{\text{acc}} = \arcsin\frac{a_x}{g}, \tag{5.22}$$

$$\hat{\phi}_{\text{gyro}} = \int \omega_\phi - b_\phi \,\mathrm{dt}, \tag{5.23}$$

$$\hat{\theta}_{\text{gyro}} = \int \omega_\theta - b_\theta \,\mathrm{dt}. \tag{5.24}$$

These two estimates for each axis were transmitted via the telemetry link and stored on a file. This file was then later imported into Matlab and the results were analyzed. The filter performance for $k_P = 0.0104$, $k_I = 0.0031$ and $T_s = 0.02$ [s] is shown in Figure 5.1. The sampling time was set to the lowest possible value for data logging, which made it possible to tune the filter parameters in Matlab.

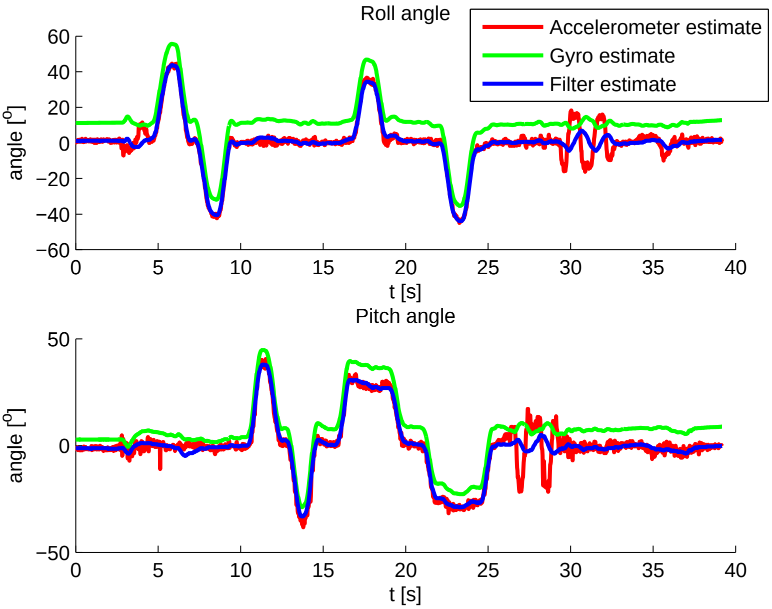


***Figure 5.1:** The computed filter estimate compared with raw sensor estimates*

The figure illustrates why the raw accelerometer and gyroscope estimates cannot be used separately. The accelerometer estimate is very noisy and also sensitive to external accelerations (occuring around 30 seconds), hence it should only be trusted during hover and slow maneuvering. The gyroscope estimate is reliable during quick movements but is suffering from drift due to the fact that the gyro bias is integrated. However, as seen in Figure 5.1, the raw gyro estimate offset does not decrease or increase during the test which indicates that the bias estimation is working. The constant offset is due to bias that is integrated when the filter bias estimate has not yet converged.

The octorotor is subjected to external accelerations in different directions at $t \approx 28$ s and at $t \approx 31$ s. These accelerations will be misinterpreted by the accelerometer estimate as a change in angular position. Due to the filtering of the accelerometer measurement in Algorithm 1, the resulting filter estimate is not affected as much. Figure 5.2 is a zoomed in version of Figure 5.1 which further illustrates how the filter estimate successfully tracks the high frequency part of the gyroscope estimate and the low frequency part of the accelerometer estimate.

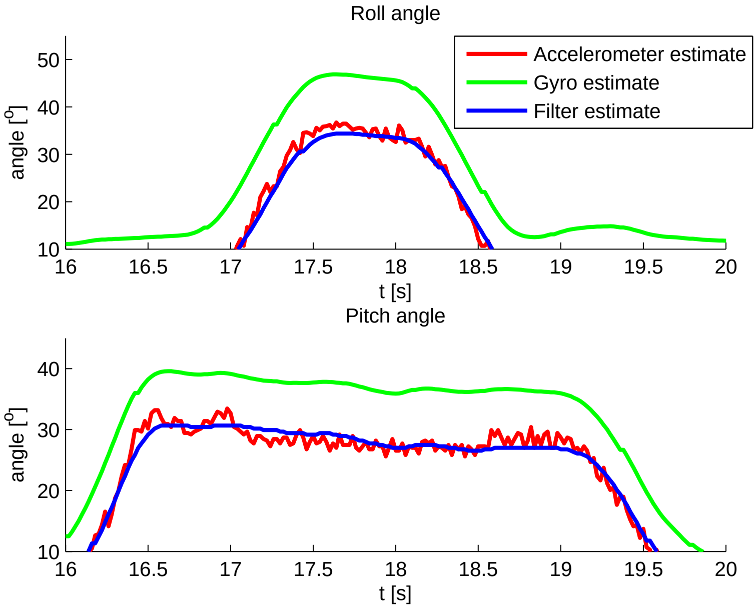


Figure 5.2: *Zoomed in version of Figure 5.1*

6 Attitude control

The main goal of this thesis is to design and evaluate different attitude control algorithms for an octorotor. The controllers are designed using a linearized model of the octorotor dynamics, and evaluated in a simulation environment based on the nonlinear octorotor model presented in Chapter 4. This chapter presents the proposed control schemes.

- Section 6.1 describes the hierarchical control structure that is commonly used in multirotor control systems.
- Section 6.2 contains a description of how the nonlinear model derived in Chapter 4 is simplified and linearized to provide a model used for the controller design.
- Section 6.3 describes how the two different attitude controllers are designed and tuned in this thesis.
- Section 6.4 includes a comparison of the designed controllers with respect to different performance measures.

6.1 Controlling the octorotor

In principle, a multirotor is controlled by varying the rotation speeds of the platform's rotors according to some strategy. However, a multirotor is an unstable system which needs a stabilizing closed-loop controller. It would not be possible for a human to fly the aircraft by manually controlling the rotation rates of the motors [19].

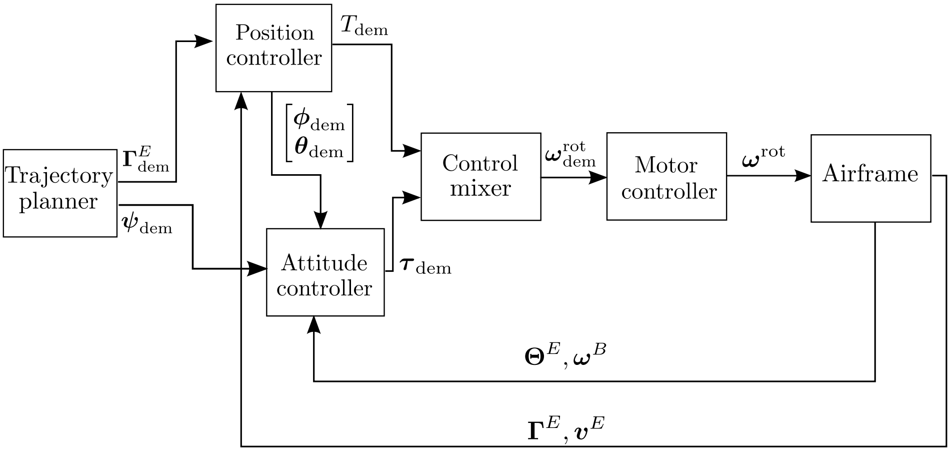


Figure 6.1: *An overview of the hierarchical controller structure often used in multirotor autopilots.*

Hierarchical control approaches are commonly used for multirotors, which means that the controller architecture is divided into different levels that form nested feedback loops [21]. The different levels in the hierarchical architecture are described in more detail in the following sections. A schematic overview of the control levels can be seen in Figure 6.1.

6.1.1 Motor controller

The innermost loop in the architecture is the motor controller, which controls the rotation rates of the motors. The motor controllers receive commands from the control mixer and tries to track the rotation rates as fast as possible. On the platform used in this thesis, the motor controllers are integrated into the BL-control boards described in Section 3.3. Since the source code to the software running on these control cards is closed, the motor controllers are modeled with a grey-box model described in Section 4.4.

6.1.2 Control mixer

The control mixer is used to translate desired force and torque commands from the outer control loops into rotation rate commands. Given the desired force and torque commands as input, the mixer distributes rotation rate commands to the motor controllers according to a specific strategy. A more detailed description of this block can be found in Section 6.2.2.

6.1.3 Attitude controller

The attitude controller regulates the attitude of the octorotor. Inputs to the attitude controller are the desired roll, pitch and yaw angle and the estimated attitude and angular rates provided by the attitude estimation algorithm. The output from this controller is the desired torque given with respect to the body-fixed frame. This command is translated into rotation rate commands by the control mixer and then fed to the motor controllers. The different attitude controllers that have been developed during this thesis work are described in Section 6.3.

6.1.4 Position controller

The position controller runs in the outermost control loop. The input to this controller is the desired linear position of the aircraft given with respect to the inertial frame. The position controller outputs desired roll and pitch angles together with a desired thrust. The desired roll and pitch angles will be fed to the attitude controller while the thrust command will be sent directly to the control mixer. A position controller was not implemented in this thesis work.

6.1.5 Trajectory planner

A trajectory planner can be used to create the desired position and attitude demands required to make the octorotor follow a trajectory in 3D-space. The desired position in the current trajectory will be sent to the position controller and the yaw angle demand is sent to the attitude controller. A trajectory planner was not implemented in this thesis work.

6.2 Model simplifications for controller design

The dynamic model of the octorotor used in this thesis is thoroughly described in Chapter 4. Since one of the main goals of this thesis is to design attitude control algorithms, it is the angular part of the kinematic and dynamic equations of motion that are of interest. These equations are given by

$$\dot{\boldsymbol{\Theta}} = \mathbf{T}_{\mathrm{B}\rightarrow\mathrm{E}}\boldsymbol{\omega}, \tag{6.1}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}_G^{-1}(\boldsymbol{\omega} \times \mathbf{I}_G\boldsymbol{\omega} + \boldsymbol{\tau}_{\mathrm{gyr}} + \boldsymbol{\tau}_{\mathrm{rot}} + \boldsymbol{\tau}_{\mathrm{aero}}). \tag{6.2}$$

Here, $\boldsymbol{\tau}_{\mathrm{rot}}$ represents the torque provided by the rotors. This model can be simplified and linearized to provide a linear model that can be used for control design. The first simplification is to neglect the gyroscopic and aerodynamic torques ($\boldsymbol{\tau}_{\mathrm{gyr}}$ and $\boldsymbol{\tau}_{\mathrm{aero}}$ in (6.2)) since these will typically be much smaller than the torque provided by the rotors [21]. The next step is to extract a linear model from the nonlinear model.

6.2.1 Linearized dynamic model

A general nonlinear system can be described by

$$\dot{x} = f(x, u), \tag{6.3}$$

$$y = h(x, u). \tag{6.4}$$

Here, x is the system's state vector, u is the input signal and y is the system's output. The functions $f(\cdot)$ and $h(\cdot)$ can be arbitrary nonlinear functions. In general, both the input and the output signals can be vector valued [28].

This nonlinear system can be linearized around an equilibrium point given by

$$f(x_0, u_0) = 0, \tag{6.5}$$

$$h(x_0, u_0) = y_0. \tag{6.6}$$

By introducing the new variables

$$z = x - x_0, \tag{6.7}$$

$$v = u - u_0, \tag{6.8}$$

$$w = y - y_0 \tag{6.9}$$

and performing a first-order Taylor-series expansion around the equilibrium point and neglecting higher order terms, one obtains the approximation

$$\dot{z} = f_x(x_0, u_0)z + f_u(x_0, u_0)v, \tag{6.10}$$

$$w = h_x(x_0, u_0)z + h_u(x_0, u_0)v. \tag{6.11}$$

Here, f_x, f_u, h_x and h_u are Jacobian matrices. The element (i, j) in these matrices denoted $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ and $d_{i,j}$ respectively are given by

$$a_{i,j} = \frac{\partial}{\partial x_j} f_i(x, u)|_{(x_0,u_0)}, \tag{6.12}$$

$$b_{i,j} = \frac{\partial}{\partial u_j} f_i(x, u)|_{(x_0,u_0)}, \tag{6.13}$$

$$c_{i,j} = \frac{\partial}{\partial x_j} h_i(x, u)|_{(x_0,u_0)}, \tag{6.14}$$

$$d_{i,j} = \frac{\partial}{\partial u_j} h_i(x, u)|_{(x_0,u_0)}. \tag{6.15}$$

Since the octorotor will not be used for aerobatic maneuvers and will spend most of its time close to a hovering state it is reasonable to linearize the nonlinear model around the equilibrium point in hover. In this state, the Euler angles are zero as well as the angular rates expressed with respect to the body-fixed frame. When the gyroscopic and aerodynamic torque is neglected according to the above discussion the remaining nonlinear system is

$$\dot{\mathbf{\Theta}} = \mathbf{T}_{\mathrm{B}\to\mathrm{E}}\boldsymbol{\omega}, \tag{6.16}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}_G^{-1}(\boldsymbol{\omega} \times \mathbf{I}_G\boldsymbol{\omega} + \boldsymbol{\tau}_{\mathrm{rot}}). \tag{6.17}$$

These equations can now be linearized around

$$x_0 = \begin{bmatrix} \mathbf{\Theta}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} = \mathbf{0}_{6\times1}, \tag{6.18}$$

$$u_0 = \boldsymbol{\tau}_0 = \mathbf{0}_{3\times1}, \tag{6.19}$$

which gives the resulting linear system

$$\dot{\mathbf{\Theta}} = \mathbf{I}_{3x3}\boldsymbol{\omega}, \tag{6.20}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}_G^{-1}\boldsymbol{\tau}_{\text{rot}}. \tag{6.21}$$

These equations can also be written as

$$\dot{\phi} = p, \tag{6.22a}$$

$$\dot{\theta} = q, \tag{6.22b}$$

$$\dot{\psi} = r, \tag{6.22c}$$

$$\dot{p} = \frac{\tau_x}{I_{xx}}, \tag{6.22d}$$

$$\dot{q} = \frac{\tau_y}{I_{yy}}, \tag{6.22e}$$

$$\dot{r} = \frac{\tau_z}{I_{zz}}. \tag{6.22f}$$

From (6.22) it can be seen that the three different Euler angles are completely decoupled in the linearized model which is valid near the hovering state. Hence, a decentralized controller using one control loop for each angle was chosen as the attitude control structure. The torque provided by the rotors can be seen as a virtual input to the system, since the torque can not be controlled directly. The relationship between the torque and the rotation speed of the eight rotors is explained further in Section 6.2.2.

