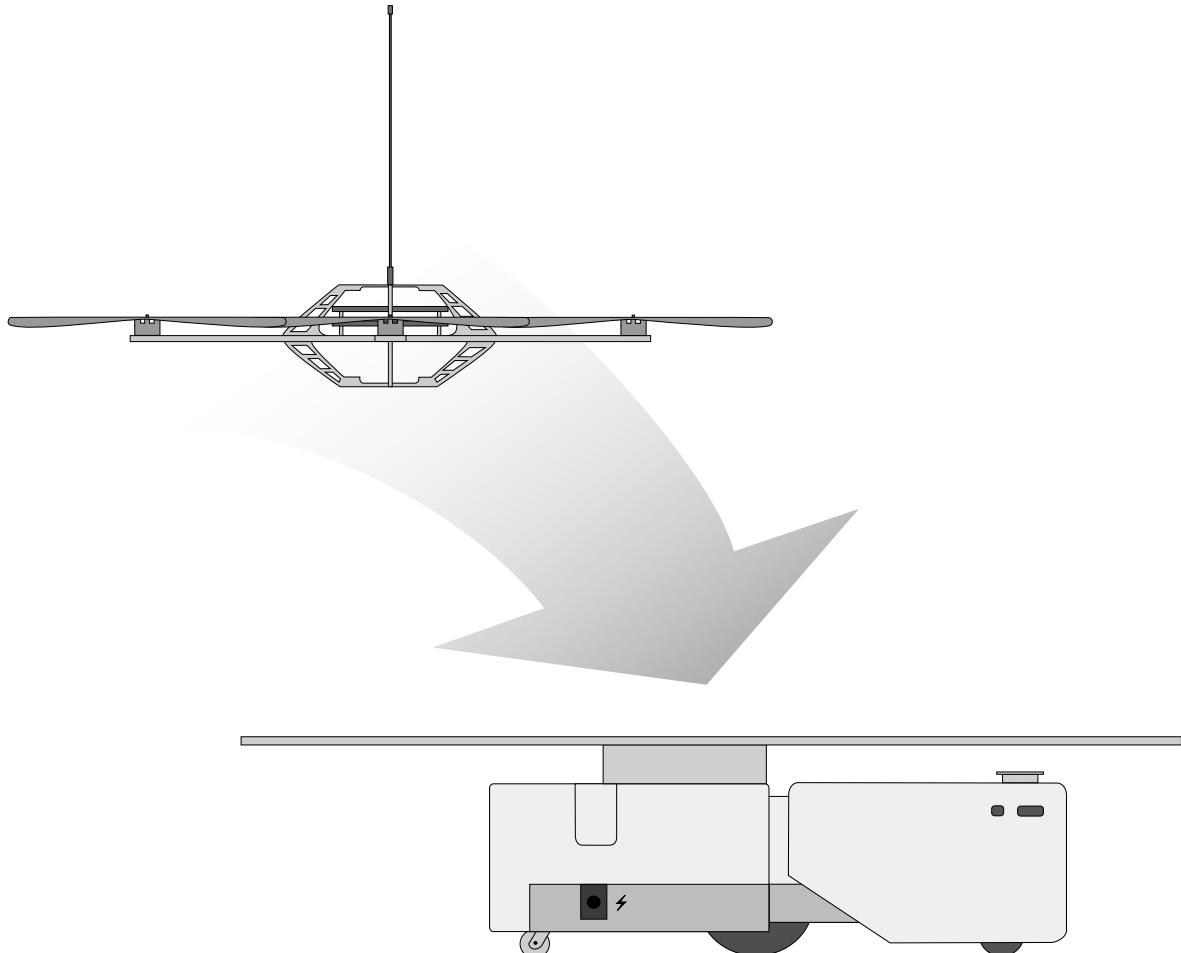


Autonomous Landing on a Moving Platform

8th SEMESTER PROJECT, AALBORG UNIVERSITY 2009

THE DEPARTMENT OF ELECTRONIC SYSTEMS



Group 830

Johannes Friis

Ebbe Nielsen

Rasmus Foldager Andersen

Jesper Bønding

Anders Jochumsen

Anders Friis Sørensen

Title:

Autonomous Landing on a Moving Platform

Theme:

Modelling and control

Project period:

Spring 2009

Project group:

09gr830

Group members:

Johannes Friis
Ebbe Nielsen
Rasmus Foldager Andersen
Jesper Bønding
Anders Jochumsen
Anders Friis Sørensen

Supervisors:

Assistant Professor Morten Bisgaard
Associate Professor Anders la Cour-Harbo

Publications: 9

Pages: 96 (126 including appendix)

Finished: June 2nd, 2009

Synopsis:

The purpose of this project is to make a quad rotor helicopter (X-3D-BL) follow, and land on, a moving platform (iRobot). A Vicon MX system is used to measure positions and orientations.

In order to understand the physical dynamics a model of the X-3D-BL is developed. The model is made as a first principles model where internal attitude controller is modelled using system identification. The data used for estimating the attitude controller is collected doing manual flights. The models is combined into a linear state space model, suited for controller design.

Two controllers are designed for the system, a hand-tuned PID and a model based LQ controller. The designed controllers are simulated and tested doing hover, steps, and tracking scenarios, and performs satisfactory.

Tracking of a randomly moving object is achieved (with deviations of approx 10 cm), though a proper landing algorithm has not been implemented. It can be concluded under the giving conditions it is possible to follow a platform, but it has not yet been proven that it is possible to land with the developed controllers.

Preface

This report concerns the modelling and control of a X-3D-BL quad rotor, making it able to track, and land on, a moving platform. The project is composed during the period from February 1st to June 2nd 2009 at Aalborg University by six 8th semester students in the Section for Automation and Control at the Department of Electronic Systems.

Group 830, Aalborg University

Johannes Friis

Ebbe Nielsen

Rasmus Foldager Andersen

Jesper Bønding

Anders Jochumsen

Anders Friis Sørensen

Contents

1	Introductory	1
1.1	Introduction	1
1.2	Problem Statement	2
1.3	Motion Tracking Lab (MTLab)	2
I	Modelling of a X-3D-BL Quad Rotor	5
2	Modelling Introduction	9
2.1	Concept and Generalities	9
3	First Principles Model	13
3.1	Model Structure	13
3.2	Rigid Body Dynamics and Kinematics	14
3.3	Forces and Torques	16
3.4	Revised Model Structure	18
3.5	Internal Controller and Motor Dynamics	19
4	System Identification and Parameter Estimation	20
4.1	Model Structure and Estimation Method	20
5	Additional Non-linearities	23
5.1	Battery Discharge	23
5.2	Measurement Noise	24
6	Linearisation of Model	25
6.1	Linearisation Method	25
6.2	Operating Point	25
6.3	Linearisation of Body Dynamics	26
6.4	Linearisation of Forces	27
6.5	Linearised Equations of Motion	28
7	State Space Model	29
7.1	Discrete state space model	31
8	Model Verification	34

8.1 Validation of Estimated Transfer Functions	34
8.2 Validation of Discrete State Space Model	37
8.3 Overall Model Evaluation	41
II Controlling X-3D-BL Quad Rotor	43
9 Introduction to Controllers	47
10 PID Controller	48
10.1 Designing the PID Controller	48
10.2 Implementation	49
11 Linear Quadratic Controller	51
11.1 The Discrete State Space Controller	51
11.2 Discrete Time LQ Control	52
11.3 Full Order Observer	55
11.4 Introducing Reference	58
11.5 Integral Action	61
11.6 Gain Sheding as Yaw Compensation	63
11.7 Simulations	64
12 Reference Tracking	68
12.1 Tracking	68
13 Linear Controller Analysis	71
14 Evaluation of Controllers	74
14.1 Hover Evaluation	74
14.2 Step Input Evaluation	80
14.3 Sinusoidal Input Evaluation	83
14.4 Tracking Evaluation	85
14.5 Summary of Controller Evaluation	88
III Epilogue	89
15 Conclusion	93
16 Future Work	94
16.1 Implementing Take-off and Landing Algorithms	94
16.2 Predicting Platform Movement	94
16.3 Implementing Noise Model	94
16.4 Onboard Micro Controller and Sensors	95
Bibliography	96
IV Appendices	97
A Measurements Journals	98

A.1 Data for Parameter Estimation	98
A.2 Polynomial Relation of S_{col} and the Thrust Force	102
A.3 Estimation of Induced Inflow Emulation Coefficient	105
A.4 Regulator Performance Tests	107
A.5 Estimation of Battery Discharge Coefficients	120
B Description of X-3D-BL	122
C CD Contents	126

Nomenclature

Θ	Roll, pitch and yaw Euler angles
A	System matrix continuous time
B	Input matrix continuous time
C	Output matrix continuous time
F_{1-4}	Force generated by the respective rotors
F_{body}	Total force affecting the quad roter in body frame
F_{rot}	Force generated by the rotors
F_{tot}	Resulting force acting on the body
H	Output matrix discrete time
H_r	Reference model output matrix discrete time
J	Inertia
K_i	Feedback matrix of integration
K_r	Reference compensation
K_s	Feedback matrix that minimises the performance function
L	Observer feedback
Q_1	Weight matrix of the states
Q_2	Weight matrix of the inputs
Q_N	Weight matrix of the last input value
u	Input vector
x	State vector
x_r	Reference state vector
y	Output vector

χ	Phase shift
Γ	Input matrix discrete time
λ	Constant used to hand tune the velocity compensated reference
\mathcal{H}	Abbreviation of the elements in the sum from \mathcal{I}
\mathcal{I}	LQR cost function
Ω	Angular velocity of rotors
Φ	System matrix discrete time
ϕ, θ, ψ	Roll, pitch and yaw Euler angles
ρ	Weight factor determining trade off between performance and control effort
τ_{drag}	Torque generated by rotors
τ_{gyro}	Torque caused by gyroscopic effects
τ_{tot}	Total force acting on the body
${}^b\mathbf{C}_e$	Direct cosine transformation from earth to body frame
${}^b\mathbf{F}_g$	Gravitational force acting on the body
${}^b\mathbf{H}_e$	Transformation of rates and angles from earth to body frame
${}^e\mathbf{C}_b$	Direct cosine transformation from body to earth frame
${}^e\mathbf{H}_b$	Transformation of rates and angles from body to earth frame
g	Gravitational acceleration
I_f	Induced inflow coefficient
J_r	Inertia of rotor blade
l	Length from center of mass of quad rotor to driving shaft of rotors
m	Mass of quad rotor
P	Vector of the earth frame references x,y,z
S_ω	Vector containing S_ϕ, S_θ, S_ψ
S_ϕ	Roll input to the onboard controller
S_ψ	Yaw input to the onboard controller
S_θ	Pitch input to the onboard controller
S_{col}	Collective input to the onboard controller
x, y, z	Position in earth frame
${}^b\boldsymbol{\omega}$	Body rates
${}^b\mathbf{v}$	Translatory velocity in body frame
${}^e\boldsymbol{\omega}$	Euler rates
${}^e\mathbf{v}$	Translatory velocity of body frame in earth frame

Chapter 1

Introductory

This chapter will introduce the reader to the project, which this report concerns. The first section is a general introduction to the problem, quad rotors and other research done in this field. The following section contains the problem statement, and aims necessary to fulfil the stated problem.

1.1 Introduction

This report concerns the subject of autonomous landing of an Uninhabited Aerial Systems (UAS) on a moving platform. An analogy to this, can be landing an aircraft on a moving vehicle, such as a scout aircraft taking off from an ambulance and flying ahead, spotting traffic conditions, or take-off and landing on a ship, in high seas.

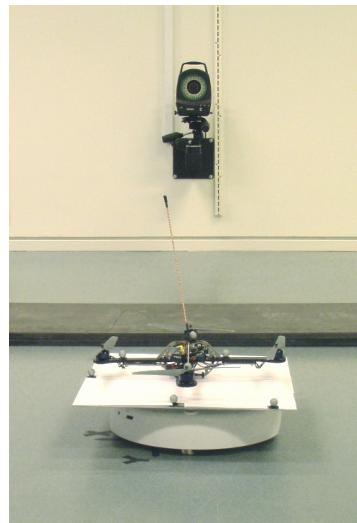


Figure 1.1: The X-3D-BL quad rotor on top of the moving platform. In the background one of the motion tracking cameras of the Vicon system is seen

A quad rotor helicopter differs from a conventional helicopter in the sense, that it has four rotors, as opposed to the conventional two. The quad rotor used in this project can be seen on top of the moving platform on Figure

1.1. The four rotors of the quad rotor are mounted in a X, with thrust of all rotors in an upwards direction. On a conventional helicopter the main rotor has upwards thrust, and the tail rotor is used for compensation and yaw. A quad rotor controls directionality and yaw, by increasing or decreasing the thrust on the rotors in pairs. The angular velocity of the rotors is the only variable controlling the quad rotor attitude.

The UAS used in the project is the commercial available X-3D-BL developed by Ascending Technologies. This quad rotor is used for research and by hobbyists around the world. This platform has been chosen, because of its availability, use in other research, and documentation. During the project it is chosen to let the quad rotor fly on batteries as opposed to connecting it to a stationary power supply. For details regarding the physical attributes of the X-3D-BL see Appendix B.

In the field of autonomous flight, several studies have been made, both at Aalborg University, and in other universities. Some contributors to the autonomous quad rotor field of studies are MIT and Stanford University. Many project have concerned the achievement of stable hover of a quad rotor. The MIT research concerns autonomous hovering and following a moving object [1, p.51-64]. The Stanford University STARMAC project concerns stable hover, trajectory tracking and collision avoidance [2]. A combination of the two will be used as primary sources of inspiration for this project.

1.2 Problem Statement

The scope of this project is to develop controllers for the X-3D-BL quad rotor. Enabling it to track, and land on, a moving platform. In order to develop an overall control strategy, several controller problems must be solved. Firstly stable hover must be achieved, additionally the quad rotor must be able to track a moving object. This is achieved by implementing controllers for tracking position and then, if necessary, make modifications of these controllers to obtain satisfactory tracking of a trajectory of unknown movements (i.e: the moving platform).

The above stated, is formulated in to the following problem statement for the project:

Is it possible to design and implement (various) controllers, enabling a X-3D-BL quad rotor to autonomously track and land on a flat platform with unknown movement in the horizontal plane?

In order to solve the above stated problem, several partial aims must be achieved:

1. Achieve stable hover
2. Enable the X-3D-BL to move to a way-point (x,y,z)
3. Follow a continuous trajectory
4. Enable the X-3D-BL to track a moving object on the ground, which movement are not known a priori
5. Implement landing and take-off algorithms

1.3 Motion Tracking Lab (MTLab)

All in-flight measurements and experiments are conducted in the Motion Tracking Lab (MTLab) at Aalborg University.

The MTLab is a 5x6 m room with 8 cameras mounted on the walls and connected to a Vicon MX system. The cameras are able to track small markers in real-time. When an object is equipped with the markers the Vicon MX system can calculate the position and orientation of the object. The information can be send to any computer with Matlab through a TCP/IP connection. The Vicon MX system is, with the current configuration, capable of sending the position and orientation of objects with an update frequency of 100 Hz.

The MTLab is mainly used for experiments with UAS'. The Vicon MX system is used for measuring the position and orientation of the UAS while a Futaba T9CAP transmitter is used for controlling the UAS. The transmitter is connected to a PC through a training cable. This makes it possible to send out RC-commands directly from the interface PC.

A MFTech USB dongle attached to Schulze receiver receives all transmitted commands send by the Futaba transmitter. The interface PC is in this way capable of sending commands to the UAS and receiving all commands send by a human pilot.

All the information in the interface PC is available to any control PC connected to the network. On the control PC the information is accessed by using Matlab. The Vicon information is available through a Simulink block called Vicon. The transmitter information is available through a Simulink block called ConLink. An overview of the MTLab are shown in Figure 1.2.

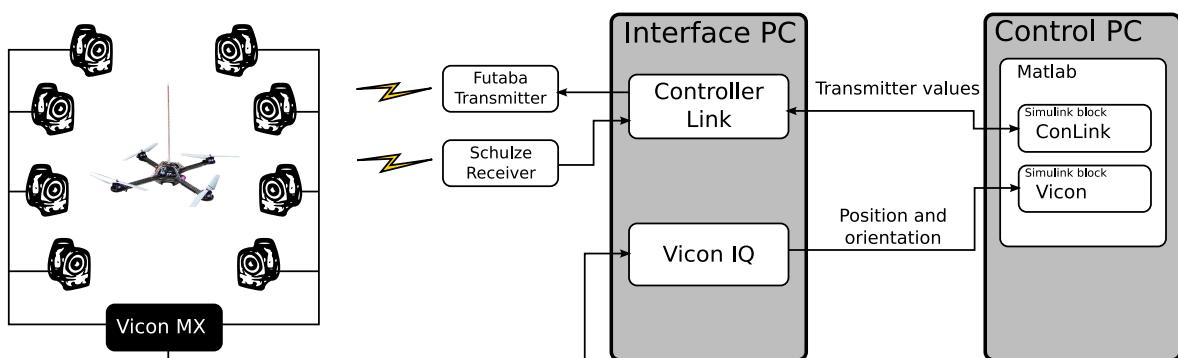


Figure 1.2: Overview of the MTLab

Data units

The data received and send from the interface PC are all in predetermined units. Some of the units are determined by the MTLab some are chosen for this project. An overview of the units and boundaries of the data are listed in Table 1.1.

1.3.1 Noise in MTLab

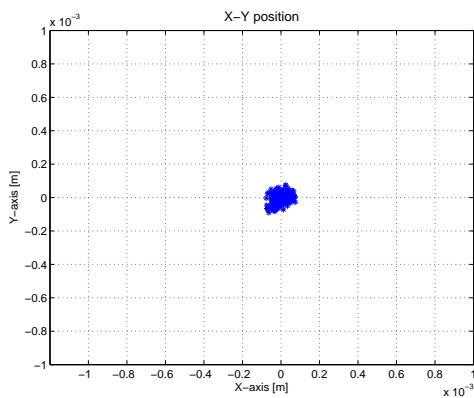
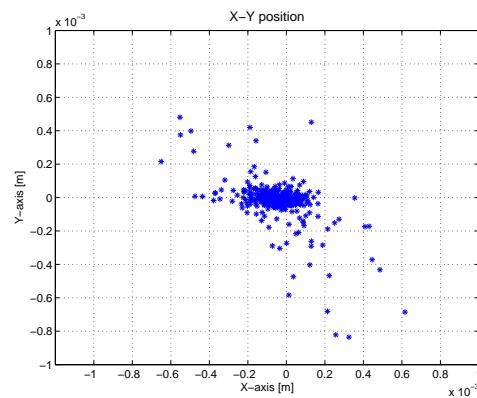
To obtain knowledge of the noise in the Vicon MX system and the precision of the measurements the position and orientation of the X-3D-BL has been measured in different states. With the motors off to examine the precision of a non moving object and the noise of the Futaba transmitter combined with the Schulze receiver. Then with the motors on to estimate the amount of noise the motors introduce as well as to confirm that all markers are visible for the cameras.

Data	Unit	Boundary
Position x	m	± 1.5
Position y	m	± 1.5
Position z	m	0 to 2
Orientation	rad	$\pm \pi$
Input:		
- roll, pitch, yaw	-	± 100 (100 is max)
- collective	-	± 100 (100 is min)

Table 1.1: Data units and boundaries

Precision of x,y Position

The position of the X-3D-BL have been measured by the Vicon MX system both with motors on and off. This is done with the quad rotor standing on the ground with the motors turned off, and with the motors turned on spinning in idle speed. The result of the measurements are shown in Figure 1.3 and 1.4.

**Figure 1.3:** Position measured by the Vicon system (motors off)**Figure 1.4:** Position measured by the Vicon system (motors on)

When the motors are off, the precision of the position measurements are within $\pm 10^{-4}$ m = ± 0.1 mm. When the motors are turned on, the position is no longer as steady. The cause of the more scattered measurements is difficult to determine. It can be the X-3D-BL moving, but this is very unlikely because it is equipped with rubber feet that prevents slipping. More likely it is either the vibrations in the frame or the rotation of the rotors that slightly moves the markers. No matter the cause the precision of the X-3D-BL are still more than adequate with a precision of approximately ± 1 mm, when the motors are on.

Part I

Modelling of a X-3D-BL Quad Rotor

Table of Contents

2 Modelling Introduction	9
2.1 Concept and Generalities	9
3 First Principles Model	13
3.1 Model Structure	13
3.2 Rigid Body Dynamics and Kinematics	14
3.3 Forces and Torques	16
3.4 Revised Model Structure	18
3.5 Internal Controller and Motor Dynamics	19
4 System Identification and Parameter Estimation	20
4.1 Model Structure and Estimation Method	20
5 Additional Non-linearities	23
5.1 Battery Discharge	23
5.2 Measurement Noise	24
6 Linearisation of Model	25
6.1 Linearisation Method	25
6.2 Operating Point	25
6.3 Linearisation of Body Dynamics	26
6.4 Linearisation of Forces	27
6.5 Linearised Equations of Motion	28
7 State Space Model	29
7.1 Discrete state space model	31
8 Model Verification	34
8.1 Validation of Estimated Transfer Functions	34
8.2 Validation of Discrete State Space Model	37
8.3 Overall Model Evaluation	41

Chapter 2

Modelling Introduction

This part concerns the modelling of the X-3D-BL quad rotor. A model of the quad rotor is necessary in order to develop a model based controller, and gain understanding of the physical principles concerning the quad rotor. The first section introduces a few concepts needed to understand the modelling. These includes the definition of the coordinate systems used and a description of how a quad rotor manoeuvres. It also contains a list of assumptions to simplify the modelling. After this a first principles model is created using rigid body dynamics and kinematics and by determining the forces and torques acting on the quad rotor.

This first principles model is then expanded by introducing transfer functions, which describes the relation from the TC-transmitter to angular velocities of the quad rotor. The parameters for these are determined, using parameter estimation. The modelling strategy using system identification is further elaborated in section 3.1. Additional non-linearities are then added to describe the discharging of the battery.

The finalised model is then linearised in a set of operation points and put on state space form. The state space model is then discretised to be used as part of a discrete time model based controller.

2.1 Concept and Generalities

In the following section general concepts used in the modelling of the X-3D-BL are explained.

2.1.1 Coordinate Systems

Figure 2.1 shows the two right hand coordinate systems used in the project. The left most coordinate system in Figure 2.1, which is denoted E, is the earth frame. Vectors seen from the earth frame will be denoted with an e in front of the variable. Vectors seen from the body frame will be denoted with a b . The earth frame is seen as the reference coordinate system where the xy-plane is aligned with the floor and the z-axis pointing down. The body frame, which is denoted B in Figure 2.1, is aligned with the body of the quad rotor. This means the centre of mass of the quad rotor is aligned with origo and the front rotor (rotor 1) is pointing in the positive x-axis.

Furthermore the z-axis of the body frame is placed as in Figure 2.2. On the Figure it can be seen that the origo of the frame is placed at the same height of the rotor blades. The body frame exists within the earth frame. The

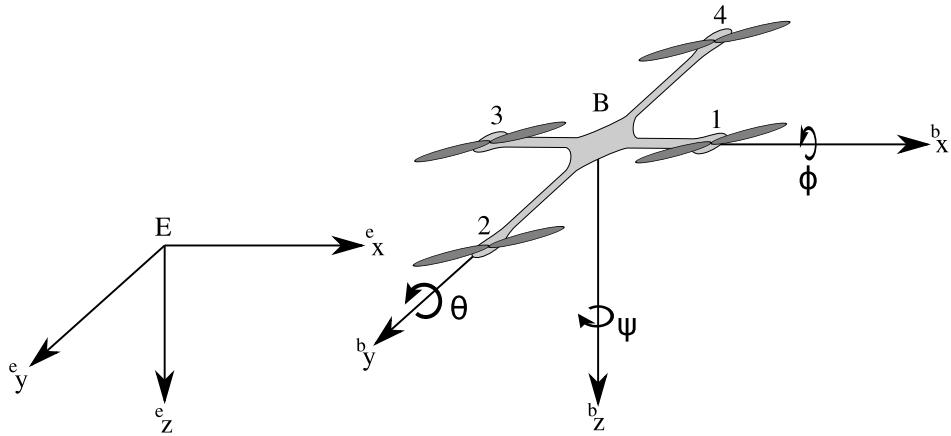


Figure 2.1: The coordinate systems as defined in this project

position of the body frame is denoted $\mathbf{P} = [x, y, z]^T$, the orientation as the 3-2-1 Euler $\Theta = [\phi, \theta, \psi]^T$ and the angular velocity of \mathbf{B} ${}^B\omega$.

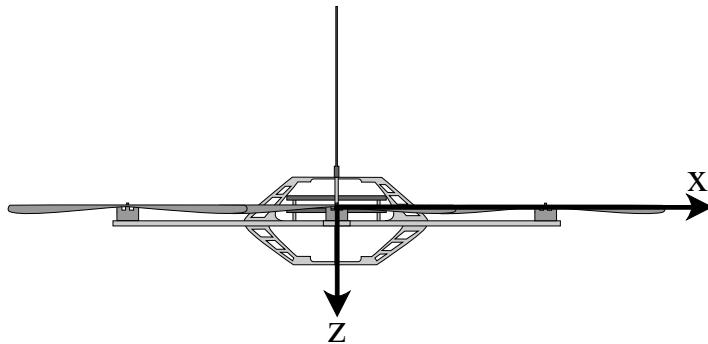


Figure 2.2: The X-3D-BL and the placement of origo in the body coordinate system

2.1.2 Quad Rotor Manoeuvring

Manoeuvring the quad rotor is done by altering the angular velocity Ω of the rotors and thereby the thrust and torque from each rotor. As illustrated in Figure 2.3 the rotors rotate in opposite directions in pairs. Rotor 1 and 3 rotate clockwise, while rotor 2 and 4 rotate counter-clockwise. When the quad rotor is in hover, Ω is the same for all four rotors, being positive for rotor 1 and 3. This will be accommodated for in the equations that involve Ω . The equal rotor velocity results in equal thrust from these, and the torque created by the rotors in the x-body-axis is cancelled by the torque created by the rotors positioned in the y-body-axis. It should be noted that, in the figures, the size of the arrows indicating the direction of the rotation of the rotor is proportional to the angular velocity of the rotor.

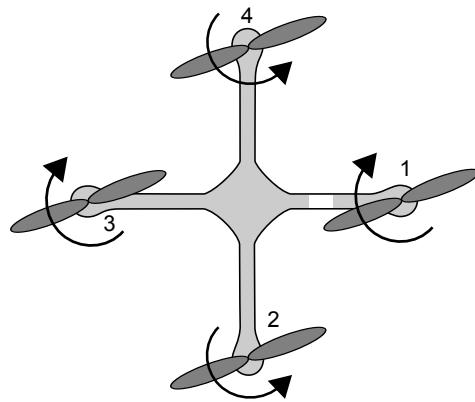


Figure 2.3: Diagram of a quad rotor hovering

Roll ϕ is defined as a rotation around the x-body-axis. This is obtained by increasing the thrust produced by one of the rotors placed in the y-body-axis and decreasing the thrust of the other, while maintaining thrust on the remaining rotors. Figure 2.4 illustrates the direction of positive roll, which means that rotor 4 increases thrust while rotor 2 decreases thrust.

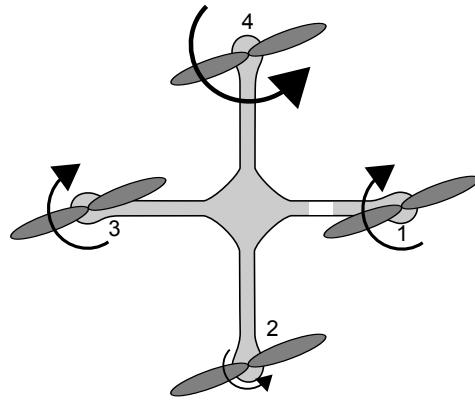


Figure 2.4: Diagram of a quad rotor performing a positive roll motion

Pitch θ is defined as the rotation around the y-body-axis. The rotation is obtained by increasing the thrust on one of the rotors placed on the x-body-axis and decreasing the thrust on the other rotor, while maintaining thrust on the remaining rotors. The positive θ is illustrated on Figure 2.5, where rotor 1 increases thrust while rotor 3 decreases thrust.

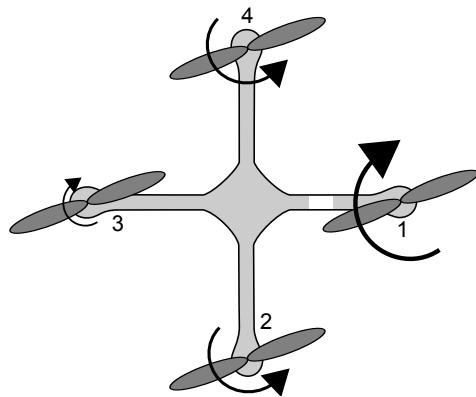


Figure 2.5: Diagram of a quad rotor performing a positive pitch motion

Yaw ψ is defined as the rotation around the z-body-axis. Yaw is obtained by altering the angular velocity Ω of the rotors on the x-body-axis compared to the angular velocity on the rotors on the y-body-axis. The resulting torque generated from the drag on the rotors, will no longer be zero and as a result the body will rotate around the z-body-axis. If the rotors on the y-body-axis is rotating faster than the rotors on the x-body-axis the body will rotate counter clockwise as illustrated in Figure 2.6.

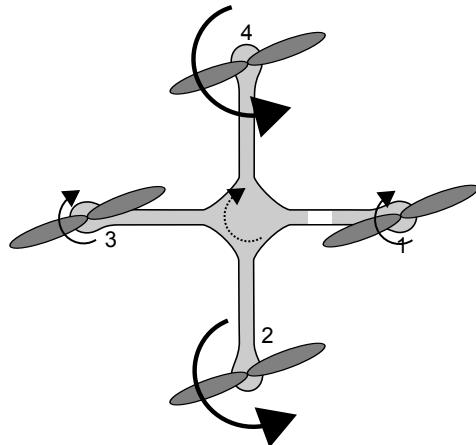


Figure 2.6: Diagram of a quad rotor performing a positive yaw motion

2.1.3 General Assumptions

In the derivation of the model for the X-3D-BL the following assumptions are made.

- Structure is symmetrical along the x and y-axis and is rigid.
- The rotors flapping effects are ignored.
- The drag effects caused by body-motion in the xy-plane is neglected.
- The effect on the rotor velocity caused by movement of the X-3D-BL is ignored.

Chapter 3

First Principles Model

3.1 Model Structure

The structure of the model can be seen on Figure 3.1. The input to the model comes from a RC-transmitter which outputs angular velocity references to an internal controller that controls pitch, yaw and roll velocities of the quad rotor. The input is contained in the vector S_ω . Furthermore the RC-transmitter controls a collective output S_{col} . The output is a position vector P and an attitude vector Θ .

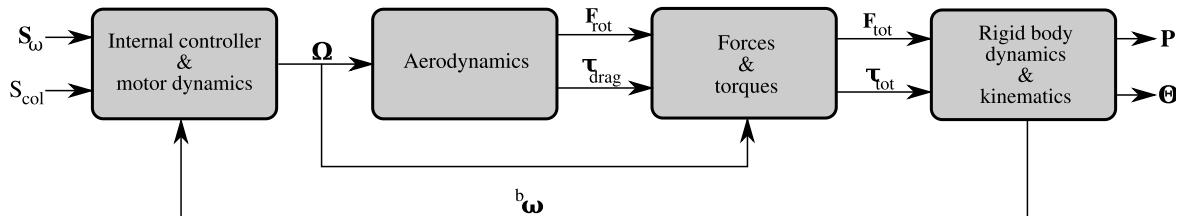


Figure 3.1: Overall structure of the model for the X-3D-BL quad rotor

As the Figure shows, the model has been split into four parts which each contains a part of the total model. Internal controller and motor dynamics covers the built-in attitude and motor controllers and the dynamics of the brushless motors. As this block is the first part of the model, it takes the input vector S_ω and a collective scalar S_{col} as input and produces rotor angular velocity of all rotors as output Ω . The aerodynamics block includes the dynamics introduced by the rotors. Its output is the force vector F_{rot} which points in the negative z-direction in the body frame. It describes the force generated by the rotors. The torque vector for each rotor is τ_{drag} which is caused by the air resistance drag on the rotors and the inertia of the blades. All forces and torques are throughout this chapter seen in body frame.

The forces and torques block converts the force vector F_{rot} and drag torque τ_{drag} to a total force F_{tot} and total torque τ_{tot} . The final block contains the rigid body dynamics and kinematics which converts the incoming forces and torques to positions P and angles Θ in the earth frame.

In the following sections each of the blocks will be explained in further detail starting with the last and moving backwards towards the input.

3.2 Rigid Body Dynamics and Kinematics

This block can be divided into two parts. The generation of angular acceleration and translatory acceleration. Both parts will be described in the following. To derive expressions for the angular and translatory acceleration it is necessary to know how transformation between the body and earth frame are done. This is explained in the following section.

3.2.1 Transformation Matrices

The orientation of one cartesian coordinate system, with respect to another, can always be described by three successive rotations. A quad rotors orientation can be described by the z,y,x (also called 3-2-1) right-hand rotation sequence that is required to get from earth frame in to alignment with the body frame. To do this following sequence is used.

- Right-hand rotation about the z-axis (positive ψ)
- Right-hand rotation about the new y-axis (positive θ)
- Right-hand rotation about the new x-axis (positive ϕ)

Matrices describing the rotations about the three axes are defined by [3, p. 15] as

$$C_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.1)$$

$$C_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.2)$$

$$C_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

The successive rotation is hereby gives as

$$\begin{aligned} {}^b\mathbf{C}_e &= C_x(\phi)C_y(\theta)C_z(\psi) \\ &= \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ (-c\phi s\psi + s\phi s\theta s\psi) & (c\phi c\psi + s\phi s\theta s\psi) & s\phi c\theta \\ (s\phi s\psi + c\phi s\theta c\psi) & (-s\phi c\psi + c\phi s\theta s\psi) & c\phi c\theta \end{bmatrix} \end{aligned} \quad (3.4)$$

where $c = \cos$ and $s = \sin$. Because ${}^b\mathbf{C}_e$ is orthonormal [3, p. 13] the inverse transformation can be described at the transpose of ${}^b\mathbf{C}_e$.

$${}^b\mathbf{C}_e^{-1}(\Theta) = {}^b\mathbf{C}_e^T(\Theta) \quad (3.5)$$

The rotation from body to earth frame is given by

$${}^e\mathbf{C}_b(\Theta) = {}^b\mathbf{C}_e^T(\Theta) = \begin{bmatrix} c\theta c\psi & (-c\phi s\psi + s\phi s\theta s\psi) & (s\phi s\psi + c\phi s\theta c\psi) \\ c\theta s\psi & (c\phi c\psi + s\phi s\theta s\psi) & (-s\phi c\psi + c\phi s\theta s\psi) \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (3.6)$$

where Θ is the vector of Euler angles

Euler Angles

The following show the transformation from the derivative of the Euler angles ($\dot{\Theta}$) to the angular velocity of the body frame (${}^b\omega$) [3, p.25].

$$\begin{aligned} {}^b\omega &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + C_x(\phi) \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ &= {}^bH_e(\Theta)\dot{\Theta} \end{aligned} \quad (3.7)$$

Where C_x , C_y are presented in Equation 3.1 and 3.2. bH_e is the transformation matrix from earth frame to body frame. By inverting bH_e the transformation matrix from body rates to the derivative of the Euler angles is found.

$$\begin{aligned} {}^eH_b(\Theta) &= {}^bH_e^{-1}(\Theta) \\ &= \begin{bmatrix} 1 & t\theta s\phi & t\theta c\phi \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \end{aligned} \quad (3.8)$$

Where $t = \tan$, and Θ are Euler angles. It can be shown that [4, p. 28]

$$\begin{aligned} \dot{\Theta} &= {}^eH_b(\Theta){}^b\omega \\ &= \begin{bmatrix} 1 & t\theta s\phi & t\theta c\phi \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} {}^b\omega \end{aligned} \quad (3.9)$$

3.2.2 Angular Acceleration

The purpose of this section is to describe the angular acceleration with respect to the body frame (${}^b\dot{\omega}$) as a function of all external torques seen from the body frame (${}^b\tau$).

From physics it is known that torque can be expressed as the derivative of angular momentum and as the product of inertia and acceleration [5, p.340]. This is shown in Equation 3.10.

$$\tau = \dot{\mathbf{L}} = \mathbf{J}\dot{\omega} \quad (3.10)$$

Angular momentum can be described in earth frame as Equation 3.11 and the derivative as in Equation 3.12. The derivative consists of two parts since both eC_b and ${}^b\mathbf{L}$ changes over time, the derivative is calculated using the differentiation rule of a product.

$${}^e\mathbf{L} = {}^eC_b {}^b\mathbf{L} = {}^eC_b \mathbf{J} {}^b\omega \quad (3.11)$$

$${}^e\tau = {}^e\dot{\mathbf{L}} = {}^eC_b {}^b\dot{\mathbf{L}} + {}^e\dot{C}_b {}^b\mathbf{L} \quad (3.12)$$

The derivative of the direct cosine matrix can be expressed as the cross product seen in Equation 3.13 [6, p.23].

$${}^e\dot{C}_b {}^b\mathbf{L} = {}^eC_b ({}^b\omega \times {}^b\mathbf{L}) \quad (3.13)$$

Equations 3.10 and 3.13 are inserted in 3.12. Then the torque is transformed into body frame and additional rotational matrices removed. Finally ${}^b\dot{\omega}$ is derived in Equation 3.17.

$${}^e\tau = {}^eC_b J^b \dot{\omega} + {}^eC_b ({}^b\omega \times J^b \omega) \quad (3.14)$$

$${}^b\tau = {}^bC_e {}^eC_b J^b \dot{\omega} + {}^bC_e {}^eC_b ({}^b\omega \times J^b \omega) \quad (3.15)$$

$$= J^b \dot{\omega} + ({}^e\omega \times J^b \omega) \quad (3.16)$$

$${}^b\dot{\omega} = J^{-1} ({}^b\tau - ({}^b\omega \times J^b \omega)) \quad (3.17)$$

3.2.3 Translatory Acceleration

An expression for the translatory acceleration of the body frame with respect to the body frame (${}^b\dot{v}$) will now be derived in the same way that the angular acceleration has been considered in the previous section. ${}^e v$ describes the translatory velocity of the body frame relative to the earth frame. Newtons second law states that acceleration is force divided by mass [7, p. 15].

$$\dot{v} = \frac{1}{m} F \quad (3.18)$$

The velocity ${}^e v$ is described relative to the earth frame in Equation 3.19 and the derivative in Equation 3.20.

$${}^e v = {}^e C_b {}^b v \quad (3.19)$$

$${}^e \dot{v} = {}^e C_b {}^b \dot{v} + {}^e \dot{C}_b {}^b v \quad (3.20)$$

Equation 3.18 and Equation 3.13 is inserted into Equation 3.20 and the expression is transformed to body frame and simplified.

$$\frac{1}{m} {}^e F = {}^e C_b {}^b \dot{v} + {}^e C_b {}^b \omega \times {}^b v \quad (3.21)$$

$$\frac{1}{m} {}^b C_e {}^e F = {}^b \dot{v} + {}^b \omega \times {}^b v \quad (3.22)$$

$${}^b \dot{v} = \frac{1}{m} {}^b F - {}^b \omega \times {}^b v \quad (3.23)$$

Both the translatory acceleration and the angular acceleration are described in body frame.