6.2.2 Control mixer

According to the linearized model of the system, the octorotor's attitude is controlled by varying the torque that is exerted on the system. From Chapter 4 it follows that the force and torque provided by the rotors are related to the rotor speeds according to

$$\begin{bmatrix} \mathbf{F} \\ \boldsymbol{\tau} \end{bmatrix} = \mathbf{\Lambda}\omega_{\text{rot}}^2. \tag{6.23}$$

By calculating the pseudo-inverse of the 6 × 8-matrix $\mathbf{\Lambda}$ it is possible to relate the four basic control inputs: F_z, τ_x, τ_y and τ_z to the corresponding desired rotation rates

$$\omega_{\text{rot}}^2 = \mathbf{\Lambda}^{\dagger} \begin{bmatrix} 0 \\ 0 \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \tag{6.24}$$

The multirotor used in this thesis has tilted rotor axes and hence the rotors' thrust causes weak forces in the X^{B}- and Y^{B}-directions. However, the aircraft's movement in the horizontal plane is not directly controlled by these forces. Instead, the linear position is indirectly controlled by varying the aircraft's attitude.

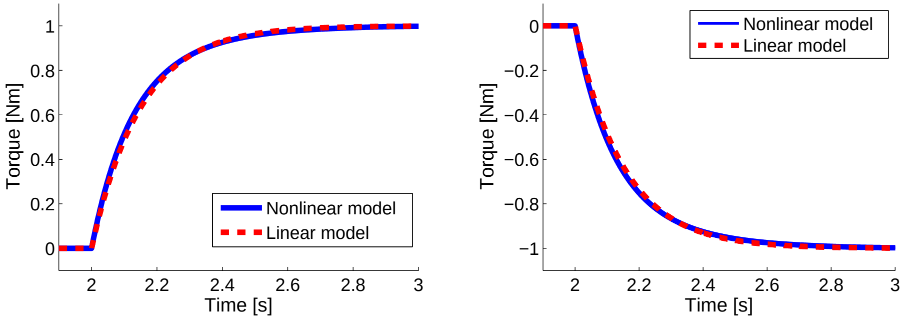
Hence, $F_x = F_y = 0$ in (6.24). The resulting vector $\boldsymbol{\omega}_{\text{rot}}^2$ is converted into rotation rate demands to each motor by calculating the square root of each element in the vector

$$\omega_{\text{rot}_i} = \sqrt{\omega_{\text{rot}_i}^2}, \quad i = 1, 2, \ldots 8. \tag{6.25}$$

6.2.3 Linearized torque dynamics

The motors cannot obtain the desired rotation rates instantaneously. Each motor has a dynamic response to a rate command that is thoroughly described in Section 4.4. This response is nonlinear and does not describe the dynamic properties of the exerted torque on the platform. In order to get a complete linear model of the attitude dynamics the torque dynamics needed to be included in this model.

A step in the desired torque input to the control mixer was simulated. The resulting rate demands were provided as input to the nonlinear motor dynamics and the obtained rotation rates were converted into a resulting torque on the platform using the total model given by (4.14) and (4.15). The result was compared to the step response from a first order linear system with unit static gain. The time constant of the first order system was changed until a good match between the simulated nonlinear system response and the response from the linear system was obtained.



(a) Positive step responses *(b) Negative step responses*

Figure 6.2: *A comparison between the nonlinear and linear systems' step responses for a positive and negative step in τ_x. The linear system has a time constant $T_{\tau_{xy}} = 0.15$ [s].*

A time constant of $T_{\tau_{xy}} = 0.15$ [s] gave good results for a step in τ_x and τ_y respectively. The response was a bit faster for the torque around the Z^{B}-axis and a time constant of $T_{\tau_z} = 0.12$ [s] gave the best match. The comparison between the two systems' responses for a unit step in τ_x is shown in Figure 6.2. Even though the dynamic behavior of the motors is different for a positive compared to a negative command, the torque dynamics behave like a linear first order system. This is due to that a torque command is realized by changing the rotation rate of the motors in a symmetric way, see Section 3.6. For any given command, the same amount of motors will increase and decrease their rotation rate.

6.2.4 Total linear model

By combining the linear model of the torque dynamics described in Section 6.2.3 with the linearized dynamic model in (6.22) a total linear model of the controlled system can be obtained. The model is converted from state-space form to a transfer function representation and the result is

$$\phi(s) = \frac{1}{s}P(s), \tag{6.26a}$$

$$P(s) = \frac{1}{I_{xx}s(T_{\tau_{xy}}s+1)}U_{\tau_x}(s), \tag{6.26b}$$

$$\theta(s) = \frac{1}{s}Q(s), \tag{6.26c}$$

$$Q(s) = \frac{1}{I_{yy}s(T_{\tau_{xy}}s+1)}U_{\tau_y}(s), \tag{6.26d}$$

$$\psi(s) = \frac{1}{s}R(s), \tag{6.26e}$$

$$R(s) = \frac{1}{I_{zz}s(T_{\tau_z}s+1)}U_{\tau_z}(s). \tag{6.26f}$$

Here, $u_{\tau_x}(t)$, $u_{\tau_y}(t)$ and $u_{\tau_z}(t)$ are the control signals from the decentralized controller. The model described in (6.26) is the model that is used for the controller design.

6.3 Controller design

One of the main goals of this thesis is to create an attitude controller that allows a human operator to control the aircraft's roll angle, pitch angle and yaw rate. This controller can then be used in future work to create a position controller and trajectory planner as described in Section 6.1. The attitude controller can be created using a variety of different methods. In this thesis, two different controllers are designed, tuned and evaluated with respect to a number of different performance measures. The first controller is designed using PID techniques. The second controller is designed as a PD controller augmented with an $\mathcal{L}_1$ adaptive controller using the linear model described in Section 6.2.

6.3.1 PID controller

This section describes the decentralized PID controller that is used for controlling the octorotor's roll angle, pitch angle and yaw rate. The controller is based on the generalized PID structure described in Section 2.9.2.

Roll and pitch angle control

The PID controllers for the octorotor's roll and pitch angles utilize the fact that the derivatives of the controlled variables $\phi(t)$ and $\theta(t)$ are directly measured by the gyroscope. This makes it possible to include derivative action in the PID controller without having to explicitly compute the derivative of the estimated roll and pitch angles. It was decided that the derivative of the reference signal $r(t)$ should not be included when forming the control signal which is a common choice for 2DoF PID controllers [1]. Hence, quick and large changes in the reference signal will not give raise to large control inputs, which is a desired behavior. This is equivalent to setting the parameter β to zero in the generalized PID controller given by (2.78). The control algorithm that is used in the PID controllers for the roll and pitch angles is hence given by

$$u(t) = K_P(\alpha r(t) - y(t)) + K_I \int_0^t e(\tau)\,\mathrm{d}\tau - K_D \frac{\mathrm{d}y(t)}{\mathrm{d}t}. \tag{6.27}$$

Variants of this controller structure with $\alpha = 1$ is common in multirotor projects and the process of using gyroscope measurements instead of explicitly computing the derivative of the control error is referred to as rate feedback [19]. By letting $\beta = 0$, only one zero instead of two are introduced to the closed-loop transfer function.

The transfer functions describing the controllers and the linear system are

$$F_r(s) = \alpha K_P + \frac{K_I}{s}, \tag{6.28}$$

$$F_y(s) = K_P + \frac{K_I}{s} + K_D s, \tag{6.29}$$

$$G^{\phi,\theta}(s) = \frac{1}{Is^2(T_{\tau_{xy}} s + 1)}. \tag{6.30}$$

Here, $G^{\phi,\theta}(s)$ is the transfer function from $U_{\tau_x}(s)$ and $U_{\tau_y}(s)$ to $\phi(s)$ and $\theta(s)$. The time constant $T_{\tau_{xy}}$ is described in Section 6.2.3. The moment of inertia I depends on whether the roll or pitch angle is controlled. The values for I_{xx} and I_{yy} respectively are given in Section 4.5. The controller transfer functions together with the linear model of the system give the following transfer functions

$$G_c^{\phi,\theta}(s) = \frac{1}{IT_{\tau_{xy}}} \cdot \frac{\alpha K_P s + K_I}{s^4 + \frac{1}{T_{\tau_{xy}}} s^3 + \frac{K_D}{IT_{\tau_{xy}}} s^2 + \frac{K_P}{IT_{\tau_{xy}}} s + \frac{K_I}{IT_{\tau_{xy}}}}, \tag{6.31}$$

$$S^{\phi,\theta}(s) = \frac{s^4 + \frac{1}{T_{\tau_{xy}}} s^3}{s^4 + \frac{1}{T_{\tau_{xy}}} s^3 + \frac{K_D}{IT_{\tau_{xy}}} s^2 + \frac{K_P}{IT_{\tau_{xy}}} s + \frac{K_I}{IT_{\tau_{xy}}}}, \tag{6.32}$$

$$S_u^{\phi,\theta}(s) = \frac{1}{IT_{\tau_{xy}}} \cdot \frac{s}{s^4 + \frac{1}{T_{\tau_{xy}}} s^3 + \frac{K_D}{IT_{\tau_{xy}}} s^2 + \frac{K_P}{IT_{\tau_{xy}}} s + \frac{K_I}{IT_{\tau_{xy}}}}, \tag{6.33}$$

$$T^{\phi,\theta}(s) = \frac{1}{IT_{\tau_{xy}}} \cdot \frac{K_D s^2 + K_P s + K_I}{s^4 + \frac{1}{T_{\tau_{xy}}} s^3 + \frac{K_D}{IT_{\tau_{xy}}} s^2 + \frac{K_P}{IT_{\tau_{xy}}} s + \frac{K_I}{IT_{\tau_{xy}}}}. \tag{6.34}$$

All transfer functions have the same denominator polynomial but different numerator polynomials. Hence, all transfer functions will have the same poles given by K_P, K_I and K_D, but the zeros will differ. The zero in $G_c^{\phi,\theta}(s)$ is possible to modify using the reference weighting parameter α.

It was decided that the controller design should be carried out using pole placement techniques. The decision was based on the fact that each closed-loop controller is a SISO system with relatively few poles compared to the number of free parameters in the controller. The closed-loop system has four poles which gives three possible pole placement options

1. four real-valued poles
2. two real-valued poles and one complex conjugate pole pair
3. two complex conjugate pole pairs.

To find the corresponding pole polynomials, the four first order polynomials for each case were multiplied.

1. Poles in $s = -a, -b, -c$ and $-d$ give the following polynomial

$$\begin{aligned} &(s+a)(s+b)(s+c)(s+d) = \\ &s^4 + (a+b+c+d)s^3 \\ &+ (ab+ac+ad+bc+bd+cd)s^2 \\ &+ (abc+abd+acd+bcd)s + abcd. \end{aligned} \tag{6.35}$$

2. Poles in $s = -a, -b$ and $-c \pm di$ give the following polynomial

$$\begin{aligned} &(s+a)(s+b)(s+c+di)(s+c-di) = \\ &s^4 + (a+b+2c)s^3 \\ &+ (ab+2ac+2bc+c^2+d^2)s^2 \\ &+ (2abc+ac^2+ad^2+bc^2+bd^2)s + abc^2 + abd^2. \end{aligned} \tag{6.36}$$

3. Poles in $s = -a \pm bi$ and $-c \pm di$ give the following polynomial

$$\begin{aligned} &(s+a+bi)(s+a-bi)(s+c+di)(s+c-di) = \\ &s^4 + (2a+2c)s^3 \\ &+ (2a^2c+2ac^2+2ad^2+2b^2c)s^2 \\ &+ (a^2+b^2+c^2+d^2+4ac)s \\ &+ a^2c^2 + a^2d^2 + b^2c^2 + b^2d^2. \end{aligned} \tag{6.37}$$

From (6.35), (6.36) and (6.37) it can be seen that the coefficient for s^3 is equal to the negative sum of the real parts of the poles. A comparison to the pole polynomial of the transfer functions in (6.31) through (6.34) shows that the negative sum of the real parts of the poles must always equal the inverse of the time constant for the motor dynamics. This will limit the attainable performance for this controller structure. However, as long as this constraint is met there is full freedom in specifying the imaginary part of the complex conjugate pole pairs.

In general, the position of a real pole or a complex conjugate pole pair in the complex plane is related to the closed loop system's properties in the following way:

- Poles far into the left half plane, i.e. poles with a large distance to the origin, correspond to a fast system with low rise time and high bandwidth.
- For complex conjugate pole pairs a large quotient between the imaginary and the real part gives an oscillative system with low stability margins.
- For higher order systems it is the pole or pole pair closest to the origin that dominates the system's response.

According to the above reasoning it is common to try to place the poles in an area spanned by two bisectors in the left complex plane [28], see Figure 6.3 for an illustration.

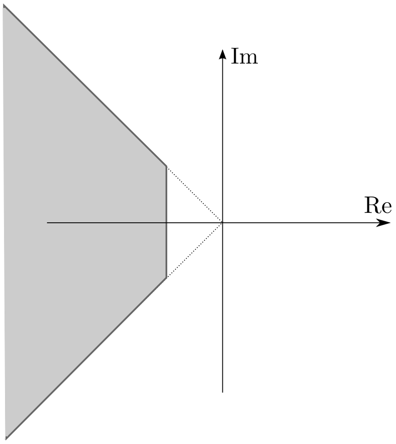


Figure 6.3: Illustration of a desirable pole placement. It is common to place the poles of the closed-loop system inside the gray area spanned by the two bisectors.

Since the sum of the real parts of the poles is constrained it was decided that the third option with two complex conjugate pole pairs should be used when placing the poles of the closed-loop system transfer functions. The system responses obtained using the other two options proved to be too slow.

For option three, the PID parameters K_P, K_I and K_D can be computed from the desired pole placement according to Algorithm 2.

Algorithm 2 Method for computing the parameters K_P, K_I and K_D given a specific pole placement.

```
  Choose desired negative real parts $a$ and $c$, and imaginary parts $b$ and $d$ for the closed-loop poles.
**Ensure:** that the constraint $2a + 2c = 1/T_{\tau_{xy}}$ holds.
  Compute the corresponding values of $K_P$, $K_I$ and $K_D$ according to
  $K_P \leftarrow IT_{\tau_{xy}}(a^2 + b^2 + c^2 + d^2 + 4ac)$
  $K_I \leftarrow IT_{\tau_{xy}}(a^2c^2 + a^2d^2 + b^2c^2 + b^2d^2)$
  $K_D \leftarrow IT_{\tau_{xy}}(2a^2c + 2ac^2 + 2ad^2 + 2b^2c)$
```

For each choice of pole placement, the system's step response was simulated for several different values of α in the range [0 1]. The value of α that gave the fastest response with an overshoot less than 10 % was chosen. Figure 6.4 shows the result of one such simulation.

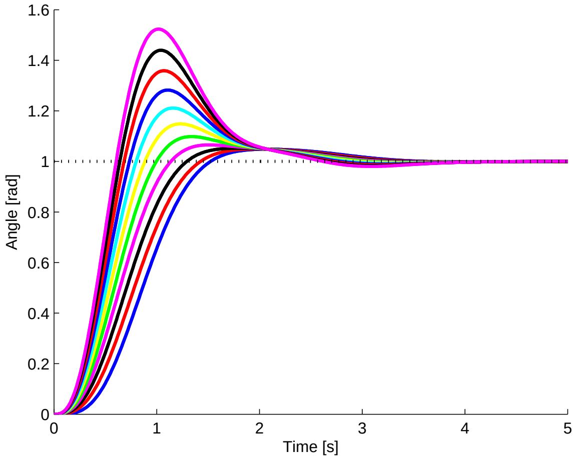


Figure 6.4: *Simulated step responses for the closed-loop system with poles in $s = -1.52 \pm 1.52$ and $s = -1.82 \pm 4.50$ for different values of α. The fastest step response corresponds to $\alpha = 1$ and the slowest to $\alpha = 0$.*

The first pole placement that was evaluated was to put both complex conjugate pole pairs in the same position, with the value of the real part equal to the imaginary part. Hence, a double pole pair was placed on the bisectors illustrated in

Figure 6.3. The green markers in Figure 6.5 illustrates this pole placement choice.

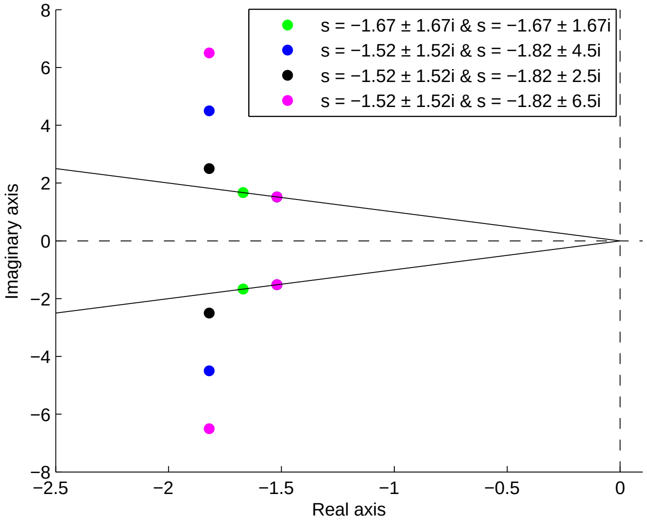


Figure 6.5: *Different pole placements evaluated for the PID roll and pitch angle controllers. The solid lines correspond to the bisectors in Figure 6.3.*

This gave a system with a rise time around one second which was deemed insufficient. In order to decrease the system's response time it was necessary to allow poles with larger imaginary part compared to the real part, i.e. less damped poles. However, while providing better response times this also decreases the stability of the system.

The properties of the system are most affected by the poles closest to the origin, the dominant poles. Hence, to obtain a well damped system which still performed well in terms of response time the two pole pairs were moved away from each other and the pole pair closest to the origin was placed on the bisectors in Figure 6.3. This made it possible to increase the imaginary part of the non-dominant pole pair to improve speed and increase the loop gain while still keeping the system well damped. The spacing could only be increased up to a certain level to prevent the dominant poles from slowing down the system response when they move closer to the origin. When the imaginary part of the non-dominant poles were increased, the system's loop gain increased which in turn led to better input disturbance rejection and a faster response. However, this also made the system more oscillative. If the imaginary part was decreased, the system became less oscillative but the input disturbance rejection and response time worsened. Three different pole placements were evaluated, see the black, blue and magenta colored markers in Figure 6.5. The resulting step response and input disturbance rejection for each pole placement is illustrated in Figure 6.6.

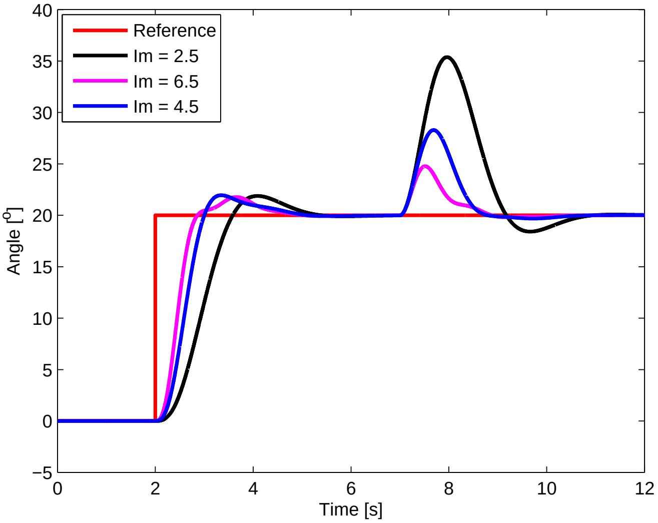


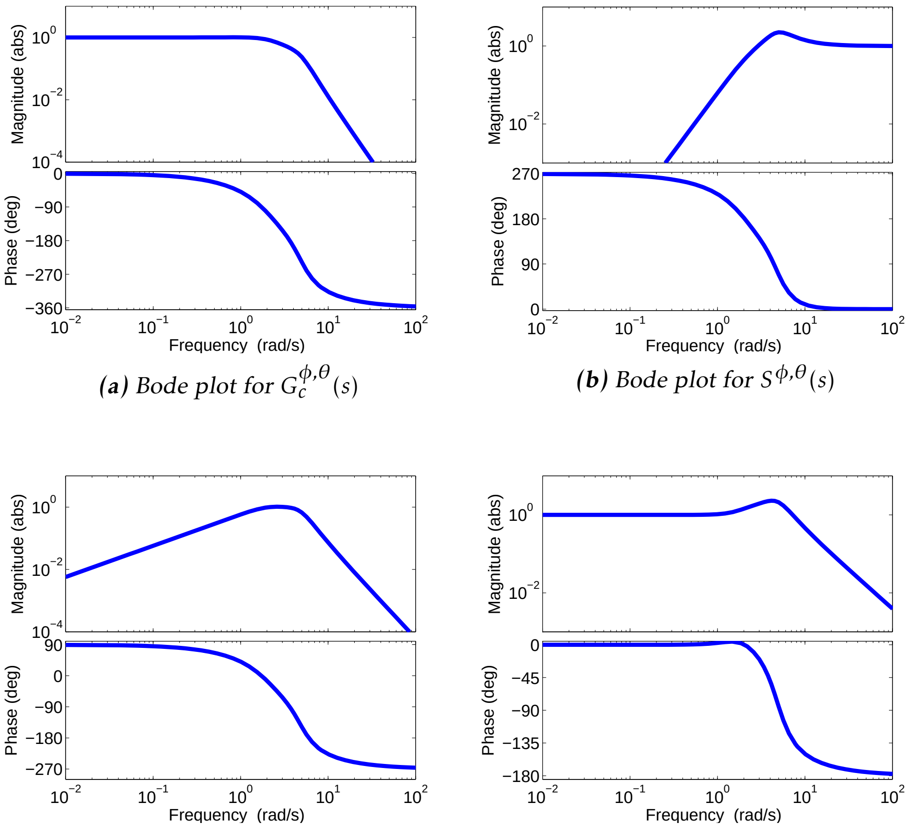
Figure 6.6: *Simulated step responses and disturbance rejection obtained for three different pole placements with differing imaginary parts for the non-dominant pole pair (see* Im *in the plot legend and Figure 6.5). The PID parameters were calculated using Algorithm 2. A reference step of* 20° *in roll angle is introduced at* $t = 2$ *[s]. A step input disturbance of* $\tau_x = 0.2$ *[Nm] is introduced at* $t = 7$ *[s].*

The final choice of pole placement gives a closed-loop system with a response that has no oscillations and less than 10 % overshoot. The system's response and its input disturbance rejection is illustrated by the blue line in Figure 6.6. The poles were placed in $s = -1.52 \pm 1.52$ and $s = -1.82 \pm 4.50$. The same pole placement was chosen for both the roll and pitch angle controller. This gave the following PID parameters

$$K_P^{\phi} = 1.42, \quad K_I^{\phi} = 1.74, \quad K_D^{\phi} = 0.63, \quad \alpha^{\phi} = 0.4, \tag{6.38}$$

$$K_P^{\theta} = 1.40, \quad K_I^{\theta} = 1.73, \quad K_D^{\theta} = 0.62, \quad \alpha^{\theta} = 0.4. \tag{6.39}$$

Bode plots for $G_c^{\phi,\theta}(s)$, $S^{\phi,\theta}(s)$, $S_u^{\phi,\theta}(s)$ and $T^{\phi,\theta}(s)$ corresponding to the roll angle system are shown in Figure 6.7. The Bode plots for the pitch angle system are identical except for a slight change in static gain for $S_u^{\phi,\theta}(s)$. This is due to that the pole polynomial is identical for the two systems. It can be seen that $S^{\phi,\theta}(s)$ and $T^{\phi,\theta}(s)$ have complementary frequency characteristics as previously discussed. The bode plot for $G_c^{\phi,\theta}(s)$ is monotonically decreasing and does not include any resonance peaks. This means that the step response should be well damped which is also the case. The bode plot for $S_u^{\phi,\theta}(s)$ never goes above 1.0 which means that no frequency in an input disturbance signal is amplified.



*(**a**) Bode plot for $G_c^{\phi,\theta}(s)$*

*(**b**) Bode plot for $S^{\phi,\theta}(s)$*

*(**c**) Bode plot for $S_u^{\phi,\theta}(s)$*

*(**d**) Bode plot for $T^{\phi,\theta}(s)$*

Figure 6.7: *Bode plots for the closed-loop transfer functions obtained when using the roll and pitch angle PID controller.*

The bode plot for the open-loop system obtained when using the roll angle controller is shown in Figure 6.8. The system has a phase margin of 26.2°. The gain margin is 0.32 which means that the system will lose its stability if the loop gain is decreased by more than 68 %. This is reasonable since the octorotor system needs a closed-loop controller in order to be stable. Increasing the loop gain will decrease the phase margin but the system will always be stable since the phase curve never goes below -180° for increasing frequencies.