3.3 Forces and Torques

From the derivation of the rigid body dynamics and kinematics it is seen that the total forces and torques acting on the body of the X-3D-BL is needed. This section will deal with the generation of these forces and torques in turn.

3.3.1 Forces

The total force acting on the body is a summation of all external forces. The forces chosen to include in this model is shown in Equation 3.24.

$$F_{tot} = {}^b F_1 + {}^b F_2 + {}^b F_3 + {}^b F_4 + {}^b F_g \quad (3.24)$$

where ${}^b\mathbf{F}_1$ through ${}^b\mathbf{F}_4$ are the lifts generated by the rotors. In the defined body frame they consists only of a negative component in the z-body-axis. ${}^b\mathbf{F}_g$ is the gravitational force acting on the body. The gravitational force can be expressed as in Equation 3.25.

$${}^b\mathbf{F}_g = {}^b\mathbf{C}_e \cdot \mathbf{g} \cdot m \quad (3.25)$$

where m is the mass of the X-3D-BL, \mathbf{g} is a vector that has the gravitational constant as its last element ($\mathbf{g} = [0 \ 0 \ g]^T$) and ${}^b\mathbf{C}_e$ is the transformation matrix from earth to body frame.

The force generated by the quad rotor is depending on the angular velocity of the rotors. The angular velocity of the rotors is chosen not to be modelled, because it is controlled by the internal controllers. When the quad rotor is in a near hover state the resulting force of $\mathbf{F}_{rot} = {}^b\mathbf{F}_1 + {}^b\mathbf{F}_2 + {}^b\mathbf{F}_3 + {}^b\mathbf{F}_4$ can be estimated as a polynomial of the input S_{col} . The derivation and experiments leading to this polynomial is described in Appendix A.2. The polynomial is shown in Equation 3.26.

$$\mathbf{F}_{rot}(S_{col}) = \begin{bmatrix} 0 \\ 0 \\ -0.0000018 \cdot S_{col}^3 + 0.000055 \cdot S_{col}^2 + 0.064 \cdot S_{col} - 6.075 \end{bmatrix} \quad (3.26)$$

By modelling the force \mathbf{F}_{rot} as a polynomial, and \mathbf{F}_{tot} as in Equation 3.24, Newton's second law would suggest that the quad rotor would be in a constant acceleration when \mathbf{F}_{rot} does not exactly cancel out the gravitational force. By observing the quad rotor flying with steps applied on collective, this can not be confirmed.

Observations suggests that a step in the collective value result in an approximately constant velocity. Because of this observation, it has been chosen to add a part to Equation 3.24 that emulates the effect of induced inflow through the rotors during translatory movement in the z-axis.

The term induced inflow should be interpreted as the total airflow through the rotors minus the airflow generated by the rotation of the rotors. This means that when in stable hover the induced inflow is zero and when the quad rotor is moving e.g. vertically, the induced inflow will be non zero [8, p.2]. The subject of induced inflow is a complex matter and will not be treated in further detail since observations have shown that the used model of induced inflow is sufficient. Further information about the induced inflow emulation can be found in Appendix A.3. The expression of the total external force affecting the quad rotor is shown in Equation 3.27.

$${}^b\mathbf{F}_{tot} = \mathbf{F}_{rot} + {}^b\mathbf{F}_g + I_f \cdot {}^b\mathbf{v}_3 \quad (3.27)$$

Where I_f is the induced inflow emulation coefficient and ${}^b\mathbf{v}_3$ is the translatory velocity of the quad rotor in the body z-axis.

In Appendix A.3 is I_f found to be -0.3559 . Adding this term, the quad rotor will no longer accelerate to infinite velocity, but settle at a velocity were the effect of the induced airflow will equal $\mathbf{F}_{rot} + {}^b\mathbf{F}_g$.

3.3.2 Torques

The total torque acting on the body of the X-3D-BL is, like the forces, a summation of all the external torques, which is shown in Equations 3.28, 3.29 and 3.30.

$${}^b\boldsymbol{\tau}_{tot\ x} = {}^b\boldsymbol{\tau}_x + {}^b\boldsymbol{\tau}_{gyro\ x} \quad (3.28)$$

$${}^b\boldsymbol{\tau}_{tot\ y} = {}^b\boldsymbol{\tau}_y + {}^b\boldsymbol{\tau}_{gyro\ y} \quad (3.29)$$

$${}^b\boldsymbol{\tau}_{tot\ z} = {}^b\boldsymbol{\tau}_z \quad (3.30)$$

where ${}^b\boldsymbol{\tau}_x$, ${}^b\boldsymbol{\tau}_y$ and ${}^b\boldsymbol{\tau}_z$ are the torques generated by the rotor lifts and ${}^b\boldsymbol{\tau}_{gyro\ x}$, ${}^b\boldsymbol{\tau}_{gyro\ y}$ are the gyro effects caused by the rotation of the rotors. The reason why there is no gyro effect contribution on the z-axis is that the rotors only rotates in xy-plane of the body.

The torques generated by the rotor lift can be described as in Equations 3.31, 3.32 and 3.33.

$${}^b\boldsymbol{\tau}_x = (-\mathbf{F}_2 + \mathbf{F}_4) \cdot l \quad (3.31)$$

$${}^b\boldsymbol{\tau}_y = (\mathbf{F}_1 - \mathbf{F}_3) \cdot l \quad (3.32)$$

$${}^b\boldsymbol{\tau}_z = (-{}^b\boldsymbol{\tau}_1 + {}^b\boldsymbol{\tau}_2 - {}^b\boldsymbol{\tau}_3 + {}^b\boldsymbol{\tau}_4) \quad (3.33)$$

Where \mathbf{F}_1 through \mathbf{F}_4 are the forces generated by rotors and l is the distance from the center of mass to the hub of the rotor. ${}^b\boldsymbol{\tau}_1$ through ${}^b\boldsymbol{\tau}_4$ are the individual torques generated by the rotors. The signs of the τ are defined by the rotational direction of the individual rotor e.g. rotor 1 is rotating clockwise.

The second term of Equations 3.28 and 3.29 describes the gyro effects of the rotors which can be expressed as in Equations 3.34 and 3.35 [9, p.103].

$${}^b\boldsymbol{\tau}_{gyro\ x} = J_r \cdot \omega_y \cdot (\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4) \quad (3.34)$$

$${}^b\boldsymbol{\tau}_{gyro\ y} = J_r \cdot \omega_x \cdot (\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) \quad (3.35)$$

Where J_r is the inertia of the rotor, ω_x and ω_y is the angular velocity of the quad rotor around the x and y-axis. Ω_1 to Ω_4 is the angular velocity of the rotors.

In order to determine the torque acting on the quad rotor it is seen from Equation 3.31, 3.32, and 3.33 that the individual rotor thrust and therefore velocity is needed. Because the rotor velocity is not available during flight and can not be controlled directly, it is chosen to use system identification and parameter estimation to determine the relationship between the RC-transmitter input and the angular velocity of the quad rotor. This means that the model structure has to be revised which is done in the following section.

3.4 Revised Model Structure

Since the angular velocity of the rotors is not available during flight of the X-3D-BL a full first principle model cannot be derived without knowledge of the internal controllers. Therefore the modelling approach is changed. The new approach can be seen on Figure 3.2 where the parts that will be covered by the system identification are boxed in.

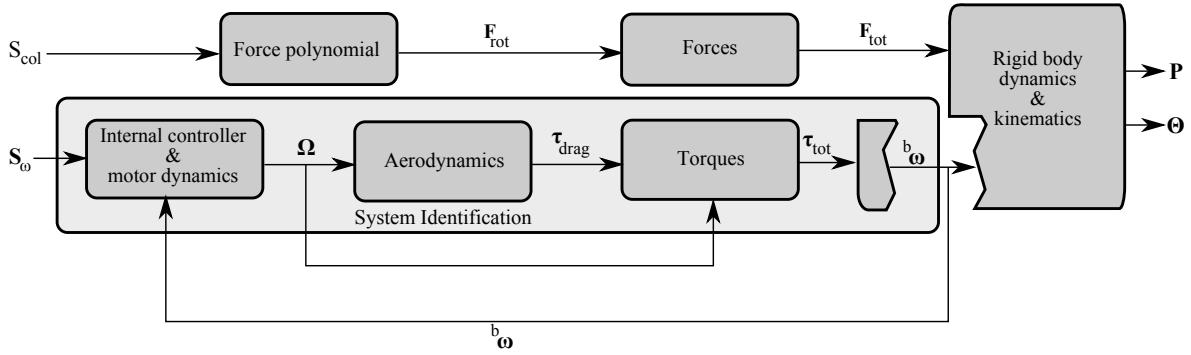


Figure 3.2: Overall structure of the model for the X-3D-BL quad rotor using system identification

This means that the interface to the Rigid body dynamics and kinematics has to be redefined. The new input to Rigid body dynamics and kinematics is now the angular velocity and the force acting on the body generated by the rotors and gravity. This redefinition imply that Equation 3.17 no longer will be used. Since system identification and parameter estimation is used now, the complex area of aerodynamics will not be further investigated.

The following chapter will concern the system identification and parameter estimation, but before this the internal controllers and motor dynamics of the X-3D-BL introduced.

3.5 Internal Controller and Motor Dynamics

This block contains the conversion from the input signal generated by the RC-transmitter to the angular velocity of the rotors. The contents of the block can be seen in Figure 3.3.

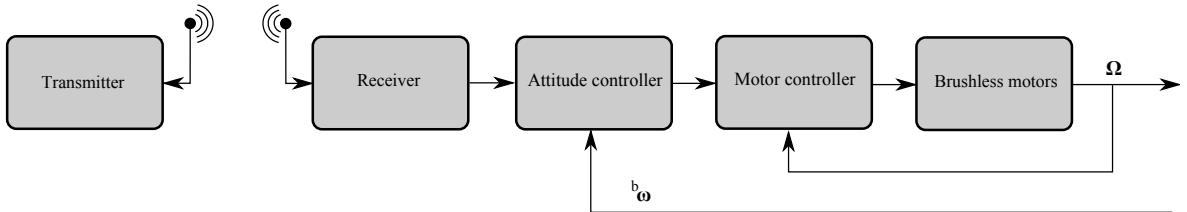


Figure 3.3: Content of the internal controller and motor dynamics block

The first block is the transmitter which transmits orientation references and thrust to the receiver. These references are used by the attitude controller to generate a reference signal to the motor controller which in turn makes the motors hold the desired angular velocity. The attitude controller uses a feedback from three piezo gyroscopes and one accelerometer placed on the Fun-Pilot board. More information of this can be found in Appendix B. Therefore the dynamics that needs to be estimated is approximately the dynamics of this attitude controller controlling the attitude of the X-3D-BL.

Since the motor is connected to the rotors, the rotors will create a load on the motor which will decrease the angular velocity. This effect is countered by the motor controller and is therefore neglected on the figure.

Since the basic model and model approach of the X-3D-BL now is covered, the system identification and parameter estimation can now be conducted.

System Identification and Parameter Estimation

In this chapter parameters for models of the roll, pitch and yaw input/output relations are estimated. The actuation of the control sticks on the RC-transmitter S_ϕ , S_θ and S_ψ are all references to an internal controller maintaining an angular velocity. More information of this controller is found in Appendix B describing the X-3D-BL quad rotor. The models describe the dynamics from the position of the stick on the RC-controller to the angular velocity of roll, pitch and yaw. As described in Section 1.3 it is only possible to record the Euler angles and position of the quad rotor. The Euler angles are differentiated and the resulting Euler rates are transformed into body rates. The body rates are used as basis for the system identification.

4.1 Model Structure and Estimation Method

The relation between input from the RC-transmitter and the angular velocity of the quad rotor is estimated as a grey-box model. Grey-box in the sense that some behaviours of the quad rotor is considered when deciding the structure.

The structure of the model is chosen to be a linear transfer function. The order of the transfer functions is chosen to be as small as possible. From early measurements a first order system is found inadequate since the initial dynamics of a step is not described. Therefore a second order system is used as basis for the parameter estimation.

One observation is that the quad rotor reacts less to fast changes of the input. This indicates that the dynamic can be described as a low-pass filter.

Further it is observed that the RC-transmitter also acts as a low-pass filter. This dampens the high frequency inputs and thereby reduces their significants. With very little high frequent test signals it is likely that the model will only be a good approximation in the low frequency range.

The structure of the transfer function is chosen to be a second order transfer function with no zeros as seen in

Equation 4.1.

$$H(s) = \frac{\phi_0}{s^2 + \theta_1 s + \theta_0} \quad (4.1)$$

Where $\phi_0, \theta_0, \theta_1$ is the parameters that need to be estimated.

The structure describes the relation from actuation of the RC-transmitter stick to body angle rates of the quad rotor.

For the estimation of the parameters the prediction error method is used. This method minimizes the sum of squared residuals in a measured flight to find the parameters $\hat{\boldsymbol{\theta}}^T = (\phi_0, \theta_0, \theta_1)$ [10, p.163].

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \left\{ \sum_{k=p+1}^N \varepsilon_k^2(\boldsymbol{\theta}) \right\} \quad (4.2)$$

Where

$\hat{\boldsymbol{\theta}}$ are the estimated parameters

p is the number of parameters in the denominator of the transfer function.

N is the number of samples in the measurement used for the estimation.

$\varepsilon_k(\boldsymbol{\theta})$ is the residual.

The offset of $p + 1$ is necessary to calculate the estimated output needed in the residuals. The residuals are the difference between the measured output and the output calculated with the estimated parameters $\hat{\boldsymbol{\theta}}$.

$$\varepsilon_k(\boldsymbol{\theta}) = y[k] - H_k(\boldsymbol{\theta}) * u[k] \quad (4.3)$$

In practice when the residuals are calculated they are calculated as the measured output sample minus the discrete impulse response convoluted with the input samples. The discrete impulse response comes from the discrete transfer function $H_z(\boldsymbol{\theta})$. For analysis purpose the transfer functions from here will only be denoted as the continuous transfer function $H(s)$.

The implementation used for estimating $\hat{\boldsymbol{\theta}}$ is found in the Matlab system identification toolbox as a function named `pem` [11].

The Matlab function `pem` uses either the Gauss-Newton, adaptive Gauss-Newton or the Levenberg-Marquardt search algorithm. They are all iterative methods to solve least squares problems. Matlab automatically chooses the fastest algorithm as default. The algorithm stops when either the maximum number of iterations is reached or the improvement in $\hat{\boldsymbol{\theta}}$ is less than the tolerance. All estimations are made with the default iteration number (20) and the default tolerance (0.01).

The tolerance is a value expressing how much improvement there have been since the last iteration.

The data used for the parameter estimation is collected doing a normal flight performed by a human pilot. The intention of the flight was to apply as big input as possible (e.g. roll) to increase the Signal to Noise Ratio. While at the same time keep the quad rotor within range of the Vicon system and applying as little input on the other inputs (e.g. pitch and yaw) as possible.

As other before have proved (e.g. [9, p.51]) there is a cross coupling between the three axis of motion. This cross coupling have not been clearly observed doing the manually flight and is believed to be insignificant due to the internal controllers. It is therefore neglected when estimating the transfer functions.

Further information of the data used for the estimation is found in the measurement journal in Appendix A.1.

The estimated transfer functions are shown in Equations 4.4 to 4.6. The transfer functions express the relation from input to angular velocity.

$$H_{b\omega_\phi}(s) = \frac{14.56}{s^2 + 67.08s + 870.6} \quad (4.4)$$

$$H_{b\omega_\theta}(s) = \frac{10.67}{s^2 + 49.55s + 613.8} \quad (4.5)$$

$$H_{b\omega_\psi}(s) = \frac{6.353}{s^2 + 27.42s + 187.9} \quad (4.6)$$

A further analysis of the estimated transfer functions is found in Chapter 8. In the next chapter further nonlinearities are introduced.

Chapter 5

Additional Non-linearities

The purpose of a model is to describe and simulate relevant parts of the real world. In this section further non-linearities are added to the model for this purpose. The model with those non-linearities is to be used as a test environment where controllers can be tested and verified before they are tried out on the X-3D-BL.

5.1 Battery Discharge

As mentioned in Section 1.1 the X-3D-BL is powered by battery. During flight it has been observed that the level of collective input required to keep the quad rotor hovering is slowly increasing with time. To document this an experiment has been conducted. The measurement journal in Appendix A.5 describes how a polynomial have been derived to describe the increasing collective input that is needed over time when the quad rotor is kept in hover. Figure 5.1 shows the measured and estimated collective input during a hover flight. During the flight the quad rotor was controlled by a PID controller.

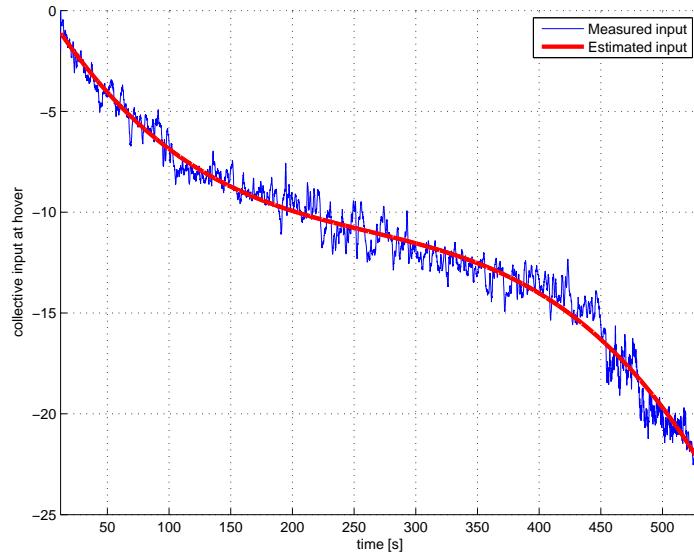


Figure 5.1: The collective input increases (100 minimum, -100 maximum) over time as a result of the battery discharge. The red line is an estimation of the discharge. The graph is shown relative to hover.

The polynomial seen in Equation 5.1 is implemented into the model by adding this to the input collective value in the model.

$$\hat{S}_{col} = S_{col} + (-0.0000003883 \cdot t^3 + 0.0003061 \cdot t^2 - 0.09536 \cdot t) \quad (5.1)$$

5.2 Measurement Noise

Section 1.3 regarding the Motion Tracking lab determines that measurements noise exists. It is estimated with a Gaussian distribution with mean 0 and variance 0.002. This is implemented in the model by adding the noise to the calculated positions and angles.

Chapter 6

Linearisation of Model

The equations detailing the kinematics, dynamics, forces and torques of the system have been derived in chapter 3. To use these equations in model based linear controllers, these equations have to be linearised. This is done in the following sections.

6.1 Linearisation Method

To linearise the equations, a first order Taylor approximation is used. Each variables will be substituted with an operating point value (steady-state value), and a small signal gain. The operating point, for this particular operation will be described in the following section.

A first order Taylor approximation with one variable, can be described as in Equation 6.1 [12, p.707].

$$f(x) \approx f(\bar{x}) + \frac{\delta f(\bar{x})}{\delta x} \tilde{x} \quad (6.1)$$

Where:

$f(x)$ is a function of the variable x

\bar{x} is the operating point value

\tilde{x} is $(x - \bar{x})$.

In the case of a first order Taylor approximation with multiple variables:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) \approx & f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) + \left. \frac{\delta f(x_1, x_2, \dots, x_n)}{\delta x_1} \right|_{x_1=\bar{x}_1} \tilde{x}_1 + \\ & \dots \left. \frac{\delta f(x_1, x_2, \dots, x_n)}{\delta x_2} \right|_{x_2=\bar{x}_2} \tilde{x}_2 + \left. \frac{\delta f(x_1, x_2, \dots, x_n)}{\delta x_n} \right|_{x_n=\bar{x}_n} \tilde{x}_n \end{aligned} \quad (6.2)$$

6.2 Operating Point

In order to perform the linearisation, an operating point must be specified. The operating point for the X-3D-BL is set to be a hover condition, specifying the angles, angular velocities and translatory velocities, being constant.

The operating points are specified as follows:

$$\bar{\phi} = \bar{\theta} = \bar{\psi} = 0 \quad (6.3)$$

$${}^b\bar{\omega} = 0 \quad (6.4)$$

$${}^b\bar{v} = 0 \quad (6.5)$$

These specified conditions are true when the quad rotor is in a perfect hover.

6.3 Linearisation of Body Dynamics

In the following, the equations concerning the motion of the body are linearised. There are two equations, which needs to be linearised, respectively:

$${}^e v = {}^e C_b(\Theta) {}^b v \quad (6.6)$$

$${}^e \omega = {}^e H_b(\Theta) {}^b \omega \quad (6.7)$$

The above stated equations will be linearised using the method stated in Section 6.1.

6.3.1 Linearising the Translatory Velocity

The linearisation of the Translatory Velocity is done by applying the first order Taylor approximation described in 6.1 on the direct cosine matrix ${}^e C_b$ yielding that $\sin(a) = a$, $\tan(a) = a$ and $\cos(a) = 1$, and by assuming that the small signal gains are so small that multiplication of two small signal gains effectively are zero.

$$\begin{aligned} {}^e C_b(\Theta) &= \begin{bmatrix} c\theta c\psi & (-c\phi s\psi + s\phi s\theta s\psi) & (s\phi s\psi + c\phi s\theta c\psi) \\ c\theta s\psi & (c\phi c\psi + s\phi s\theta s\psi) & (-s\phi c\psi + c\phi s\theta s\psi) \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (6.8) \\ &\approx \begin{bmatrix} 1 & -\psi + \phi\theta & \phi\psi + \theta \\ \psi & 1 + \phi\theta\psi & -\phi + \theta\psi \\ -\theta & \phi & 1 \end{bmatrix} \\ &\approx \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} \end{aligned}$$

Insertion of 6.8 into 6.6 gives the following ${}^e v$:

$$\begin{aligned} {}^e v &= {}^e C_b(\Theta) \cdot {}^b v \quad (6.9) \\ &\approx \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} \cdot {}^b v \\ &= \begin{bmatrix} {}^b v_1 - \psi \cdot {}^b v_2 + \theta \cdot {}^b v_3 \\ \psi \cdot {}^b v_1 + {}^b v_2 - \phi \cdot {}^b v_3 \\ -\theta \cdot {}^b v_1 + \phi \cdot {}^b v_2 + {}^b v_3 \end{bmatrix} \\ &\approx \begin{bmatrix} {}^b v_1 \\ {}^b v_2 \\ {}^b v_3 \end{bmatrix} = {}^b v \end{aligned}$$

6.3.2 Linearising the Euler Rates

The linearisation of the Euler rates is done as in the previous section by implying the small angle approximation on the transform matrix ${}^e\mathbf{H}_b$. The Euler transform, found in section 3.2.1 linearised becomes the following.

$${}^e\mathbf{H}_b(\Theta) = \begin{bmatrix} 1 & \tan(\theta)\sin(\phi) & \tan(\theta)\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \approx \begin{bmatrix} 1 & \theta\phi & \theta \\ 0 & 1 & -\phi \\ 0 & \phi & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & -\phi \\ 0 & \phi & 1 \end{bmatrix} \quad (6.10)$$

Insertion of 6.10 into 6.7 gives the following $\dot{\Theta}$:

$$\begin{aligned} \dot{\Theta} &= {}^e\mathbf{H}_b(\Theta) \cdot {}^b\omega \\ &\approx \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & -\phi \\ 0 & \phi & 1 \end{bmatrix} \cdot {}^b\omega \\ &= \begin{bmatrix} {}^b\omega_\phi + \theta \cdot {}^b\omega_\psi \\ {}^b\omega_\theta - \phi \cdot {}^b\omega_\psi \\ \phi \cdot {}^b\omega_\theta + {}^b\omega_\psi \end{bmatrix} \\ &\approx \begin{bmatrix} {}^b\omega_\phi \\ {}^b\omega_\theta \\ {}^b\omega_\psi \end{bmatrix} = {}^b\omega \end{aligned} \quad (6.11)$$

6.4 Linearisation of Forces

The following will concentrate on the linearisation of the generated forces of the system. In this, a second order Taylor approximation will be used, in the rotational matrices, mapping the force of gravity to the body frame, due to the Translatory force's high dependency on the body angles.

6.4.1 Forces

The equation for the generated lift is:

$$\mathbf{F}_{tot} = \mathbf{F}_{rot} + {}^b\mathbf{F}_g + I_f \cdot {}^b\mathbf{v}_3 \quad (6.12)$$

The gravitational force ${}^b\mathbf{F}_g$ can, as previously mentioned, be expressed as:

$${}^b\mathbf{F}_g = {}^b\mathbf{C}_e(\Theta) \cdot \mathbf{g} \cdot m \quad (6.13)$$

Since all the expressions are independent of each other, they are linearised separately. First the gravitational force will be linearised, taking Θ as a input for the rotational matrix ${}^b\mathbf{C}_e$:

As previously the direct cosine matrix is linearised by using first order Taylor approximation and assuming that small signal gains multiplied with each other gives approximately zero.

The rotational matrix ${}^b\mathbf{C}_e$ becomes as follows.

$${}^b\mathbf{C}_e = {}^e\mathbf{C}_b^T \approx \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \quad (6.14)$$

Since the gravity in the earth frame only affects the z-axis, the equation can be rewritten:

$${}^b\mathbf{F}_g = {}^b\mathbf{C}_e \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \cdot m \quad (6.15)$$

Inserting the linearised expression yields:

$${}^b\mathbf{F}_g = \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \cdot m = \begin{bmatrix} -\theta \\ \phi \\ 1 \end{bmatrix} g \cdot m \quad (6.16)$$

The linearised gravitational force is with the remaining forces inserted in the expression for the translatory acceleration in the body frame ${}^b\dot{\mathbf{v}}$.

$${}^b\dot{\mathbf{v}} = \frac{\mathbf{F}_{tot}}{m} = \frac{\mathbf{F}_{rot} + {}^b\mathbf{F}_g + I_f \cdot {}^b\mathbf{v}_3}{m} \quad (6.17)$$

When ${}^b\dot{\mathbf{v}}$ is expressed as components the input force \mathbf{F}_{rot} and $I_f \cdot v_z$ is only part of the third components since they only have an contribution in the z-axis of the body frame.

$${}^b\dot{v}_1 = -\theta \cdot g \quad (6.18)$$

$${}^b\dot{v}_2 = \phi \cdot g \quad (6.19)$$

$${}^b\dot{v}_3 = \frac{\mathbf{F}_{rot} + g \cdot m + I_f \cdot {}^b\mathbf{v}_3}{m} \quad (6.20)$$

Because the expression in ${}^b\dot{v}_3$ is not linear, it is chosen to use the expression $\mathbf{F}_z = \mathbf{F}_{rot} + g \cdot m$ as input. Finally the expression ${}^b\dot{v}_3$ becomes.

$${}^b\dot{v}_3 = \frac{\mathbf{F}_z}{m} + \frac{I_f}{m} \cdot {}^b\mathbf{v}_3 \quad (6.21)$$

6.5 Linearised Equations of Motion

The above state linearisation of the equations of motion, are presented here.

$$\dot{x} = {}^b\dot{v}_1 \quad (6.22)$$

$$\dot{y} = {}^b\dot{v}_2 \quad (6.23)$$

$$\dot{z} = {}^b\dot{v}_3 \quad (6.24)$$

$$\dot{\phi} = {}^b\omega_\phi \quad (6.25)$$

$$\dot{\theta} = {}^b\omega_\theta \quad (6.26)$$

$$\dot{\psi} = {}^b\omega_\psi \quad (6.27)$$

$${}^b\dot{v}_1 = -\theta \cdot g \quad (6.28)$$

$${}^b\dot{v}_2 = \phi \cdot g \quad (6.29)$$

$${}^b\dot{v}_3 = \frac{\mathbf{F}_z}{m} + \frac{I_f}{m} \cdot {}^b\mathbf{v}_3 \quad (6.30)$$

In the following chapter a state space model, which is based on the linearised equations, will be developed.

Chapter **7**

State Space Model

The purpose of this section is to put the model equations derived in the above chapters into state space form. In order to do this the following states has been chosen:

$$[x, y, z, \phi, \theta, \psi, {}^b v_1, {}^b v_2, {}^b v_3, {}^b \omega_\phi, {}^b \omega_\theta, {}^b \omega_\psi]^T \quad (7.1)$$

Where x, y, z are the position of the quad rotor's center of mass, ϕ, θ, ψ are the 3-2-1 Euler angles, ${}^b v_1, {}^b v_2, {}^b v_3$ are translatory velocities in the body frame and ${}^b \omega_\phi, {}^b \omega_\theta, {}^b \omega_\psi$ are the angular velocities in the body frame. The system inputs are S_ϕ, S_θ, S_ψ and F_z , where S_ϕ, S_θ, S_ψ are references to the onboard angular velocity controller and F_z are a mapping of S_{col} . For details regarding F_z , see Section 3.3.1. The system outputs are $x, y, z, \phi, \theta, \psi$ which can be measured by the Vicon system. The equations required to create the state space description are derived and linearised in the above sections and are for clarity repeated below:

$$\dot{x} = {}^b v_1 \quad (7.2)$$

$$\dot{y} = {}^b v_2 \quad (7.3)$$

$$\dot{z} = {}^b v_3 \quad (7.4)$$

$$\dot{\phi} = {}^b \omega_\phi \quad (7.5)$$

$$\dot{\theta} = {}^b \omega_\theta \quad (7.6)$$

$$\dot{\psi} = {}^b \omega_\psi \quad (7.7)$$

$${}^b \dot{v}_1 = -\theta \cdot g \quad (7.8)$$

$${}^b \dot{v}_2 = \phi \cdot g \quad (7.9)$$

$${}^b \dot{v}_3 = \frac{F_z}{m} + \frac{I_f}{m} \cdot {}^b v_3 \quad (7.10)$$

$$\frac{{}^b \omega_\phi(s)}{S_\phi(s)} = \frac{14.56}{s^2 + 67.08s + 870.6} \quad (7.11)$$

$$\frac{{}^b \omega_\theta(s)}{S_\theta(s)} = \frac{10.67}{s^2 + 49.55s + 613.8} \quad (7.12)$$

$$\frac{{}^b \omega_\psi(s)}{S_\psi(s)} = \frac{6.353}{s^2 + 27.42s + 187.9} \quad (7.13)$$

To have Equations 7.11, 7.12 and 7.13 on state space form, three extra states are introduced. The states are denoted ${}^b\dot{\omega}_\phi$, ${}^b\dot{\omega}_\theta$ and ${}^b\dot{\omega}_\psi$ and corresponds to the respective derivatives of the angular velocities in body frame. The introduction of these three additional states yields the following state vector:

$$[x, y, z, \phi, \theta, \psi, {}^b v_1, {}^b v_2, {}^b v_3, {}^b \omega_\phi, {}^b \omega_\theta, {}^b \omega_\psi, {}^b \dot{\omega}_\phi, {}^b \dot{\omega}_\theta, {}^b \dot{\omega}_\psi]^T \quad (7.14)$$

With the three additional states the transfer functions can be transformed into differential equations. This is done as seen in Equations 7.15 to 7.18.

$$H_\phi(s) = \frac{^b\omega_\phi(s)}{S_\phi(s)} = \frac{14.56}{s^2 + 67.08s + 870.6} \quad (7.15)$$

$$14.56 \cdot S_\phi(s) = s^2 \cdot {}^b\omega_\phi(s) + 67.08s \cdot {}^b\omega_\phi(s) + 870.6 \cdot {}^b\omega_\phi(s) \quad (7.16)$$

$$s^2 \cdot {}^b\omega_\phi(s) = -67.08s \cdot {}^b\omega_\phi(s) - 870.6 \cdot {}^b\omega_\phi(s) + 14.56 \cdot S_\phi(s) \quad (7.17)$$

$${}^b\ddot{\omega}_\phi = -67.08 \cdot {}^b\dot{\omega}_\phi - 870.6 \cdot {}^b\omega_\phi + 14.56 \cdot S_\phi \quad (7.18)$$

The differential equations arising from the transfer functions for angular velocity of pitch and yaw are presented in Equations 7.19 and 7.20. The derivations are left out because the approach is similar to roll.

$${}^b\ddot{\omega}_\theta = -49.55 \cdot {}^b\dot{\omega}_\theta - 613.8 \cdot {}^b\omega_\theta + 10.67 \cdot S_\theta \quad (7.19)$$

$${}^b\ddot{\omega}_\psi = -27.42 \cdot {}^b\dot{\omega}_\psi - 187.9 \cdot {}^b\omega_\psi + 6.353 \cdot S_\psi \quad (7.20)$$

The above equations are combined in the state space model shown in Equations 7.21 and 7.22.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (7.21)$$

$$y(t) = Cx(t) \quad (7.22)$$

The result is the following A , B and C matrices.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g \\ 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{I_f}{m} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -871 & 0 & 0 & -67.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -614 & 0 & 0 & -49.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -188 & 0 & 0 & 0 & -27.4 \end{bmatrix} \quad (7.23)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 14.6 & 0 & 0 & 0 \\ 0 & 10.7 & 0 & 0 \\ 0 & 0 & 0 & 6.35 \end{bmatrix} \quad (7.24)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.25)$$

7.1 Discrete state space model

The system represented by the matrices \mathbf{A} , \mathbf{B} and \mathbf{C} above is in continuous time. To implement the system in practice, a discrete representation is needed.

A solution to the continuous state space model can be stated as Equation 7.26 [13, p.630].

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \quad (7.26)$$

When making a discretisation the continuous time variable t is replaced with the sample number k . In this derivation $t = kT + T$ and $t_0 = kT$ where T is the sampling time. The solution over the k -th sample is shown in Equation 7.27

$$\mathbf{x}(kT + T) = e^{\mathbf{A}T}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \quad (7.27)$$

In this discretisation the method Zero Order Hold (ZOH) is used. The input $\mathbf{u}(\tau)$ is kept constant throughout the sample period.

$$\mathbf{u}(\tau) = \mathbf{u}(kT) \quad kT \leq \tau < kT + T \quad (7.28)$$

To ease the derivation the variable η is introduced.

$$\eta = kT + T - \tau \quad (7.29)$$

This reduces Equation 7.27 to the following equation.

$$\mathbf{x}(kT + T) = e^{\mathbf{A}T}\mathbf{x}(kT) + \left(\int_0^T e^{\mathbf{A}\eta} d\eta \right) \mathbf{B}\mathbf{u}(kT) \quad (7.30)$$

From this expressions for the discrete state space can be extracted. By defining the following, the discrete state space model is simplified.

$$\Phi = e^{\mathbf{A}T} \quad (7.31)$$

$$\Gamma = \left(\int_0^T e^{\mathbf{A}\eta} d\eta \right) \mathbf{B} \quad (7.32)$$

$$\mathbf{H} = \mathbf{C} \quad (7.33)$$

The discrete state space model is described in Equation 7.34 and 7.35.

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k) \quad (7.34)$$

$$\mathbf{y}(k) = \mathbf{H} \mathbf{x}(k) \quad (7.35)$$

Computation of Φ and Γ by use of Equations 7.31 and 7.32 is implemented in the Matlab function `c2d(sysc,Ts,'zoh')`, where `sysc` are the continuous state space description and `Ts` the sample time. By use of the `c2d` function the following discrete state space representation is found.

Chapter 8

Model Verification

In this chapter the model is analysed and compared to measured flights. Firstly the transfer functions found in chapter 4.1 are analysed and secondly the linearised model from Chapter 7. The stability of the models are analysed using step responses, pole-zero mapping and the systems frequency responses are analysed through bode plots.

8.1 Validation of Estimated Transfer Functions

In this section the three transfer functions are further analysed. It is the purpose of this section to validate them as usable models of the internal attitude controllers, motor controllers and quad rotor dynamics combined. In other words, it is investigated how the transfer functions resemble measured characteristics and behaviour of the quad rotor.

When estimating the transfer functions, data from a flight with a human pilot was used. The maximum angle of pitch and roll measured during this flight was ± 20 degrees in roll and pitch.

8.1.1 Step Response

As a first step of the validation the step response is analysed. The step response can give an indication of whether the overall tendencies of the models are correct. Furthermore it indicates the response time of the systems. On Figure 8.1 is a step response of each of the models for roll, pitch and yaw shown.