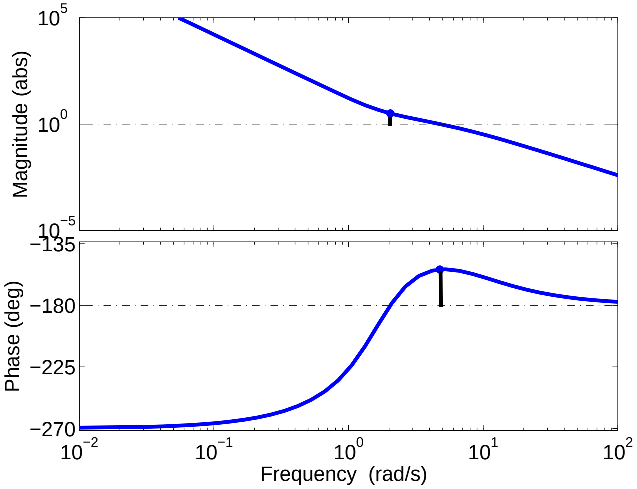


Figure 6.8: *Bode plot for the open-loop system obtained when using the roll and pitch angle PID controller. The system has a gain margin of 0.32 and a phase margin of 26.2°.*

Yaw rate control

The design of the controller for the octorotor's yaw rate was based on the linear model of the yaw rate dynamics

$$G^{\psi}(s) = \frac{1}{I_{zz}s(T_{\tau_z}s+1)}. \tag{6.40}$$

It was decided that the yaw rate controller should be designed as a 2DoF PI controller similar to the controller for the roll and pitch angle. The derivative action in the controller was omitted to avoid having to explicitly compute the derivative of the noisy yaw rate measurements.

The transfer functions for the yaw rate controller are

$$F_r(s) = \frac{\alpha K_P s + K_I}{s}, \tag{6.41}$$

$$F_y(s) = \frac{K_P s + K_I}{s}. \tag{6.42}$$

This controller structure gives the following closed-loop transfer functions

$$G_c^{\psi}(s) = \frac{1}{I_{zz}T_{\tau_z}} \cdot \frac{\alpha K_P s + K_I}{s^3 + \frac{1}{T_{\tau_z}}s^2 + \frac{K_P}{I_{zz}T_{\tau_z}}s + \frac{K_I}{I_{zz}T_{\tau_z}}}, \tag{6.43}$$

$$S^{\psi}(s) = \frac{s^3 + \frac{1}{T_{\tau_z}}s^2}{s^3 + \frac{1}{T_{\tau_z}}s^2 + \frac{K_P}{I_{zz}T_{\tau_z}}s + \frac{K_I}{I_{zz}T_{\tau_z}}}, \tag{6.44}$$

$$S_u^{\psi}(s) = \frac{1}{I_{zz}T_{\tau_z}} \cdot \frac{s}{s^3 + \frac{1}{T_{\tau_z}}s^2 + \frac{K_P}{I_{zz}T_{\tau_z}}s + \frac{K_I}{I_{zz}T_{\tau_z}}}, \tag{6.45}$$

$$T^{\psi}(s) = \frac{1}{I_{zz}T_{\tau_z}} \cdot \frac{K_P s + K_I}{s^3 + \frac{1}{T_{\tau_z}}s^2 + \frac{K_P}{I_{zz}T_{\tau_z}}s + \frac{K_I}{I_{zz}T_{\tau_z}}}. \tag{6.46}$$

Similar to the closed-loop system obtained for the roll and pitch angle controller, it is the inverse of the motor time constant that limits the attainable performance. The closed-loop system has three poles which gives two possible pole placement options

1. three real-valued poles
2. one real-valued pole and one complex conjugate pole pair.

Again, the sum of the negative real parts of the three poles must always equal the inverse motor time constant. Hence, the second option was chosen in order to get a system with a sufficient response time. The pole polynomial for one real valued pole in $s = -a$ and a complex conjugate pole pair in $s = -b \pm ci$ is

$$\begin{aligned} &(s+a)(s+b+ci)(s+b-ci) = \\ &+ s^3 + (a+2b)s^2 + (2ab + b^2 + c^2)s \\ &+ ab^2 + ac^2 \end{aligned} \tag{6.47}$$

Algorithm 3 was used to compute the PI parameters. For each pole placement choice, the system's step response was simulated and the value of α that gave the fastest response with an overshoot less than 10 % was selected. The same pole placement strategy used for the roll and pitch controller was also used for the yaw rate controller.

Algorithm 3 Method for computing the parameters K_P and K_I given a specific pole placement.

```
  Choose desired negative real parts $a$ and $b$, and imaginary part $c$ for the closed-loop poles
Ensure: that the constraint $a + 2b = 1/T_{\tau_z}$ holds.
  Compute the corresponding values of $K_P$ and $K_I$ according to
  $K_P \leftarrow I_{zz}T_{\tau_z}(2ab + b^2 + c^2)$
  $K_I \leftarrow I_{zz}T_{\tau_z}(ab^2 + ac^2)$
```

First, all poles were given the same real part and the complex pole pair was placed on the bisectors in Figure 6.9 (the green markers).

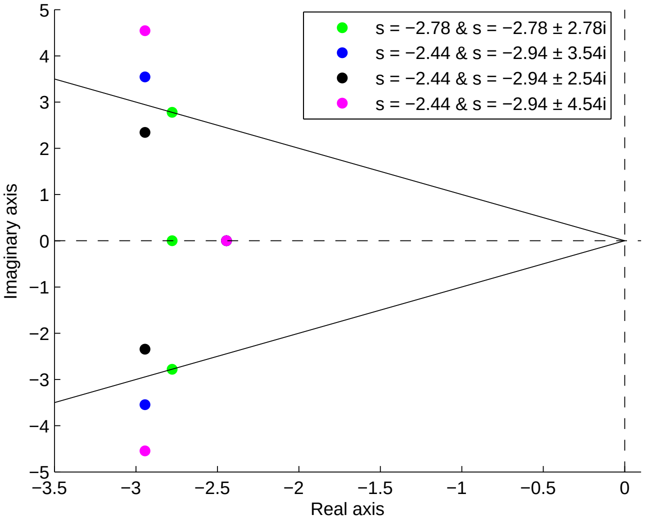


Figure 6.9: *Different pole placements evaluated for the PI yaw rate controller. The solid lines correspond to the bisectors in Figure 6.3.*

In order to be able to increase the loop gain and still maintain good reference tracking without too much oscillations it was necessary to move the real valued pole closest to the origin. Hence, the real valued pole becomes the dominant pole and will therefore have the most impact on the system's behavior. As for the roll and pitch controller, the dominant pole could not be moved too close to the origin in order to maintain sufficient speed in the system. When the imaginary part of the non-dominant pole pair was increased, the input disturbance rejection improved but the system became more oscillative. The tradeoff is illustrated in Figure 6.10 with corresponding pole placements in Figure 6.9.

The final choice of pole placement gives a closed-loop system with a response that has very small oscillations and less than 10 % overshoot. The system's response and its input disturbance rejection is illustrated by the blue line in Figure 6.10. The poles were placed in $s = -2.44$ and $s = -2.94 \pm 3.54$. This gave the following PI parameters

$$K_P^{\psi} = 0.89, \quad K_I^{\psi} = 1.29, \quad \alpha^{\psi} = 0.6. \tag{6.48}$$

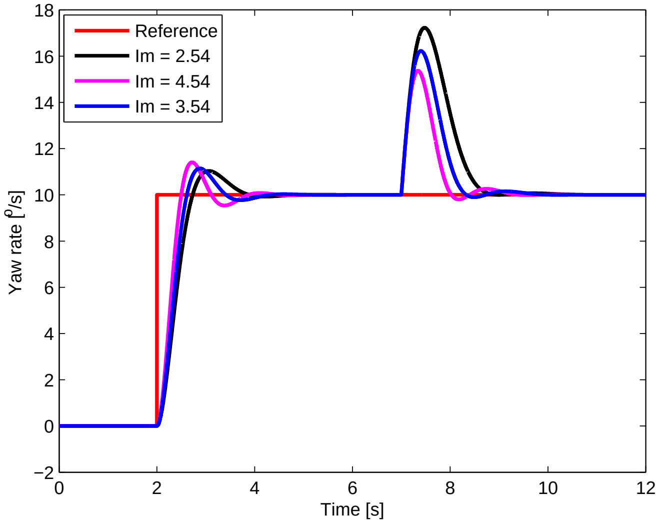
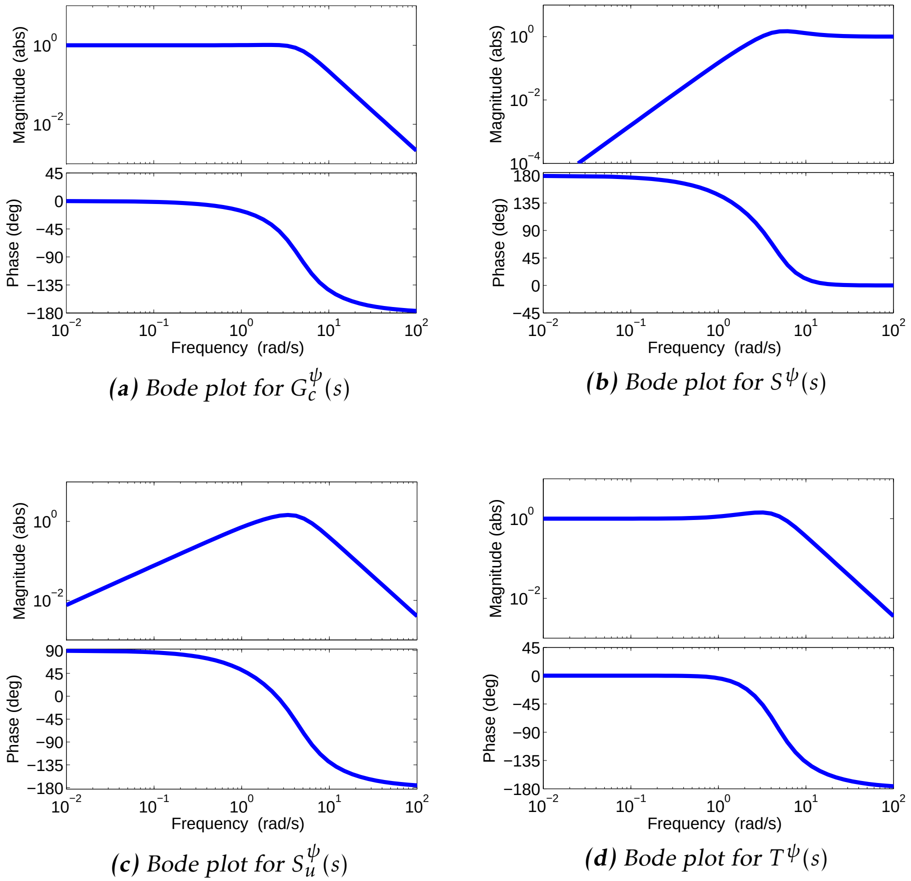


Figure 6.10: *Simulated step responses and disturbance rejection obtained for three different pole placements with differing imaginary parts for the non-dominant pole pair (see* Im *in the plot legend and Figure 6.9). The PI parameters were calculated using Algorithm 3. A reference step of 10°/s in yaw rate is introduced at* $t = 2$ *[s]. A step input disturbance of* $\tau_z = 0.1$ *[Nm] is introduced at* $t = 7$ *[s].*

The bode plots for the yaw rate closed-loop system are shown in Figure 6.11 and the bode plot for the open-loop system is shown in Figure 6.12. The system has a phase margin of 44.3° and an infinite gain margin.



(a) *Bode plot for $G_c^{\psi}(s)$*

(b) *Bode plot for $S^{\psi}(s)$*

(c) *Bode plot for $S_u^{\psi}(s)$*

(d) *Bode plot for $T^{\psi}(s)$*

Figure 6.11: *Bode plots for the closed-loop transfer functions obtained when using the yaw rate PI controller.*

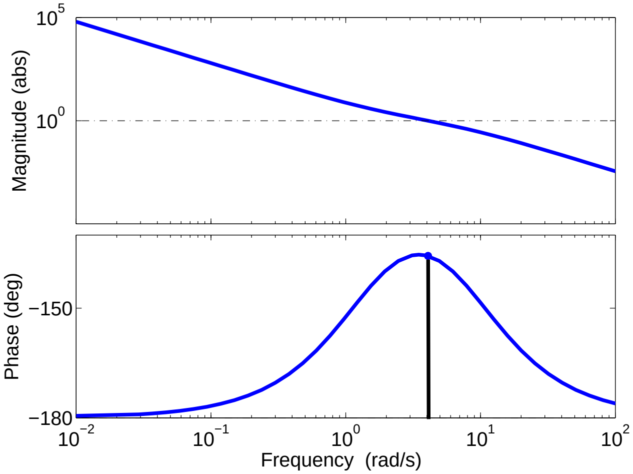


Figure 6.12: *Bode plot for the open-loop system obtained when using the yaw rate PI controller. The system has a phase margin of 44.3° and an infinite gain margin.*

6.3.2 PD controller with $\mathcal{L}_1$ adaptive control augmentation

In addition to the PID controller described in Section 6.3.1 another decentralized controller was designed. This controller is based on a PD controller augmented with an $\mathcal{L}_1$ adaptive controller to handle input disturbances and model uncertainties. Hence, the integral action of the PID controller is replaced with the adaptive controller.

Roll and pitch angle control

By setting $K_I = 0$, $\alpha = 1$ and $\beta = 0$ in (2.78) a PD controller is obtained that utilizes the fact that the derivative of the controlled variable is directly measured by the gyroscopes. This type of PD controller is used as a baseline controller for control of the octorotor's roll and pitch angles. As for the PID controller described in Section 6.3.1, pole placement was used to obtain the control parameters K_P and K_D. The poles were placed such that the PD controllers' reference tracking was similar to the closed-loop tracking obtained when using the PID controllers. The PD parameters that are used are

$$K_P^\phi = 0.87, \quad K_D^\phi = 0.48, \tag{6.49}$$

$$K_P^\theta = 0.86, \quad K_D^\theta = 0.47. \tag{6.50}$$

The PD controller structure can be interpreted as two P controllers forming a cascaded control loop according to Figure 6.13.

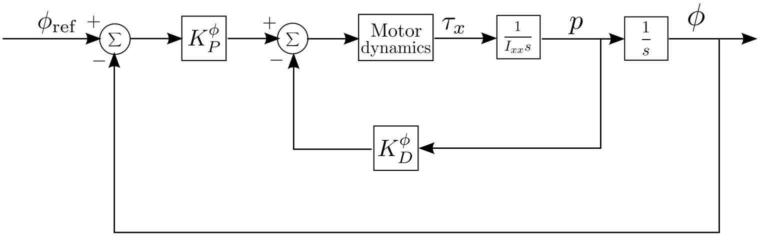


Figure 6.13: *Block diagram of the linearized octorotor roll dynamics controlled by a PD controller. This control structure can be interpreted as a cascaded control system with two P controllers.*

The inner loop, the rate-feedback loop, is augmented with an $\mathcal{L}_1$ adaptive controller to provide input disturbance rejection and robustness against modeling errors. A similar approach is used in [22]. A key issue is how to handle the motor dynamics and incorporate it into the $\mathcal{L}_1$ adaptive control framework. In [22] the motor dynamics is neglected and hence the system needs to be designed with a sufficient time delay margin for the closed-loop system. In this thesis, an alternative approach is used. By placing the actuator dynamics model before the state predictor given by (2.89) it is possible to include the actuator dynamics without the need of an additional observer. This approach is also used in [25]. No unmatched compensation is necessary in this design which leads to a simplified state predictor, adaptation law and control law, thus requiring less computational power.

If the actuator model is moved from the inner loop according to the above discussion, the inner closed-loop system (the rate-feedback loop) can be written in state space form as

$$\dot{p}(t) = -\frac{K_D^\phi}{I_{xx}} p(t) + \frac{1}{I_{xx}} u(t) + f(t, p(t)), \tag{6.51}$$

where $u(t)$ is the control input signal from the outer loop when no disturbances are present. Hence, the inner closed-loop system can be seen as a first order SISO system affected by general nonlinear disturbances $f(t, p(t))$. By comparing this expression to (2.83) it is possible to identify the matrices A_m and B_m. This gives

$$A_m = -\frac{K_D^\phi}{I_{xx}}, \quad B_m = \frac{1}{I_{xx}}, \tag{6.52}$$

which corresponds to the desired rate loop dynamics. Since the matrix B_m is a scalar there are no unmatched uncertanties. Hence, the problem formulation in (6.51) can be rewritten as

$$\dot{p}(t) = -\frac{K_D^\phi}{I_{xx}} p(t) + \frac{1}{I_{xx}} \left(u(t) + f_1(t, p(t))\right), \tag{6.53}$$

where $f_1(t, p(t))$ can be interpreted as modeling errors and input disturbances.

The $\mathcal{L}_1$ adaptive controller will strive to make the inner loop behave like a first order system with A_m and B_m according to (6.52). As soon as $\hat{p}(t)$ estimated by the state predictor deviates from the measured roll rate, the $\mathcal{L}_1$ controller will try to compensate for this deviation via the control signal.

The $\mathcal{L}_1$ adaptive controller for the roll rate loop is described by:

State predictor

$$\dot{\hat{p}}(t) = -\frac{K_D^{\phi}}{I_{xx}}\hat{p}(t) + \frac{1}{I_{xx}}\left(u^{*}(t) + \hat{\sigma}_1(t)\right), \tag{6.54}$$

$$U^{*}(s) = \frac{1}{T_{\tau_{xy}}s + 1}U(s). \tag{6.55}$$

Adaptation law

$$\hat{\sigma}_1(t) = \hat{\sigma}_1(kT_s),\ t \in [kT_s,\ (k+1)T_s), \tag{6.56}$$

$$\hat{\sigma}_1(kT_s) = M_{\phi}\tilde{p}(kT_s), \tag{6.57}$$

$$\tilde{p} \triangleq \hat{p}(kT_s) - p_{\text{meas}}(kT_s), \tag{6.58}$$

$$M_{\phi} \triangleq -B_m^{-1}\Phi^{-1}e^{A_m T_s}. \tag{6.59}$$

Control law

$$U(s) = K_P^{\phi}\left(\phi_{\text{ref}}(s) - \phi_{\text{meas}}(s)\right) - C(s)\hat{\sigma}_1(s), \tag{6.60}$$

$$C(s) = \frac{K_{\phi}}{s + K_{\phi}}. \tag{6.61}$$

The control signal from the proportional action of the outer loop controller is interpreted as the reference signal to the inner loop $\mathcal{L}_1$ adaptive controller. It was decided that this signal should not be low pass filtered through $C(s)$ since it was found to give better performance. This was done also when designing the controller in [25]. The control signal input to the state predictor is formed as the difference between the reference input and the low pass filtered estimation of $\sigma_1(t)$ fed through the motor dynamics. The total control signal from the PD controller with $\mathcal{L}_1$ augmentation is formed as

$$U_{\tau_x}^{\mathcal{L}_1}(s) = U_{\tau_x}^{PD}(s) - C(s)\hat{\sigma}_1(s), \tag{6.62}$$

$$U_{\tau_x}^{PD}(s) = K_P^{\phi}\left(\phi_{\text{ref}}(s) - \phi_{\text{meas}}(s)\right) - K_D^{\phi}P_{\text{meas}}(s). \tag{6.63}$$

Equation (6.62) and (6.63) show that the $\mathcal{L}_1$ adaptive controller augments the baseline PD controller. An overview of the entire control system for the roll angle is shown in Figure 6.14. For the pitch angle controller, an identical structure but with different PD parameters and moment of inertia was used.

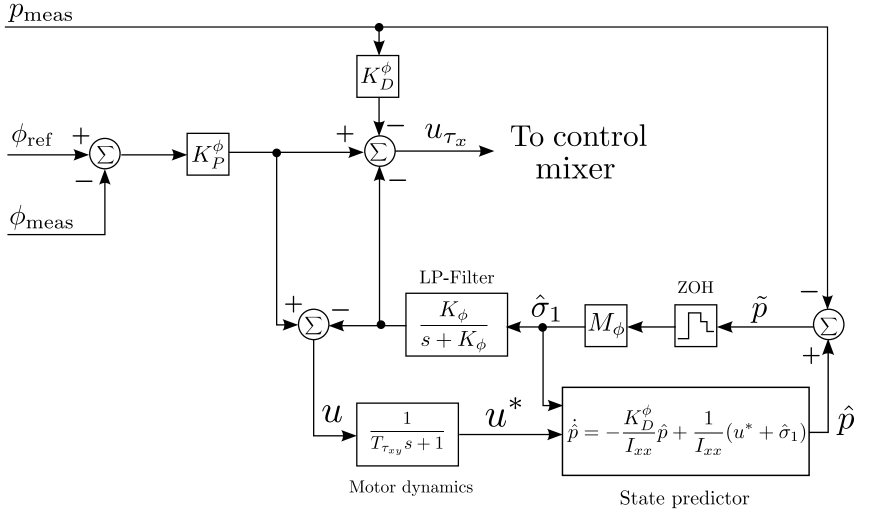


Figure 6.14: *Block diagram of the roll angle control system. The $\mathcal{L}_1$ adaptive control augmentation consists of a state predictor, low pass filter and a motor dynamics model. The reference signal from the outer loop to the inner loop is not filtered through the low pass filter.*

The $\mathcal{L}_1$ adaptive controller is tuned by varying the sample time T_s and the bandwidth of the lowpass filter $C(s)$. In general, lower sampling time and higher bandwidth give better performance. However, the sample time is limited by the hardware and the bandwidth of the lowpass filter need to be sufficiently low so that measurement noise is attenuated and the controller does not try to compensate for frequencies outside the model's valid range. The sample time was set to $T_s = 0.004$ [s] corresponding to 250 Hz when the simulations of the system was performed. The filter bandwidth in [rad/s] is equal to the value of the parameter K. This value was set to 15 which gave a good tradeoff between noise attenuation and controller performance.

Yaw rate control

As for the roll and pitch angle controllers the integral action in the yaw rate PI controller was omitted and replaced with an $\mathcal{L}_1$ adaptive controller augmentation. This gives a baseline P controller that is used for controlling the octorotor's yaw rate. The P parameter value for the baseline controller was obtained using pole placement aiming for similar reference tracking as the PI controller in Section 6.3.1. The P parameter was chosen as

$$K_P^\psi = 0.70. \tag{6.64}$$

The $\mathcal{L}_1$ controller augmentation was performed in a similar way as for the roll and pitch angle controllers. The controller for the yaw rate loop is described by

State predictor

$$\dot{\hat{r}}(t) = -\frac{K_P^{\psi}}{I_{zz}}\hat{r}(t) + \frac{1}{I_{zz}}\left(u^*(t) + \hat{\sigma}_1(t)\right), \tag{6.65}$$

$$U^*(s) = \frac{1}{T_{\tau_z}s + 1}U(s). \tag{6.66}$$

Adaptation law

$$\hat{\sigma}_1(t) = \hat{\sigma}_1(kT_s),\ t \in [kT_s,\ (k+1)T_s), \tag{6.67}$$

$$\hat{\sigma}_1(kT_s) = M_{\psi}\tilde{r}(kT_s), \tag{6.68}$$

$$\tilde{r} \triangleq \hat{r}(kT_s) - r_{\text{meas}}(kT_s), \tag{6.69}$$

$$M_{\psi} \triangleq -B_m^{-1}\Phi^{-1}e^{A_m T_s}. \tag{6.70}$$

Control law

$$U(s) = K_P^{\psi}\dot{\psi}_{\text{ref}}(s) - C(s)\hat{\sigma}_1(s), \tag{6.71}$$

$$C(s) = \frac{K_{\psi}}{s + K_{\psi}}. \tag{6.72}$$

The total control signal from the P controller with $\mathcal{L}_1$ augmentation is formed as

$$U_{\tau_z}^{\mathcal{L}_1}(s) = U_{\tau_z}^{P}(s) - C(s)\hat{\sigma}_1(s), \tag{6.73}$$

$$U_{\tau_z}^{P}(s) = K_P^{\psi}\left(\dot{\psi}_{\text{ref}}(s) - R_{\text{meas}}(s)\right). \tag{6.74}$$

An overview of the control system for the yaw rate is shown in Figure 6.15. The sample time T_s and filter bandwidth parameter K_{ψ} was set to the same values as for the roll and pitch angle controllers.

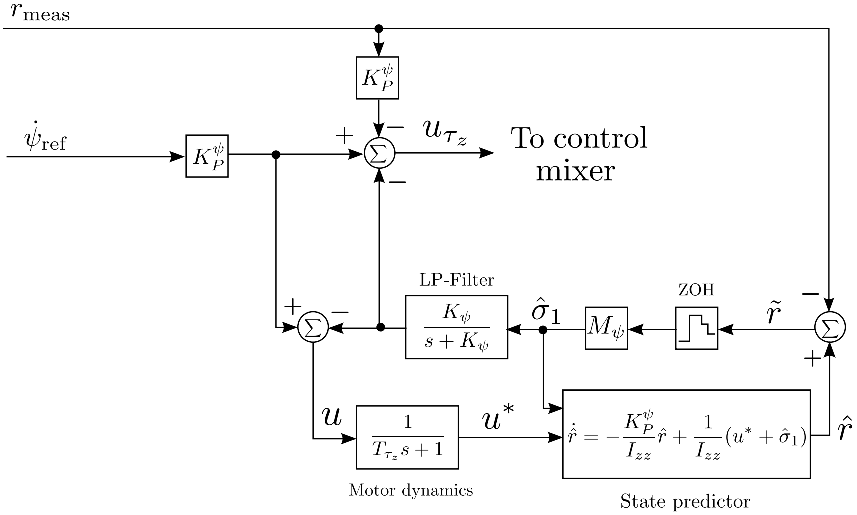


Figure 6.15: *Block diagram of the yaw rate control system. The $\mathcal{L}_1$ adaptive control augmentation consists of a state predictor, low pass filter and a motor dynamics model. The reference signal from the outer loop to the inner loop is not filtered through the low pass filter.*

6.4 Controller comparison and results

In this section a comparison of the two designed controller structures is performed using a simulation environment based on the nonlinear model of the octorotor described in Chapter 4. The implemented attitude filter algorithm from Chapter 5 depends on unmodeled aerodynamic effects such as blade flapping, induced drag and air resistance. Since these effects are not included in the octorotor model according to the discussion in Section 4.1.4 it is not possible to include the filter algorithm in the simulation environment. Instead, the roll and pitch angle outputs from the model are used as measurement input to the controllers. For the purpose of comparing the two different control strategies this is sufficient. Additive white noise has been added to the rate measurement inputs. The power of this noise is set to model the actual noise from the octorotor's gyroscopes. Since the output from the attitude filter described in Chapter 5 is very smooth compared to the gyroscope measurements no noise was added to the attitude measurement inputs. At the start of each simulation, the octorotor is in a stationary hovering state. The controller comparison was made with respect to three different aspects:

- reference tracking
- input disturbance rejection
- robustness against modeling errors.