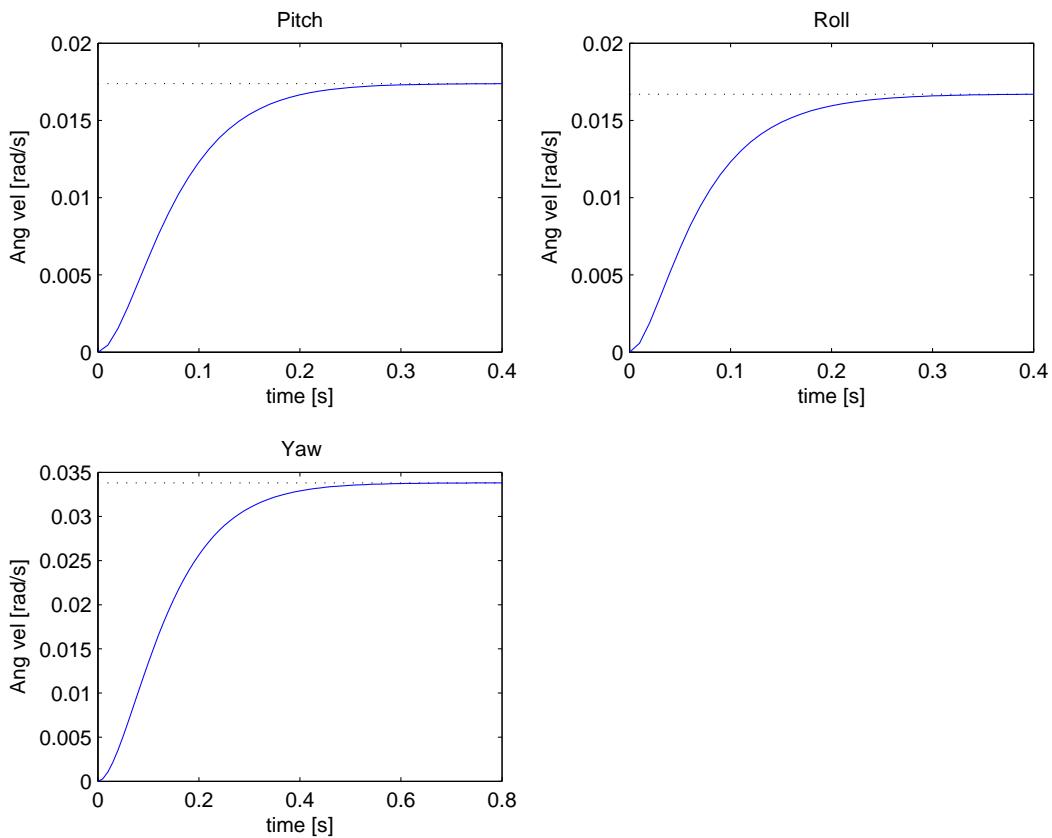


Figure 8.1: Step response of the three transfer functions (pitch, roll and yaw)

When activating the sticks on the RC-controller the output has a value between -100 and 100. Positive directions are indicated on Figure 8.2. All positive input should result in positive steps as well. From the step responses this tendency can be observed.

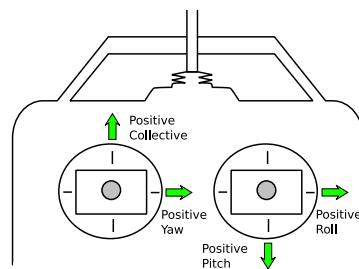


Figure 8.2: Indication of positive inputs from the RC-controller

The step response for pitch and roll are much alike, as expected because of the symmetry of the quad rotor. Whereas the step response of yaw is slower. This corresponds very well to the behaviour observed when flying manually.

8.1.2 Position of the Poles

The poles can be seen in Table 8.1. From the Table it is shown that all poles are in the negative half plane which indicates the models are stable. This agrees with observations.

Transfer function	Poles
$H_{\dot{\phi}}(s)$	-49.5, -17.6
$H_{\dot{\theta}}(s)$	-24.7, -24.7
$H_{\dot{\psi}}(s)$	-13.7, -13.7

Table 8.1: The poles of the estimated transfer functions

From the Table it is also seen that there is no complex part in the poles. This is normal unlikely when describing a system of this kind. The transfer functions are estimated with no restrictions on real pole placement, so it is likely that the described poles are true. An explanation can be that the poles observed are the poles of the closed loop system consisting of the quad rotor dynamics and the internal controllers. The poles of the closed loop system would likely be designed as real.

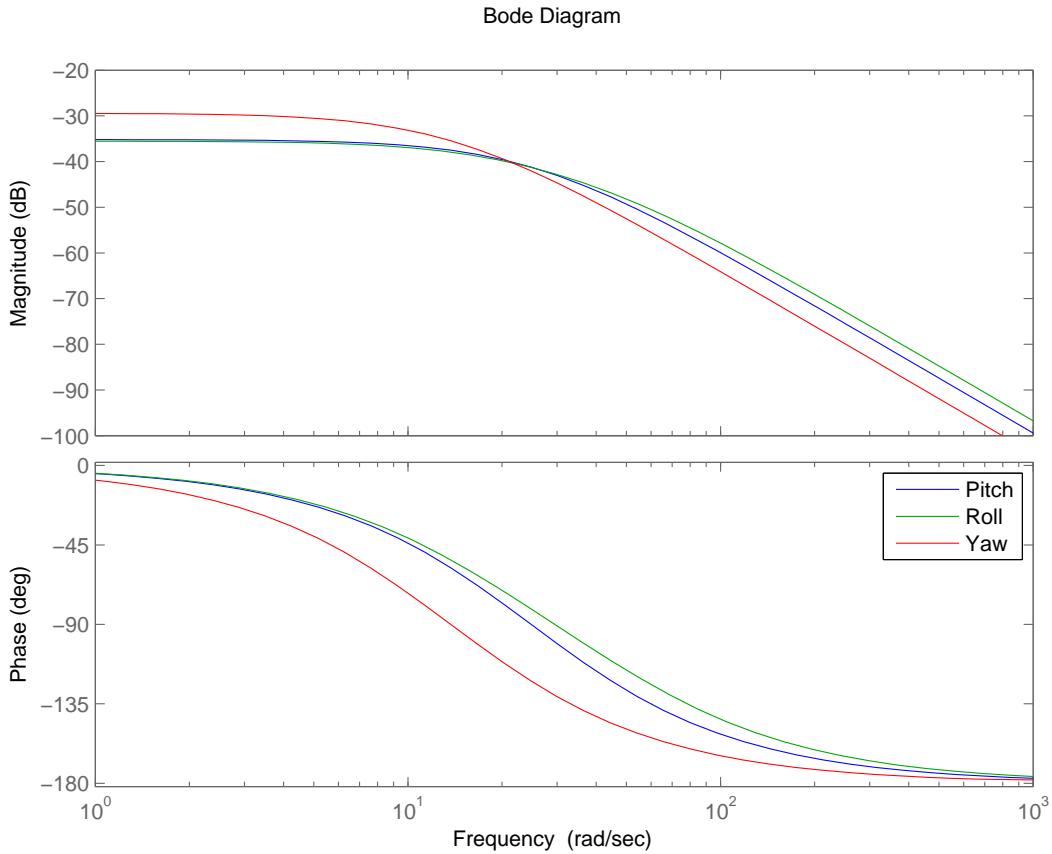


Figure 8.3: The frequency response of the three model transfer functions

On Figure 8.3 the frequency response is shown. As previously assumed the model looks like a low-pass filter with a low cut-off frequency. Also here it is seen that the model for yaw is the slowest of the models. From the phase plot it can be observed that yaw has a significant phase delay already at low frequencies (approx -60° at 10 Hz). Whereas the other models have a lower phase delay.

8.1.3 Simulation of the Model

A good way of investigating whether the model approximates reality is to directly compare the simulated output with reality. This have been done and the result is shown in Figure 8.4.

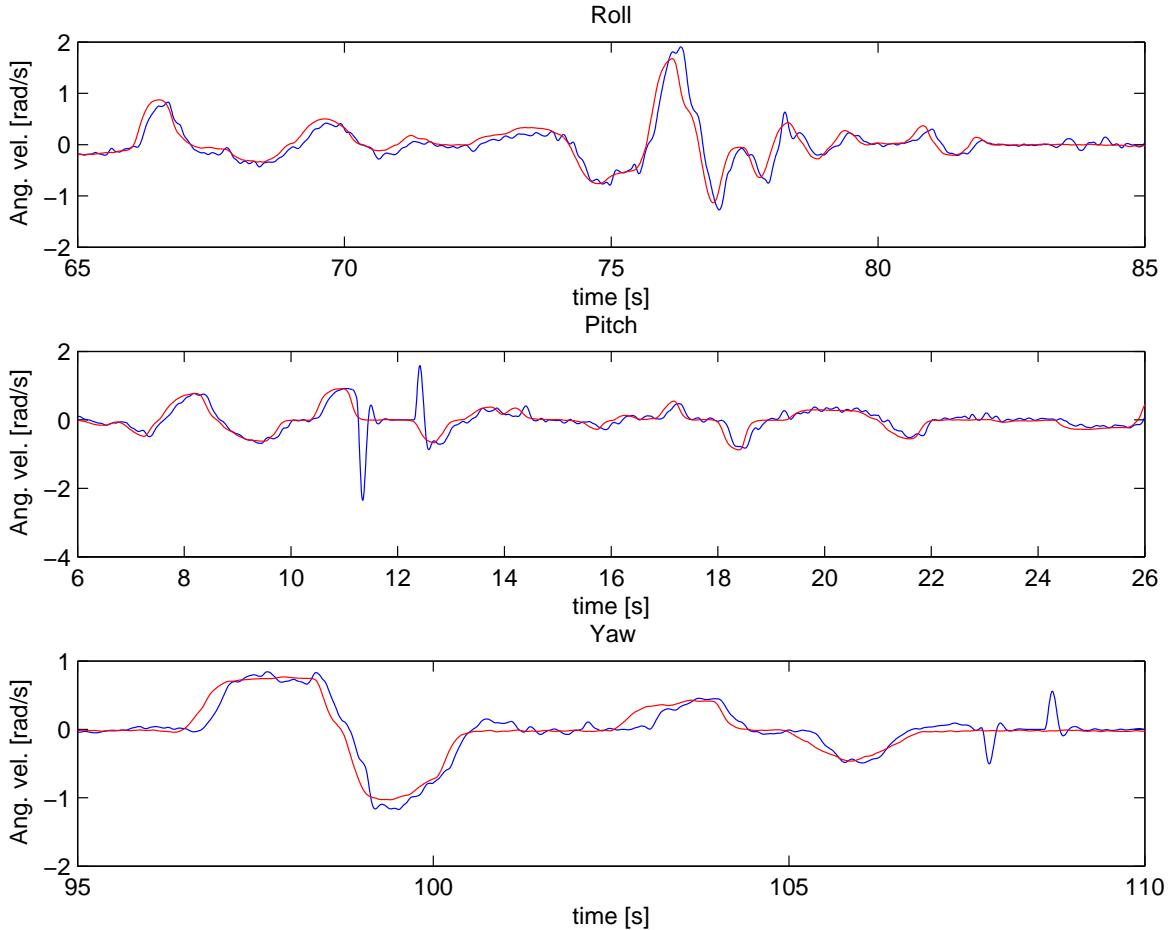


Figure 8.4: The response of the models (red lines) and the measured body rates (blue lines)

In Figure 8.4 the model output is seen, with the measured body rates. It is seen that the roll, pitch and yaw model follows both the dynamics and the amplitude of the measured signal. Two abnormal spikes can be seen in the plots for pitch and yaw. These are the effect of the Vicon system, which lost the orientation of the quad rotor for a short period of time. When this happen the orientation is set to zero. This differentiated and filtered signal then tends to have the abnormal spikes.

8.2 Validation of Discrete State Space Model

In this section the entire linearised state space model is analysed. The purpose of this analysis is to observe in what way the linearisation affects the system behaviour. This analysis is done using frequency response and eigenvalue plots.

8.2.1 Position of the poles

The positions of the poles of the discretised system are shown in Figure 8.5. From this it can be seen that all the poles are within the unit circle. This indicates that the system is stable, which makes sense as it is based on stable transfer functions. The poles are listed in Equation 8.1.

$$\text{poles} = [1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 0.78, 0.78, 0.87, 0.87, 0.87, 0.84, 0.61, 1.00, 1.00] \quad (8.1)$$

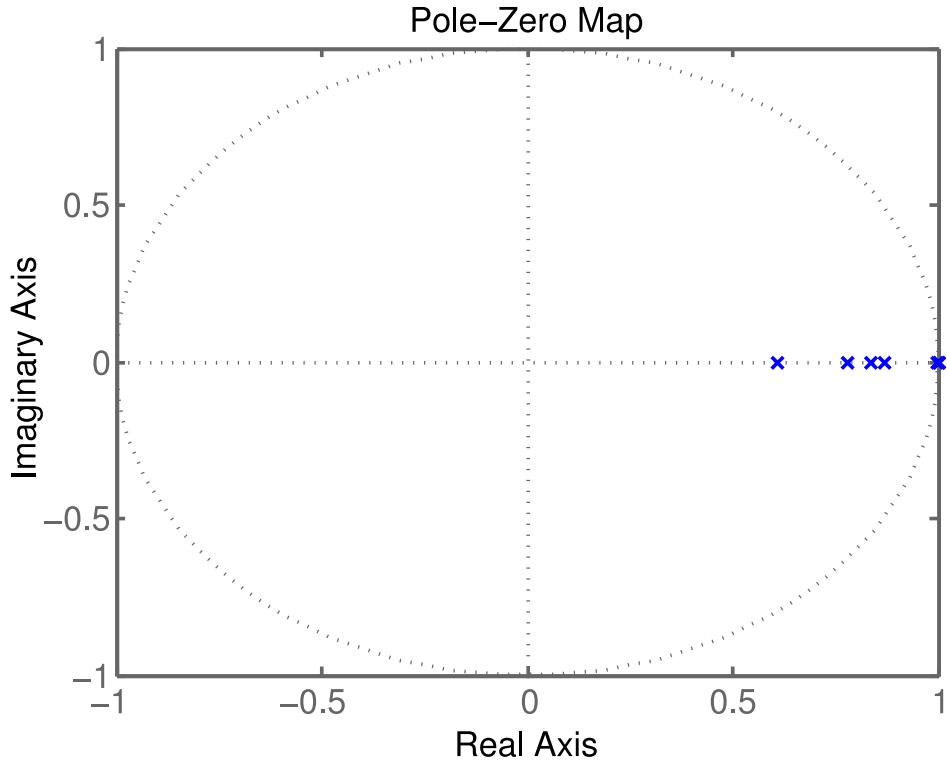


Figure 8.5: Map of the poles and zeroes in the system

In Figure 8.6 a bode plot is showing the frequency response from the roll, pitch and yaw control inputs to the roll, pitch and yaw angle outputs. As these are based on the transfer functions calculated in Chapter 4.1, but with some cross-couplings and an extra integral, it looks similar the frequency response of the transfer functions as shown in Figure 8.3, but the phase is shifted of 90° which is due to the integral.

In Figure 8.7 the bode plot of the transfer function from the roll and pitch inputs to the x and y-positions is shown. It can be observed that the magnitude of the two functions follow each other nicely, but that pitch is phase shifted by -180° . This can be explained by the fact that a positive pitch gives a negative movement in the x-axis, while a positive roll gives a positive movement in the y-axis.

In Figure 8.8 the response from the Collective input to the z-axis position output is shown. Here it can be seen that the phase margin is only 12.8° . This means that if the gain for the controller for the collective input is too large, the system can become unstable.

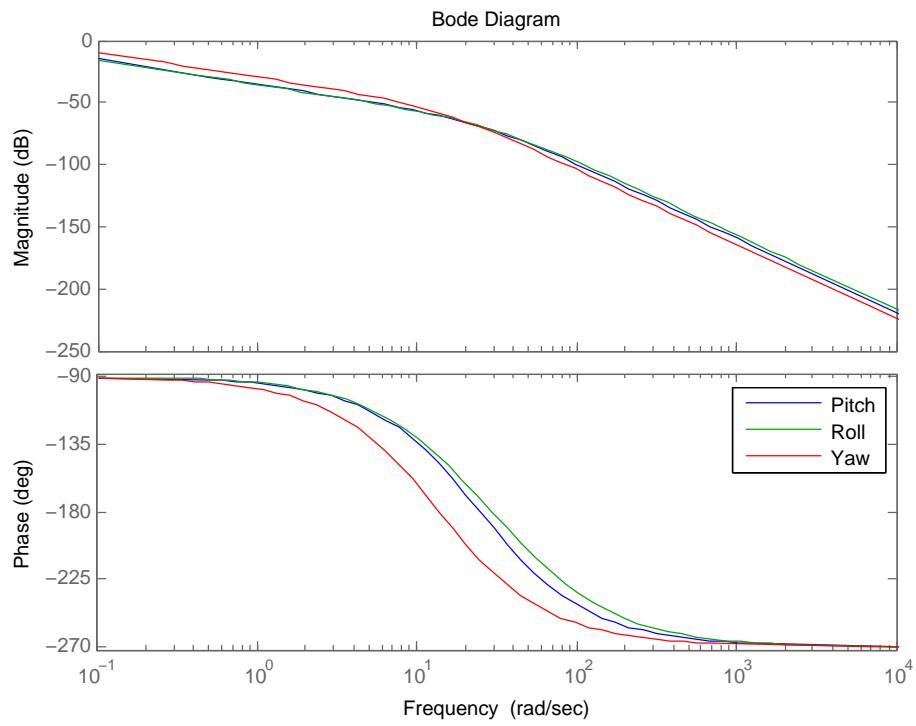


Figure 8.6: Bode plot of the response from the inputs roll, pitch and yaw to the outputs roll, pitch and yaw angles on the linearised system

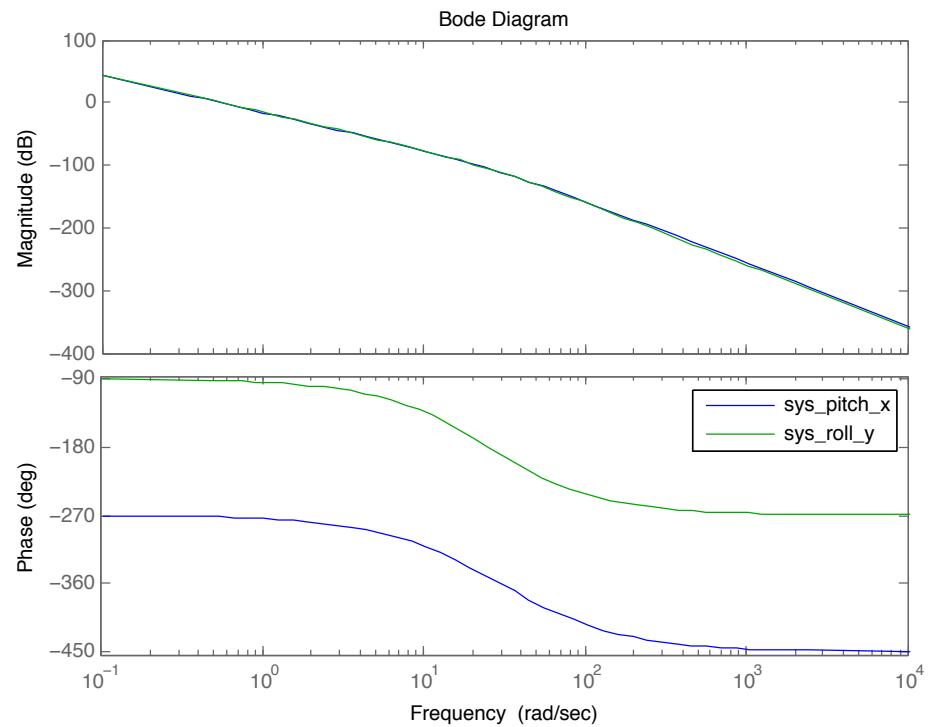


Figure 8.7: Bode plot of the response from the inputs roll and pitch to the outputs x and y-position on the linearised system

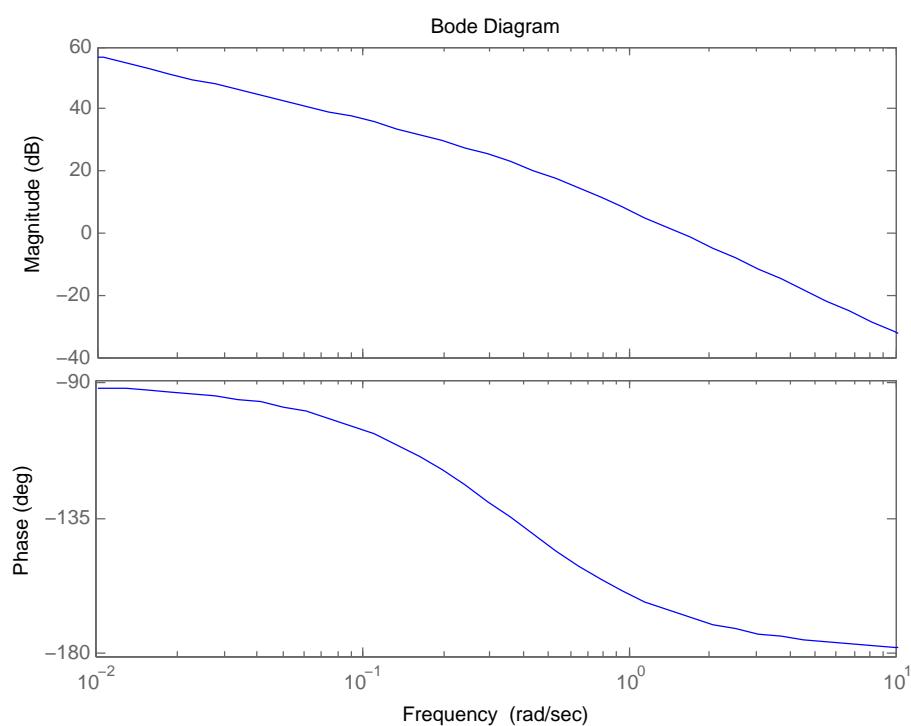


Figure 8.8: Bode plot of the response from the input "Collective" to the output "z-position" on the linearised system

8.3 Overall Model Evaluation

Through this chapter several aspects of the models have been analysed. A model was created based on a first principles model and expanded by transfer functions obtained using system identification.

It was concluded that the model is probably not valid with high frequent input signals, since it have not been possible to use a high frequency signal for the parameter estimation. This was due to a low-pass filter found in the RC-transmitter hardware. The models are also estimated with no attitude excitations higher than ± 20 degrees. Therefore only in this range is the model considered valid.

As presumed earlier the structure of the transfer functions as a series of low-pass filters seems as a good estimation. Furthermore simulations of the model follows the measured orientation of the real flights.

The model was the linearised and discretised, and it was found that the frequency responses from the linearised model has the same characteristics as the parameter estimated transfer functions.

Overall the models found is presumed sufficient for usage in control calculations and simulations.

Part II

Controlling X-3D-BL Quad Rotor

Table of Contents

9	Introduction to Controllers	47
10	PID Controller	48
10.1	Designing the PID Controller	48
10.2	Implementation	49
11	Linear Quadratic Controller	51
11.1	The Discrete State Space Controller	51
11.2	Discrete Time LQ Control	52
11.3	Full Order Observer	55
11.4	Introducing Reference	58
11.5	Integral Action	61
11.6	Gain Scheduling as Yaw Compensation	63
11.7	Simulations	64
12	Reference Tracking	68
12.1	Tracking	68
13	Linear Controller Analysis	71
14	Evaluation of Controllers	74
14.1	Hover Evaluation	74
14.2	Step Input Evaluation	80
14.3	Sinusoidal Input Evaluation	83
14.4	Tracking Evaluation	85
14.5	Summary of Controller Evaluation	88

Chapter 9

Introduction to Controllers

The objective of the controller is to keep the X-3D-BL quad rotor at a reference. To achieve this two controller strategies are considered. The first is a hand tuned, model independent, PID controller, and the second is a model based LQ controller. The two controllers are chosen, in order to analyse the differences between model dependant and model independent control.

The PID controller is a hand tuned controller, only the implementation of this will be discussed. Since the LQ controller is model based, the calculations and design considerations are closely based on model performance. Since the LQ controller is model based and highly dependant on the model performance, more effort will be put in the derivations of different substructures of this.

The two controllers are designed independently, but the same tests are performed on both controller designs, in order to investigate similarities and differences of the two controllers. The tests for both controllers are:

- **Hover tests:** The quad rotor is set to hover at a height of one meter and the systems long term ability to keep the fixed position is investigated.
- **Step tests:** The quad rotor is set in hover and a series of steps are performed in x, y and z positions.
- **Known continuous movement:** In order to test the continuous dynamics of the system, a series of sinusoidal are set as references, along with ramp signals for yaw.
- **Random continuous movement:** The platform is set to roam within a 2x2 meter space, and the quad rotor is set to track the reference generated by the position of the platform.

The initial objective includes landing on the platform. It has been chosen to focus more on the controllers making the quad rotor fly and not focus on a landing algorithm. If a landing algorithm was to be implemented it is necessary to handle external influences occurring when flying near ground such as ground effect.

PID Controller

The priority of this project has been to design a model based controller. But to gain an understanding of how the system (Vicon, ConLink, X-3D-BL, etc.) works in practice and to get the quad rotor flying relatively early in the project process, PID controllers have been designed for position and orientation, respectively.

10.1 Designing the PID Controller

The PID controllers are designed with intuitive choices based on experiments and hand tuning and should thus be seen as a preliminary controller. As a start P-controllers for control of the attitude and vertical velocity were designed/tuned individually with one degree of freedom at a time, with a pilot controlling the remaining three input signals.

Together those four P-controllers enabled the quad rotor to stay airborne, autonomously, except when drifting out of Vicon range.

As a final step position control where implemented in the form of three PI-controllers (one for each axis x,y and z) cascade coupled with the previously designed controllers for yaw, pitch, roll and collective. Because of the relatively short distances in the limited flying space of the motion tracking lab, it has been chosen to give the PI-controllers position references directly. The output of the three PI-controllers are via a direct cosine matrix transformed into the body frame and passed on as references to a simple velocity P-controller whose output goes to the Roll, Pitch and collective -controllers, respectively. The equations from the x-reference to the controller input, where the angles for simplicity are set to 0 to avoid the rotations, are:

$$S_{\omega_\phi} = k_\phi \cdot (\phi_{ref} - \phi) \quad (10.1)$$

$$\phi_{ref} = v_{x,ref} - (K \cdot v_x) \quad (10.2)$$

$$v_{x,ref} = (Kp_x + \frac{1}{s}Ki_x) \cdot (x_{ref} - x) \quad (10.3)$$

The control structure is depicted on Figure 10.1.

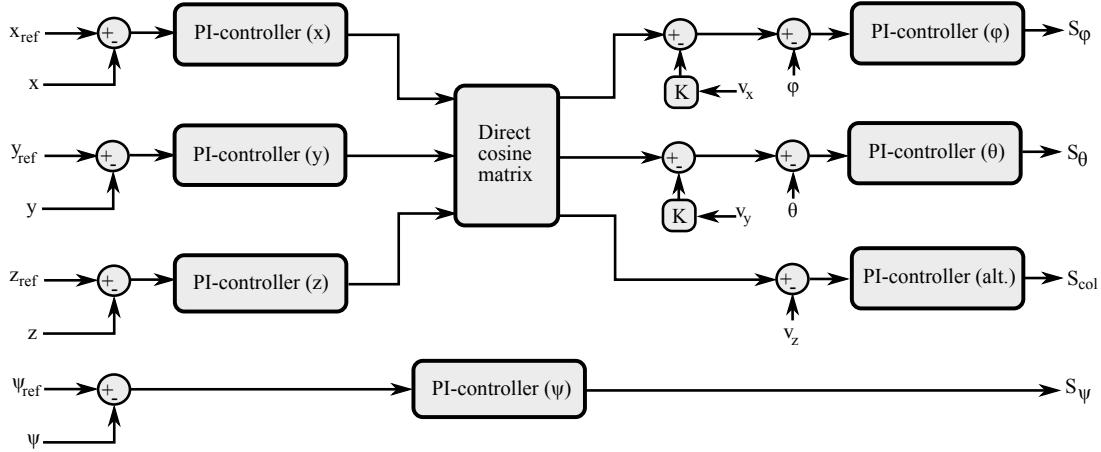


Figure 10.1: Structure of the PID controller for the X-3D-BL

10.2 Implementation

The PID-control of the X-3D is implemented in Simulink, communicating with the X-3D via the Conlink block and receiving sensor inputs from the Vicon block. The control setup is depicted in Figure 10.2. The Conlink

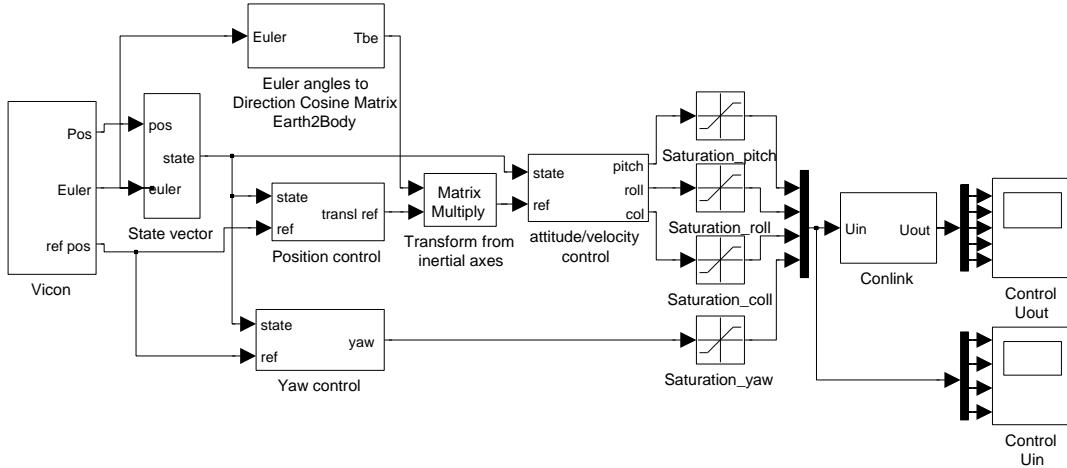


Figure 10.2: Implemented PID controller in Simulink blocks

block takes 6 signals as input, however only 4 are used (collective, pitch, yaw and roll). The Vicon block provides the Euler angles and the position measured by the Vicon MX System, whilst the reference signal is a coordinate set based on the measured position of the moving platform that the X-3D-BL is supposed to land on. The state vector block calculates a 12 element vector with position, body frame velocity, Euler angles and body rates, based on the measured position and attitude.

The content of the state vector block can be seen in Figure 10.3. Two second order Butterworth filters are implemented to reduce the noise amplification caused by the derivations. Both filters has a cutoff frequency of 25 Hz, which has been found to be sufficient to gain the desired noise suppression.

The control parameters of the control setup displayed in Figure 10.2 are contained in the position, attitude/velocity and yaw control blocks, respectively. Each of the controller blocks contains the general PID implementation,

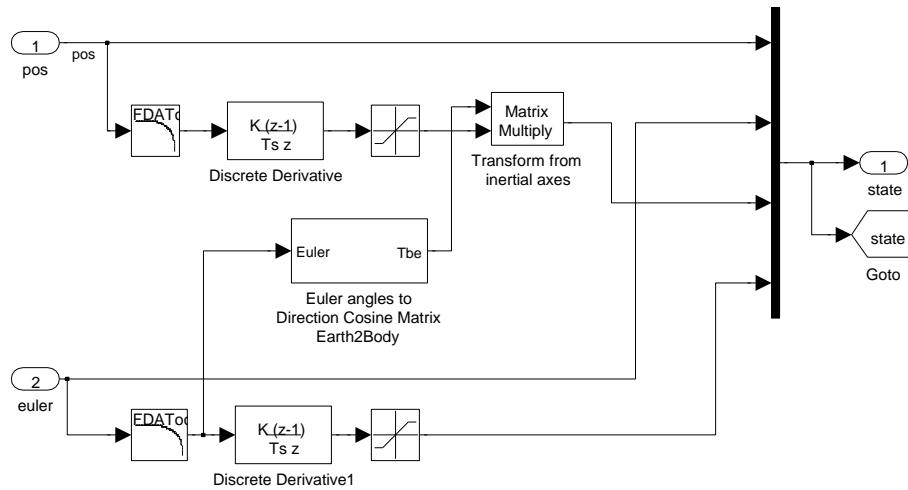


Figure 10.3: State vector simulink block

shown on Figure 10.4, the gain values are provided in Table 10.1.

In Simulink Block:	Control Variable	Kp	Ki	Kd
Position control	Position X	0.2	0.03	0
Position control	Position Y	0.2	0.03	0
Position control	Position Z	0.45	0.15	0
Attitude/velocity control	Roll	200	0	0
Attitude/velocity control	Pitch	200	0	0
Attitude/velocity control	velocity Z	30	0	0
Yaw control	Yaw	50	5	0

Table 10.1: Controller gains

The input to the Discrete time integrator, named auto, resets the integrator at the instance a transaction from manual flight to autonomous flight occurs. The subtraction of horizontal velocities, shown on Figure 10.1 is included in the Attitude/Velocity Control block, with $K = 0.2$.

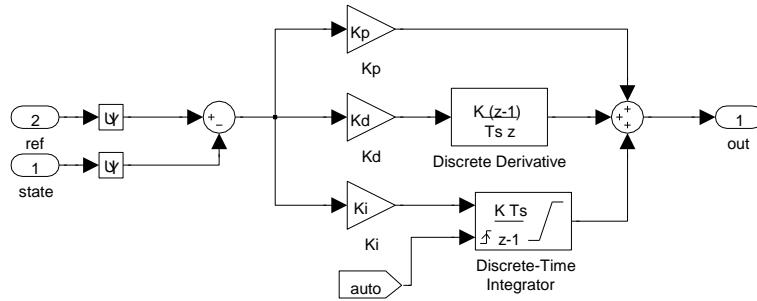


Figure 10.4: PID structure in simulink

The controller setup described in the above should as mentioned previously be seen as a preliminary controller and is used primarily for comparison purposes in testing the LQ Controller, whose design is treated in the next chapter.

Linear Quadratic Controller

The purpose of this section is to describe the LQ controller, developed for the X-3D-BL quad rotor. First a general introduction to LQ control design is presented. Later a LQ feedback controller is designed latterly adding reference, full-order observer and integral action. Finally simulations of the system with controllers are presented, showing the flight tests compared with the simulation.

11.1 The Discrete State Space Controller

In earlier chapters a linear model for the quad rotor were developed. The system can be written on the following state space representation:

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) \quad (11.1)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \quad (11.2)$$

The control law of the discrete time system can be described as a linear feedback function of the states \mathbf{x} , given by the feedback matrix \mathbf{K}

$$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k) \quad (11.3)$$

An illustration of the discrete time state space system, with the feedback matrix \mathbf{K} can be seen in Figure 11.1. A description of the closed loop system can be performed by inserting the control law in the state space system equations:

$$\mathbf{x}(k+1) = (\Phi - \Gamma\mathbf{K})\mathbf{x}(k) \quad (11.4)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \quad (11.5)$$

The closed loop poles of the system are the eigenvalues of the matrix $\Phi - \Gamma\mathbf{K}$.

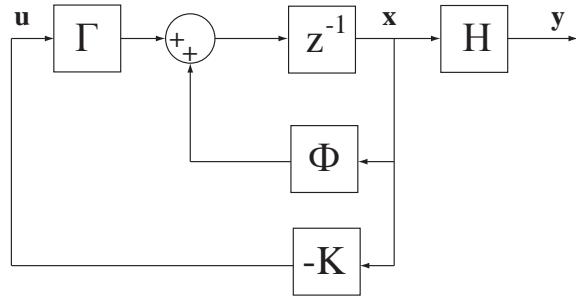


Figure 11.1: The LQ state feedback

11.2 Discrete Time LQ Control

As the objective of the developed controller is to keep the output of the system as close to a reference, with the least amount of control effort, the LQR method is used for controller design. This controller is chosen, because it offers a balance between tight use of control signal[14, p. 15]

The purpose of this method is to determine a input signal $u(k)$ such that the performance function (11.6) is minimised. The performance function used in linear quadratic control, in discrete time, is a sum of weighted squared states $x(k)$ and weighted squared inputs $u(k)$ [14, p. 4]:

$$\mathcal{I} = \sum_{k=0}^N x^T(k) Q_1 x(k) + u^T(k) Q_2 u(k) \quad (11.6)$$

where N is the finite time horizon, Q_1 is the weighting matrix of the states and Q_2 is the weighting matrix of the inputs. This can be written with a shorter notation:

$$\mathcal{I} = \sum_{k=0}^N \mathcal{H}(x(k), u(k)) \quad (11.7)$$

The performance function and parameters in this is where the weighting of larger control signal versus large states is determined. Thus the feedback is calculated by using the control law in equation 11.3 to minimise the performance function.

In a system with n states and p inputs the performance function for a given system can be expanded to:

$$\mathcal{I} = \sum_{k=0}^N (x^T(k) Q_1 x(k) + u^T(k) Q_2 u(k)) + x^T(N) Q_N x(N) \quad (11.8)$$

Q_N is introduced to perform a separate weighting of the last value of the states, i.e. the initialisation of the states. Q_N will have dimensions $(n \times n)$, Q_2 will have dimensions $p \times p$ and Q_1 will have dimensions $n \times n$. All three matrices are positive semi definite[14, p. 9].

It can be found that for a first order system that the inputs, $u(k)$, are proportional to the current state, $x(k)$, for each step. Thus the performance function can be rewritten as a quadratic function of the current state $x(k)$ for the samples $k \dots N$:

$$u = -K x(k) \quad (11.9)$$

$$\mathcal{J}_k^N(x(k)) = x^T(k) S(k) x(k) \quad (11.10)$$

where the matrices $\mathbf{K}(k)$ and $\mathbf{S}(k)$ describes the linear properties from $\mathbf{u}(k)$ to $\mathbf{x}(k)$ and the quadratic solution to $\mathbf{x}(k)$ respectively. $\mathbf{K}(k)$ is of dimension $(p \times n)$ and $\mathbf{S}(k)$ is of dimension $(n \times n)$. The general dynamic expression for the quadratic solution of $\mathbf{x}(k)$ can be described as [14, p. 10]:

$$\mathcal{J}_k^N(\mathbf{x}(k)) = \min_{\mathbf{u}(k)} [\mathcal{H}(\mathbf{x}(k), \mathbf{u}(k)) + \mathcal{J}_{k+1}^N(\mathbf{x}(k+1))] \quad (11.11)$$

which describes the iterative properties of the solution to the quadratic form of $\mathbf{x}(k)$.