For each aspect, results for the roll angle and yaw rate controllers are provided. The pitch dynamics is almost identical to the roll dynamics and is hence not included for brevity.

6.4.1 Reference tracking

The reference tracking comparison is carried out using two different reference inputs to the simulation environment. The first input is a series of steps and the second input is a stored reference signal from a real flight. Each simulation is carried out using the two different controllers and the resulting output and control signals for each controller are shown in the same figure. The step series simulation result for the roll angle is shown in Figure 6.16.

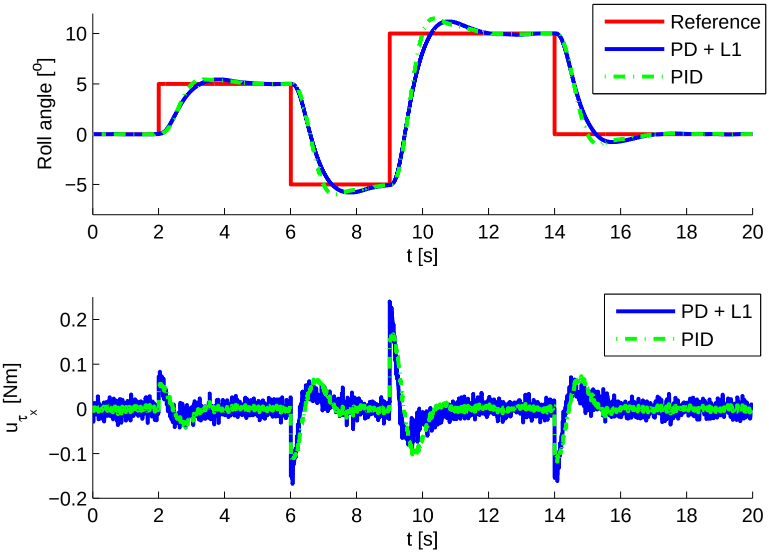


Figure 6.16: *Simulation result for the roll angle controllers using a series of steps as reference input. The lower plot shows the control signal outputs. The simulations are carried out using the nominal system model.*

It can be seen that the two controller structures behave similarly with respect to reference tracking which was one of the design goals. The PID controller gives a lower rise time compared to the PD + $\mathcal{L}_1$ controller but it has a slightly higher overshoot. The settling time is almost identical. The control signals are comparable, the PD + $\mathcal{L}_1$ controller uses slightly higher signals and is more affected by the measurement noise. This will not affect the motors as much since the control signal is filtered through the motor dynamics. It is possible to lower the effect of the noise on the control signal by lowering the filter constant K_ϕ in (6.61). However, this also lowers the overall performance of the controller.

The results obtained using the stored reference signal (from a live flight test) is given by Figure 6.17. The results are comparable to those obtained for the step series input. The two controllers structures give similar reference tracking. The control signal output by the PD + $\mathcal{L}_1$ controller is more affected by the noise.

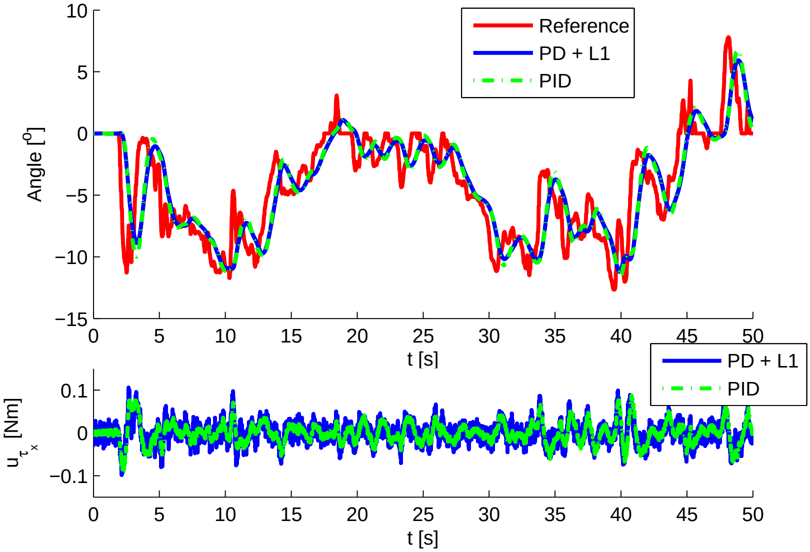


__Figure 6.17:__ Simulation result for the roll angle controllers using a stored reference signal from a multirotor flight. The lower plot shows the control signal outputs. The simulations are carried out using the nominal system model.

The same type of simulation was performed for the yaw rate controllers and the results are given by Figures 6.18 and 6.19. For the yaw rate controllers, the difference in reference tracking is more easily distinguished. The system controlled with the PI controller behaves like a higher order system with some overshoot. The reference tracking of the P + $\mathcal{L}_1$ controller on the other hand resembles that of a first order system. This is explained by the fact that the $\mathcal{L}_1$ augmentation strives to make the rate loop behave like a first order system, which is not the case in practice due to the motor dynamics. The $\mathcal{L}_1$ augmentation for the roll and pitch angle controllers will also strive to make the rate loop behave like a first order system, but since it is the absolute angles that are controlled in these channels this is not explicitly seen in the reference tracking in Figure 6.16. The PI controller has a lower rise time but the settling time is the same for the two controllers. The control signals are related in the same way as for the roll angle controllers. The P + $\mathcal{L}_1$ controller uses a slightly higher control signal and the signal is more affected by the measurement noise.

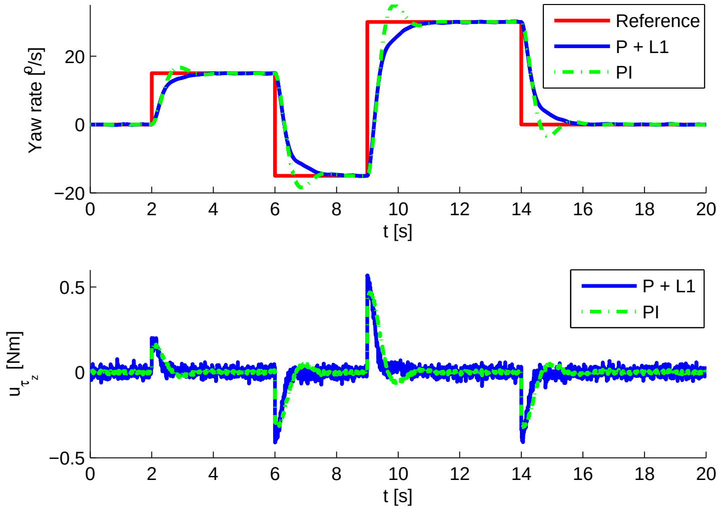


Figure 6.18: Simulation result for the yaw rate controllers using a series of steps as reference input. The lower plot shows the control signal outputs. The simulations are carried out using the nominal system model.

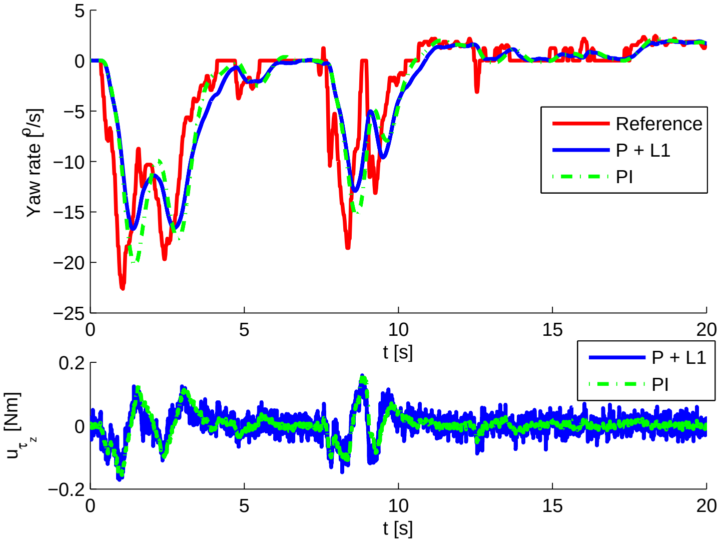


Figure 6.19: Simulation result for the yaw rate controllers using a using a stored reference signal from a multirotor flight. The lower plot shows the control signal outputs. The simulations are carried out using the nominal system model.

To study the effect of the cross coupling in the system a simulation was performed using a simultaneous step reference input for the roll and pitch angles. The result of this simulation is shown in Figure 6.20.

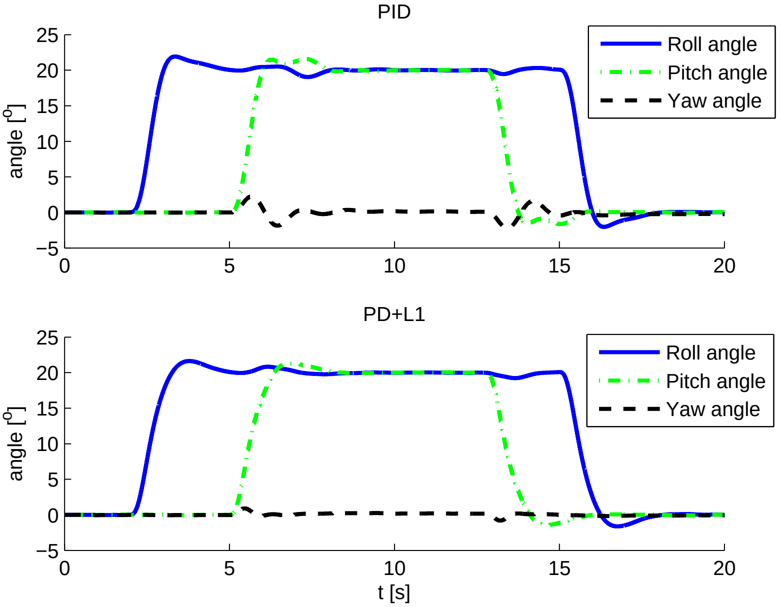


__Figure 6.20:__ Simulation of the reference tracking obtained when using a 20° step in roll angle between $t = 2$ and $t = 15$ [s], and a 20° step in pitch angle between $t = 5$ and $t = 13$ [s]. The upper plot corresponds to the PID controller and the lower plot to the PD + $\mathcal{L}_1$ controller.

It can be seen that the cross coupling introduces oscillations in the system using both controllers when more than one Euler angle is excited at a time. However, the $\mathcal{L}_1$ adaptive controller is better at handling the system's cross coupling. It can also be seen that the yaw angle is affected the most. This is due to that there is no absolute yaw angle controller.

6.4.2 Input disturbance rejection

When flying, the octorotor will be affected by wind disturbances which will exert forces and torques on the platform. The wind torque will affect the attitude of the aircraft and the designed controllers need to be able to reject this type of input disturbance. To compare the two controller structures' input disturbance rejection, two different simulations were performed. The first simulation is carried out using a step input disturbance. In addition, another scenario is simulated were the input disturbance is provided as a triangle pulses to imitate wind gusts. Figure 6.21 and 6.22 show the results for the step disturbance.

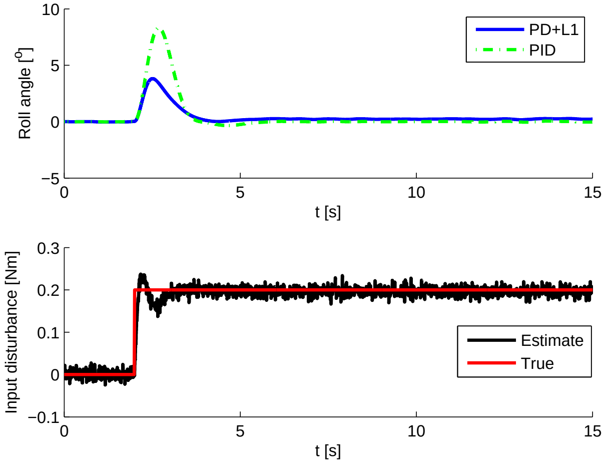


Figure 6.21: *Simulation result for the roll angle controllers using an input disturbance $\tau_x = 0.2$ [Nm] after 2 seconds. The lower plot shows how the low pass filtered value of $\hat{\sigma}_1$ in the $\mathcal{L}_1$ adaptive controller estimates the disturbance.*

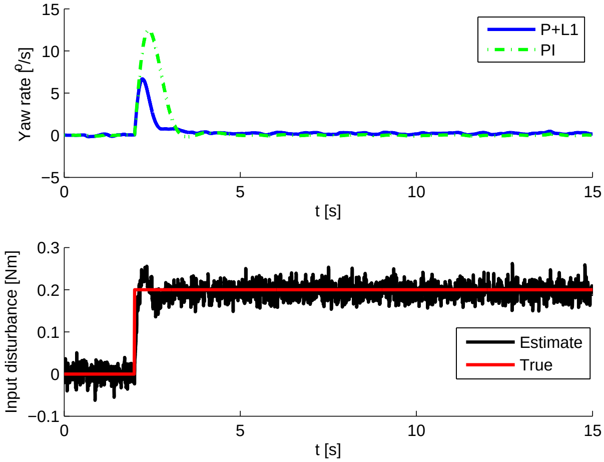


Figure 6.22: *Simulation result for the yaw rate controllers using an input disturbance $\tau_z = 0.2$ [Nm] after 2 seconds. The lower plot shows how the low pass filtered value of $\hat{\sigma}_1$ in the $\mathcal{L}_1$ adaptive controller estimates the disturbance.*

The results show that there is a significant performance gain when using the $\mathcal{L}_1$ augmented controllers compared to the standard PID and PI architectures.

The $\mathcal{L}_1$ adaptive controller is able to estimate the input disturbance quickly and compensate for it by modifying the control signal. The results for the triangle pulse simulations are shown in Figure 6.23 and 6.24.

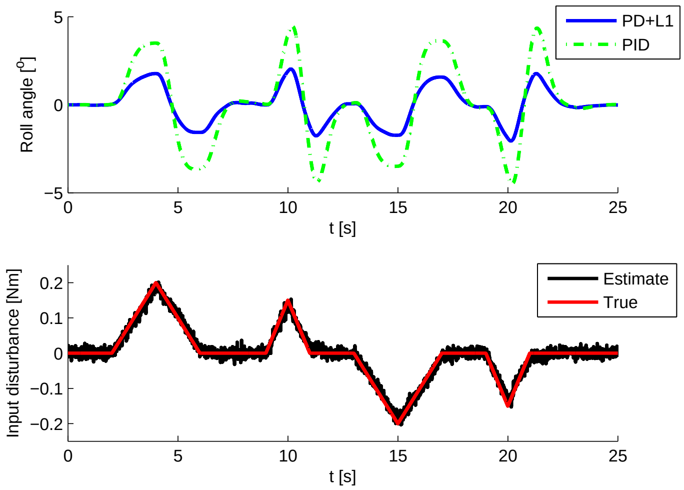


Figure 6.23: *Simulation result for the roll angle controllers using triangle-shaped input disturbances. The lower plot shows how the low pass filtered value of $\hat{\sigma}_1$ in the $\mathcal{L}_1$ adaptive controller estimates the disturbance.*

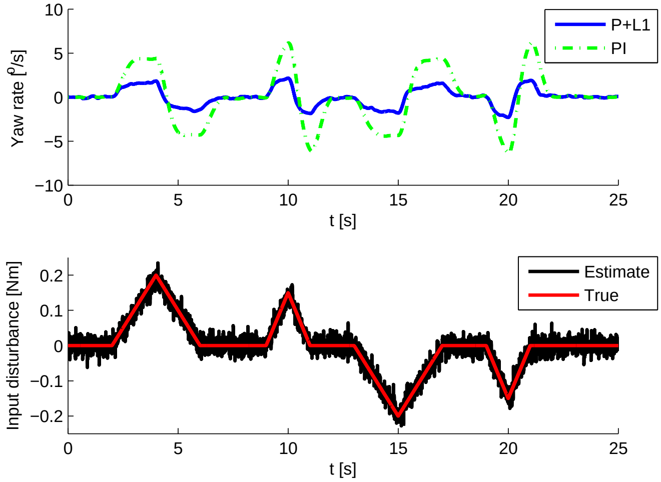


Figure 6.24: *Simulation result for the yaw rate controllers using triangle-shaped input disturbances. The lower plot shows how the low pass filtered value of $\hat{\sigma}_1$ in the $\mathcal{L}_1$ adaptive controller estimates the disturbance.*

Again, the disturbance rejection performance of the $\mathcal{L}_1$ adaptive controller is better compared to the standard controller architectures.

6.4.3 Robustness against modeling errors

An important aspect to consider is how robust the different controller structures are against different types of modeling errors. For the octorotor model, the main model uncertainties are

- the moment of inertia
- the time constants of the motor dynamics
- the thrust and torque constants.

To compare the robustness of the two controllers the nominal model from Chapter 4 was perturbed with different modeling errors and a series of steps were used as the input reference signal to the controller. The resulting reference tracking is compared with the one obtained when the nominal model is used. First, only a single type of modeling error was introduced to identify the effect that the particular error has on the reference tracking performance. The results for the roll angle and yaw rate controllers are shown in Figure 6.25 to 6.30. These figures show one of the main strengths of the $\mathcal{L}_1$ augmentation, strong robustness against modeling errors. While the PID and PI controllers are more or less affected by the different types of modeling errors, the controllers utilizing the $\mathcal{L}_1$ augmentation are almost unaffected. The roll angle PID controller is most sensitive to errors in the thrust constant c_T while the PI yaw rate controller is most sensitive to errors in the moment of inertia I_{zz}.

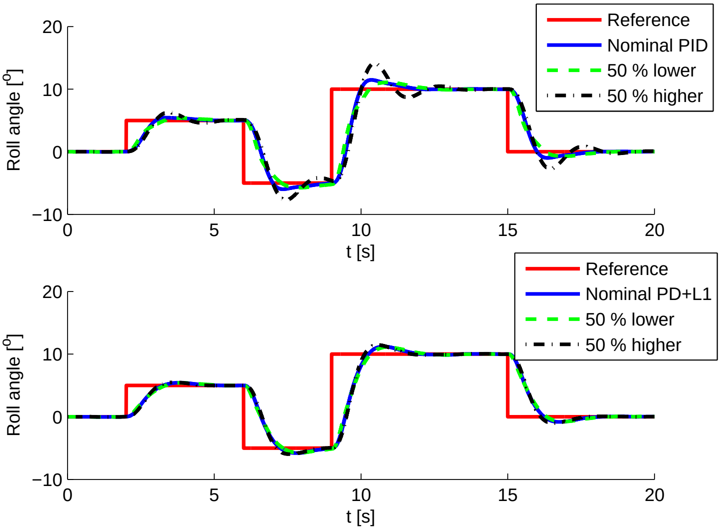


Figure 6.25: *Simulation result for the roll angle controllers using a model error of ± 50 % for the moment of inertia I_{xx}. The upper plot corresponds to the PID controller and the lower plot to the PD + $\mathcal{L}_1$ controller.*

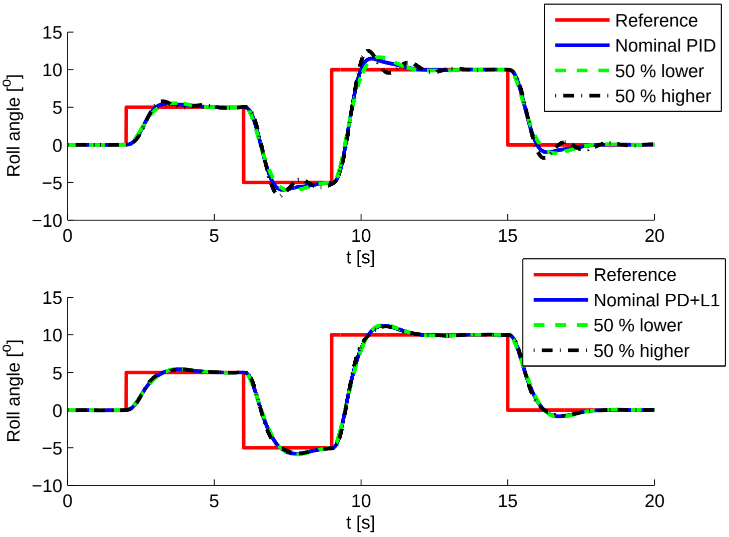


Figure 6.26: *Simulation result for the roll angle controllers using a model error of ± 50 % for T_{pos} and T_{neg} in the motor dynamics model. The upper plot corresponds to the PID controller and the lower plot to the PD + $\mathcal{L}_1$ controller.*

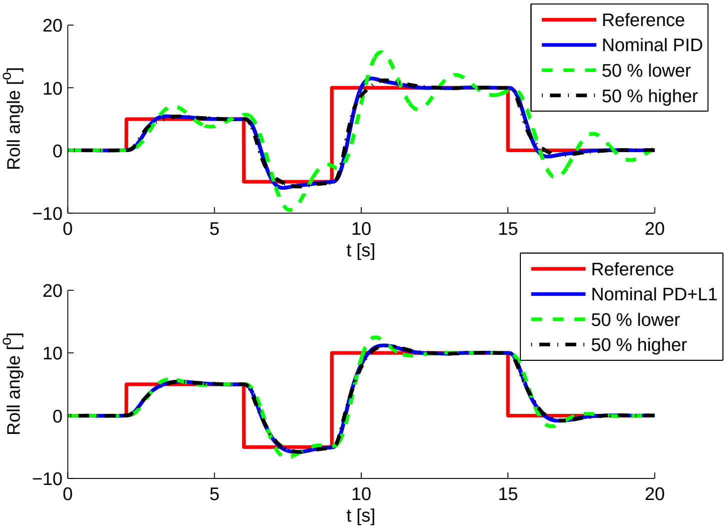


Figure 6.27: *Simulation result for the roll angle controllers using a model error of ± 50 % for the thrust constant c_T. The upper plot corresponds to the PID controller and the lower plot to the PD + $\mathcal{L}_1$ controller.*

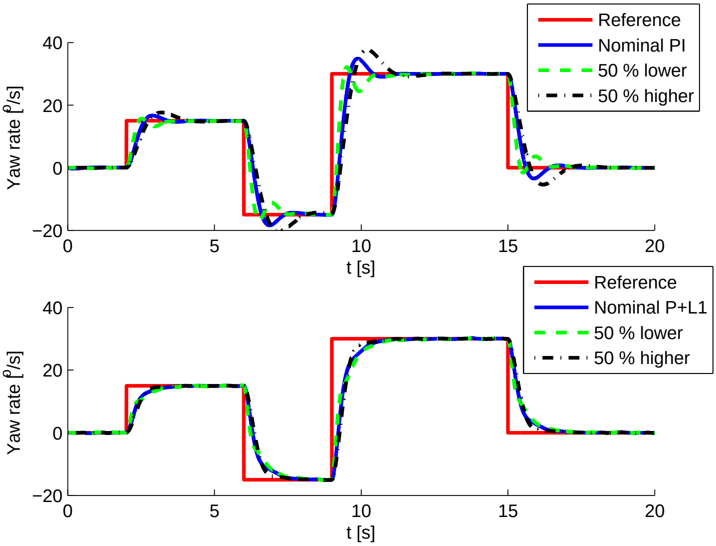


Figure 6.28: *Simulation result for the yaw rate controllers using a model error of ± 50 % for the moment of inertia I_{zz}. The upper plot corresponds to the PI controller and the lower plot to the P + $\mathcal{L}_1$ controller.*

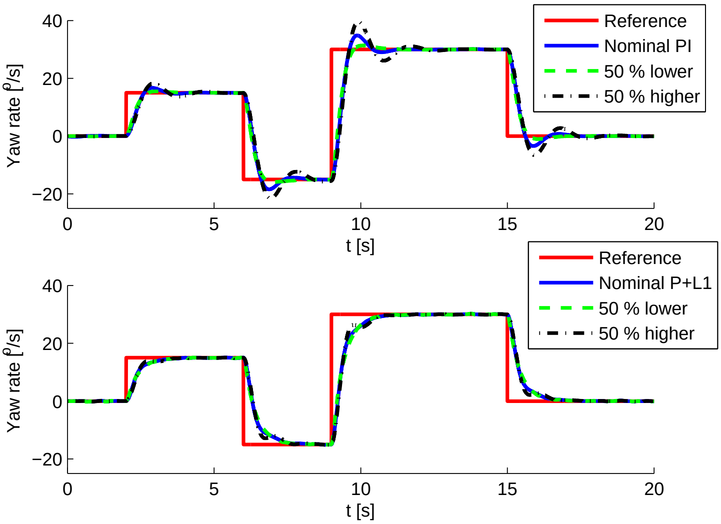


Figure 6.29: *Simulation result for the yaw rate controllers using a model error of ± 50 % for T_{pos} and T_{neg} in the motor dynamics model. The upper plot corresponds to the PI controller and the lower plot to the $P + \mathcal{L}_1$ controller.*

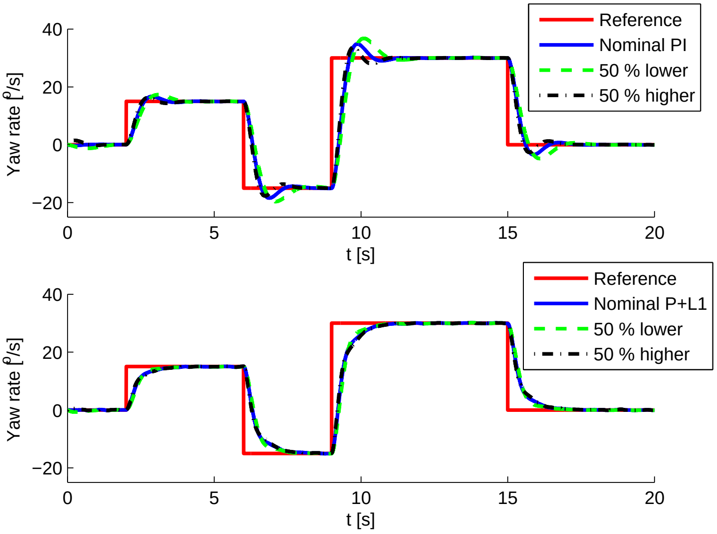


Figure 6.30: *Simulation result for the yaw rate controllers using a model error of ± 50 % for the torque constant c_Q. The upper plot corresponds to the PI controller and the lower plot to the $P + \mathcal{L}_1$ controller.*

A more realistic scenario is that there are more than one type of modeling error. From Figure 6.25 to 6.30 a worst case was identified where the moment of inertia and motor time constants are increased and the thrust and torque constants are decreased. I_{xx}, I_{zz} and T_{pos} were increased by 20 %. T_{neg} was increased by 30 % to model the lesser knowledge of this constant. The thrust and torque constants were decreased by 20 %. The results from these simulations are shown in Figure 6.31 and 6.32.