Inserting the expressions for $\mathcal{H}(\mathbf{x}(k), \mathbf{u}(k))$ and $\mathcal{J}_{k+1}^N(\mathbf{x}(k+1))$ and solving this for the control signal $\mathbf{u}(k)$, the equation will yield the minimum performance[14, p. 10]:

$$\mathcal{J}_k^N(\mathbf{x}(k)) = \min_{\mathbf{u}(k)} [\mathbf{x}^T(k) \mathbf{Q}_1 \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k) + \mathbf{x}^T(k+1) \mathbf{S}(k+1) \mathbf{x}(k+1)] \quad (11.12)$$

$$= \min_{\mathbf{u}(k)} [\mathbf{x}^T(k) \mathbf{Q}_1 \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k) + (\Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k))^T \mathbf{S}(k+1) (\Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k))] \quad (11.13)$$

To find the optimal control signal \mathbf{u}^* , which minimises the performance function at time k , it is differentiated and put equal to zero (for convenience the argument k is omitted)[14, p. 11]:

$$\frac{\delta \mathcal{J}_k^N(\mathbf{x})}{\delta \mathbf{u}} = 2\mathbf{Q}_2 \mathbf{u} + 2\Gamma^T \mathbf{S}(k+1) (\Phi \mathbf{x} + \Gamma \mathbf{u}) = 0 \quad (11.14)$$

If $[\mathbf{Q}_2 + \Gamma^T \mathbf{S}(k+1) \Gamma]$ is invertible, it can be solved with respect to the optimal control signal \mathbf{u}^* :

$$\mathbf{u}^*(k) = -[\mathbf{Q}_2 + \Gamma^T \mathbf{S}(k+1) \Gamma]^{-1} \Gamma^T \mathbf{S}(k+1) \Phi \mathbf{x}(k) \quad (11.15)$$

It is this optimal value, \mathbf{u}^* , which minimises the performance function and the proportional gain $\mathbf{K}(k)$ can then be rewritten as:

$$\mathbf{K}(k) = [\mathbf{Q}_2 + \Gamma^T \mathbf{S}(k+1) \Gamma]^{-1} \Gamma^T \mathbf{S}(k+1) \Phi \quad (11.16)$$

If Equation 11.16 is inserted into Equation 11.12 $\mathbf{S}(k)$ can be recursively expressed as:

$$\mathbf{S}(k) = \mathbf{Q}_1 + \mathbf{K}^T(k) \mathbf{Q}_2 \mathbf{K}(k) + ((\Phi - \Gamma \mathbf{K}(k))^T \mathbf{S}(k+1) (\Phi - \Gamma \mathbf{K}(k))) \quad (11.17)$$

11.2.1 Stationary LQ Control

Most often LQ control will use values of $\mathbf{K}(0)$ and $\mathbf{S}(0)$ determined for $N \rightarrow \infty$. This yields, that the expressions for controller calculations:

$$\mathbf{K}(k) = [\mathbf{Q}_2 + \Gamma^T \mathbf{S}(k+1) \Gamma]^{-1} \Gamma \mathbf{S}(k+1) \Phi \quad (11.18)$$

$$\mathbf{S}(k) = \mathbf{Q}_1 + \Phi^T \mathbf{S}(k+1) \Phi - \Phi^T \mathbf{S}(k+1) \Gamma [\mathbf{Q}_2 + \Gamma^T \mathbf{S}(k+1) \Gamma]^{-1} \Gamma^T \mathbf{S}(k+1) \Phi \quad (11.19)$$

will tend to go towards steady state, when the performance index's sum tends towards infinity[14, p. 19]. As N goes towards infinity, $\mathbf{K}(0) = \mathbf{K}(k) = \mathbf{K}(k+1) = \mathbf{K}$ and $\mathbf{S}(0) = \mathbf{S}(k) = \mathbf{S}(k+1) = \mathbf{S}$, are solutions to the steady state Riccati equations:

$$K = [\mathbf{Q}_2 + \Gamma^T \mathbf{S} \Gamma]^{-1} \Gamma^T \mathbf{S} \Phi \quad (11.20)$$

$$S = \mathbf{Q}_1 + \Phi^T \mathbf{S} \Phi - \Phi^T \mathbf{S} \Gamma [\mathbf{Q}_2 + \Gamma^T \mathbf{S} \Gamma]^{-1} \Gamma^T \mathbf{S} \Phi \quad (11.21)$$

11.2.2 Choice of Weight-Matrices

When using the LQ method for feedback control, the design parameters are the \mathbf{Q}_1 and \mathbf{Q}_2 matrices. The number of design parameters becomes very large ($r^2 + n^2$), but the solution to the control problem, the feedback matrix \mathbf{L} , will be uniquely determined.

The weighting matrices \mathbf{Q}_1 and \mathbf{Q}_2 , and their relationship determines, the trade-off between performance and control effort. By reducing these matrices to diagonal, the number of design parameters will be lower to $r + n$. In this case, the diagonal elements of $\mathbf{Q}_1(i, i)$ and $\mathbf{Q}_2(j, j)$ can be used as measures of the relative punishment of large deviations in $\mathbf{x}_i(k)$ and $\mathbf{u}_j(k)$.

11.2.3 Controllability

Before a linear controller can be designed, the controllability of the system must be investigated. This is done to ensure that all states are controllable through the inputs. The controllability is examined by computing the controllability matrix and ensuring, that the rank of this matrix is full, meaning that the rank is equal to the number states in the system (n). The controllability matrix is computed as [13, p.845]:

$$\mathcal{C} = [\Gamma \Phi\Gamma \dots \Phi^{n-1}\Gamma] \quad (11.22)$$

$$\text{rank}(\mathcal{C}) = \text{rank}([\Gamma \Phi\Gamma \dots \Phi^{n-1}\Gamma]) = n \quad (11.23)$$

The system is computed to have:

$$\text{rank}(\mathcal{C}) = 15 \quad (11.24)$$

which is equal to the number of states in the system, hence all states in the system are controllable through the inputs.

11.2.4 Designing LQ Controller

When designing the LQ controller, the design parameters are, as previous mentioned, the \mathbf{Q}_1 and \mathbf{Q}_2 matrices. Thus appropriate choices of the weight in these matrices have to be made. At this point the actual \mathbf{Q}_1 and \mathbf{Q}_2 matrices are not determined but will be tuned in a later section.

The stationary LQ controller is calculated solving the Riccati equation shown in Equation 11.20. The calculations is made as an algorithm as seen in Algorithm 1.

Input: Discrete system matrices and weight matrices \mathbf{Q}_1 and \mathbf{Q}_2

Output: Feedback matrix \mathbf{K}_s

$\mathbf{S} = \mathbf{Q}_1$;

$\mathbf{K}_{old} = 0$;

$\mathbf{K} = (\mathbf{Q}_2 + \mathbf{\Gamma}^T \mathbf{S} \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T \mathbf{S} \mathbf{\Phi}$;

$\mathbf{S} = \mathbf{Q}_1 + \mathbf{\Phi}^T \mathbf{S} \mathbf{\Phi} - \mathbf{K}^T \mathbf{\Gamma}^T \mathbf{S} \mathbf{\Phi}$;

while $|\mathbf{K} - \mathbf{K}_{old}| > 10^{-6} \cdot |\mathbf{K}|$ **do**

$\mathbf{K}_{old} = \mathbf{K}$;

$\mathbf{K} = (\mathbf{Q}_2 + \mathbf{\Gamma}^T \mathbf{S} \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T \mathbf{S} \mathbf{\Phi}$;

$\mathbf{S} = \mathbf{Q}_1 + \mathbf{\Phi}^T \mathbf{S} \mathbf{\Phi} - \mathbf{K}^T \mathbf{\Gamma}^T \mathbf{S} \mathbf{\Phi}$;

end

Algorithm 1: Solves the Riccati equation shown in Equation 11.20

The implementation, calculates the feedback matrix \mathbf{K}_s , which minimises the performance function. The function takes the weight, and system matrices as inputs, setting the initial value of $\mathbf{S}(k)$ as the value of the \mathbf{Q}_1 weight matrix.

11.3 Full Order Observer

In order to obtain full state feedback the non-measurable states, must be estimated by an observer, utilising the model of the system. The only states measurable by the system are the positions and the Euler angles, hence the rest of the states must be estimated. An illustration of the full order observer are shown in Figure 11.2.

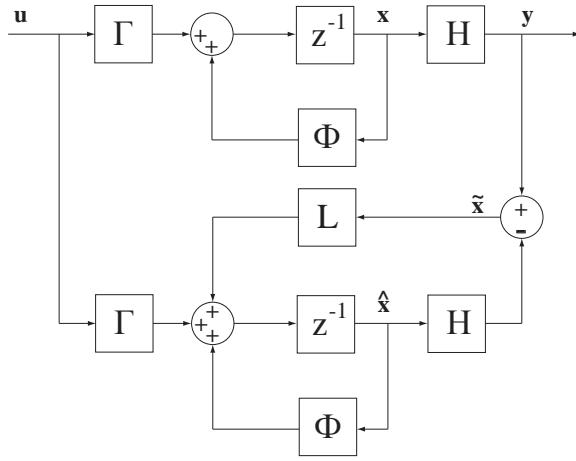


Figure 11.2: The full order observer, and the regular system

Since the non-measurable states are estimated, a difference between the estimated states (\hat{x}) and the true states (x), will exist. This error \tilde{x} is defined as:

$$\tilde{x} = x - \hat{x} \quad (11.25)$$

Because of inaccuracies in the developed model or faulty initialisation of the observer, it is possible that the estimated states can deviate largely from the true states. As a compensation, a feedback of the output error is introduced. This feedback matrix L is introduced to ensure that the estimated states will converge to the true states. The equation describing the estimated states is given by:

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + L(y(k) - H\hat{x}(k)) \quad (11.26)$$

By inserting $y(k) = Hx(t)$ and substituting the system equations, an equation for the estimator dynamics is by [13, p.634] found to be:

$$\tilde{x}(k+1) = (\Phi - LH)\tilde{x}(k) \quad (11.27)$$

In order to obtain a good estimate of the states, the error \tilde{x} should converge towards zero. This can be achieved by determining L such that $\Phi - LH$ has eigenvalues within the unit circle (stable poles). It is also required, that the observer, hence the dynamics of the state errors, is significantly faster than the dynamics of the system. As a consequence of this, the eigenvalues of the observer should be numerically larger than the controller poles.

The design of the controller and the observer can be done separately, due to the separation principle. This allows the feedback matrix K to be fed back from the observed states, rather than the measured states, giving full state feedback.

11.3.1 Observability

Before designing the full order observer, the observability of the system must be investigated. This is done, to make sure, that all states are observable through the observer. The observability is examined, by computing the rank of the observability matrix. If the rank is equal to the number of states, having full rank, the system is observable [13, p.634]:

$$\text{rank}(\mathcal{O}) = \text{rank} \begin{bmatrix} H \\ H\Phi \\ \vdots \\ H\Phi^{n-1} \end{bmatrix} = n \quad (11.28)$$

By using Matlab, the observability is investigated, and the system is found to have:

$$\text{rank}(\mathcal{O}) = 15 \quad (11.29)$$

Which is equal to the number of states in the system, meaning that all states are observable though the full order state observer.

11.3.2 Designing Full Order Observer

When designing the full order observer, the task is to construct the feedback matrix \mathbf{K} , such that $\Phi - \mathbf{H}\mathbf{L}$, has stable eigenvalues. As the system is presented in discrete time, the poles must be within the unit circle. In order to achieve fast error dynamics, a rule of thumb is, to place the poles of the observer 2-6 times faster than the fastest open loop system pole.

The closed loop system poles are the eigenvalues of $\Phi - \mathbf{K}\Gamma$, shown in Figure 11.3(a), and Equation 11.30.

Choosing a eigenvalue placement for the observer four times faster than the fastest pole in the loop system, yields the poles shown in Equation 11.31 and Figure 11.3(b).

$$p_{\text{open-loop}} = \begin{matrix} 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 0.7806 \\ 0.7806 & 0.8719 & 0.8719 & 0.8387 & 0.6096 & 1.0000 & 0.9964 \end{matrix} \quad (11.30)$$

$$p_{\Phi-\mathbf{H}\mathbf{L}} = \begin{matrix} 0.1534 & 0.1544 & 0.1554 & 0.1564 & 0.1574 & 0.1584 & 0.1594 & 0.1604 \\ 0.1614 & 0.1624 & 0.1634 & 0.1644 & 0.1654 & 0.1664 & 0.1674 \end{matrix} \quad (11.31)$$

The selected poles are used to calculate the observer gain \mathbf{L} . This is done by use of the pole placement method. This method is commonly used to select the feedback matrix \mathbf{K} , which places the eigenvalues of the closed loop system $\Phi - \mathbf{K}\Gamma$ at the specified locations. Since $\Phi - \mathbf{K}\Gamma$ is equal to $\Phi - \mathbf{H}\mathbf{L}$, the same method can be used for designing the observer, taking the transpositions into account.

The pole placement method is available through Matlab the function `place`, which returns the feedback matrix \mathbf{L} . The inputs to the functions are Φ^T , \mathbf{H}^T and an array of the specified pole locations. Algorithm 2 describes

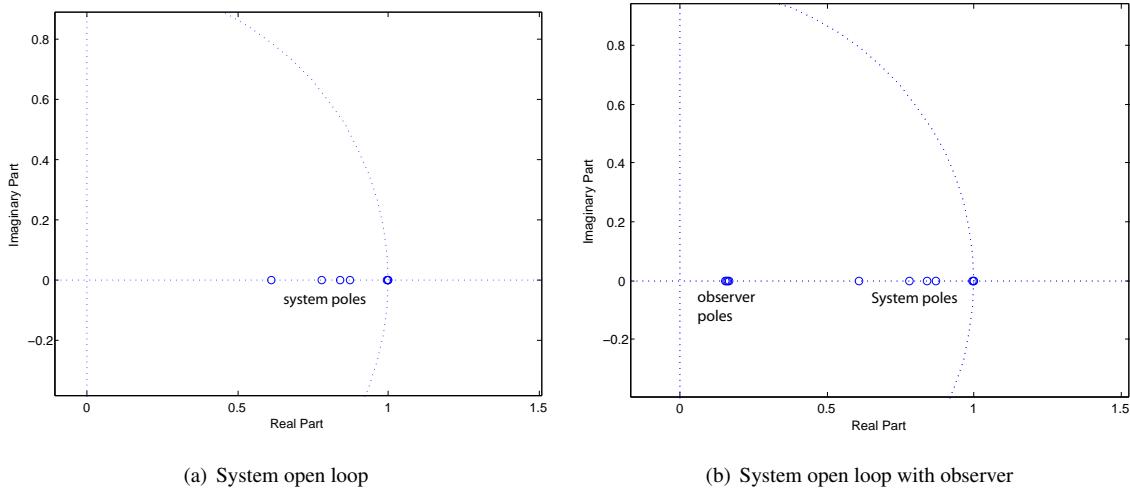


Figure 11.3: Pole/zero map of open loop and observer

how the \mathbf{L} matrix is calculated.

```

 $p_{scale} = 4;$ 
 $pole_{fastest} = \text{lowest eigenvalue of discrete system } \Phi;$ 
foreach Closed loop pole p do
  |  $poles_{cl} = (pole_{fastest}/p_{scale}) + 0.001 \cdot p;$ 
end
 $L = place(\Phi^T, H^T, poles_{cl})^T;$ 
```

Algorithm 2: Calculates the L matrix

The observer feedback \mathbf{L} have been calculated as in Equation 11.32. The values seen in the equation is rounded to nearest integer for a better overview.

$$\mathbf{L} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 70 & 0 & 0 & 0 & 0 & 0 \\ 0 & 70 & 0 & 0 & 0 & 0 \\ 0 & 0 & 70 & 0 & 0 & 0 \\ 1 & -9 & 4 & 98 & 0 & 0 \\ 0 & 1 & 2 & 0 & 113 & 0 \\ 0 & 0 & 0 & 0 & 0 & 143 \\ 22 & -135 & 65 & -141 & 0 & 0 \\ -7 & 18 & 49 & -1 & 624 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2548 \end{bmatrix} \quad (11.32)$$

The observer design can be combined with the state feedback controller, to obtain full state feedback. The feedback states are now, the estimated states from the observer, as illustrated in Figure 11.4. Full state information is achieved through observer, making the unmeasurable states suitable for feedback.

According to the separation principle, the closed loop equations are obtained by combining the dynamic equa-

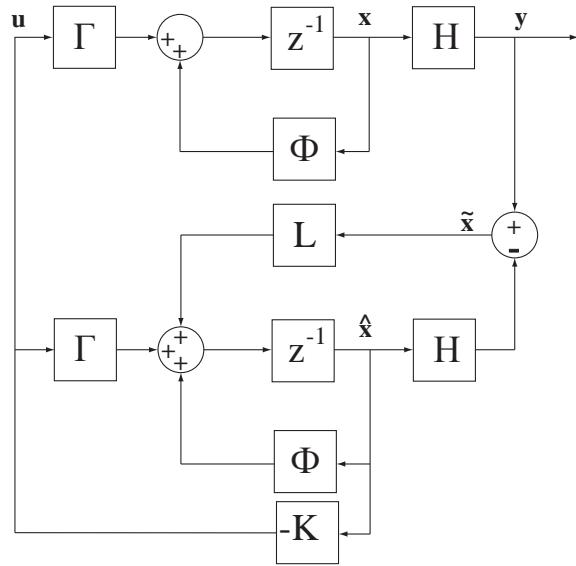


Figure 11.4: The full order observer, with state feedback from the estimated states.

tions of the system and the full order observer. As seen in Figure 11.4 the expression for the system can be rewritten to:

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) \quad (11.33)$$

$$= \Phi\mathbf{x}(k) + \Gamma\mathbf{K}\hat{\mathbf{x}}(k) \quad (11.34)$$

$$= \Phi\mathbf{x}(k) + \Gamma\mathbf{K}(\mathbf{x} - \tilde{\mathbf{x}})(k) \quad (11.35)$$

$$= (\Phi + \Gamma\mathbf{K})\mathbf{x}(k) - \Gamma\mathbf{K}\tilde{\mathbf{x}}(k) \quad (11.36)$$

By combining the observer error equation and the equation above, the complete system representation can be derived. The closed loop poles of the system are the eigenvalues of:

$$\begin{bmatrix} \tilde{\mathbf{x}}(k+1) \\ \mathbf{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \mathbf{L}\mathbf{H} & \mathbf{0} \\ -\Gamma\mathbf{K} & \Phi + \Gamma\mathbf{K} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}(k) \\ \mathbf{x}(k) \end{bmatrix} \quad (11.37)$$

11.4 Introducing Reference

The purpose of introducing a reference input, is to make it possible to give a reference value to the position and the yaw angle. By introducing this reference, the controller will be able to keep set values for the position and yaw angle.

In order to control the system with the reference, a model of the reference is introduced along the system output error defined as [14, p. 31]:

$$\mathbf{e}(k) = \mathbf{r}(k) - \mathbf{y}(k) = \mathbf{H}_r\mathbf{x}_r(k) - \mathbf{H}_s\mathbf{x}_s(k) \quad (11.38)$$

Where \mathbf{H}_r is the reference model output matrix and $\mathbf{x}_r(k)$ is the reference states.

The performance function, subject to minimisation will now be:

$$\mathcal{I} = \sum_{k=0}^{N-1} (\mathbf{e}^T(k)\mathbf{Q}_{1e}\mathbf{e}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)) \quad (11.39)$$

In order to perform this minimisation, the system state $\mathbf{x}_s(k)$ is augmented by the reference state $\mathbf{x}_r(k)$, giving the augmented state vector $\mathbf{x}(k) = [\mathbf{x}_s^T(k) \ \mathbf{x}_r^T(k)]^T$. Giving the augmented system, described by [14, p. 31]:

$$\begin{bmatrix} \mathbf{x}_s(k+1) \\ \mathbf{x}_r(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_s & \mathbf{0} \\ \mathbf{0} & \Phi_r \end{bmatrix} \begin{bmatrix} \mathbf{x}_s(k) \\ \mathbf{x}_r(k) \end{bmatrix} + \begin{bmatrix} \Gamma_s \\ \mathbf{0} \end{bmatrix} \mathbf{u}(k) \quad (11.40)$$

$$\mathbf{e}(k) = \begin{bmatrix} -\mathbf{H}_s & \mathbf{H}_r \end{bmatrix} \begin{bmatrix} \mathbf{x}_s(k) \\ \mathbf{x}_r(k) \end{bmatrix} \quad (11.41)$$

Introducing the augmented system matrices as:

$$\Phi = \begin{bmatrix} \Phi_s & \mathbf{0} \\ \mathbf{0} & \Phi_r \end{bmatrix} \quad \Gamma = \begin{bmatrix} \Gamma_s \\ \mathbf{0} \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} -\mathbf{H}_s & \mathbf{H}_r \end{bmatrix} \quad (11.42)$$

Yields the rewritten performance function as:

$$\mathcal{I} = \sum_{k=0}^{N-1} (\mathbf{e}^T(k) \mathbf{Q}_{1e} \mathbf{e}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.43)$$

$$= \sum_{k=0}^{N-1} ((\mathbf{H}\mathbf{x}(k))^T \mathbf{Q}_{1e} (\mathbf{H}\mathbf{x}(k)) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.44)$$

$$= \sum_{k=0}^{N-1} (\mathbf{x}^T(k) \mathbf{H}^T \mathbf{Q}_{1e} \mathbf{H}\mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.45)$$

$$= \sum_{k=0}^{N-1} (\mathbf{x}^T(k) \mathbf{Q}_1 \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.46)$$

Which has the same structure, as the original performance function. Thus the feedback can be determined by use of the standard calculation, performed on the augmented system [14, p. 32]. Notice, that the \mathbf{Q}_1 weight matrix is now a weighting of the controller error, calculated as:

$$\mathbf{Q}_1 = \begin{bmatrix} -\mathbf{H}_s^T \\ \mathbf{H}_r^T \end{bmatrix} \mathbf{Q}_{1e} \begin{bmatrix} -\mathbf{H}_s & \mathbf{H}_r \end{bmatrix} \quad (11.47)$$

Using a reference model of a step, shown in Figure 11.5 in the autonomous model, yields:

$$\mathbf{x}_r(k+1) = \mathbf{I}\mathbf{x}_r(k) \quad (11.48)$$

$$\mathbf{r}(k) = \mathbf{H}_r \mathbf{x}_r(k) \quad (11.49)$$

The optimal calculated feedback matrix will now be a partition of the state vector in system and reference states.

$$\mathbf{K}(k) = \begin{bmatrix} \mathbf{K}_s(k) & \mathbf{K}_r(k) \end{bmatrix} \quad (11.50)$$

The introduction of the step reference to the feedback system is illustrated in Figure 11.5 It is shown that by augmenting the system matrices, with the reference states, the reference gain, can be calculated as part of the optimal control problem, giving optimal values of the gain \mathbf{K}_r .

11.4.1 Designing Reference Input

When introducing the reference input, the performance function is altered, as described in Equations 11.43 and 11.47. The main difference to the performance function is, that instead of weighting the states, the weight is put on the difference between the feedback states, and the reference input. Thus the developed controller will punish regulator errors.

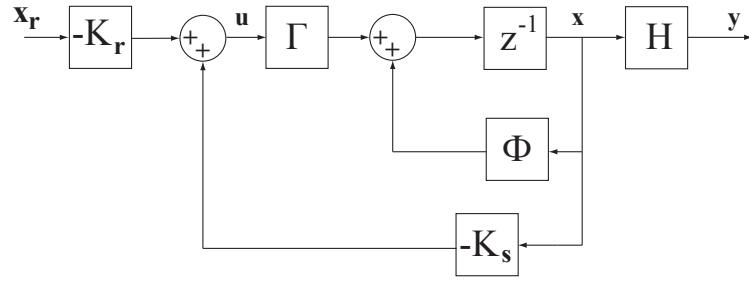


Figure 11.5: Introducing reference gain \mathbf{K}_r to the system, using autonomous model of a step input.

The performance function, subject to minimisation, through the feedback, is as stated earlier:

$$\mathcal{I} = \sum_{k=0}^{N-1} (\mathbf{x}^T(k) \mathbf{Q}_1 \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.51)$$

With the augmented system matrices, where:

$$\mathbf{Q}_1 = \begin{bmatrix} -\mathbf{H}_s^T \\ \mathbf{H}_r^T \end{bmatrix} \mathbf{Q}_{1e} \begin{bmatrix} -\mathbf{H}_s & \mathbf{H}_r \end{bmatrix} \quad (11.52)$$

Hence the design parameters for the introduction of the reference is the weight of the errors \mathbf{Q}_{1e} , the reference input matrix \mathbf{H}_r and the system input matrix \mathbf{H}_s .

The reference output matrix is constructed as, of a step reference:

$$\mathbf{H}_r = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.53)$$

As the inputs will give reference to x-position x , y-position y , z-position (elevation) z and yaw ψ of the quad rotor. The weights of the input errors, are iteratively designed as:

$$\mathbf{Q}_{1e} = \begin{bmatrix} 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix} \quad (11.54)$$

The optimal feedback is calculated, using Algorithm 1, which yields the optimal reference gain and the optimal system gain shown in Equations 11.55 and 11.56.

$$\mathbf{K}_r = \begin{bmatrix} 0 & -66.79 & 0 & 0 \\ 66.77 & 0 & 0 & 0 \\ 0 & 0 & -4.17 & 0 \\ 0 & 0 & 0 & -98.56 \end{bmatrix} \quad (11.55)$$

$$\mathbf{K}_s = \begin{bmatrix} 0 & 66.79 & 0 & 310.41 & 0 & 0 & 0 & 64.98 & 0 & 21.05 & 0 & 0 & 0.30 & 0 & 0 \\ -66.77 & 0 & 0 & 0 & 305.44 & 0 & -64.45 & 0 & 0 & 0 & 21.74 & 0 & 0 & 0.42 & 0 \\ 0 & 0 & 4.17 & 0 & 0 & 0 & 0 & 0 & 1.70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 98.56 & 0 & 0 & 0 & 0 & 13.07 & 0 & 0 & 0.45 & 0 \end{bmatrix} \quad (11.56)$$

The reference gain \mathbf{K}_r , added to the system, including the full order observer, can be seen in Figure 11.6

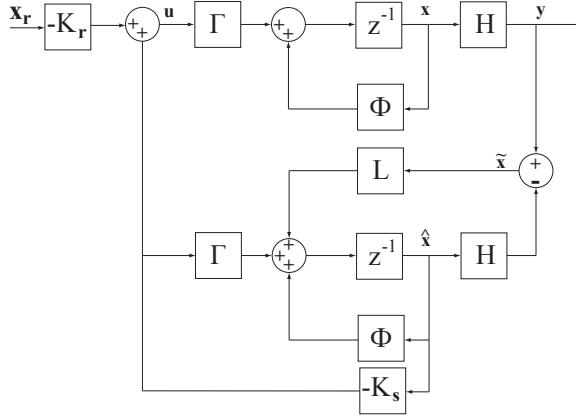


Figure 11.6: Introducing reference gain \mathbf{K}_r to the system, with observer feedback \mathbf{L} and state feedback \mathbf{K}_s

11.5 Integral Action

As the developed linearised model will have some inaccuracies, and no model and measurements of the disturbances are available, a compensation for this can be the introduction of an additional state vector, representing the integral of the control error [14, p. 42]:

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \mathbf{e}(k) \quad (11.57)$$

$$= \mathbf{x}_i(k) + \mathbf{r}(k) - \mathbf{y}(k) \quad (11.58)$$

The performance function is now constructed to punish the control error $e(k)$, the integrated control error \mathbf{x}_i and the input signal $\mathbf{u}(k)$ as the following [14, p. 45]:

$$\mathcal{I} = \sum_{K=0}^{N-1} (\mathbf{e}^T(k) \mathbf{Q}_{1e} \mathbf{e}(k) + \mathbf{x}_i^T(k) \mathbf{Q}_{1i} \mathbf{x}_i(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.59)$$

By adding extra dimensions to the system, the augmented system, with the augmented state vector $\mathbf{x}(k) = [\mathbf{x}_s^T(k) \quad \mathbf{x}_r^T(k) \quad \mathbf{x}_i^T(k)]^T$ becomes:

$$\mathbf{x}(k+1) = \begin{bmatrix} \Phi_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Phi_r & \mathbf{0} \\ -\mathbf{H}_s & \mathbf{H}_r & \mathbf{I} \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \Gamma_s \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{u}(k) = \Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k) \quad (11.60)$$

$$\mathbf{y}(k) = [\mathbf{H}_s \quad \mathbf{0} \quad \mathbf{0}] \mathbf{x}(k) \quad (11.61)$$

$$\mathbf{e}(k) = [-\mathbf{H}_s \quad \mathbf{H}_r \quad \mathbf{0}] \mathbf{x}(k) = \mathbf{H}_e \mathbf{x}(k) \quad (11.62)$$

$$\mathbf{x}_i(k) = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{I}] \mathbf{x}(k) = \mathbf{H}_i \mathbf{x}(k) \quad (11.63)$$

With this system augmentation, the performance function can be rewritten as [14, p. 43]:

$$\mathcal{I} = \sum_{k=0}^{N-1} (\mathbf{e}^T(k) \mathbf{Q}_{1e} \mathbf{e}(k) + \mathbf{x}_i^T(k) \mathbf{Q}_{1i} \mathbf{x}_i(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.64)$$

$$= \sum_{k=0}^{N-1} (\mathbf{x}^T(k) [\mathbf{H}_e^T \mathbf{Q}_{1e} \mathbf{H}_e + \mathbf{H}_i^T(k) \mathbf{Q}_{1i} \mathbf{H}_i(k)] \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.65)$$

$$= \sum_{k=0}^{N-1} (\mathbf{x}^T(k) \mathbf{Q}_1 \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)) \quad (11.66)$$

where:

$$\mathbf{Q}_1 = \mathbf{H}_e^T \mathbf{Q}_{1e} \mathbf{H}_e + \mathbf{H}_i^T \mathbf{Q}_{1i} \mathbf{H}_i \quad (11.67)$$

$$= \begin{bmatrix} \mathbf{H}_s^T \mathbf{Q}_{1e} \mathbf{H}_s & -\mathbf{H}_s^T \mathbf{Q}_{1e} \mathbf{H}_s & \mathbf{0} \\ -\mathbf{H}_r^T \mathbf{Q}_{1e} \mathbf{H}_s & \mathbf{H}_r^T \mathbf{Q}_{1e} \mathbf{H}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{1i} \end{bmatrix} \quad (11.68)$$

With the values of the matrices Φ , Γ , \mathbf{Q}_1 and \mathbf{Q}_2 , the performance function is now on standard form, and can be solved for $\mathbf{K}(k)$ and $\mathbf{S}(k)$ using the techniques discussed earlier. The optimal controller will now be on the form:

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) = [\mathbf{K}_s(0) \quad \mathbf{K}_r \quad \mathbf{K}_i] \begin{bmatrix} \mathbf{x}_s(k) \\ \mathbf{x}_r(k) \\ \mathbf{x}_i(k) \end{bmatrix} \quad (11.69)$$

$$= -\mathbf{K}_s \mathbf{x}_s(k) - \mathbf{K}_r \mathbf{x}_r(k) - \mathbf{K}_i \mathbf{x}_i(k) \quad (11.70)$$

11.5.1 Designing Integral Action

When the integral action is added to the system, the system matrices (already augmented with reference states), will be augmented with four integral states, as described in Equation 11.60. This introduces the new \mathbf{Q}_1 weight matrix seen in Equation 11.68.

The only new parameter in this weight matrix, is the \mathbf{Q}_{1i} matrix, which is the weight of the integrated output. This weight matrix have iteratively, been tuned through simulations but no satisfactory result was obtained. It was not possible to obtain enough integral effect without increasing the system feedback \mathbf{K}_s . The result was to set $\mathbf{Q}_{1i} = \mathbf{0}$ and make a hand tuned \mathbf{K}_i . The \mathbf{K}_i matrix found is shown in Equation 11.71

$$\mathbf{K}_i = \begin{bmatrix} 0 & -1.5 & 0 & 0 & 0 & 0 \\ 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.6 \end{bmatrix} \quad (11.71)$$

The discrete time integral is implemented using forward Euler. And the final implementation of the system, with observer, feedback, reference and integral action can be seen in Figure 11.7

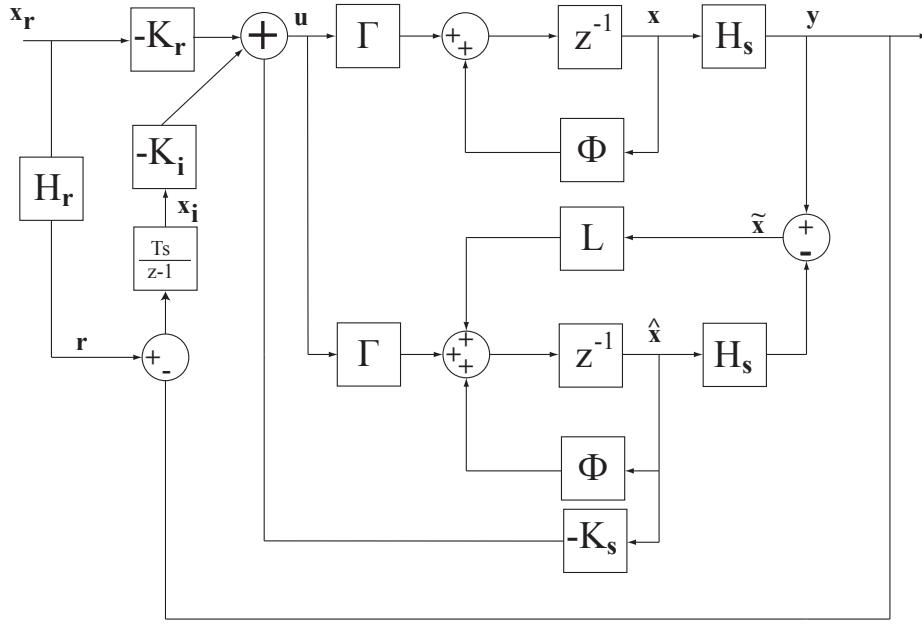


Figure 11.7: Introducing integral action K_i to the system, with observer feedback L , state feedback K_s and reference input gain K_r

11.6 Gain Sheding as Yaw Compensation

It is desired to control the yaw angle as well as the position. This means the operation point of $\psi = 0$ is no longer valid. When using $\psi \neq 0$ a non-linearity is added to the system as a consequence of the rotational matrices. To avoid this a gain scheduling system is implemented in the controller to compensate for the changing operating point. From [15, p. 489] the steps to create a gain-scheduled controller for a non-linear system are the following:

1. Linearise the non-linear model about a family of operating points.
2. Using linearisation, design a parametrised family of linear controllers to achieve the specified performance at each operation point
3. Construct a gain-scheduled controller such that
 - for each constant value of the exogenous input, the closed-loop system under the gain-scheduled controller has the same equilibrium point as the closed-loop system under the fixed-gain controller;
 - the linearisation of the closed-loop system under the gain-scheduled controller is equivalent to the linearisation of the closed-loop system under the fixed-gain controller.
4. Check the nonlocal performance of the gain-scheduled controller by simulating the non-linear closed-loop model.

In this case the non-linearity, that needs work with gain-scheduling, comes from the yaw angle. It is desired to be able to fly with the yaw angle following the reference angle. When the yaw angle is different from zero, the feedback matrix gives the wrong result. This originates in the fact that the controller is designed with all angles linearised in 0. It is observed that the change in gain needed follows the rotational matrix about the z-axis from Chapter 3.2.1.

$$C_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11.72)$$

This rotation is done for the relationship between the positions and velocities in the x-y plane and the output of the controller. The reference output matrix will also be rotated about the x-y plane according to the yaw angle.

Instead of choosing a set of operation points and designing one controller for each of them, the rotational matrix is implemented as a matrix multiplication in a part of the feedback loop. This solution requires more computational power than the operation point approach, but eliminates eventual problems when the system states are far from the chosen operation point. An illustration of the implementation of the yaw compensation for K_s can be seen in Figure 11.8 and are similar to the rotations of K_i and K_r .

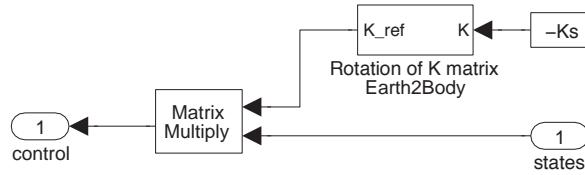


Figure 11.8: The yaw compensation, rotation of feedback gain K_s implemented in the simulink diagram.

The performance from the new controller should be similar to the standard controller as the resulting gain is not changed.

11.7 Simulations

The developed LQ controller is implemented in simulink, and simulated on the developed non-linear model. The simulations are all based on step references, as the controller is calculated with a step model reference. The implementation in simulink can be seen on Figure 11.9.

11.7.1 Simulation of step in x,y and z with yaw=0

In the first simulation the yaw angle of the quad rotor is kept at zero ($\psi = 0$). This is done to investigate the performance when the rotation of the feedback matrix and input has no effect. A series of steps are performed in x,y and z positions. The output step response of the system can be seen in Figure 11.10. This Figure also illustrates the actuation in angles to achieve the changes in position.

In Figure 11.10 it can be seen that the rise time of a step in the x-axis is approximately 1 second, with a settling time within 2% of 3.5 seconds. To achieve this step in position a negative pitch of -0.15 rad is made, to accelerate, and a positive step of 0.1 rad is made to decelerate the quadrotors motion.

The step in the y-position has a rise time of approximately 0.9 seconds, and a settling time of 3.2 seconds, within 2% of the reference. The angle actuation for achieving this, is 1.8 positive and 0.5 negative for deceleration.