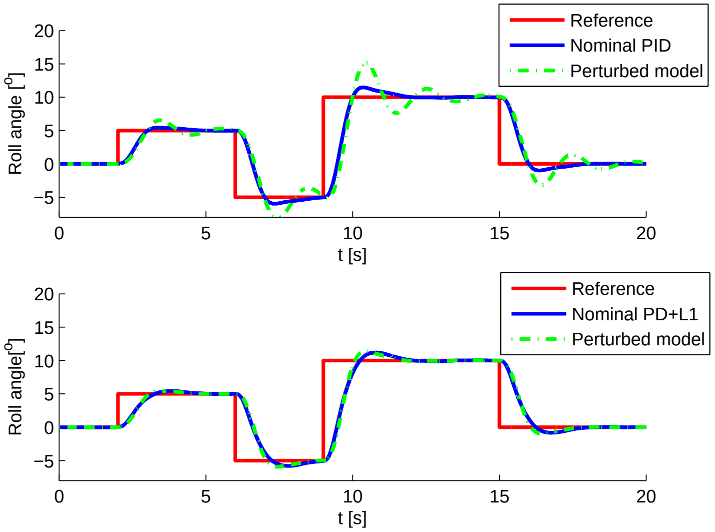


Figure 6.31: *Simulation result for the roll angle controllers using combined model perturbations that were found to create a challenging combination. The upper plot corresponds to the PID controller and the lower plot to the PD +* $\mathcal{L}_1$ *controller.*

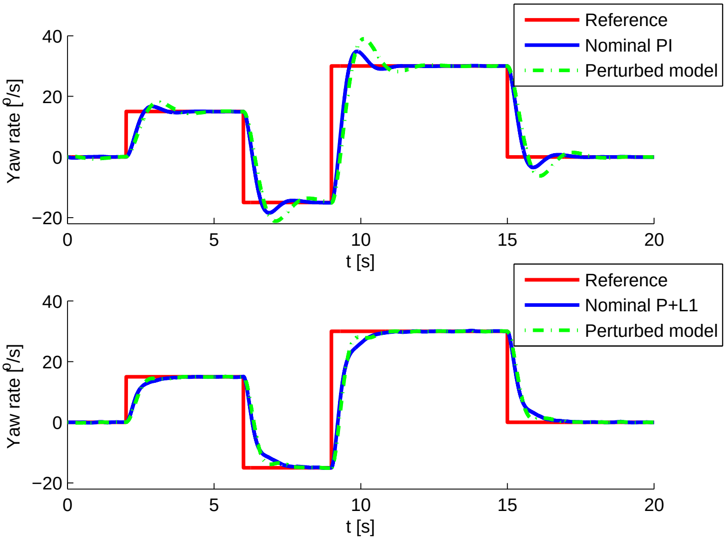


Figure 6.32: *Simulation result for the yaw rate controllers using combined model perturbations that were found to create a challenging combination. The upper plot corresponds to the PI controller and the lower plot to the P + $\mathcal{L}_1$ controller.*

7 Conclusions and future work

The results obtained in this thesis are promising and the main purposes and goals of the thesis are met. This chapter summarizes the results and presents some ideas regarding how the project can be continued in the future.

7.1 Conclusions

An extensive description of the dynamic model of the octorotor is included in this thesis. The model's parameters have been identified using several identification experiments with good results. These experiments are thoroughly documented and can be applied to any multirotor platform of similar scale. A novel approach for identifying the motor dynamics using only a microphone and signal processing software has been presented. This method is inexpensive and removes the need for more advanced measuring equipment. The time constant of the motors is in the order of hundredths of milliseconds and is hence not negligible. In fact, the motors' time constant limits the attainable performance of the PID controller structure presented in this thesis. The resulting model is implemented in Simulink and used for controller design and comparison.

Two different attitude controllers have been designed and compared using the developed model of the octorotor. The design process is general and can be applied to similar types of systems, particularly other types of multirotors. Results from simulations show that the proposed $\mathcal{L}_1$ adaptive control augmentation gives a notable performance increase compared to a PID architecture in terms of input disturbance rejection and robustness against modeling errors. This suggests that $\mathcal{L}_1$ adaptive control is a suitable control structure for UAV:s that are affected by varying uncertanties and input disturbances. However, the $\mathcal{L}_1$ adaptive controller is more complex to implement and gives less noise attenuation compared to the PID controller.

A state estimation algorithm that can be used for attitude estimation has been designed. The algorithm has been implemented and run on the target hardware using a fixed point number representation. The fixed point implementation enables the algorithm to be run at a high sample rate on hardware without floating point support. This trait makes it suitable for use in combination with the proposed $\mathcal{L}_1$ adaptive controller since this controller performs better when the sample rate is increased. Initial filter tests have been performed with promising results but more extensive testing is needed to get an accurate estimate of the filter performance.

The new flight and navigation electronics can now be used together with the Mikrokopter platform and it is easy to modify and extend the present source code with new control and estimation algorithms. The PID controller has been implemented on the platform and initial flight tests have been performed.

7.2 Future work

The developed model of the octorotor does not include aerodynamic effects such as rotor flapping and induced drag. To get an improved model, more research and parameter identification can be performed such that these effects and other aerodynamic disturbances can be included. In addition, this will make it possible to include the proposed state estimation algorithm in the simulation environment since it is based on these effects.

The $\mathcal{L}_1$ adaptive control augmentation to the PD controller should be implemented on the target hardware to get experimental verification of the simulation results. The support for fixed point arithmetic can be used when implementing the control augmentation so that the $\mathcal{L}_1$ sample time can be set to a low value for increased performance. This controller can then be used together with a position controller to make the octorotor attain and hold a desired position in the airspace.

The implemented state estimation algorithm can only be used for attitude estimation and is not evaluated against ground truth data. In order to get a complete performance evaluation either an accurate motion capture system or a proven attitude filter could be used for comparison. To further develop the control of the platform and include a position controller and trajectory planner, a more complete navigation system is needed. A natural choice is an EKF based on a simplified system model. This also requires that all onboard sensors such as the GPS-unit, magnetometer and barometer are integrated with the estimation software.

The hardware and software from the UAVP-NG project have very limited logging capabilities since it is not possible to store data on the platform during flight. A solution to this problem is to switch to a more capable hardware platform such as e.g. Raspberry Pi (see [26]). This will simplify development since a full-scale operating system can be run on this type of hardware and it is possible to integrate new sensors and other measuring equipment. However, this requires a significant amount of initial work before actual controller development can begin.

Bibliography

[1] V.M. Alfaro, R. Vilanova, and O. Arrieta. Considerations on Set-Point Weight choice for 2-DoF PID Controllers. In *7th IFAC International Symposium on Advanced Control of Chemical Processes*, Istanbul, Turkey, 2009. Cited on pages 24, 25, and 80.

[2] L.K. Baxter. *Capacitive Sensors: Design and Applications*. New York, NY: Institute of Electrical and Electronics Engineers, Inc., 1997. ISBN 0780311302. Cited on page 32.

[3] Biltema Nordic Services AB. Varvtalsräknare, laser. `http://www.biltema.se/sv/Bil---MC/Verktyg-och-Verkstadsutrustning/Testinstrument-och-Elektronik/Varvtalsraknare-laser-15294/`, March 2014. Cited on page 45.

[4] T. Bresciani. Modelling, Identification and Control of a Quadrotor Helicopter. Master's thesis, Lund University, Lund, Sweden, 2008. Cited on pages 6 and 15.

[5] C. Chéron, A. Dennis, V. Semerjyan, and C. YangQuan. A Multifunctional HIL Testbed for Multirotor VTOL UAV Actuator. In *International Conference on Mechatronic and Embedded Systems and Applications*, Qingdao, ShanDong, China, 2010. Cited on pages 44, 45, and 52.

[6] H. Corluka and H. Ohlsson. Modellering och reglering av en oktakopter. Master's thesis, Lund University, Lund, Sweden, 2014. Cited on page 2.

[7] E. Fresk and G. Nikolakopoulos. Full Quaternion Based Attitude Control for a Quadrotor. In *2013 European Control Conference (ECC)*, Zurich, Switzerland, 2013. Cited on pages 18, 19, and 21.

[8] H. Goldstein. *Classical Mechanics*. Reading, MA: Addison-Wesley Publishing Company Inc., 2nd edition, 1980. ISBN 0201029693. Cited on pages 9, 13, 15, 42, and 56.

[9] Graupner/SJ GmbH. MC-32 PRO HoTT 2.4 GHz RC-unit. `http://www.graupner.de/en/products/`

1736df13-32af-4183-aa8e-80f31a7f03cb/33032.PRO.EN/product.aspx, March 2014. Cited on page 33.

[10] F. Gustafsson. *Statistical Sensor Fusion*. Lund: Studentlitteratur AB, 2nd edition, 2012. ISBN 9789144077321. Cited on pages 9, 18, and 63.

[11] F. Gustafsson, M. Millnert, and L. Ljung. *Signal Processing*. Lund: Studentlitteratur AB, 1st edition, 2010. ISBN 9789144058351. Cited on pages 16 and 17.

[12] HiSystems GmbH. Brushless Controller version 2.0. http://www.mikrokopter.de/ucwiki/en/BL-Ctrl_2.0, March 2014. Cited on pages 33 and 44.

[13] HiSystems GmbH. Company web page. http://www.mikrokopter.de, March 2014. Cited on page 29.

[14] HiSystems GmbH. MK3638 Brushless motor. http://www.mikrokopter.de/ucwiki/MK3538, March 2014. Cited on page 32.

[15] N.V. Hoffer, C. Coopmans, A.M. Jensen, and Y. Chen. A Survey and Categorization of Small Low-Cost Unmanned Aerial Vehicle System Identification. *Journal of Intelligent and Robotic Systems*, 74(1-2):129–145, 2014. Cited on page 2.

[16] F. Hoffmann, N. Goddemeier, and T. Bertram. Attitude Estimation and Control of a Quadrocopter. In *International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010. Cited on page 45.

[17] N. Hovakimyan and C. Cao. *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010. ISBN 9780898717044. Cited on pages 25, 26, and 27.

[18] M. Hua, T. Hamel, P. Morin, and C. Samson. Introduction to Feedback Control of Underactuated VTOL Vehicles: A Review of Basic Control Design Ideas and Principles. *IEEE Control Systems Magazine*, 33(1):61 – 75, 2013. Cited on pages 1 and 3.

[19] H. Lim, J. Park, D. Lee, and H.J. Kim. Build Your Own Quadrotor: Open-Source Projects on Unmanned Aerial Vehicles. *IEEE Robotics and Automation Magazine*, 19(3):33 – 45, 2012. Cited on pages 1, 3, 17, 23, 66, 73, and 80.

[20] R. Mahony, T. Hamel, and J. Pflimlin. Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control*, 53 (5):1203 – 1218, 2008. Cited on pages 17, 19, 66, 67, and 68.

[21] R. Mahony, V. Kumar, and P. Corke. Multirotor Aerial Vehicles: Modeling, Estimation and Control of Quadrotor. *IEEE Robotics and Automation Mag-*

azine, 19(3):20–32, 2012. Cited on pages 2, 3, 31, 35, 40, 43, 63, 64, 66, 67, 74, and 75.

[22] S. Mallikarjunan, B. Nesbitt, E. Kharisov, E. Xargay, N. Hovakimyan, and C. Cao. L1 Adaptive Controller for Attitude Control of Multirotors. In *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, 2012. Cited on pages 3, 25, and 93.

[23] P. Martin and E. Salaün. The True Role of Accelerometer Feedback in Quadrotor Control. In *IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010. Cited on pages 43 and 64.

[24] Next Generation Universal Aerial Video Platform. Project web page. `http://ng.uavp.ch/`, March 2014. Cited on pages 31 and 32.

[25] A. Pettersson. *Augmenting L1 Adaptive Control of Piecewise Constant Type to Aerial Vehicles*. Licentiate thesis, Lund University, Lund, Sweden, 2013. Cited on pages 2, 93, and 94.

[26] Raspberry Pi Foundation. Project main page. `http://www.raspberrypi.org`, May 2014. Cited on page 113.

[27] F. Rinaldi, A. Gargioli, and F. Quagliotti. PID and LQ Regulation of a Multirotor Attitude: Mathematical Modelling, Simulations and Experimental Results. *Journal of Intelligent and Robotic Systems*, 73(1-4):33 – 50, 2014. Cited on page 3.

[28] Glad T. and Ljung L. *Reglerteknik: Grundläggande teori*. Lund: Studentlitteratur AB, 4th edition, 2006. ISBN 9144022751. Cited on pages 21, 22, 23, 24, 75, and 82.

[29] D. H. Titterton and J. L. Weston. *Strapdown Inertial Navigation Technology*. London: Peter Peregrinus Ltd, 1st edition, 1997. ISBN 0863412602. Cited on pages 6, 8, 20, 21, 34, and 35.

Linköpings universitet