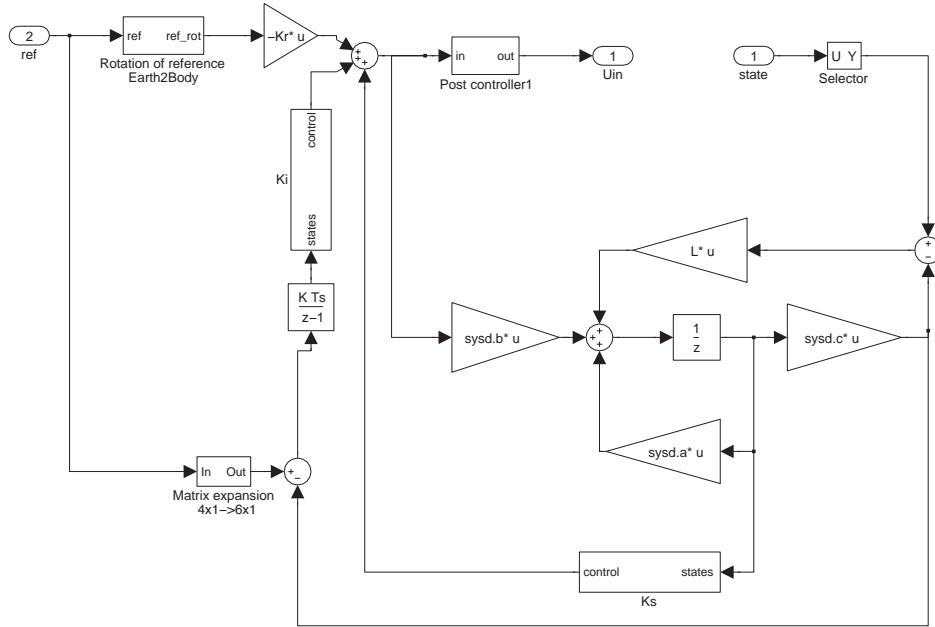


Figure 11.9: The simulink implementation of the LQ controller. The numerated ellipses, illustrates inputs and outputs from the non-linear model.

The step to one meter in elevation (z-position), has a rise time of approximately 0.8 seconds, and a settling time of 2.8 seconds. Furthermore it can be seen, that the quad rotor gains altitude, when stepping to x-, and y-positions. This increase in altitude is due to the rotation of the reference to the body frame, only is affecting yaw. If all Euler angles where considered, this additional altitude correction would not happen.

The above stated dynamic properties, including overshoot, is summerised in table 11.1

	Risetime	Overshoot	Settling time
x	1 s	10%	3,5
y	0.9 s	11%	3.2
z	0.8 s	5%	2.8

Table 11.1: Dynamic properties of LQ controller, with yaw=0

11.7.2 Simulation of step in x,y and z with yaw \neq 0

The second simulation concerns the controller dynamics, when the yaw is different from zero, here $\psi = 1$. This is done to ensure that the rotational matrices, introduced to reference and feedback, corrects the controller to perform similar, independently of the heading. Figure 11.11 shows the step responses in x,y,z and ψ .

As seen on Figure 11.11 the dynamic responses to x,y and z remain unchanged independent of yaw, meaning that the rotational matrices inserted as gain scheduling for $\psi \neq 0$, on reference and feedback, functions as intended. Furthermore it can be seen, that due to the rotation of the orientation, actuation of angles in both roll and pitch are needed to step in either x- or y-positions. The step in yaw has a rise time of approximately 2 seconds, and a 2% settling time of 2.7 seconds.

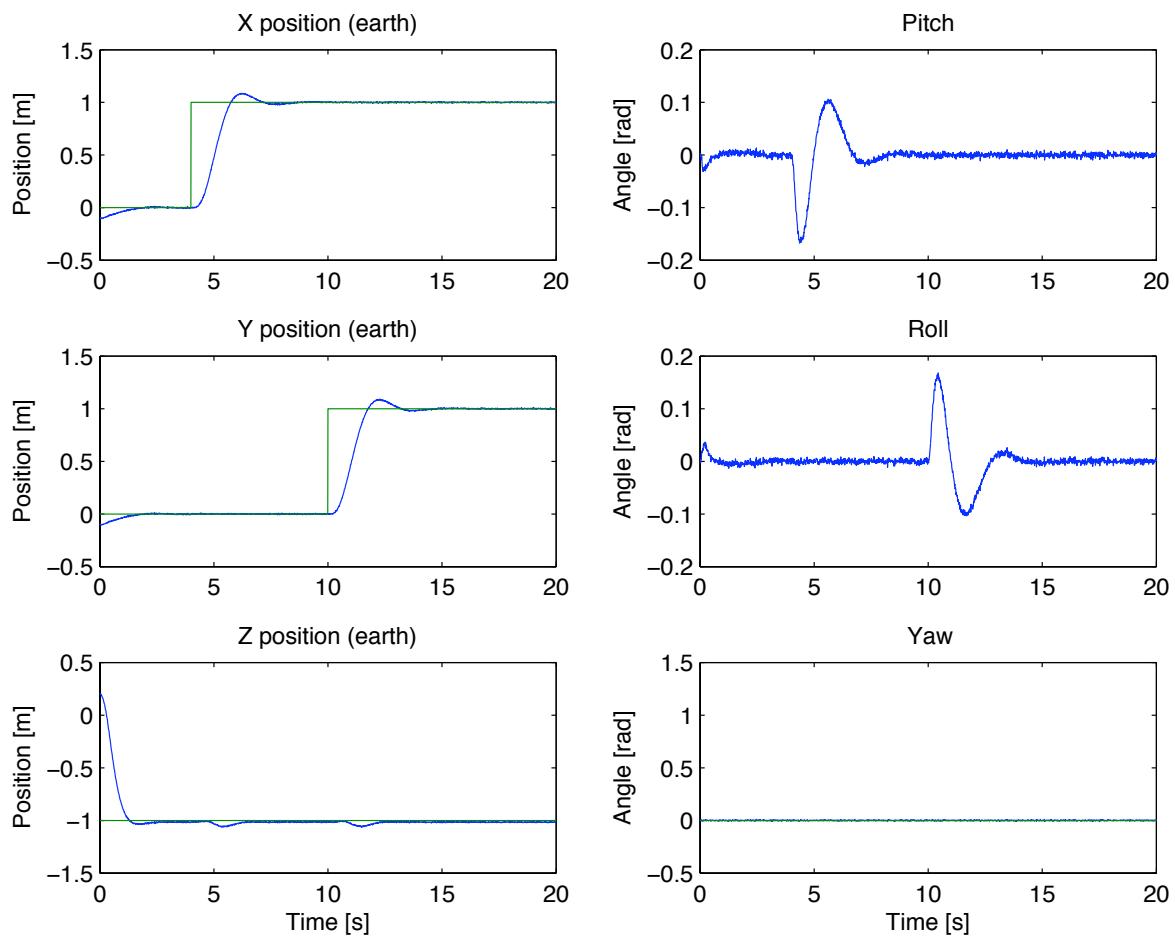


Figure 11.10: Simulations of steps in positions, with yaw=0. Step of -1 in z-position at time 0, step of 1 in x-position at time 4 and step of 1 in y-position at time 10.

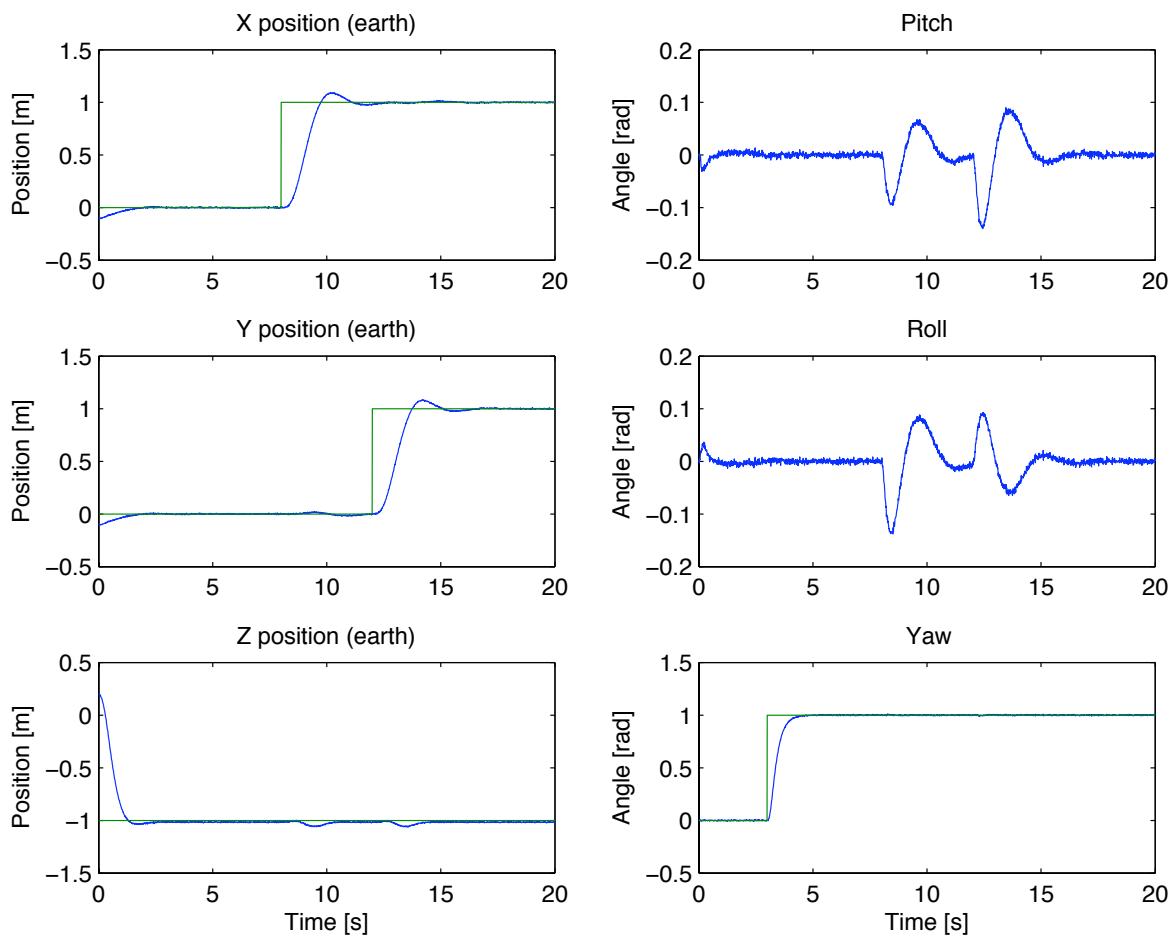


Figure 11.11: Simulations of steps in positions, with yaw=1 at time 3. Step of -1 in z-position at time 0, step of 1 in x-position at time 8 and step of 1 in y-position at time 12.

Reference Tracking

In Chapter 10 and 11 controllers have been designed with the objective to minimise the difference between a reference position and the actual position of the quad rotor. Before performing a thorough controller evaluation, the result of tracking a moving reference is considered.

12.1 Tracking

Because the controller minimises an error between a reference position and an actual position, an error must be present before the controller will act. Because the controller is not capable of predicting the reference position ahead of time, it is expected that the position of the quad rotor always will be lagging the reference position when the reference is moving with a constant velocity.

Figure 12.1 shows measurements of the quad rotors x and y position when flying with the PID controller, together with the x and y references and the velocity of the reference object in the x and y direction.

From Figure 12.1 it can be seen that the quad rotor, as expected, is lagging the reference position when the reference is moving with a constant velocity.

In order to minimise this tracking error it has been chosen to add a term to the x and y reference that compensates for the reference objects velocity:

$$\begin{bmatrix} x_{ref_c} \\ y_{ref_c} \end{bmatrix} = \begin{bmatrix} x_{ref_o} \\ y_{ref_o} \end{bmatrix} + \lambda \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (12.1)$$

Where x_{ref_c} and y_{ref_c} are the compensated references, x_{ref_o} and y_{ref_o} are the position of the reference object, and v_1 and v_2 are the x and y velocity of the reference object. λ is the tuning constant determining how much weight the velocity of the reference object shall have on the compensated reference. The reference compensation strategy is shown in Figure 12.2(a) and 12.2(b).

Figure 12.2(a) shows the flight without the reference compensation, where the quad rotor is lagging behind the

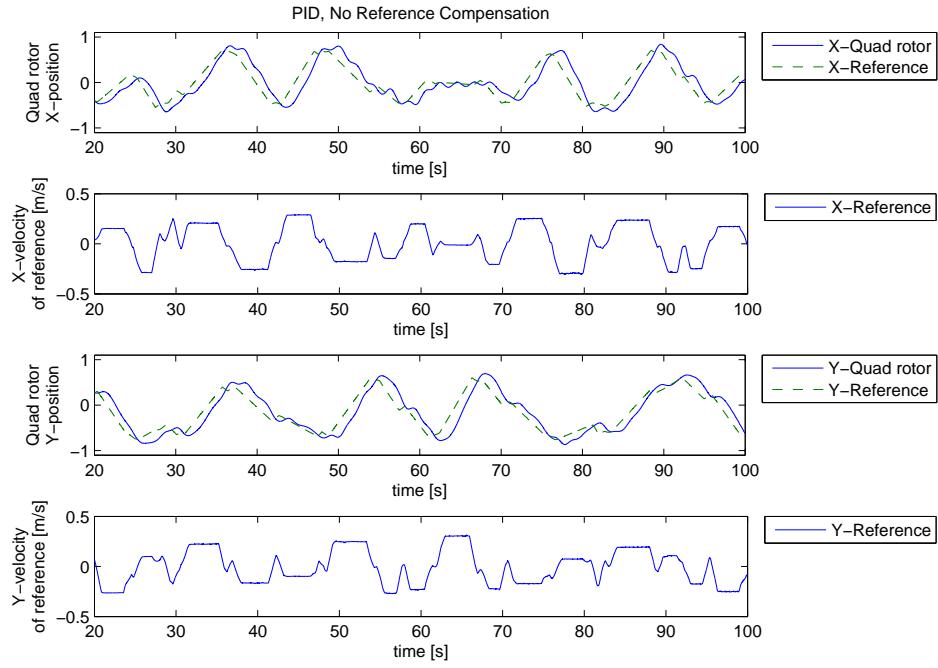


Figure 12.1: The x and y position of the quad rotor plotted together with the x and y reference. The quad rotor is controlled by the PID-regulator

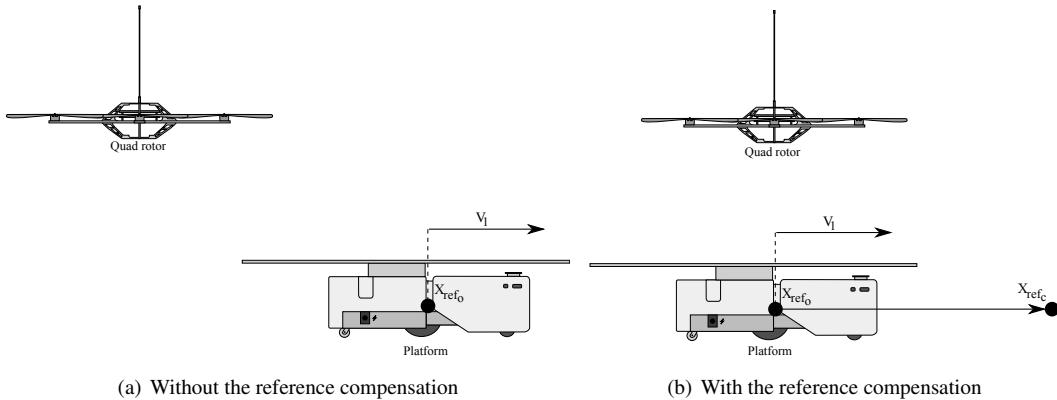


Figure 12.2: Principal sketch of flight with and without the reference compensation. Only x-axis is shown

platform. Figure 12.2(b) shows the flight with the reference compensation, where the quad rotor no longer lags the platform. It should be noted that when using this kind of reference compensation the x_{ref_c} and y_{ref_c} will be equal to x_{ref_o} and y_{ref_o} respectively when the platform is not moving, which of cause is wanted.

Some hand tuning of λ has been performed, and it is found that the best results occurs with λ around 1. Figure 12.3 shows the result of tracking the objects position using the compensated reference with $\lambda = 1$. It can be seen that the lagging has decreased but is not completely eliminated when the quad rotor is controlled by the PID-controller.

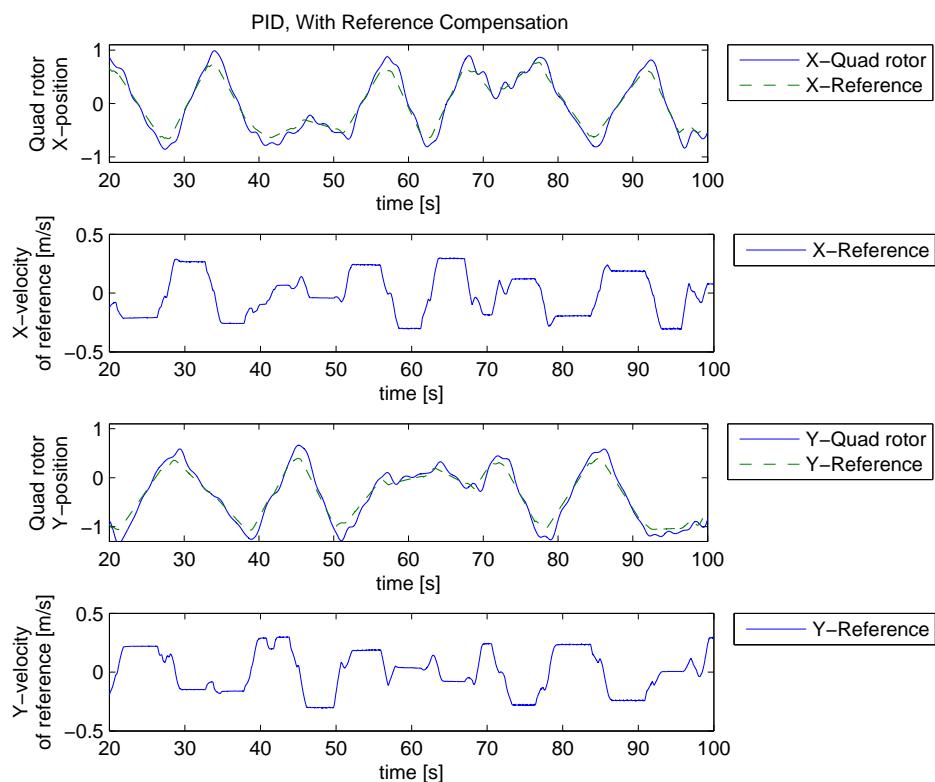


Figure 12.3: The x and y position of the quad rotor plotted together with the compensated x and y reference. The quad rotor is controlled by the PID-controller

Chapter 13

Linear Controller Analysis

In this chapter is the closed loop system with the quad rotor model and the controller analysed. In Chapter 8 the model was analysed. This analysis included bode plots of the linear system model as seen in Figures 8.6 to 8.8. The linear system model with controller contains no force polynomial which was introduced in Section 3.3.1. The effect of this is not further investigated.

The closed loop linear system model is obtained through inspection of the closed loop diagram seen on Figure 13.1.

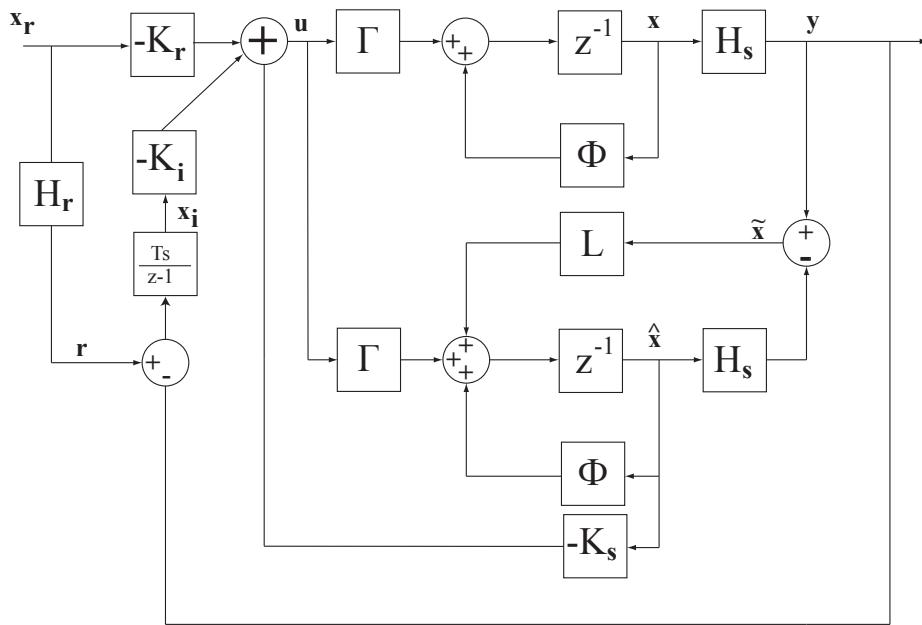


Figure 13.1: Closed loop system using LQ control with reference and integral action

To make a complete state space description of the closed loop system, equations for $x(k+1)$, $\hat{x}(k+1)$, $y(k)$, $\hat{y}(k)$, $u(k)$ and $x_i(k+1)$ is deduced.

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) \quad (13.1)$$

$$\hat{\mathbf{x}}(k+1) = \Phi\hat{\mathbf{x}}(k) + \mathbf{L} \cdot \mathbf{y}(k) - \hat{\mathbf{y}}(k) \quad (13.2)$$

$$\mathbf{x}_i(k+1) = \mathbf{r}(k) - \mathbf{y}(k) \quad (13.3)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \quad (13.4)$$

$$\hat{\mathbf{y}}(k) = \mathbf{H}\hat{\mathbf{x}}(k) \quad (13.5)$$

$$\mathbf{u}(k) = -\mathbf{r}(k) \cdot \mathbf{K}_r - \mathbf{x}_i(k)\mathbf{K}_i - \hat{\mathbf{x}}(k)\mathbf{K}_s \quad (13.6)$$

Equation 13.4, 13.5 and 13.6 are then inserted to 13.1, 13.2 and 13.3, which yields.

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) - \Gamma\hat{\mathbf{x}}(k)\mathbf{K}_s - \Gamma\mathbf{x}_i(k)\mathbf{K}_i - \Gamma\mathbf{r}(k)\mathbf{K}_r \quad (13.7)$$

$$\hat{\mathbf{x}}(k+1) = \Phi\hat{\mathbf{x}}(k) - \Gamma\hat{\mathbf{x}}(k)\mathbf{K}_s - \Gamma\mathbf{x}_i(k)\mathbf{K}_i - \Gamma\mathbf{r}(k)\mathbf{K}_r + \mathbf{L}\mathbf{H}\mathbf{x}(k) - \mathbf{L}\mathbf{H}\hat{\mathbf{x}}(k) \quad (13.8)$$

$$= (\Phi - \mathbf{L}\mathbf{H} - \Gamma\mathbf{K}_s)\hat{\mathbf{x}}(k) + (-\Gamma\mathbf{K}_i)\mathbf{x}_i + (-\Gamma\mathbf{K}_r)\mathbf{r}(k) + (\mathbf{L}\mathbf{H})\mathbf{x}(k) \quad (13.9)$$

$$\mathbf{x}_i(k+1) = -\mathbf{H}\mathbf{x}(k) + \mathbf{r}(k) \quad (13.10)$$

From these three equations the state space system can be made. This is seen in Equation 13.11.

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \hat{\mathbf{x}}(k+1) \\ \mathbf{x}_i(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & -\Gamma\mathbf{K}_s & -\Gamma\mathbf{K}_i \\ \mathbf{L}\mathbf{H} & \Phi - \mathbf{L}\mathbf{H} - \Gamma\mathbf{K}_s & -\Gamma\mathbf{K}_i \\ -\mathbf{H} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \\ \mathbf{x}_i(k) \end{bmatrix} + \begin{bmatrix} -\mathbf{H}\mathbf{K}_r \\ -\mathbf{H}\mathbf{K}_r \\ 1 \end{bmatrix} \mathbf{r}(k) \quad (13.11)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & \dots \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \\ \mathbf{x}_i(k) \end{bmatrix} \quad (13.12)$$

A bode plot of the closed loop transfer functions from x reference to x position, y reference to y position and z reference to z position is shown in Figure 13.2. From this it can be seen that all three axes have about the same cut off frequency in about 2 rad/sec. The phase margins can be seen in Equation 13.15

$$PM_x = 105^\circ \quad (13.13)$$

$$PM_y = 106^\circ \quad (13.14)$$

$$PM_z = 109^\circ \quad (13.15)$$

The gain margins can be seen in Equation (13.18).

$$GM_x = 6.69dB \quad (13.16)$$

$$GM_y = 6.82dB \quad (13.17)$$

$$GM_z = 31.2dB \quad (13.18)$$

The pole/zero plot in Figure 13.3 shows the placement of the poles and zeros. One additional zero is found in -135000 but not shown in the plot. From the plot it can be seen that no poles are found in the right half plane, but four zeros are found there. This can also be seen on the bodeplot where the initial phase is 360°.

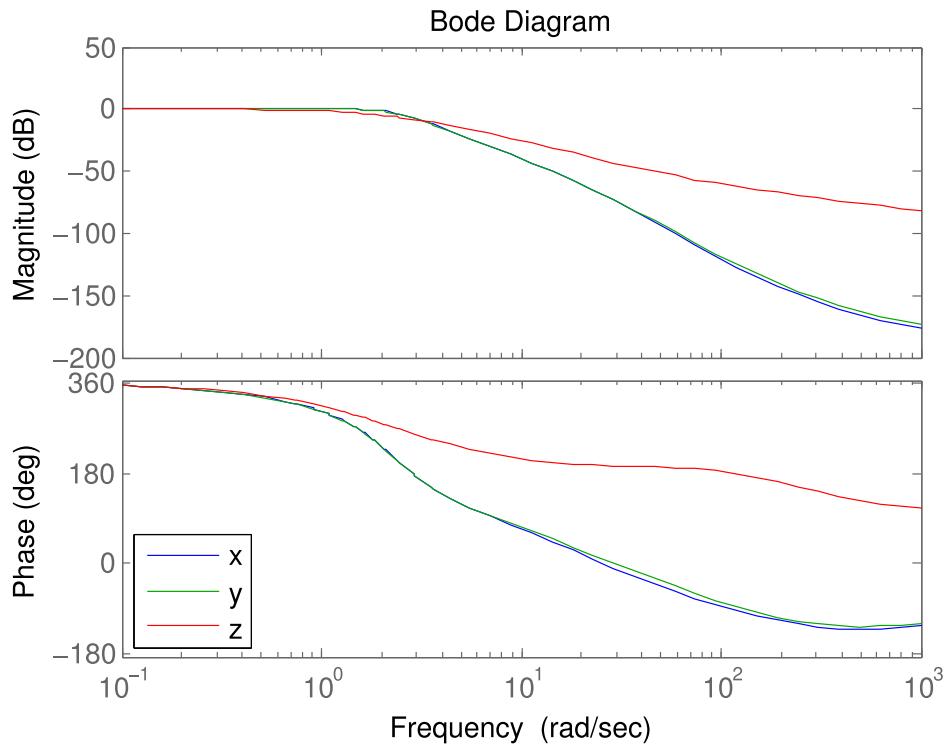


Figure 13.2: Closed loop bode plot of the system and LQ controller

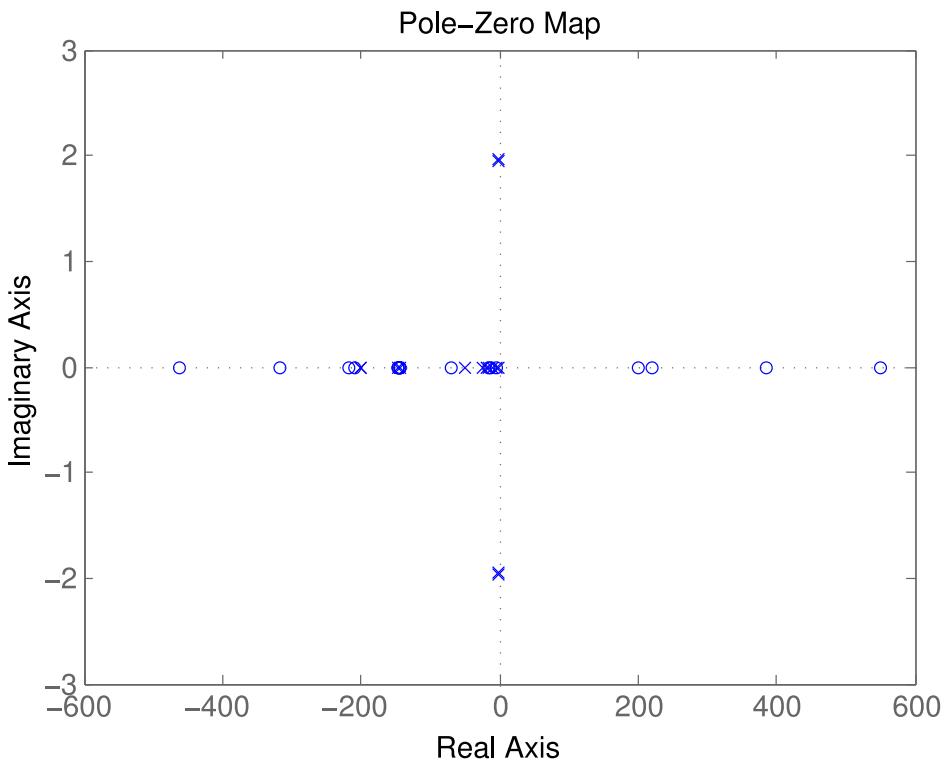


Figure 13.3: Closed loop pole/zero plot of the system and LQ controller

Evaluation of Controllers

This chapter contains evaluations of the performance of the designed controllers. The evaluations are based on the test flights described in the measurement journal found in Appendix A Section A.4.

The tests have been divided into four sections which concern four different types of reference inputs. The first section covers the hover tests where the reference is held constant over time. The second set of tests deals with step inputs. In the third section the reference is sinusoidal inputs and the final set of tests is conducted using a moving platform as reference input.

Each of the four sets of tests is conducted for both the PID and the LQ controller.

14.1 Hover Evaluation

Two types of hover tests is conducted; one with the reference set to $(0,0,-1,0)$, and one with the reference set to $(0,0,-1,\frac{3\pi}{4})$, i.e. the only difference being yaw, which is set to 0 and 135 degrees respectively.

Firstly the hover with yaw set to zero is treated. The result of the PID and LQ controller is shown in Figures 14.1 and 14.2 respectively. The figures show only a 25 second out of the 450 second test (which corresponds to the flight time of a fully charged battery). Figures of the entire test can be found in Appendix A.4.

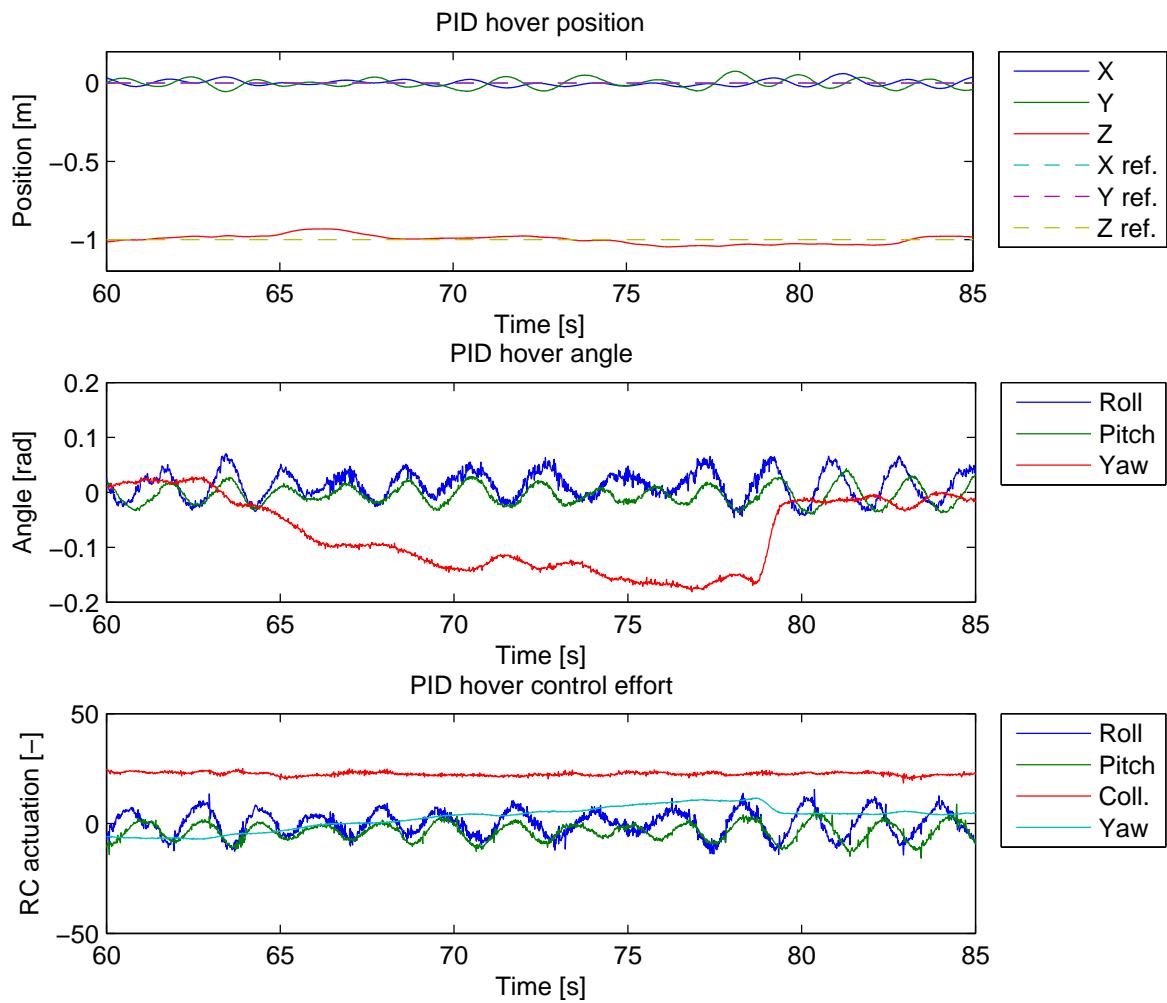


Figure 14.1: Test of hover capabilities for the PID controller. Top: Position. Middle: Angle. Bottom: Control effort.

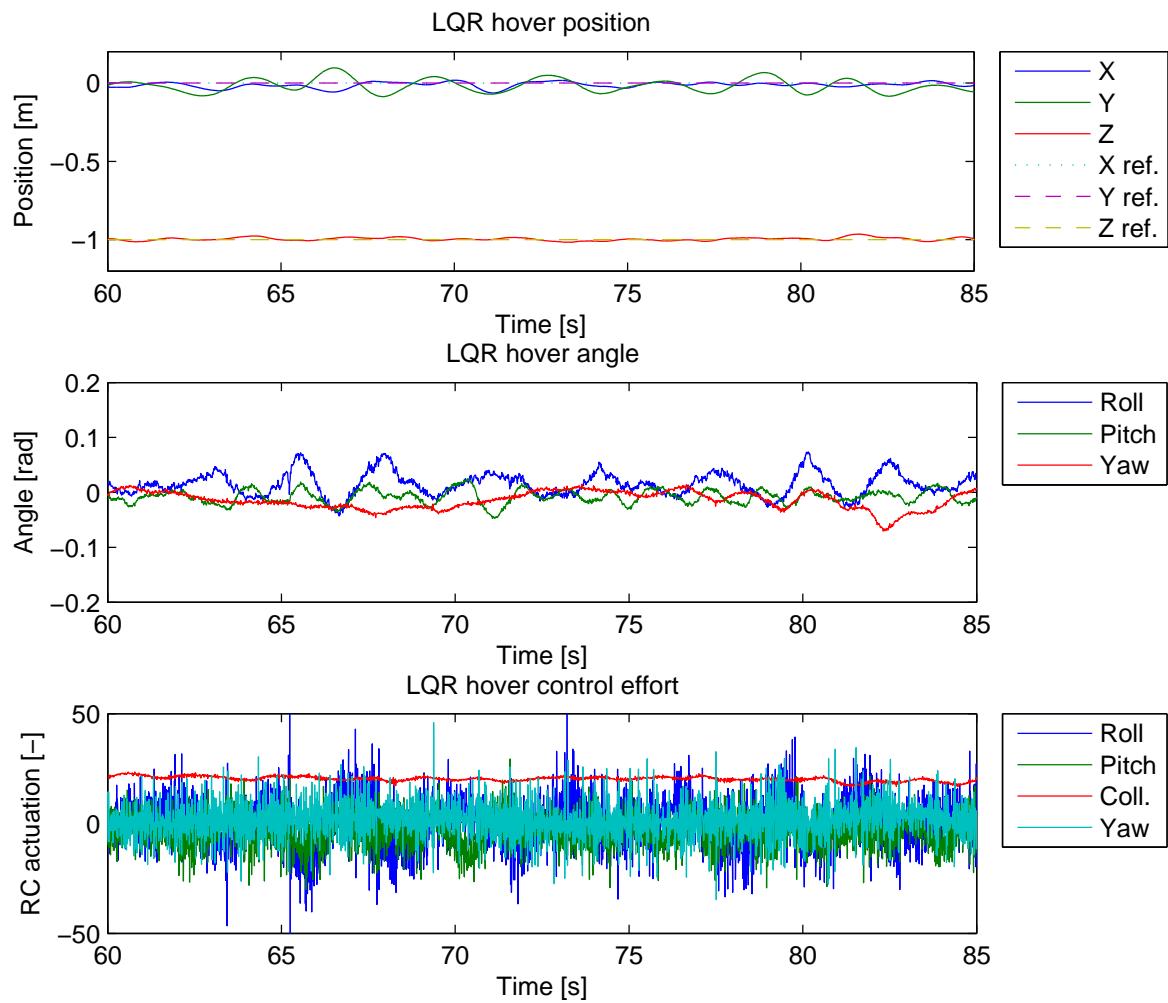


Figure 14.2: Test of hover capabilities for the LQ controller. Top: Position. Middle: Angle. Bottom: Control effort.

From Figure 14.1 it is seen that yaw angle is deviating more from zero than both roll and pitch. This is believed to be caused by a deadzone in the physical setup on the X-3D-BL. It can be seen from the control effort that the controller reacts to the offset but movement does not occur until the yaw angle reaches about 0.18 rad. The same yaw deviations is not seen in the tests of the LQ controller because it is tuned to punish deviations from the reference harder.

The position error from the hover references is found through analysis of the entire hover test. The results is presented in the Table 14.1.

	PID	LQR
mean $ r_x $	0.024 m	0.017m
mean $ r_y $	0.029 m	0.032m
mean $ r_z $	0.020 m	0.009m
max $ r_x $	0.125 m	0.097m
max $ r_y $	0.132 m	0.147m
max $ r_z $	0.095 m	0.039m

Table 14.1: Error from hover references for both PID and LQ controller were r_x , r_y , and r_z is the position error residual regarding the x- y- z-position in the earth frame

As it can be seen from Table 14.1 the overall position error during the flight using LQ control is smaller or equal to the overall position error of the flight using PID control, except for the Y position where the LQ controller cause more variance than the PID.

From Figure 14.1 and 14.2 it is seen that the control effort during the LQ controller flight is much higher than that of the PID controller flight. This is caused by the LQ controller having much higher gains than the PID, which makes noise and measurement errors more dominant and thereby increasing the control effort.

Another aspect worth mentioning, with respect to tuning of the parameters of the controllers, is that the yaw angle during the PID flight has a tendency to deviate more from its reference than it does during the LQ controller flights. This is also a result of tuning where the LQ controller has been tuned to be more aggressive than the PID.

From the hover tests it can be seen that the collective input increases over time. This is seen in Figure 14.3 which shows collective input for both PID and LQ controller flights. This is due to the discharge of the onboard lithium-ion polymer battery. Furthermore it can be seen that the last third of both measurements contains spikes which might be an effect of the battery discharge.

Figure 14.4 shows the positions of the first 80 seconds of flight for both the PID and LQ controller. From this Figure the effect of integral action can be seen for both the PID and the LQ controller.

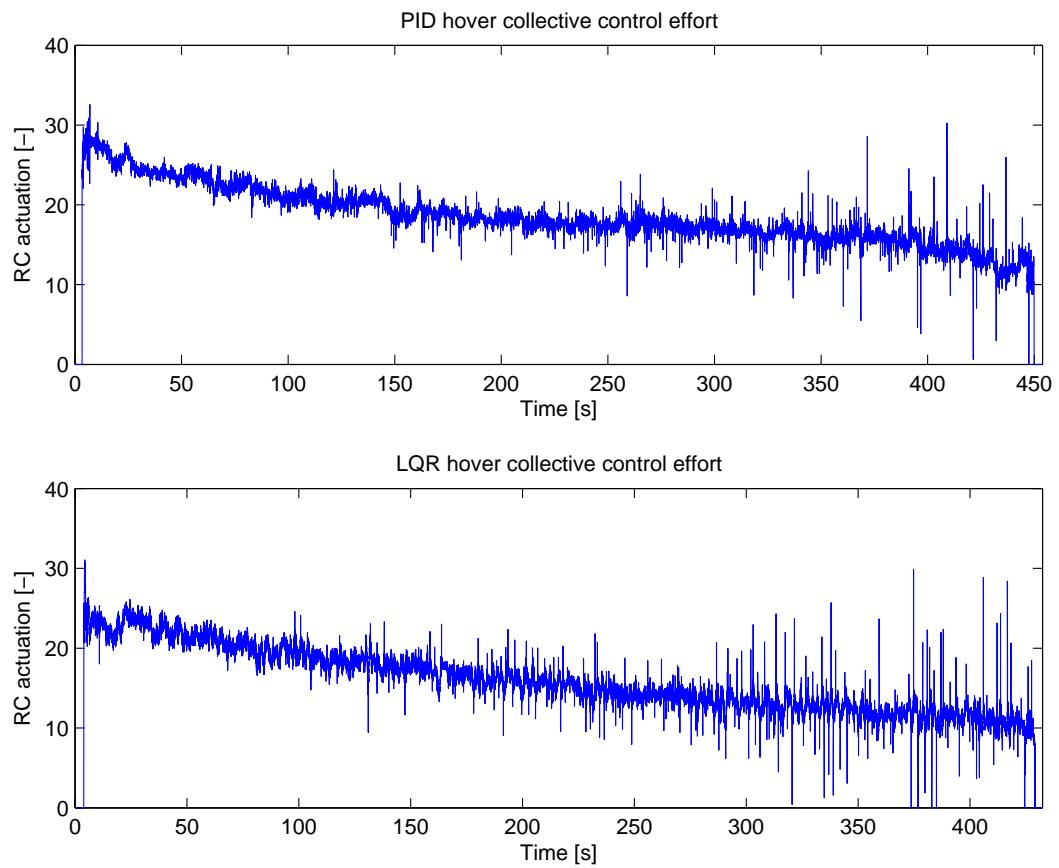


Figure 14.3: Battery discharge effect shown as increase in collective input over time. It is reminded from Section 1.2 that the scale of collective spans from min=100 to max=-100 .

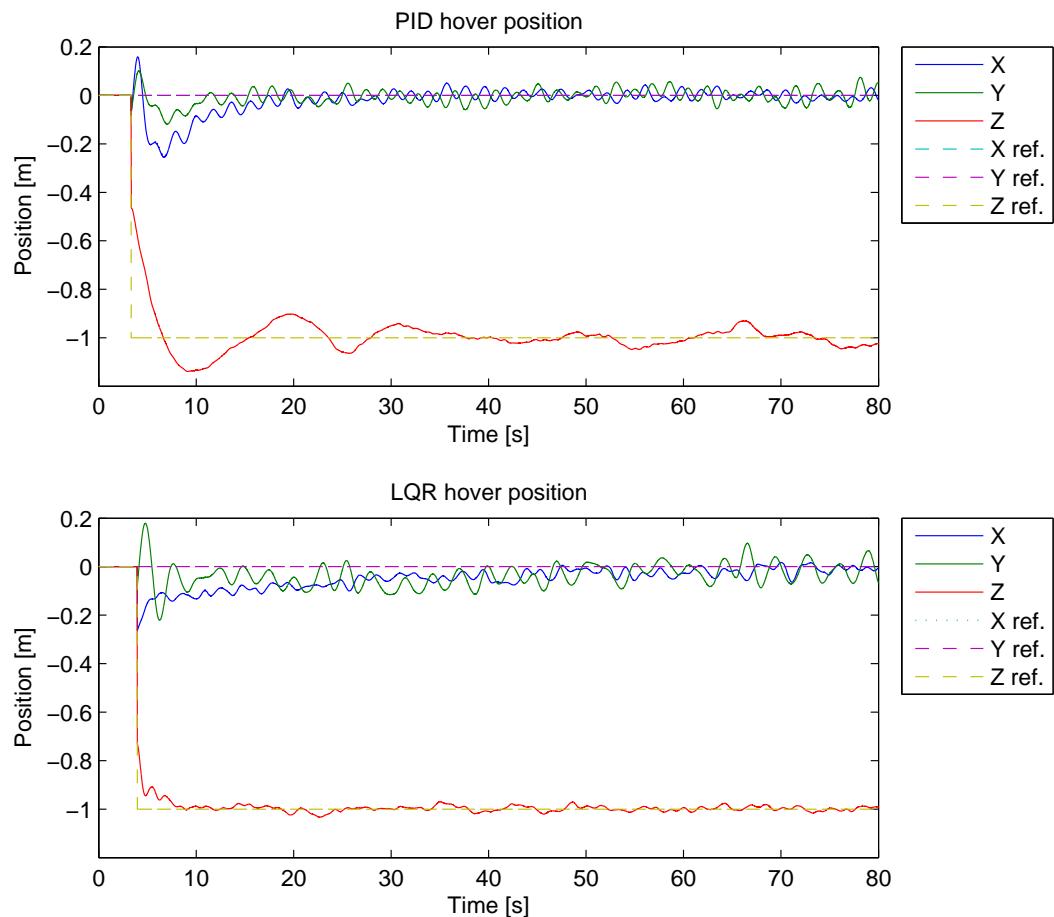


Figure 14.4: Integral effect for both PID and LQ controller.

For the PID controller the x and y positions have a relatively fast integrator time. This is not the case for the LQ controller which is much slower than the PID. For the z position the PID has an integrator time that is shorter than its settling time and is therefore hard to determine exactly. The z position integrator time for the LQ controller is relatively fast. From the plots it is difficult to accurately compare the PID to the LQ controller flight because their initial conditions are different (Autonomous flight is engaged from different points in space). To perform more accurate measurement stable hover should be reached (no offset from reference) and then a change in reference should be applied and held, until stable hover is reached again (again with no offset from reference). As will be seen later in this chapter steps are performed, but they are not held long enough to show the full effect of the integral action thereby making it impossible to assess the integrator time.

Integral action is also implemented on the yaw angle but since the reference is set to zero and the initial condition for the quadrotor is set to zero as well, integral action is not noticeable on the hover measurements where yaw is held to zero.

On Figures A.8 and A.10 in Appendix A.4 the effect of integral action is seen for the PID and LQ controller respectively. The figures show the same hover measurement as investigated earlier but this time yaw is at an 135 degree angle. It is seen that the integral action on yaw on the PID controller is much slower than that for the LQ controller. From Figures A.8 and A.10 it is also seen that the when yawing the position error is not significantly increased compared to the measurements for hover with the yaw reference set to 0 degrees.

14.2 Step Input Evaluation

Step tests are conducted in each of the three axes, for both the PID and LQ controller. The results of the step tests performed with the PID controller is seen in Figure 14.5.

For all three measurements, rise time, overshoot and settling time is evaluated. The values are determined through inspection of the plots and are presented in Table 14.2.

	Rise time (positive)	Rise time (negative)	Overshoot (positive)	Overshoot (negative)	Settling time (positive)	Settling time (negative)
X	1.5 s	1.6 s	30%	20 %	10 s	10 s
Y	1.7 s	1.6 s	20%	25 %	10 s	8+ s
Z	3.2 s	3.1 s	13%	20 %	13 s	15 s

Table 14.2: Rise time, overshoot and settling time for both positive and negative steps for the PID controller. The values are found through inspection.

The results of the step tests performed with the LQ controller is seen in Figure 14.6 and Table 14.3.

	Rise time (positive)	Rise time (negative)	Overshoot (positive)	Overshoot (negative)	Settling time (positive)	Settling time (negative)
X	2.8 s	2.2 s	0%	10 %	6 s	5 s
Y	2.6 s	2.1 s	5%	17 %	6 s	4 s
Z	1.0 s	0.7 s	55%	70 %	6 s	5 s

Table 14.3: Rise time, overshoot and settling time for both positive and negative steps for the LQ controller. The values are found through inspection.

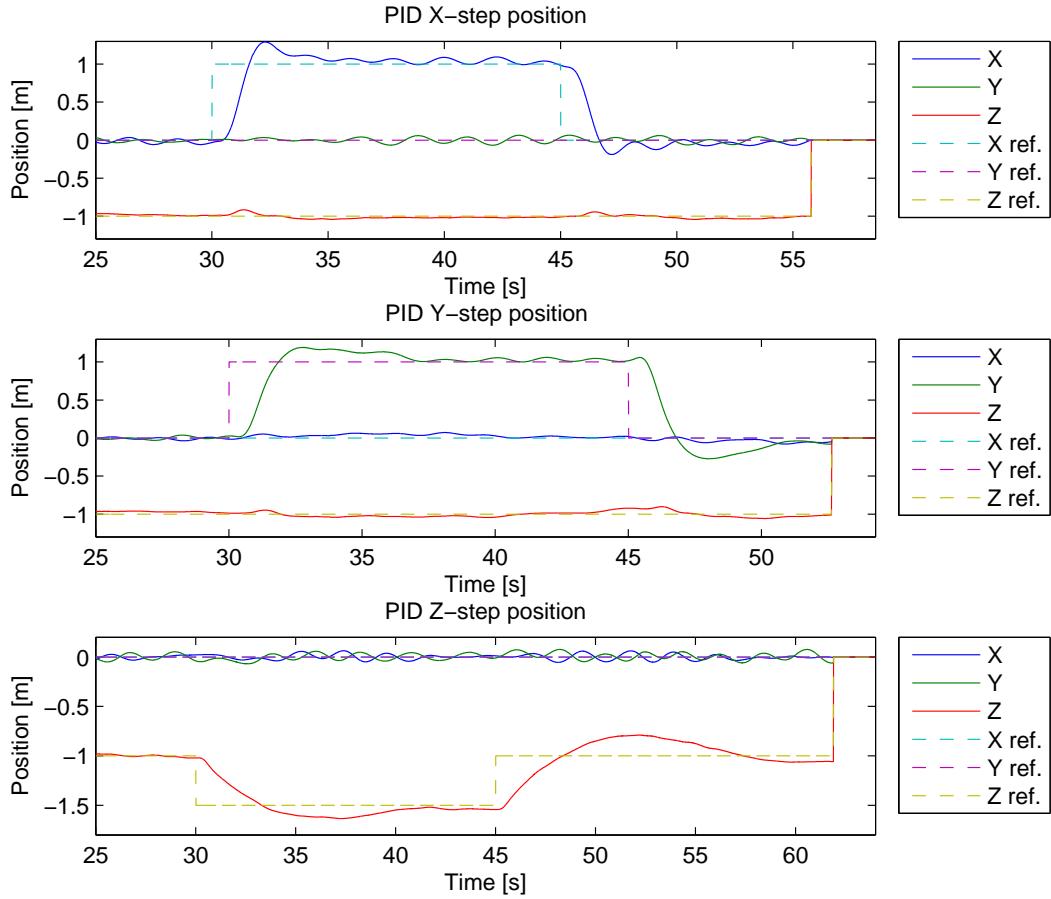


Figure 14.5: Step responses for the PID controller on the three translatory axis.

It appears from Tables 14.2 and 14.3 that the LQ-regulator performs better than the PID regulator for the x- and y-steps. The LQ-regulators overshoot is smaller and the settling time is shorter. It is however noticed that the integration effect of the LQ regulator is so slow that it has not eliminated a steady state error within the 15s step time. This problem was also confirmed in Section 14.1.

Further it can be seen that the vertical steps for the LQ controller flights generate a large overshoot compared with the PID controller flights. The PID controller is however very slow performing vertical steps resulting in a large settling time compared to the LQ-regulator.

Both the PID and LQ controllers have a delay on the position in the steps and it has roughly the same length for all the measurements. From Figure A.11 in Appendix A it is seen that the delay does not occur on the angles. It is believed that the reason for this delay is a need for the quad rotor to change attitude before translatory movement can be initiated. This behavior is assessed to be caused by the onboard controllers.

The steps done in flight performed with the LQ regulator can now be compared with the simulated ones. A plot of this can be seen in Figure 14.7, where steps in x, y and z are shown.

As seen on Figure 14.7 both the simulation and the real flight have the same tendencies, although the simulation gives overshoot on x and y steps, whereas the real flight does not generate any overshoot. The reverse is true for the z-axis, where the real flight generates more overshoot than the simulation. The model overshoot of the x- and

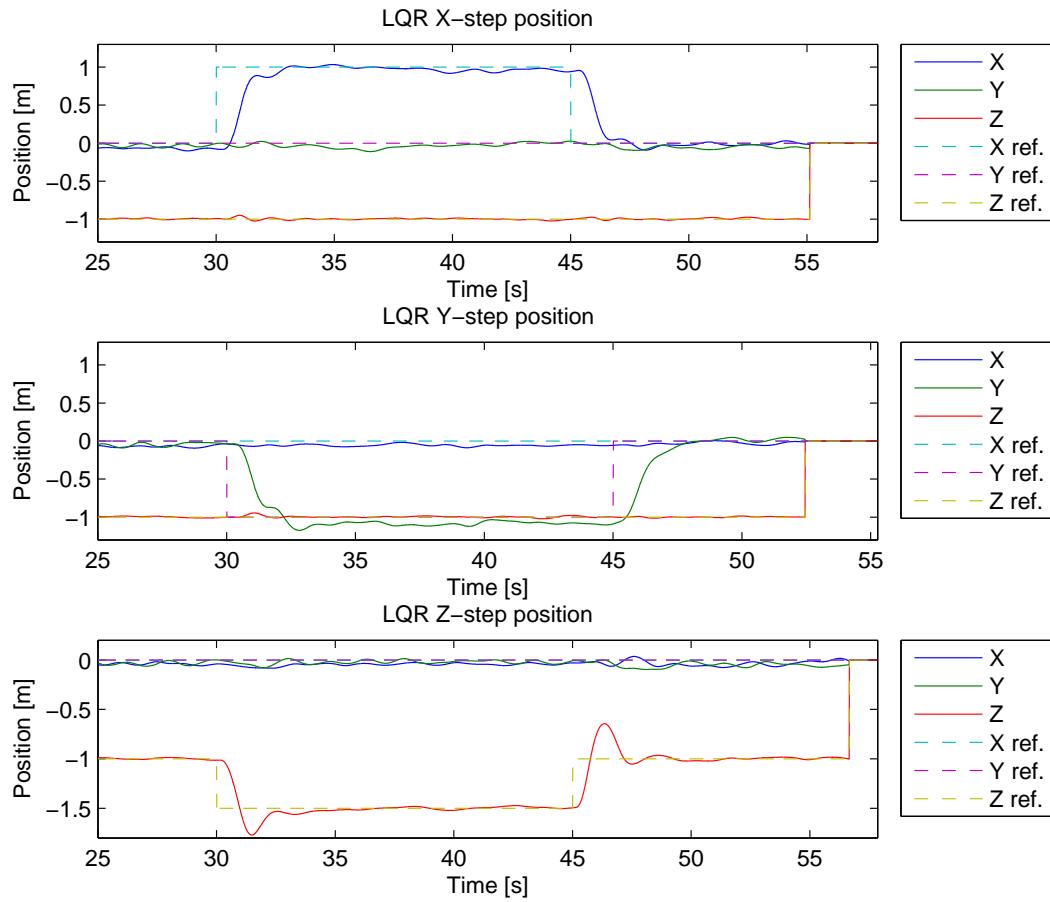


Figure 14.6: Step responses for the LQ controller on the three translatory axis.

y- step could indicate that a damping in translator x- and y-movement has not been modeled. The difference in overshoot of the real flight steps in the vertical plan, compared to the simulated overshoot, emphasises that the modelling of the total force acting on the quad rotor only is valid in near hover flight. Further hand tuning of the LQ regulator, could minimise the overshoot, but will increase the rise time.

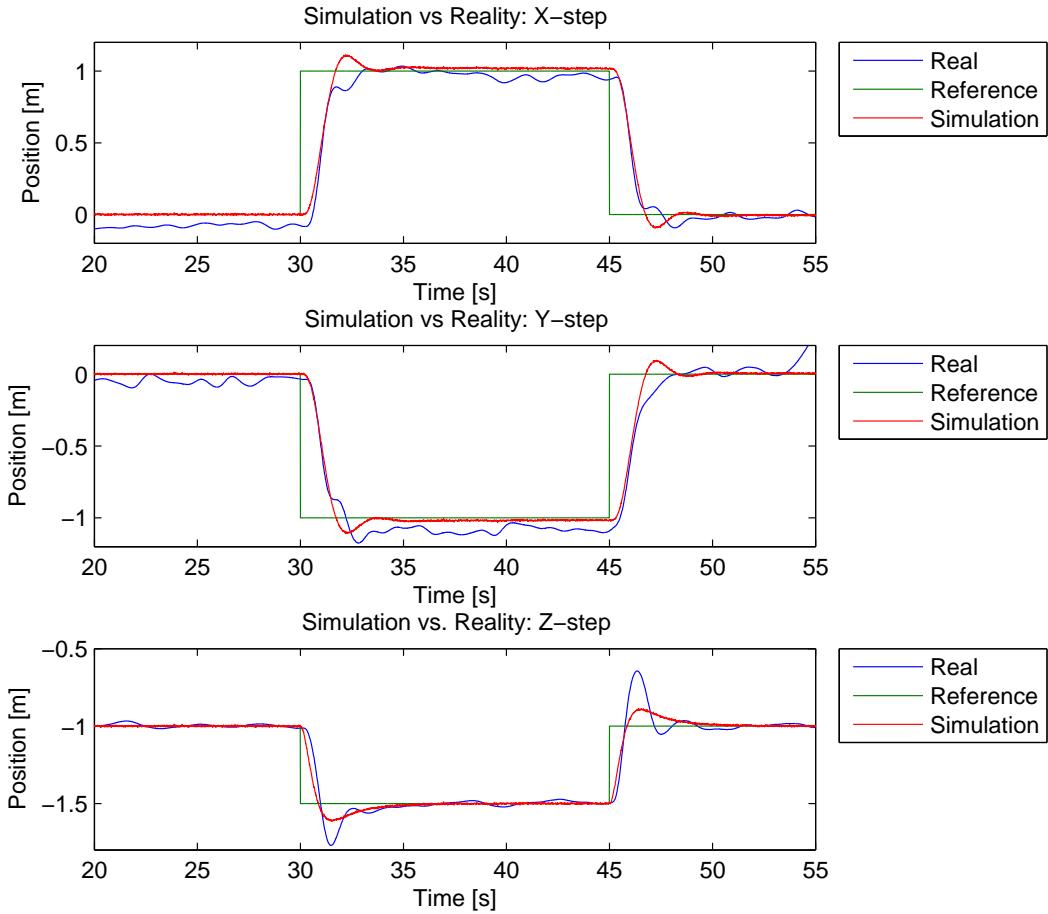


Figure 14.7: Steps in x, y and z for both real flights and simulations using the nonlinear model.

14.3 Sinusoidal Input Evaluation

Figures A.17 and A.19 in Appendix A.4 shows flights where the x, y and z reference inputs are sinusoidal. These tests shows that the actual flights lag behind the references, i.e. a phase shift exists. A section of the before mentioned plots is seen in Figure 14.8 where it can be seen that the x and y lags the reference by approximately 60 degrees and z lacks with approximately 90 degrees for the PID. The LQ controller gives rise to approximately 45 degrees of phase lag for x and y and 20 degrees for z. It should be noted these phase lags is at an angular velocity of $\frac{\pi}{4}$ rad/sec for all measurements. This phase lag agrees with the simulated one in chapter 13, Figure 13.2.

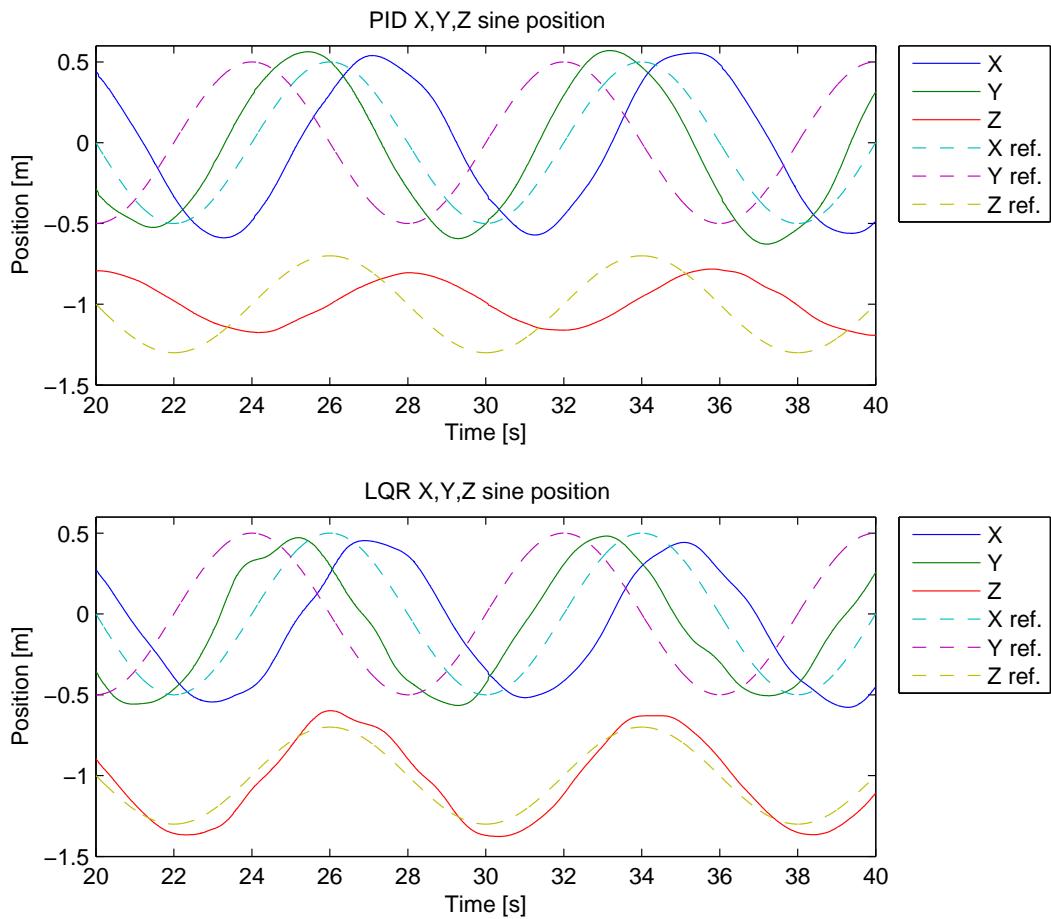


Figure 14.8: Sinusoidal input on x, y and z for both PID and LQ controller flights.

The phase lag can also be seen when plotting yz, xz and all axis in 3D. This is seen in Figure 14.9.

From the Figure it is seen that the actual path of the quad rotor is skewed with respect to the reference. This is expected because of the phase lag.

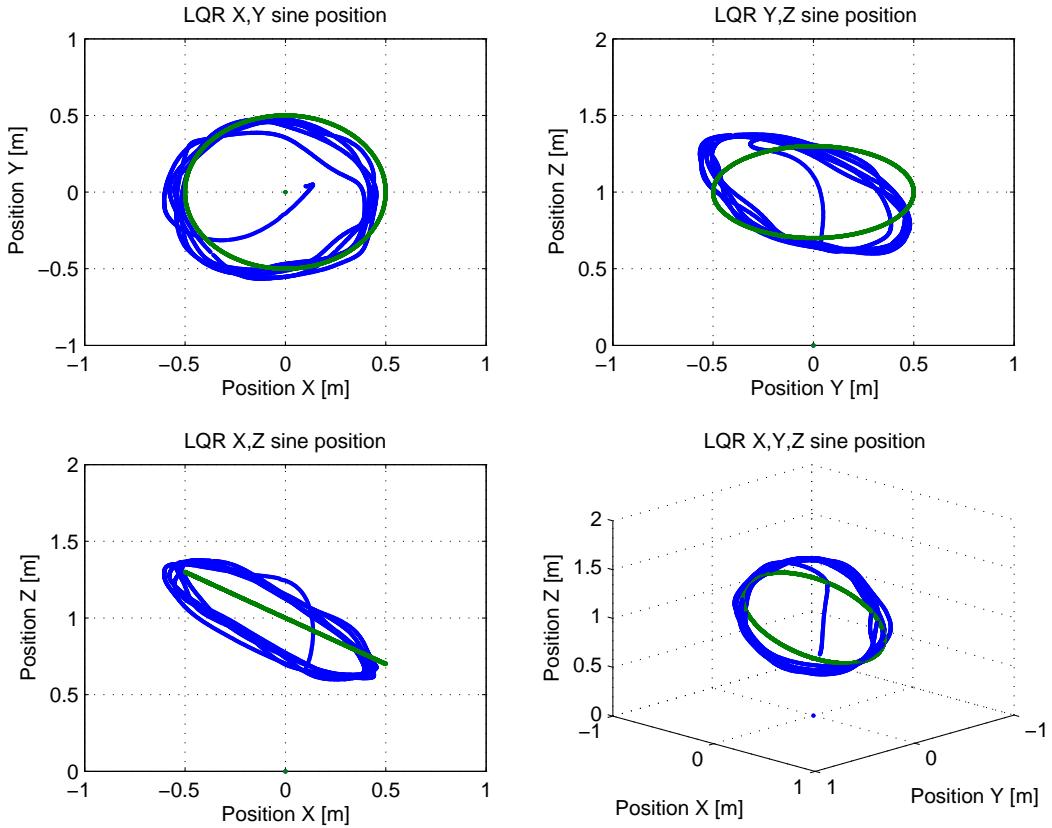


Figure 14.9: Sinusoidal input on x, y and z for LQ controller plotted as xy, yz, xz plots and a 3D plot of all axis. The blue plot is the real flight whereas the green is the reference.

14.4 Tracking Evaluation

In Chapter 12 the problems of tracking a moving platform was evaluated. In Appendix A.4 the test procedure for tracking an autonomous platform is described, along with a brief description of the platform.

Figures 14.10 and 14.11 shows the PID and LQ regulator tracking the moving platform. The PID tracking tests was performed with the quad rotor maintainning a hight of 1 m above ground, while the LQR tracking test was performed with the quad rotor maintainning an hight of 0.8 m above ground. All though the tests was performed at different heights it is assumed that the quad rotor, in both tests was clear of the ground effect, and that the tests for this reason are comparable.

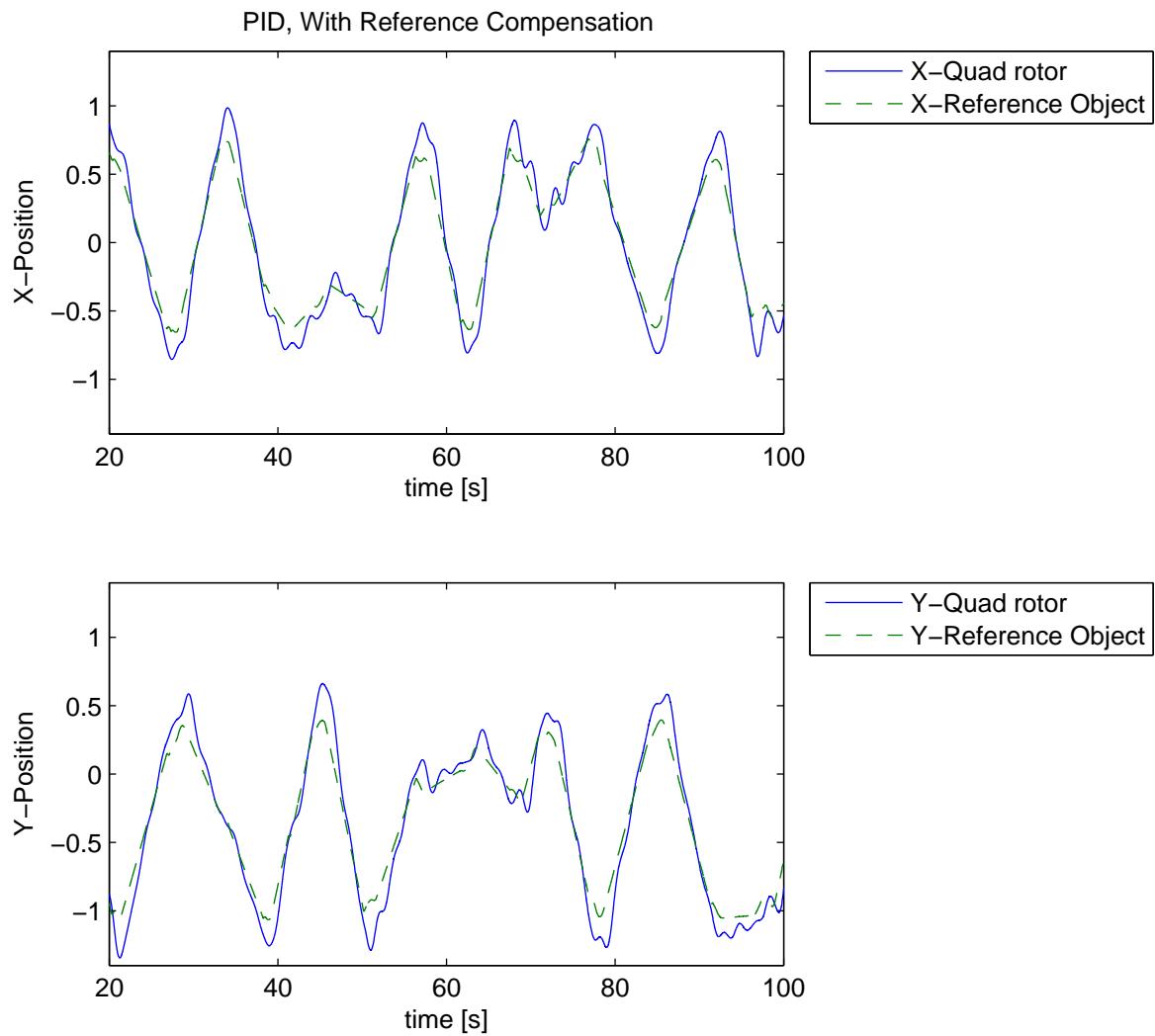


Figure 14.10: The PID controller tracking a moving platforms x- and y-position

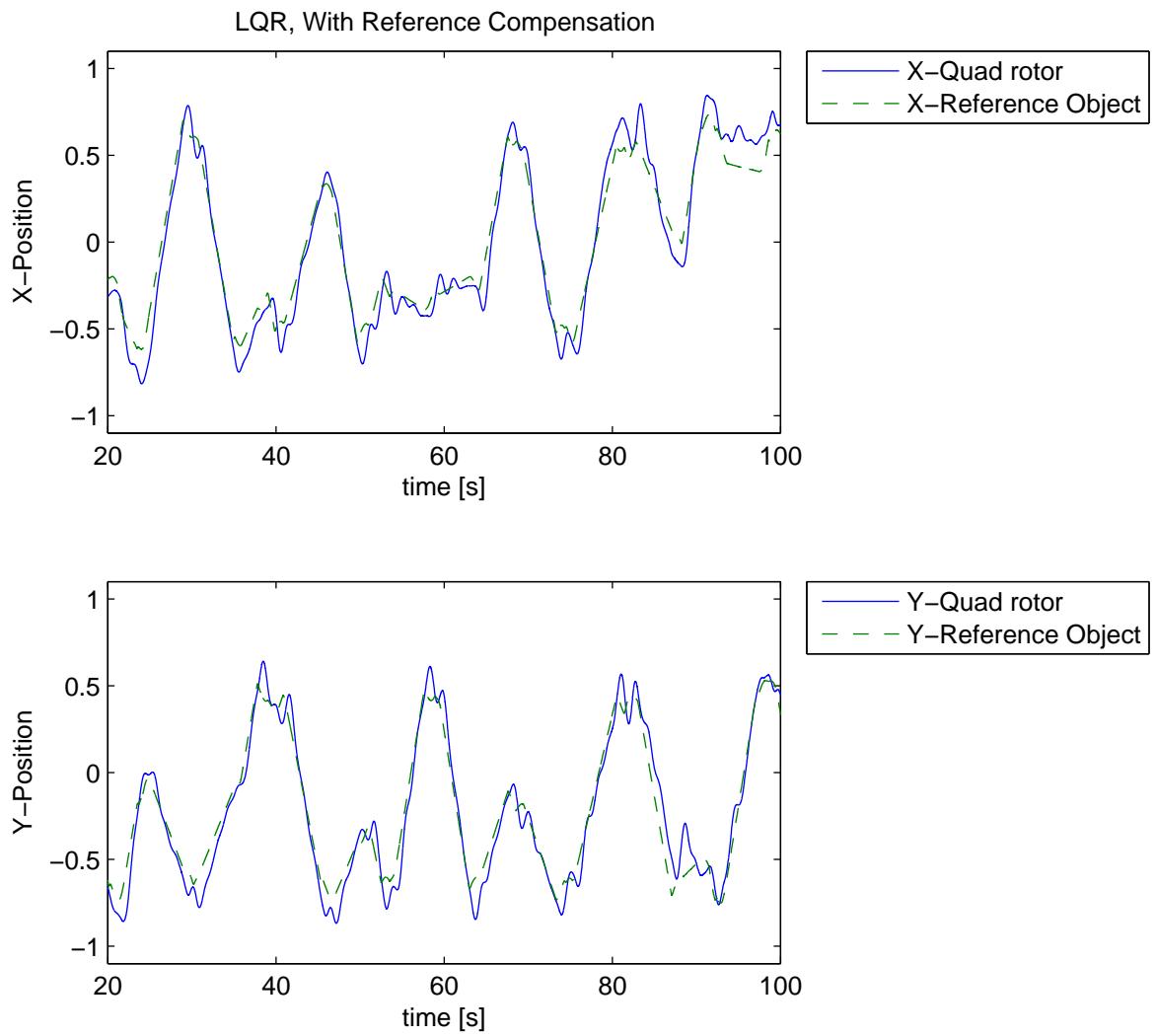


Figure 14.11: The LQR controller tracking a moving platforms x- and y-position

Table 14.4 shows the absolute mean and maximum error of the quad rotors position compared to the objects position.

	PID	LQR
Mean $ r_x $	0.10 m	0.09 m
Mean $ r_y $	0.11 m	0.08 m
Max $ r_x $	0.37 m	0.28 m
Max $ r_y $	0.38 m	0.29 m

Table 14.4: The mean and max tracking error regarding the x- and y-position where r_x and r_y are the sample residuals of the platforms position minus the quad rotors position

It is seen that the LQ regulator is capable of tracking the moving platform with a average error of 0.08 and 0.09 [m] where the PID regulator has a slightly larger average error. The Maximum error of the position gives an indication of how well the regulators respond to a rapid change in reference velocity and direction. This will be the case every time the reference platform reaches the boundary of its territory. Again it is seen that the LQ regulator has a smaller maximum error than the PID regulator.

It should be noted that the reference compensation performed in Chapter 12 is hand tuned to the speed of the platform used in the tests. How the controllers would perform if the speed of the reference object is increased is not evaluated.

14.5 Summary of Controller Evaluation

From the tests conducted it is seen that both the PID and LQ controller is able to make the quad rotor achieve stable hover, perform steps, and track the moving platform with some what similar results, although the PID is shown to be slower tuned than the LQ controller. Another noteworthy point is that additional tuning of the integral action could be done to achieve better offset suppression, which in turn could lead to better performance.

Part III

Epilogue

Table of Contents

15 Conclusion	93
16 Future Work	94
16.1 Implementing Take-off and Landing Algorithms	94
16.2 Predicting Platform Movement	94
16.3 Implementing Noise Model	94
16.4 Onboard Micro Controller and Sensors	95
Bibliography	96

Chapter 15

Conclusion

This goal of the project was to make a X-3D-BL quad rotor helicopter autonomously track and land on a moving platform, using various types of controllers. The classic PID controller and the model based LQ controller was chosen as the controllers to be implemented. A model of the X-3D-BL was derived. This model was used both in the model based LQ controller, but also as a tool for simulation of the developed controllers. Both a non linear as well as a linear model was made. Part of the model was made as a first principle model while the rest was modelled using system identification and parameter estimation. The parts included in the parameter estimation was the generation of the forces and torques acting on the quad rotor, the aerodynamics and the internal controller and motor dynamics. This leaves the derivation of the rigid body kinematics.

The main focus regarding the controllers was the LQ controller. This controller was implemented using a full order observer. A reference was also introduced which makes it possible for the quad rotor to follow a user specified reference. Furthermore integral action was implemented in the LQ controller design. Since the integral action did not perform satisfactory using the chosen design method, hand tuning was applied afterwards. Flying with the yaw angle different from zero was not possible using the developed controller directly, but after implementing a rotation of the feedback gain matrix yawing could be made without affecting the controller.

The PID controller was implemented as three position controllers, three velocity/attitude controllers and a yaw controller. Both the position and the yaw controllers were implemented as PI controllers while the velocity/attitude controllers merely consisted of proportional gains. All of these controllers were tuned by hand.

Tests showed that both the PID and LQ controller was able to achieve stable hover, perform steps, and track a randomly moving platform satisfactorily. The performance of the LQ controller exceeded the performance of the PID controller in some areas. This is due to more time was spent tuning the parameters of the LQ controller as the PID controller merely was seen as a preliminary controller. In the light of the tests and experiences in the laboratory, it is estimated that both controllers can be improved through further tuning of the parameters, especially the integral performance of the LQ controller.

From experience in the laboratory it is ascertained that it was not possible to perform autonomous landing on the moving platform using the developed controllers. This was due to the rather large deviations from the reference when the quad rotor is lowered to the height where ground effect is present.

Future Work

As previously shown not all goal of the project was achieved throughout this report. This is the base for some future work and development.

16.1 Implementing Take-off and Landing Algorithms

Reasonable take-off and landing algorithms must be developed in order for the X-3D-BL quad rotor to do this autonomously. The standard controller can not be used, as it gets unstable in near ground positions, due to ground effect. The ground effect, is caused by the turbulence created by the downward thrust of the rotors. This effect has a large impact on the performance in near ground scenarios.

By implementing take-off and landing algorithms, the quad rotor will then be able to decide by itself, when to land, for instance, if the tracking error is below a pre-set tolerance.

16.2 Predicting Platform Movement

In this report, a random moving platform is considered, with a simple compensation for offset in tracking. By developing a model of the movement of the platform, advanced prediction algorithms for the movement can be developed. This prediction can be used as a base for generating inputs for the controller, more accurately than the current position of the platform.

16.3 Implementing Noise Model

By implementing a noise model in the LQR controller, measurement noise from the system will be suppressed through the feedback loop. For now the Vicon system does not have much noise on measurement, but even small noise signal propagate through the feedback gains, amplifying it to a considerable amount of input noise and therefore implementation of a noise model could benefit the results.

16.4 Onboard Micro Controller and Sensors

For now, the range of the Vicon system is the limit of the movement of both the platform and quad rotor. If for instance, this project should apply to a practical situation, outdoor flight will have to be made possible.

Outdoor flight can only be made possible by adding onboard computational units and sensors. The most apparent sensor to add must be a GPS receiver, to get a position and altitude. Further, the radio must be expanded to receive GPS coordinates of the platform, on which the quad rotor will land.

Bibliography

- [1] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, “Real-time indoor autonomous vehicle test environment,” in *IEEE Control Systems Magazine*. Massachusetts Institute of Technology, Massachusetts, USA: IEEE, April 2008.
- [2] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, “The stanford testbed of autonomous rotorcraft for multi-agent control,” in *Digital Avionics System Conference 2004*, Massachusetts Institute of Technology, Massachusetts, USA, November 2004.
- [3] T. Bak, *Lecture Notes - Modeling of Mechanical Systems*, 1st ed. AAU, 2002.
- [4] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, 2nd ed. Wiley, 2003.
- [5] R. A. Serway and J. W. Jewett, *Physics for Scientists and Engineers with Modern Physics*, 6th ed. Thomson, 2004.
- [6] P. C. Hughes, *Space Attitude Dynamics*. Wiley, 1986.
- [7] I. Newton, *Philosophiae naturalis principia mathematica*, 1st ed. G. Brookman, 1833.
- [8] R. W. Prouty, *Helicopter performance, stability, and control*, 2nd ed. Krieger publishing company, INC., 1986.
- [9] S. Bouabdallah, “Design and control of quadrotors with application to autonomous flying,” Master’s thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2007.
- [10] H. Madsen, *Time Series Analysis*. Chapman & Hall/CRC, 2008.
- [11] Matlab, March 2009, <http://www.mathworks.com>.
- [12] D. E. Penney and C. H. Edwards, *Calculus*, 6th ed. Prentice Hall, 2002.
- [13] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed. Prentice Hall, 2006.
- [14] P. Andersen, “Optimal control,” internet (pdf), February 2009.
- [15] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2000.
- [16] *Ascending Technologies*, x-3D-BL Users Manual.
- [17] “Ascending technologies website,” March 2009, <http://www.asctec.de>.

Part IV

Appendices

Measurements Journals

A.1 Data for Parameter Estimation

The objective of this measurement journal is to gather data to use for the parameter estimation. The parameter estimation data is used to derive models for the relation between the inputs from the RC-transmitter and the body rates of the quad rotor. For the parameter estimation it is decided to obtain data by sampling the response of manual flight.

The sampled data contains time correlated RC-transmitter inputs and Euler angles. Because the data needed for parameter estimation is time correlated RC-transmitter input and body rates, equation 3.8 is applied to the sampled data before used for parameter estimation. This transform is implemented in the .m file found on the CD in Appendix C

A.1.1 Procedure

The procedure for the experiment is as follows.

1. MTLab is powered on (Vicon and Conlink information must be available).
2. A control PC is started with Matlab Simulink and the x3d.mdl model is opened.
3. The X-3D-BL quad rotor is equipped with a newly charged battery and placed on the floor (0,0,0) in the MTLab.
4. The Simulink simulation x3d.mdl is started.
5. The X-3D-BL is flown manually while actuating θ , ϕ and ψ as much and as fast as possible. The flight is divided into three parts. The first part focuses on actuating θ , the second on ϕ and the third on ψ .
6. After 180 sec the simulation stops automatically. Measurements of Euler angels and position is automatically saved.
7. If the measurement is satisfactory the Euler angels is preprocessed by use of equation 3.8 and all data is saved as System Identification objects

More information about the MTLab is found in Appendix 1.3.

The x3d.mdl Simulink model used for recording the measurements is found on the CD under Parameter_Estimation/x3d.mdl.

Manual Flying

The measurement made for the parameter estimation process is performed by flying manually. During manual flight the quad rotor is exposed to an input signal of a large amplitude at one of the orientation axes, meanwhile the other inputs are kept as low as possible.

A.1.2 Measurement Results

In Figure A.1, A.2, and A.3 segments of the measurements of the flight is shown. The figures shows the input values for pitch, roll, and yaw and the respective measured body rates. All the data used for parameter estimation can be found on the CD in Appendix C

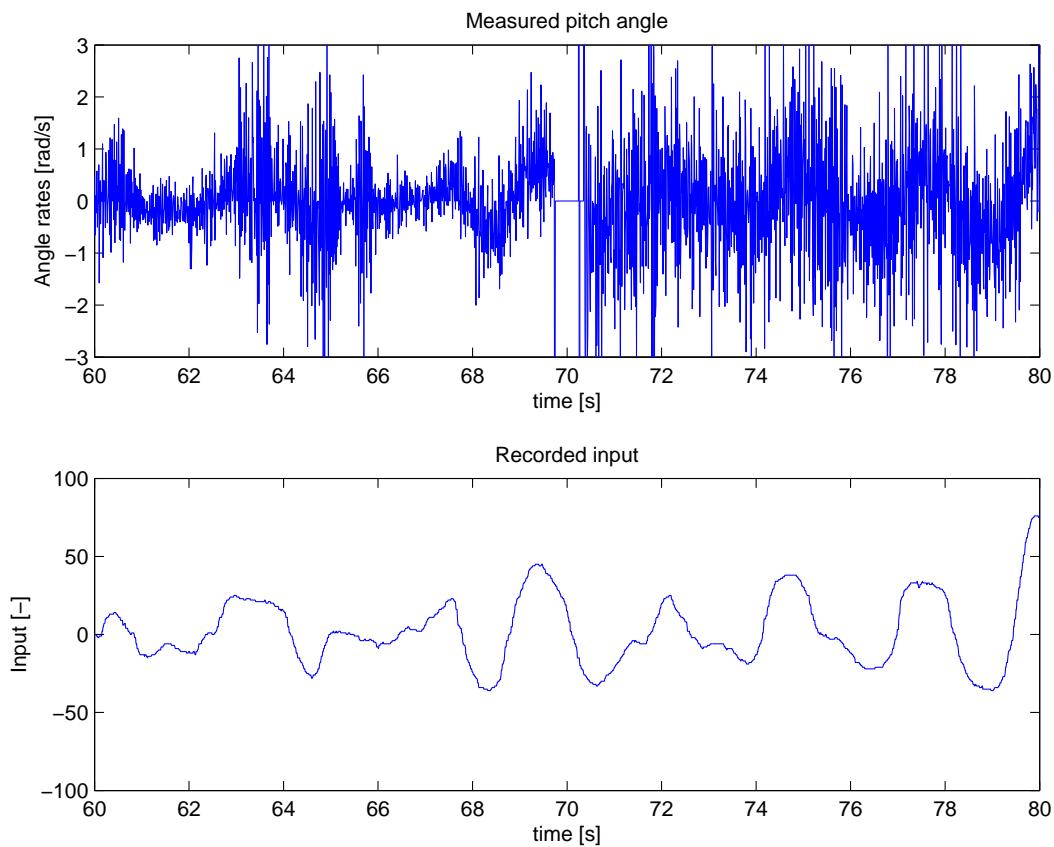


Figure A.1: Measurement of pitch response doing a manual flight

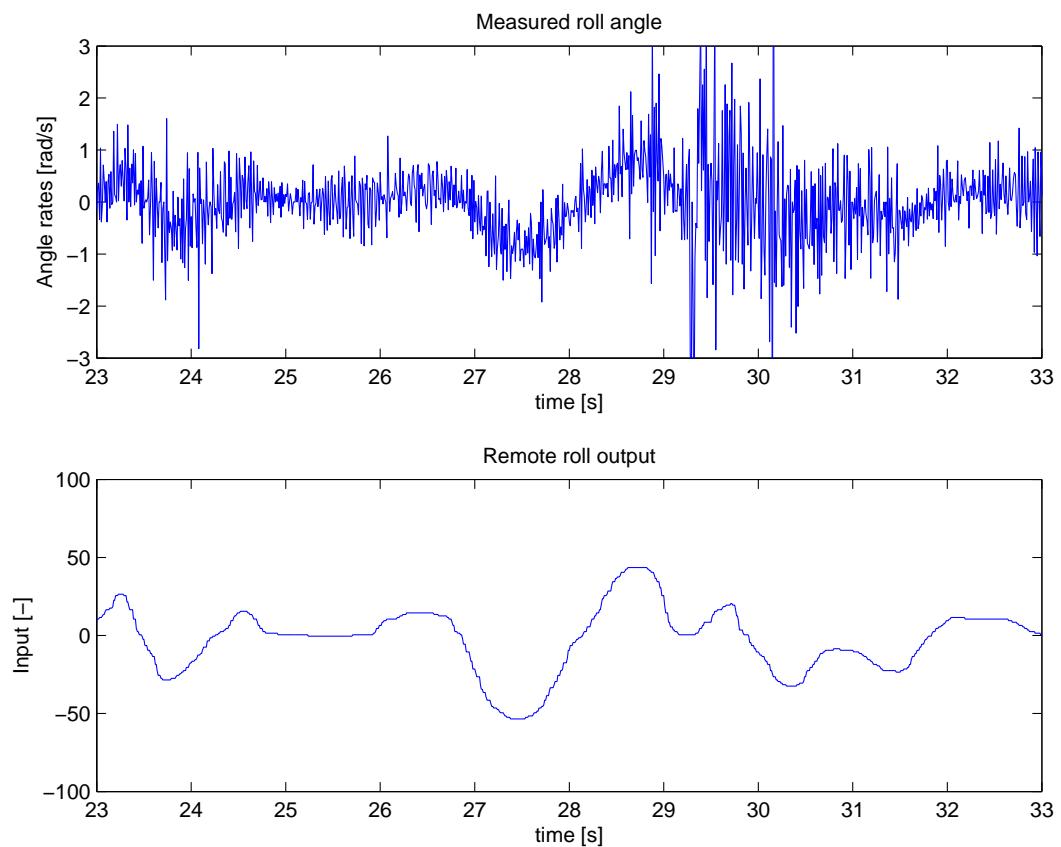


Figure A.2: Measurement of roll response doing a manual flight

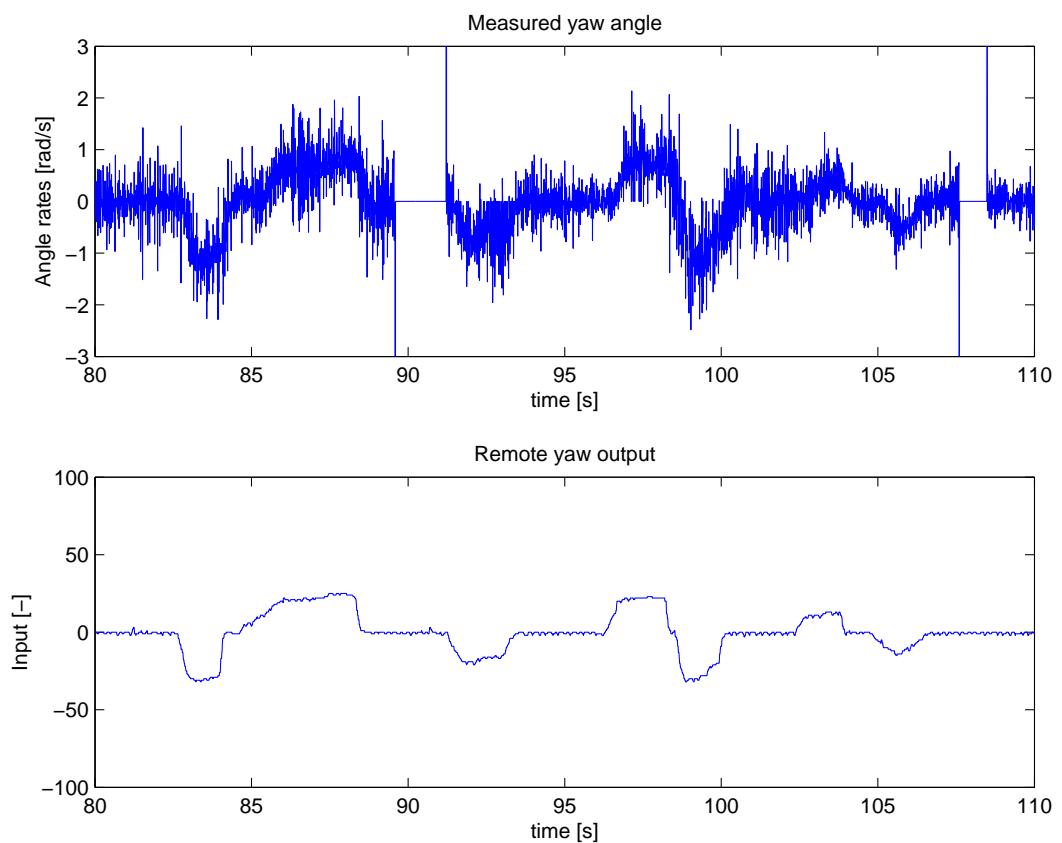


Figure A.3: Measurement of yaw response doing a manual flight

A.2 Polynomial Relation of S_{col} and the Thrust Force

The objective of this measurement journal is to find a polynomial relation from the collective input S_{col} from the RC-controller to the thrust force induced on the quad rotor in near hover flight. To find this, the quad rotor is attached to a box that is so heavy that it can not lift it. Measurements of changes in the weight are then recorded while the input is increasing. The amount of weight loss is proportional to the force induced as long as the acceleration is kept constant. This originates from Newtons second law.

$$F = m \cdot a \quad (\text{A.1})$$

Where

F is the force [N]

m is the mass [kg]

a is the acceleration [m/s^2]

A.2.1 Procedure

The procedure for the experiment is as follows.

1. MTLab is powered on (Vicon and Conlink information must be available).
2. A control PC is started with Matlab Simulink and the x3d.mdl model is opened.
3. The X-3D-BL quad rotor is equipped with a newly charged battery and attached to a box with a bigger mass than the quad rotor is capable of lifting.
4. The quad rotor and box is then placed on a scale on a elevated (0.75 m) platform as shown in Figure A.4.
The rotors needs to be free from the platform to avoid any ground effect.
5. The test signal for collective is applied while the scale is read off and recorded.
6. When the test signal is done, the motors are returning to idle and the experiment is done.

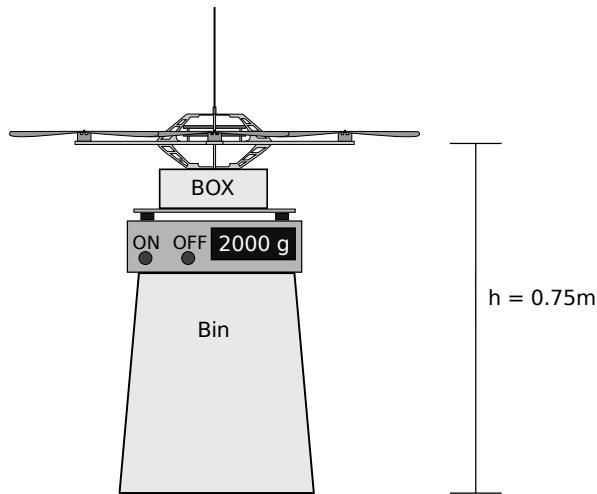


Figure A.4: The set-up used in the thrust test

The scale used is a Rådvad Elevtronics IS075 kitchen scale measuring at a resolution of 1 gram. Further information about the MTLab is found in section 1.3.

The x3d.mdl Simulink model used for applying the test signals and recording the measurements is found on the CD under Parameter_Estimation/x3d.mdl.

A.2.2 The Test Signal

The signal used for the experiment is a step signal as shown in Figure A.5. The initial input value is 100 and the final value before returning to idle is -100.

Between each step there is a pause of 5 seconds to read off the scale. Every step changes the input with 25.

A.2.3 Results

The thrust force can then be expressed as in Equation A.2

$$F_t = \Delta m * g \quad (\text{A.2})$$

Where: F_t is the thrust force [N]

Δm is the change in mass [kg]

g is the gravitational acceleration [m/s^2]

The results of the measurements can be seen in Table A.1.

The calculated force of each input step is shown on Figure A.6 as a continuous graph. To estimate the graph as a polynomial the least squares problem for the following equation is minimised.

$$a_3x^3 + a_2x^2 + a_1x + a_0 = f(x) \quad (\text{A.3})$$

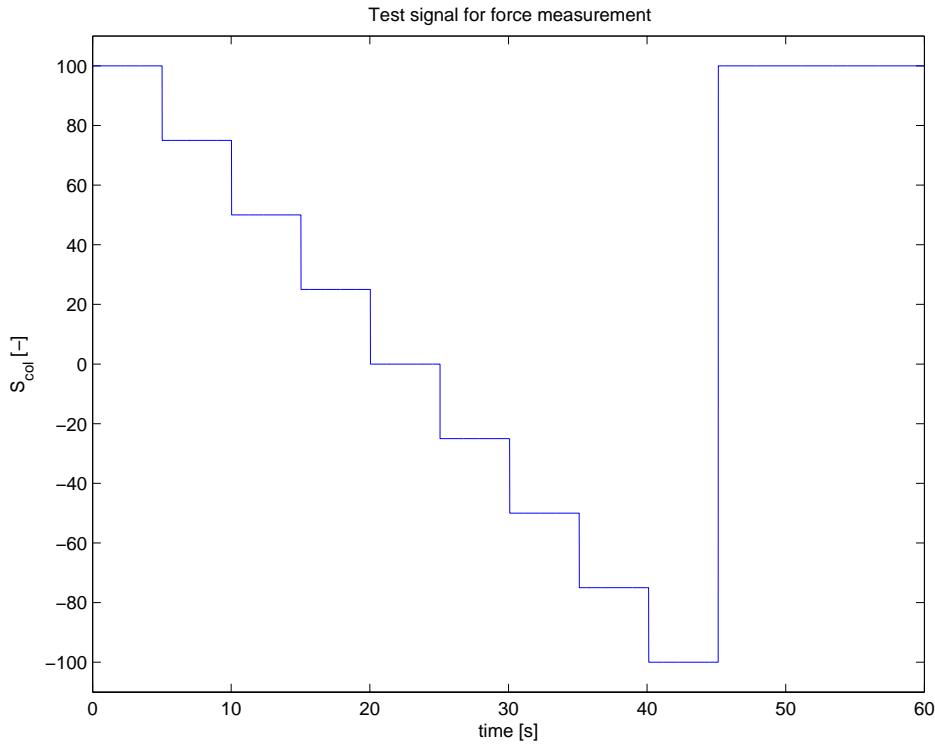


Figure A.5: The input signal used for force measurements

	Measurement 1		Measurement 2	
Initial weight [g]	2.0040		2.0050	
RC input	m [g]	Δm [g]	m [g]	Δm [g]
100	1.9060	0.0980	1.9040	0.1010
75	1.8460	0.1580	1.8490	0.1560
50	1.6950	0.3090	1.7030	0.3020
25	1.5440	0.4600	1.5400	0.4650
0	1.3780	0.6260	1.3800	0.6250
-25	1.2400	0.7640	1.2300	0.7750
-50	1.1100	0.8940	1.1100	0.8950
-75	1.0100	0.9940	1.0000	1.0050
-100	0.9800	1.0240	0.9700	1.0350

Table A.1: The results of the force measurement, where the initial weight is the total mass of the quad rotor and the box. Δm is the total mass minus the measured mass at a given input. m is the mass measured by the scale

Where $f(x)$ is the result of one measurements and $a_0 - a_3$ is parameters to a 3rd order polynomial. Figure A.6 also shows the 3rd order polynomial that is fitted to the measured graph. The polynomial can be described by the parameters $a_3 = 0.0000018$, $a_2 = -0.000055$, $a_1 = -0.064$, and $a_0 = 6.075$.

The polynomial then becomes:

$$f(x) = 0.0000018x^3 + -0.000055x^2 + -0.064x + a_0 = 6.075 \quad (\text{A.4})$$

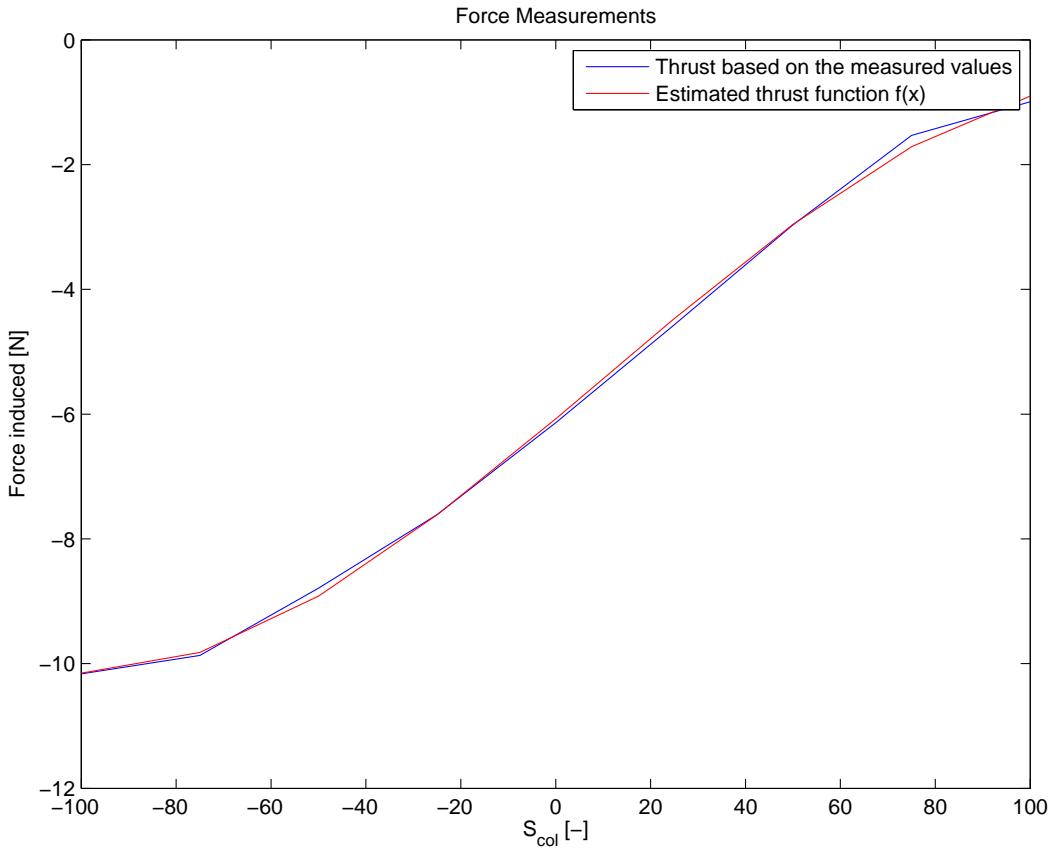


Figure A.6: The measured results plotted together with the $F_t(x)$

A.3 Estimation of Induced Inflow Emulation Coefficient

The objective of this measuring journal is to estimate a induced inflow emulation coefficient used in the linear model of the quad rotor.

Instead of modelling the effect of induced inflow by first order principle, it is chosen to estimate the effect as a function of the speed of the quad rotor in the vertical body plane.

A.3.1 Procedure

For this measurement the quad rotor will be stabilised by a human pilot. By manually applying steps to the collective value it is expected that the quad rotor will accelerate in the Z-direction. Eventually the quad rotor will reach a speed where the effect of induced inflow will cancel further acceleration. The relation between applied force, as a function of the step, and speed will be used as induced inflow emulation coefficient

A.3.2 Results

Table A.2 shows the measuring results. The mean of the relation between force and velocity is found to be -0.3559, which is used in the linear model. The variance of the results is primarily caused by the battery

Step size	Change in force	Velocity	$\frac{force}{velocity}$
4.5 [-]	0.2833 [N]	-0.7800 [$\frac{m}{s}$]	-0.3632
2.0 [-]	0.1253 [N]	-0.3000 [$\frac{m}{s}$]	-0.4178
2.0 [-]	0.1258 [N]	-0.4500 [$\frac{m}{s}$]	-0.2794
1.5 [-]	0.0940 [N]	-0.2400 [$\frac{m}{s}$]	-0.3915
3.5 [-]	0.2194 [N]	-0.6700 [$\frac{m}{s}$]	-0.3274

Table A.2: Measurements of the estimated induced airflow coefficient

discharge, but the accuracy is also limited by the pilots ability to stabilise the angles to zero.

A.4 Regulator Performance Tests

The objective of this measuring journal is to produce the data needed for verification of both the PID and LQ regulator.

A.4.1 Procedure

Each regulator will be put through the same set of tests. The set of tests will include two hover tests, three step tests and two tests with sinusoidal input. Finally the regulators ability to track a moving platform will be tested.

Test 1: Hovering until the battery is discharged in (0,0,-1) with yaw angle 0 deg (0 rad).

Test 2: Hovering until the battery is discharged in (0,0,-1) with yaw angle 135 deg ($\frac{3\pi}{4}$ rad).

Test 3: Positive step in x direction from hover.

Test 4: Positive step in y direction from hover.

Test 5: Negative step in z direction from hover.

Test 6: Sinusoidal input on x, y and z.

Test 7: Sinusoidal input on x, y, z and a ramp signal on yaw.

Test 8: Tracking of a moving platform.

Test Reference Inputs

The reference inputs used in the tests are as follows:

Test 1: Static reference of (0,0,-1,0) for about 10 minutes.

Test 2: Static reference of (0,0,-1, $\frac{3\pi}{4}$) for about 10 minutes.

Test 3: The reference for the first 30 seconds is (0,0,-1,0), next 15 seconds is (1,0,-1,0) and the last 15 seconds is (0,0,-1,0).

Test 4: The reference for the first 30 seconds is (0,0,-1,0), next 15 seconds is (0,1,-1,0) and the last 15 seconds is (0,0,-1,0).

Test 5: The reference for the first 30 seconds is (0,0,-1,0), next 15 seconds is (0,0,-1.5,0) and the last 15 seconds is (0,0,-1,0).

Test 6: The reference over 60 seconds is $(0.5 \cdot \sin(\frac{\pi}{4}t), 0.5 \cdot \sin(\frac{\pi}{4}t + \frac{\pi}{2}), 0.3 \cdot \sin(\frac{\pi}{4}t) - 1, 0)$.

Test 7: The reference over 60 seconds is $(0.5 \cdot \sin(\frac{\pi}{4}t), 0.5 \cdot \sin(\frac{\pi}{4}t + \frac{\pi}{2}), 0.3 \cdot \sin(\frac{\pi}{4}t) - 1, \text{repeating ramp function that rises from } -\pi \text{ to } \pi \text{ over 8 seconds.}$

Hover procedure

1. MTLab is powered on (Vicon and Conlink information must be available).
2. A control PC is started with Matlab Simulink and the x3d_lqr.mdl model is opened.
3. The X-3D-BL quad rotor is equipped with a newly charged battery.
4. The X-3D-BL is flown manually to the position (0,0,-1) and held there hovering.
5. The x3d_lqr_nonlinear.mdl is run and autonomous flight is engaged.
6. After approx. 10 min. of hover the autonomous flight is disengaged and the quad rotor is landed manually.
7. The simulink simulation is stopped

This procedure is used both Test 1 and Test 2

Step and sinusoidal procedure

1. MTLab is powered on (Vicon and Conlink information must be available).
2. A control PC is started with Matlab Simulink and the x3d_lqr.mdl model is opened.
3. The X-3D-BL quad rotor is equipped with a newly charged battery.
4. The X-3D-BL is flown manually to the position (0,0,-1) and held there hovering.
5. The x3d_lqrmdl is run and autonomous flight is engaged.
6. After approx. 60 sec. the autonomous flight is disengaged and the quad rotor is landed manually.

This procedure is used for tests 3 through 7.

A.4.2 Procedure for Tracking of a Moving Platform

For this test a moving platform is needed. For this purpose a "iRobot Create" is used. The iRobot is capable of moving autonomously within a confined territory with a maximum velocity of ≈ 0.3 [m/s]. Following procedure was used.

1. MTLab is powered on (Vicon and Conlink information must be available).
2. A control PC is started with Matlab Simulink and the x3d.mdl model is opened.
3. The X-3D-BL quad rotor is equipped with a newly charged battery.
4. The iRobot is placed in a physically confined territory of size $2 \cdot 2$ m and activated.
5. The X-3D-BL is flown manually to the position (0,0,-1) and held there hovering.
6. The x3dmndl is run and autonomous flight is engaged with object tracking, yaw reference = 0 and the Z-reference = -1.
7. After approx. 135 sec. the tracking of the platforms yaw is engaged.
8. After approx. 180 sec. the autonomous flight is disengaged and the quad rotor is landed manually.

A.4.3 Results

In this section the data collected during the tests will be presented. The plots shown in this section represents the entire duration of the test flights. This causes the plots of some of the measured values to be unclear. This is especially apply to many of the control effort plots. Relevant scaling and interpretation of these plots can be found in chapter 14 in the main report.

Hover results for PID controller

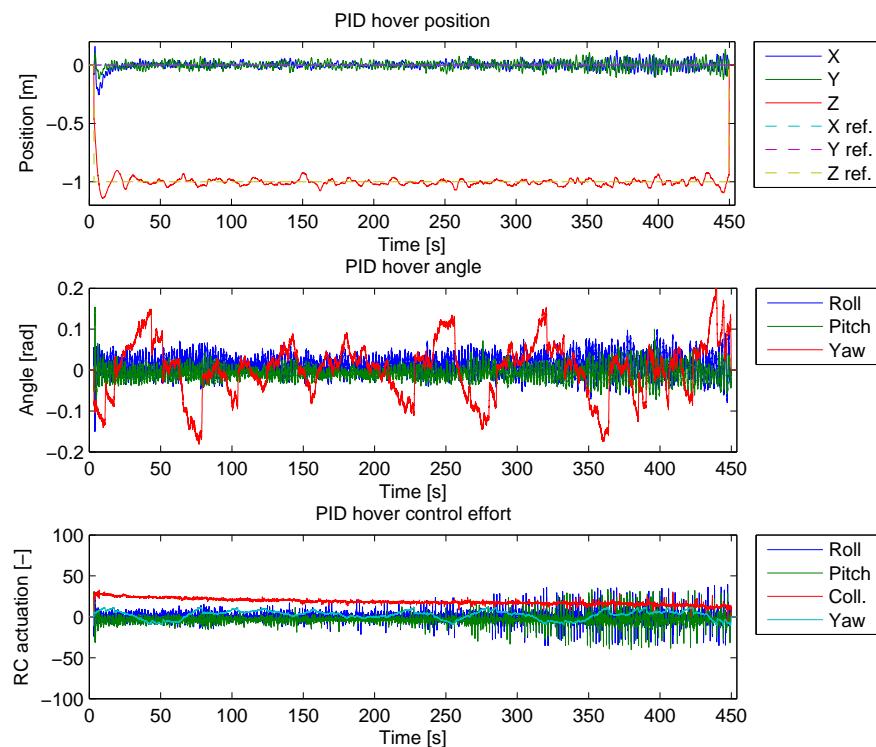


Figure A.7: Test of hover capabilities for the PID controller. Top: Position. Middle: Angle. Bottom: Control effort.

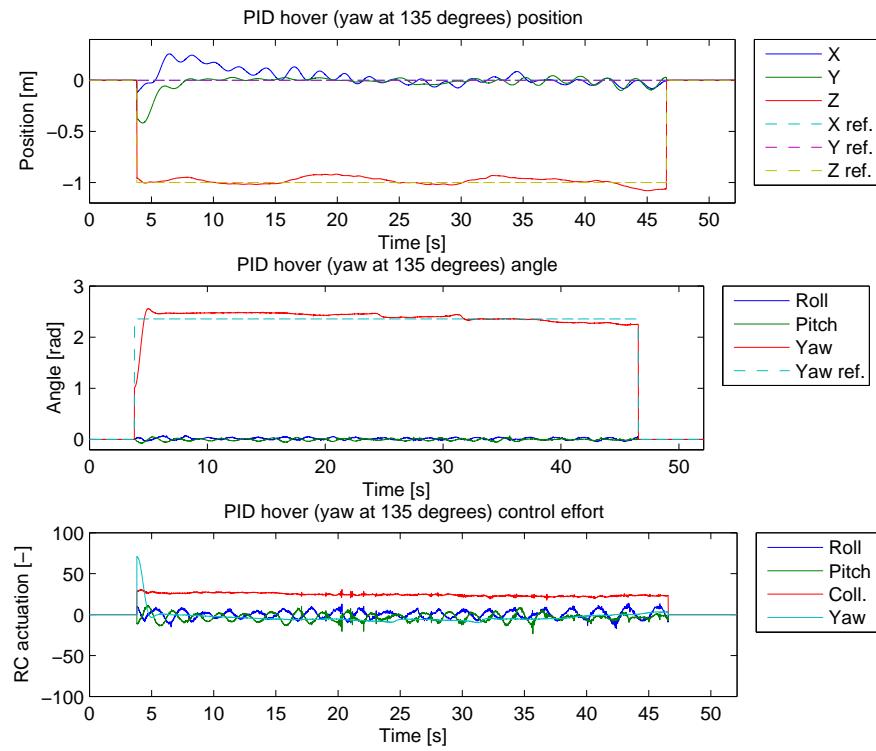


Figure A.8: Test of hover capabilities for the PID controller with yaw at angle 135 degrees. Top: Position. Middle: Angle. Bottom: Control effort.

Hover results for LQ controller

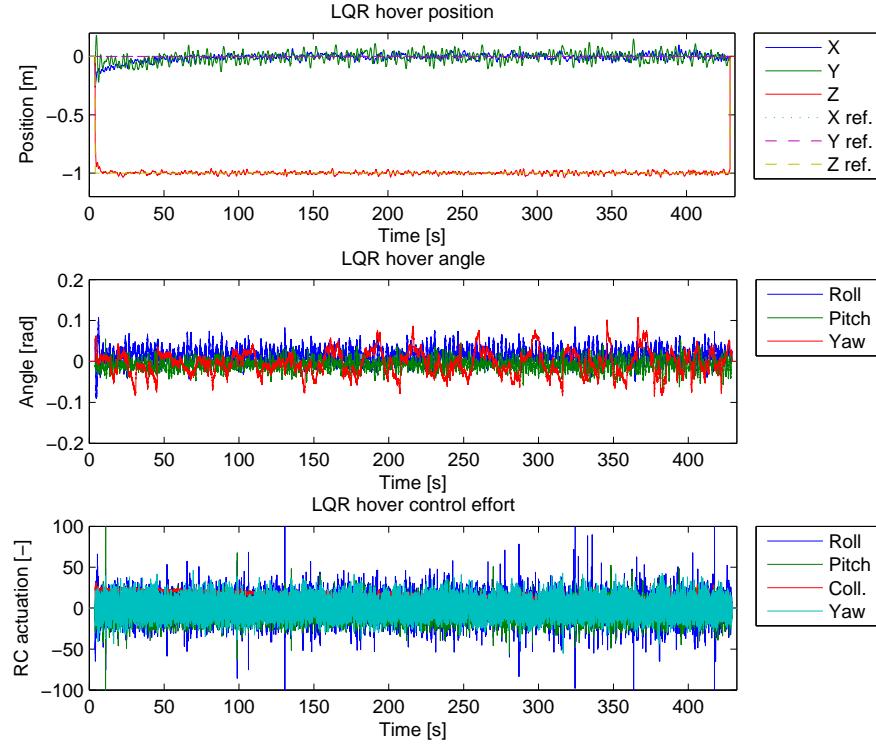


Figure A.9: Test of hover capabilities for the LQ controller. Top: Position. Middle: Angle. Bottom: Control effort.

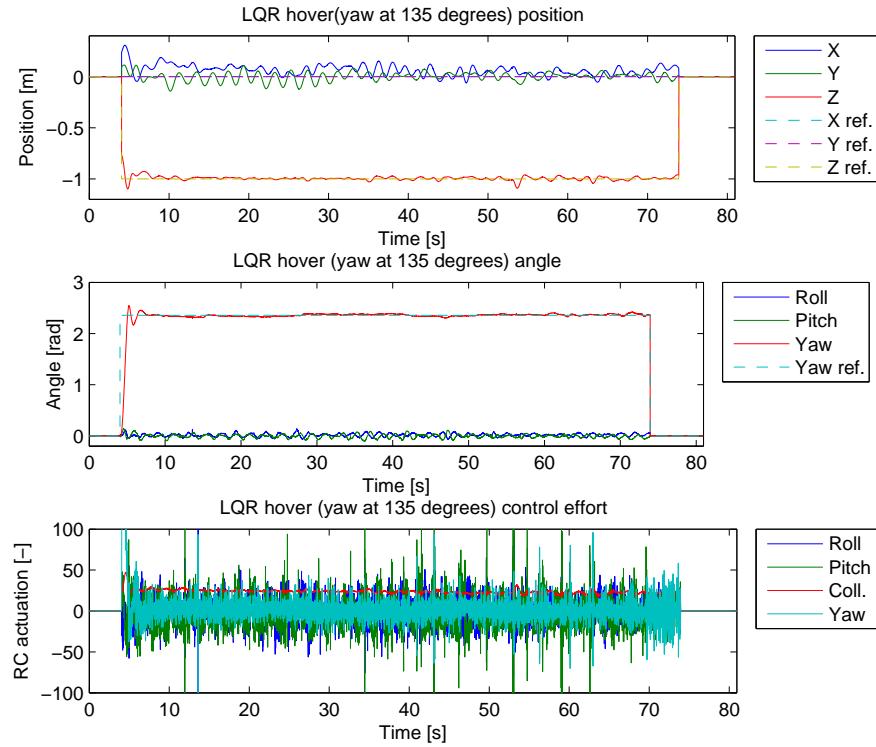


Figure A.10: Test of hover capabilities for the LQ controller with yaw at angle 135 degrees. Top: Position. Middle: Angle. Bottom: Control effort.

Step results for PID controller

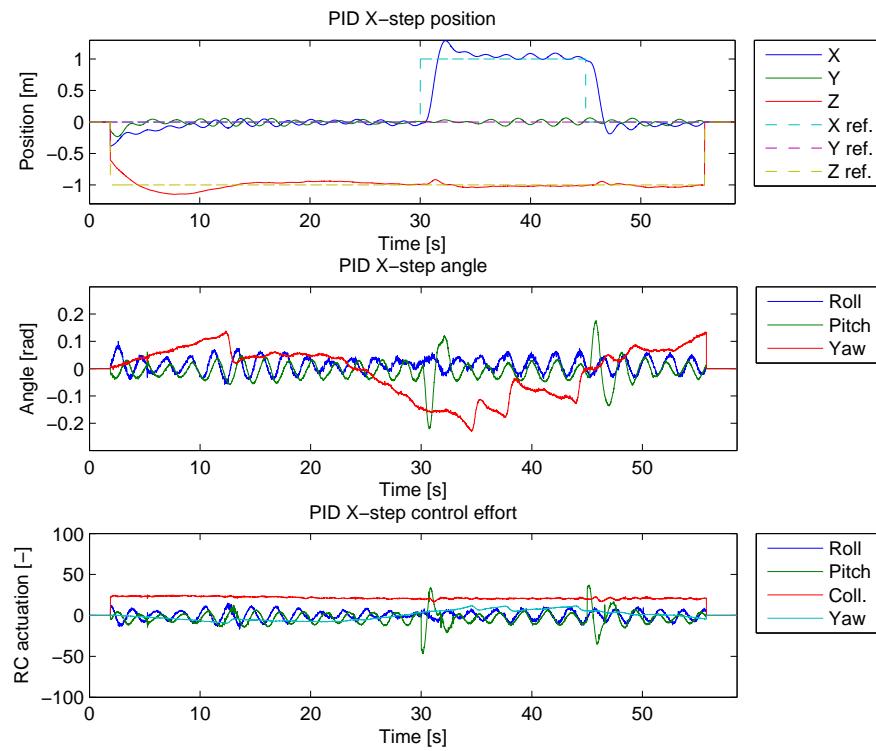


Figure A.11: Test of step capabilities in positive X direction, for the PID controller. Top: Position. Middle: Angle. Bottom: Control effort.

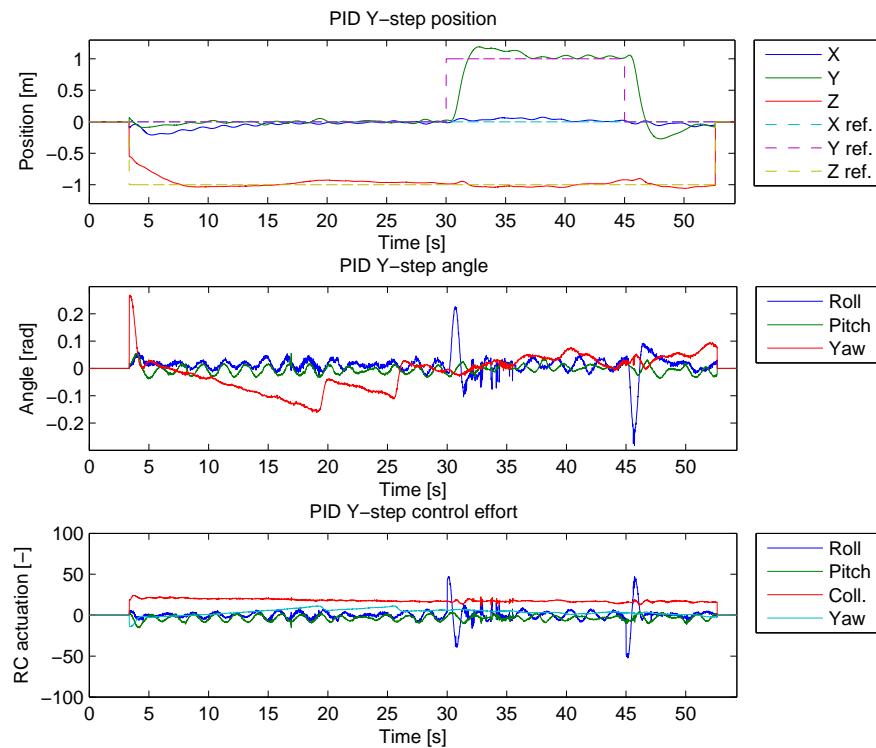


Figure A.12: Test of step capabilities in positive Y direction, for the PID controller. Top: Position. Middle: Angle. Bottom: Control effort.

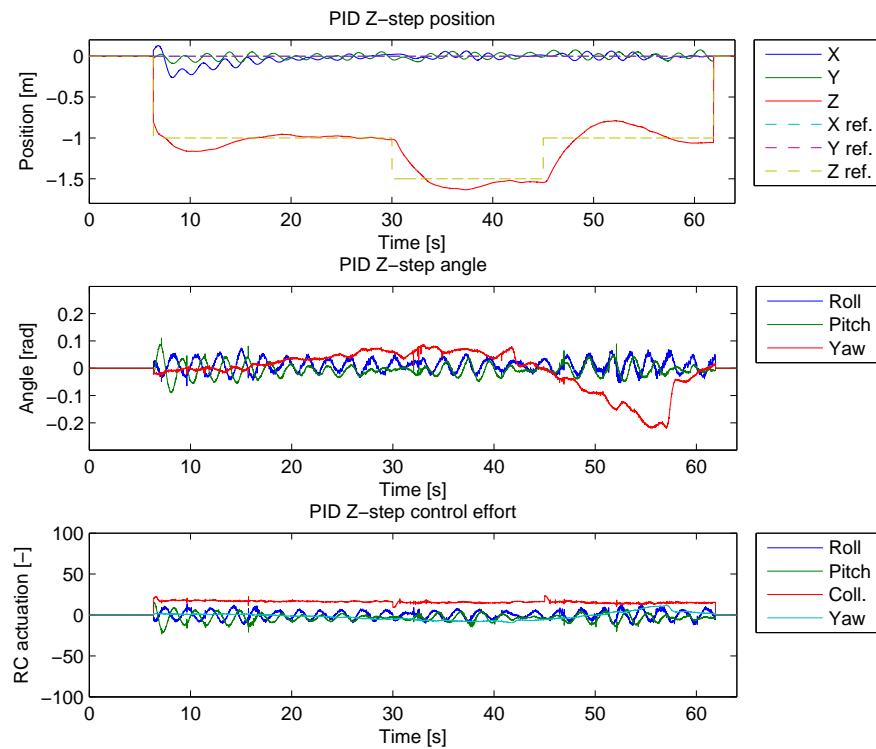


Figure A.13: Test of step capabilities in positive Z direction, for the PID controller. Top: Position. Middle: Angle. Bottom: Control effort.

Step results for LQ controller

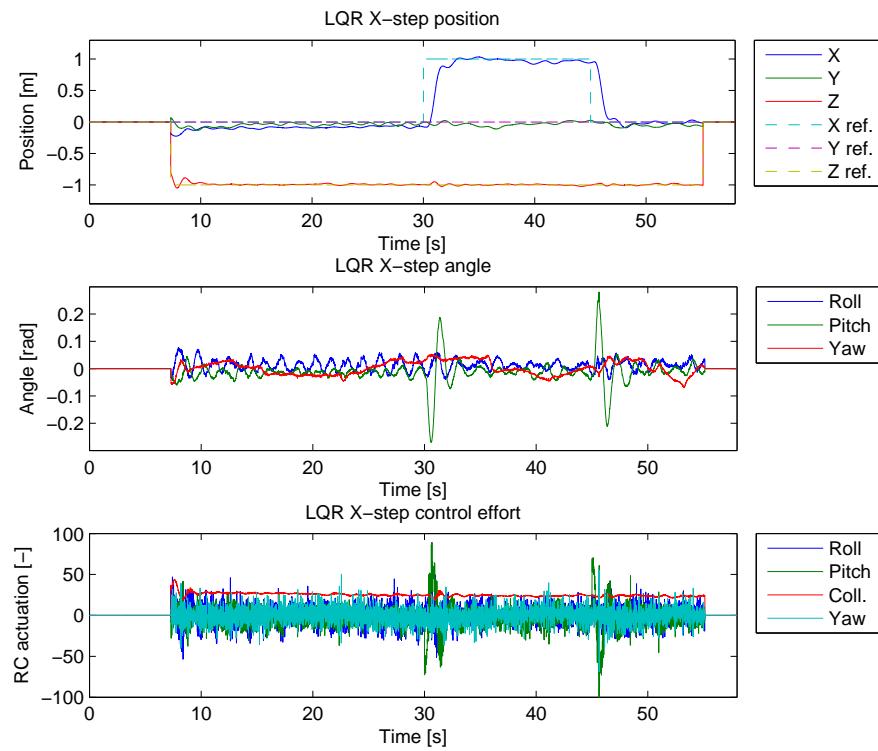


Figure A.14: Test of step capabilities in positive X direction, for the LQ controller. Top: Position. Middle: Angle. Bottom: Control effort.

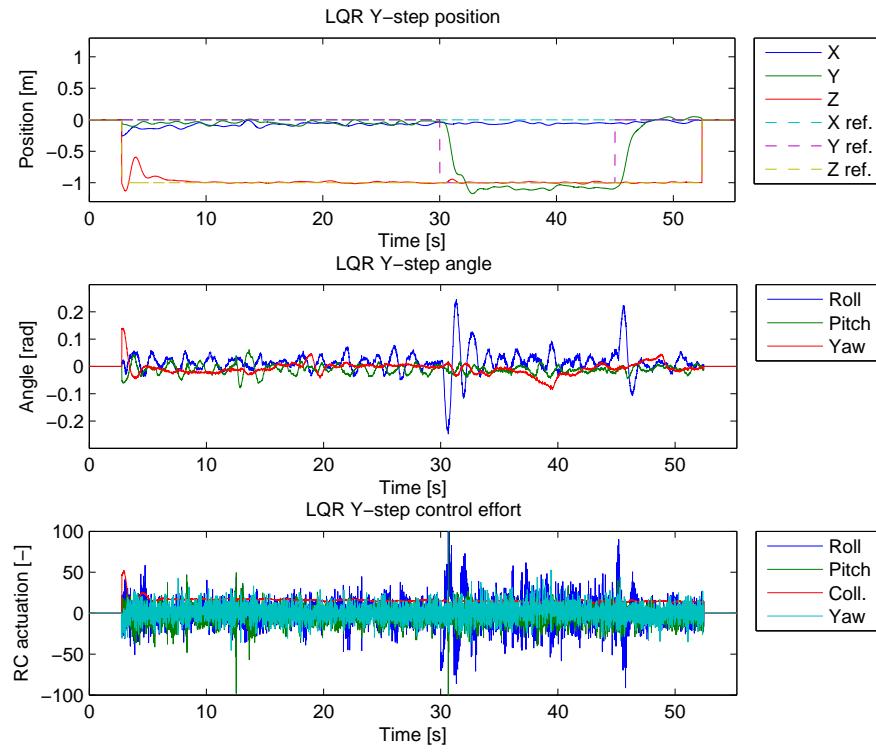


Figure A.15: Test of step capabilities in negative Y direction, for the LQ controller. Top: Position. Middle: Angle. Bottom: Control effort.

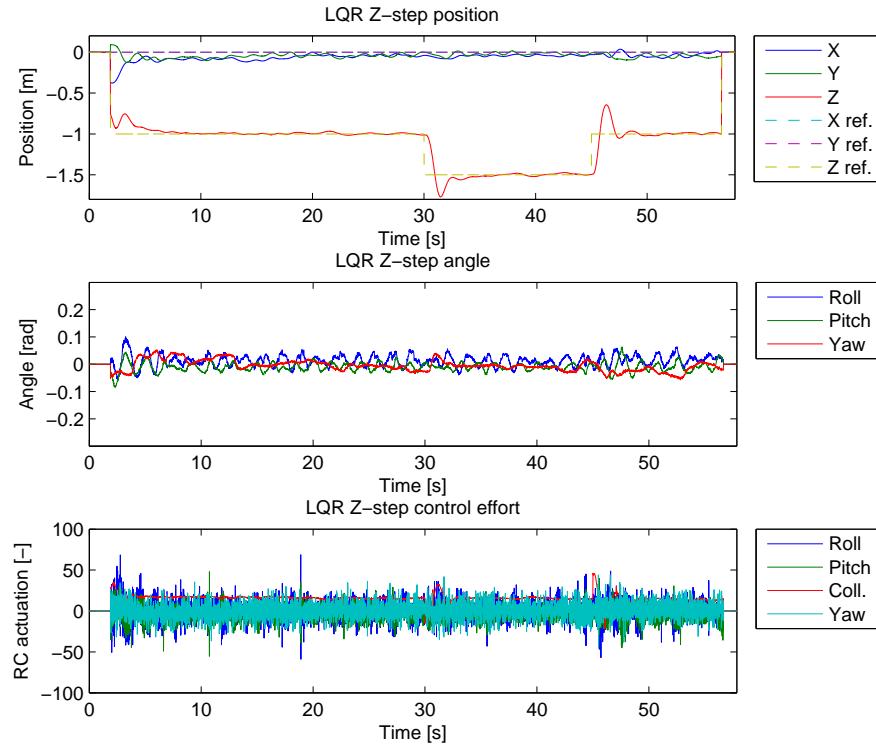


Figure A.16: Test of step capabilities in positive Z direction, for the LQ controller. Top: Position. Middle: Angle. Bottom: Control effort.

Sinusoidal results for PID controller

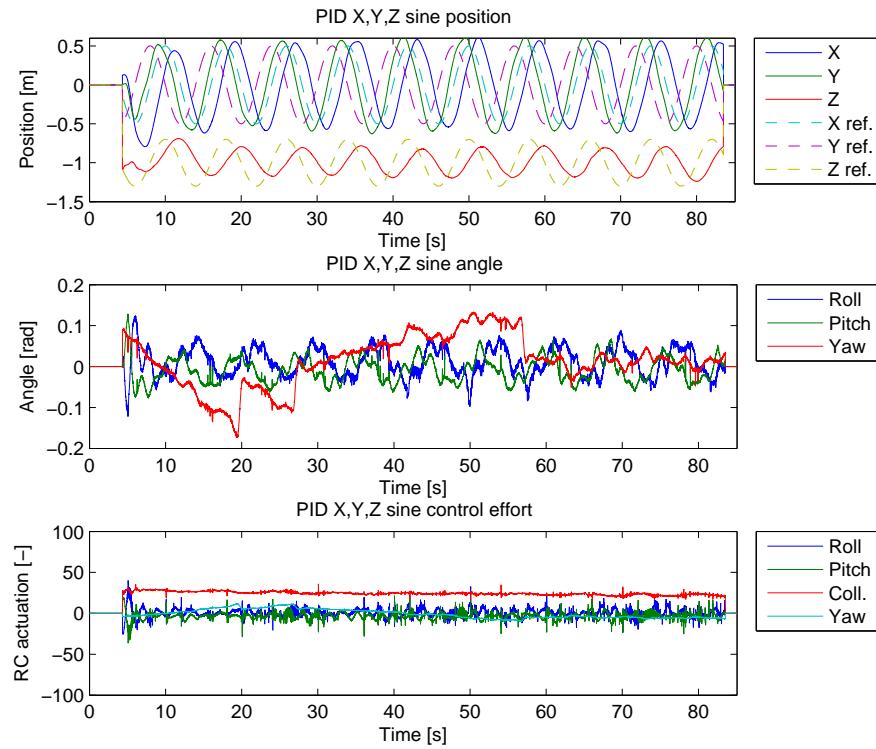


Figure A.17: Test of PID controller following a continuous sinusoidal reference on X,Y and Z positions. Top: Position. Middle: Angle. Bottom: Control effort.

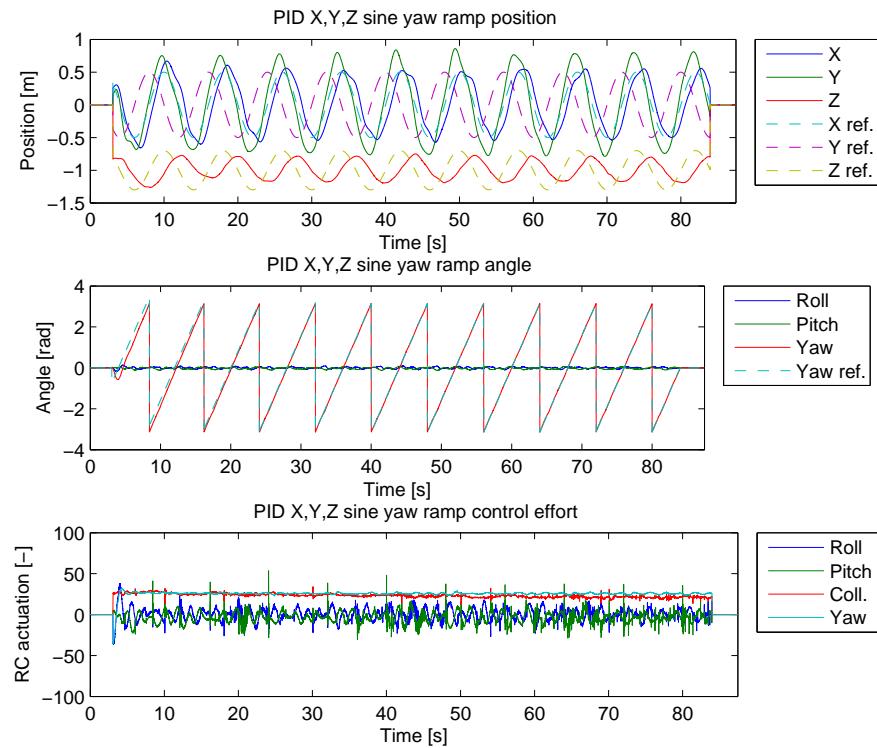


Figure A.18: Test of PID controller following a continuous sinusoidal reference on X,Y and Z positions and a ramp on yaw angle. Top: Position. Middle: Angle. Bottom: Control effort.

Sinusoidal results for LQ controller

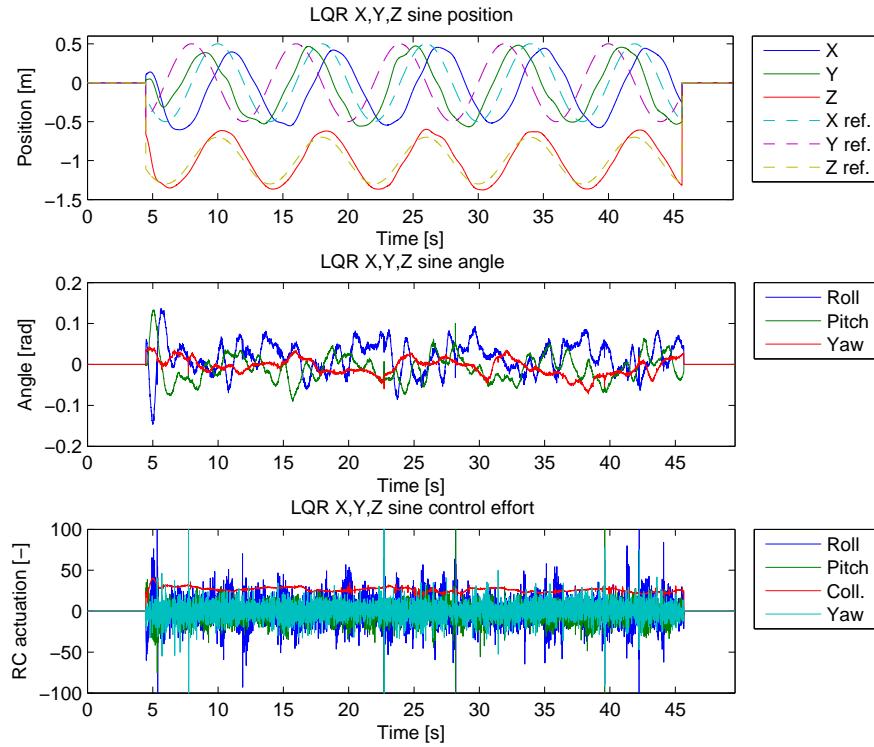


Figure A.19: Test of LQ controller following a continuous sinusoidal reference on X,Y and Z positions. Top: Position. Middle: Angle. Bottom: Control effort.

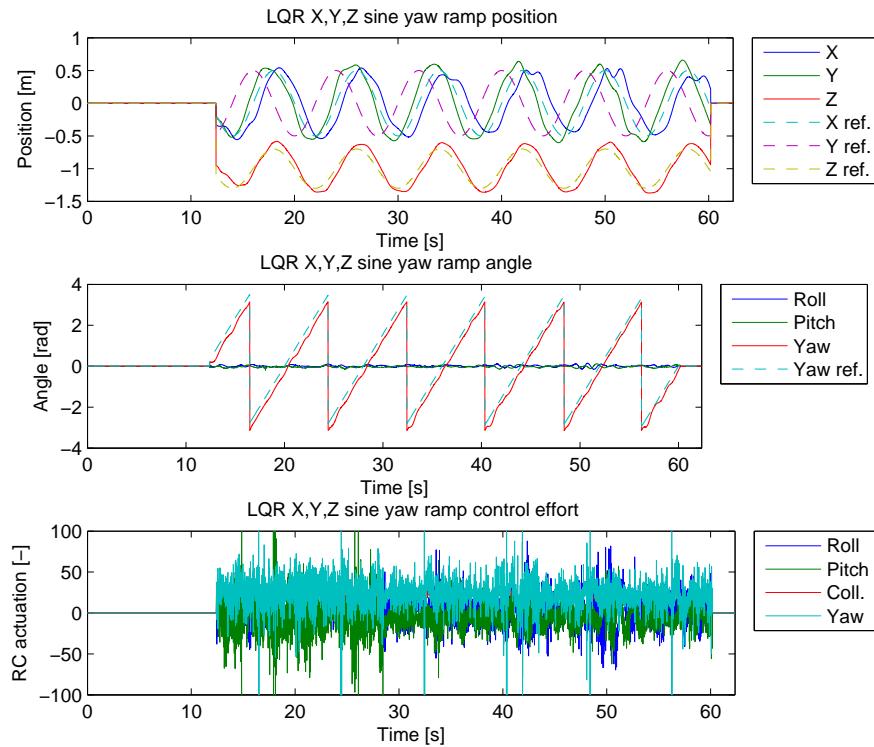


Figure A.20: Test of LQ controller following a continuous sinusoidal reference on X,Y and Z positions and a ramp on yaw angle. Top: Position. Middle: Angle. Bottom: Control effort.

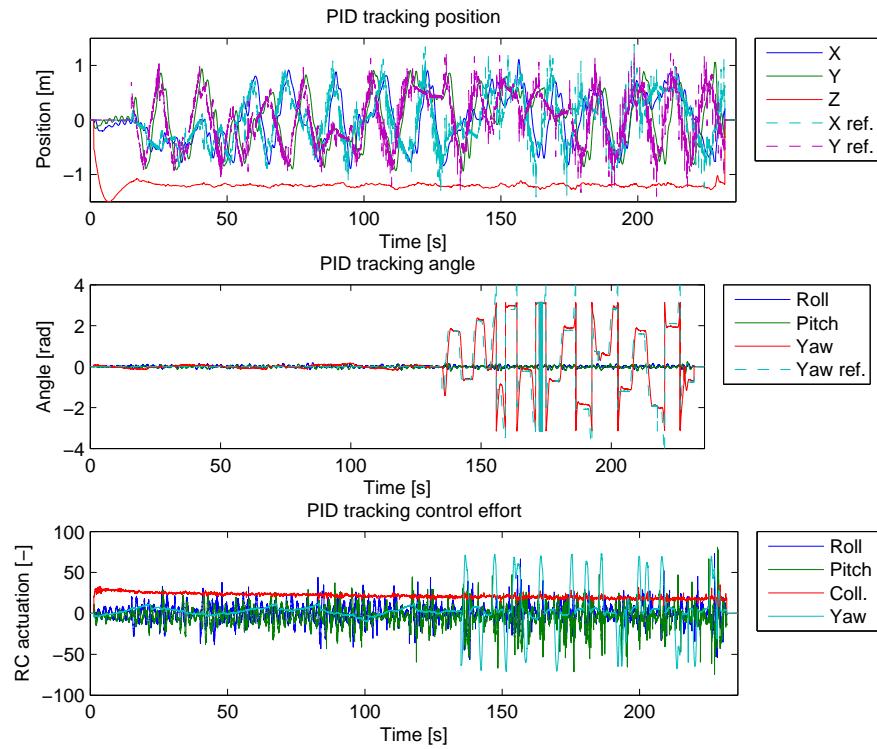
Tracking for PID controller

Figure A.21: Test of PID controller tracking a randomly moving platform. Top: Position. Middle: Angle. Bottom: Control effort.

Tracking for LQ controller

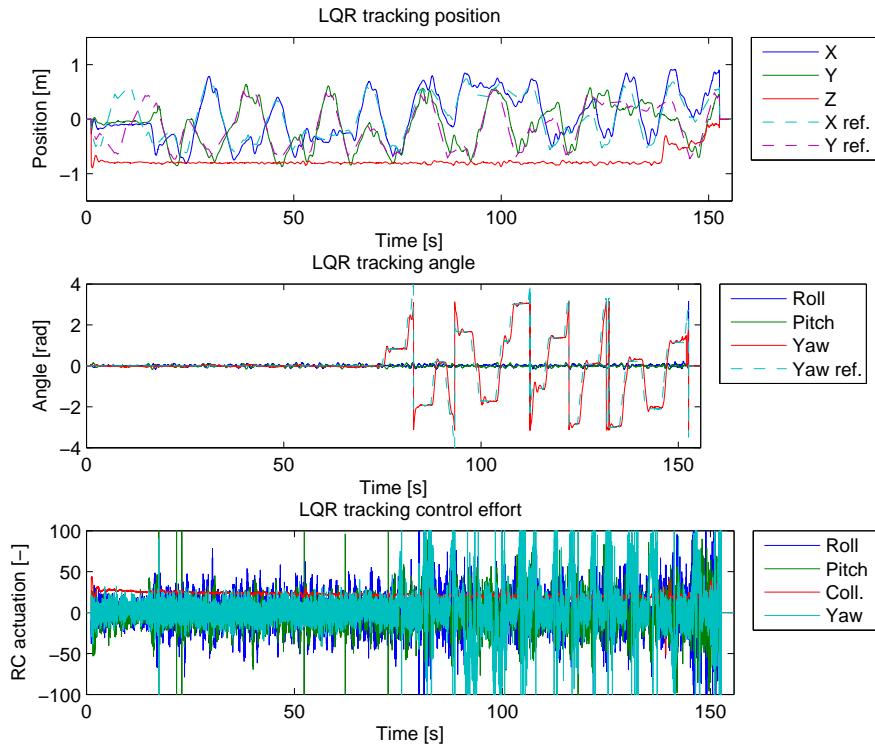


Figure A.22: Test of LQ controller tracking a randomly moving platform. Top: Position. Middle: Angle. Bottom: Control effort.

A.5 Estimation of Battery Discharge Coefficients

When flying the quad rotor manually it has been observed that over time it is necessary to increase the collective input to maintain hover. This is only observed when flying on battery and not with a power supply. It is therefore assessed that this is due to the discharging of the battery. To determine the proportions of this phenomena, this experiment is conducted. The result is a polynomial describing the changes in collective RC-controller input needed to keep the quad rotor hovering. The polynomial is to be used as part of a non-linear model.

A.5.1 Procedure

1. The quad rotor is equipped with a fully charged battery
2. The quad rotor is manually flown to position (0,0,-1)
3. The PID controller takes over control and keep the quad rotor in position
4. When the indication on low battery activates the experiment is interrupted

A.5.2 Results

The result of the flight is shown as the blue curve in Figure A.23. The curve seen has been smoothed by the smooth function in Matlab, using smooth coefficient 100. To emulate the curve a 3rd order polynomial fit has

been made in Matlab. The polynomial fit is shown as the red curve in Figure A.23.

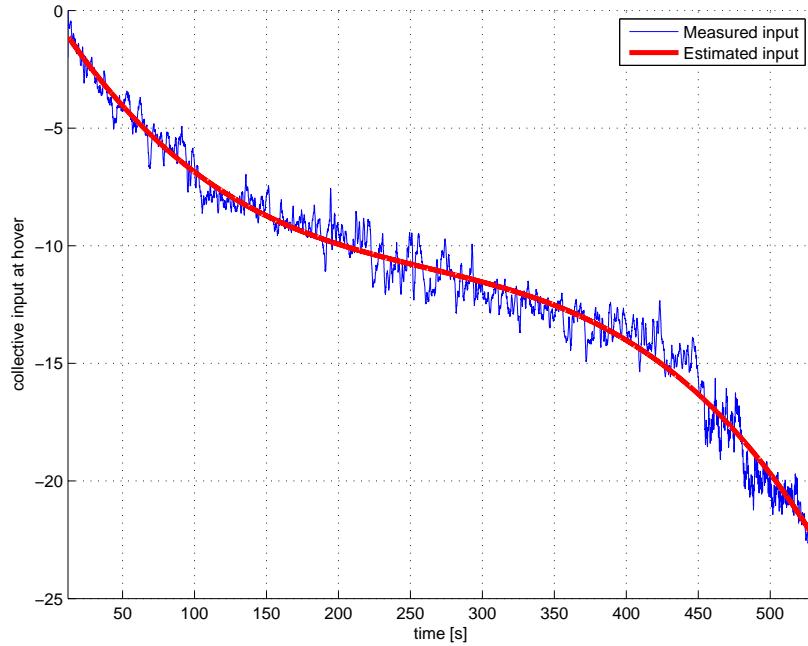


Figure A.23: The collective input increases (100 minimum, -100 maximum) over time because of the battery discharge. The red line is an estimation of the discharge

The coefficients for the polynomial is listed in Table A.3.

a3	a2	a1	a0
-0.000000388	0.000306	-0.0954	0

Table A.3: coefficients for the 3rd order polynomial describing the battery discharge curve

Appendix B

Description of X-3D-BL

The X-3D-BL is developed and manufactured by the German company Ascending Technologies. Ascending Technologies develop different kinds of remote controlled aerial vehicles. The X-3D-BL is a part of their hobby line. The X-3D-BL is a kit consisting of the parts listed in Table B.1.

X-Base	Power and signal processing
FunPilot	Sensor board and attitude controller
X-CSM	The frame
X-BL	Brush less motors, motor controllers and rotors
DSL-4top receiver	The Radio receiver
X-ACC	Gyroscope for absolute attitude hold

Table B.1: Parts in the X-3D-BL kit

In the following sections the individual parts are shortly described. All information is found in the X-3D-BL manual [16], the Ascending Technologies website [17] and the X-Control software.

B.0.3 X-base

The X-base is the central control unit connected to all other peripherals. All communication with the X-3D-BL goes through this unit. The X-base is equipped with an AtMega88 processor (16 MHz) which among other things controls the speed of the rotors. The ON/OFF button for the system is located on the X-base. A short press turns it on a press longer than 200 ms turns it off. The X-base is shown on Figure B.2.

With the software X-Control it is possible to calibrate the radio receiver, configure the directions of the motors and configure battery warnings. The Firmware currently used is version "ACC-3.2". The X-base communicates with a PC through a X-USB serial to USB interface.

B.0.4 FunPilot

The FunPilot (shown on Figure B.3) is the sensor unit that makes the stable flight of the X-3D-BL possible. The FunPilot is equipped with three piezo gyroscopes that measures the angular velocity of the body. The



Figure B.1: The X-3D-BL

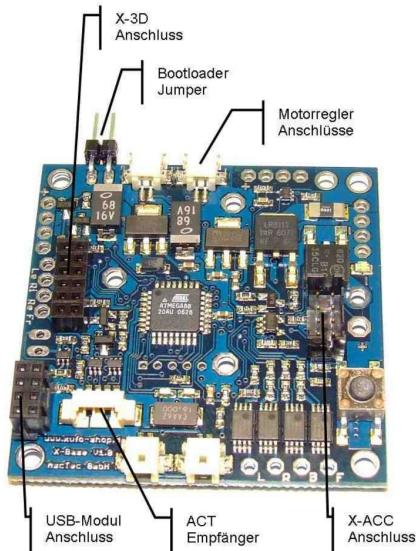


Figure B.2: The X-base

data are processed in a LPC_{2146} ARM7 processor (60 MHz). Further there is a pressure sensor that makes height control possible. As default the X-3D-BL is controlled by a heading hold controller on the FunPilot, but with the X-ACC it is possible to switch this to an alternative controller. This alternative controller keeps an horizontal attitude unless there is input from the pilot. Both controllers are updated with a frequency of 1 kHz.

All controllers can be configured from the X-Control software. The controllers can be configured with numerous constants. Among them values as K_p and K_d . Four different sets of parameters for the controller can be saved on the X-3D-BL, and one can change between them when the motors are stopped.

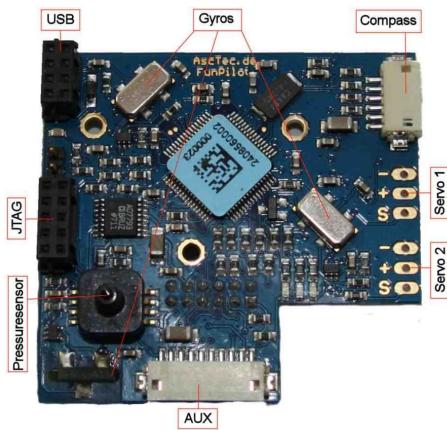


Figure B.3: The FunPilot

With the default heading hold controller the input S_ϕ, S_θ and S_ψ given from the RC-controller is angular velocity references to the controller. S_{col} is a reference to the total angular velocity of all rotors. The input S_{col} is further analysed in Appendix A.2.

B.0.5 X-CSM

X-CSM is the frame of the X-3D-BL. The frame is made of carbon fibre and magnesium to make a stable and robust frame. All electronics are protected from impact by a magnesium bar.

B.0.6 X-BL

The X-3D-BL is equipped with four brush less DC motors (X-BL-52s). Each with a brush less motor controller (X-BLDC). The controllers receive their reference values from the X-base through an I²C bus. The hardware on the controllers are equipped with an AtMega88 (15 MHz) processor and can deliver a current of 12 A continuous and 20 A peak.



Figure B.4: A brush less motor

B.0.7 DSL-4top receiver

The DSL-4top receiver from ACT Europe is the four channel receiver included in the X-3D-BL kit. The receiver supports 35 MHz and 40 MHz radio frequencies. The DSL-4top supports S-DSL and PPM connections.

B.0.8 X-ACC

The X-ACC is an add-on module that makes it possible to compute an absolute attitude and hold it. It is equipped with a 3-axial accelerometer, which measures an absolute angle for pitch and roll.

Appendix

C

CD Contents

The CD attached to the report contains relevant information, Matlab scripts, videos and measurements results. Following is a description of the folders on the CD and their content.

Model Contains all Matlab files and measurement results used throughout the project

Use: Change directory to the model folder and run “startup”. Additional commands will be available.

Report A pdf version of the report is found here

Videos Videos of the quad rotor doing various tests